SORT Special issue: Privacy in statistical databases, 2011, 59-76

Eliminating small cells from census counts tables: empirical vs. design transition probabilities

Sarah Giessing[1] and Jörg Höhne[2]

Abstract

The software SAFE has been developed at the State Statistical Institute Berlin-Brandenburg and has been in regular use there for several years now. It involves an algorithm that yields a controlled cell frequency perturbation. When a microdata set has been protected by this method, any table which can be computed on the basis of this microdata set will not contain any small cells, e.g. cells with frequency counts 1 or 2. We compare empirically observed transition probabilities resulting from this pre-tabular method to transition matrices in the context of variants of microdata key based post-tabular random perturbation methods suggested in the literature, e.g. Shlomo, N., Young, C. (2008) and Fraser, B.,Wooton, J. (2006).

MSC: 62Q05 ”Statistical tables”

Keywords: Tabular data protection, Census Frequency Tables, SAFE, Post-tabular protection methods.

1. Introduction

In preparation for the German census 2011 we have started a comparative study of several perturbation methods for census frequency counts. The German Census will partly be register based, and partly be the outcome of a sample survey. This leads of course to limitations in the amount of detail of tables that can sensibly be released, as compared to a full census. Nevertheless, a huge amount of tabular output is going to be published. Publication of tables will to a major extent be pre-planned, but there will also be some flexible, user demand driven release of tabular data.

[1] Federal Statistical Office of Germany, 65180 Wiesbaden.

[2] State Statistical Institute Berlin-Brandenburg, 14467 Potsdam, Dortustraße 46.

Received: November 2010
Accepted: March 2011

Given the size of the publication, and other complexities (like non-nested hierarchies that are foreseen for some classification variables like "age") non-perturbative methods like cell suppression do not seem to be a good choice: one of the issues to be raised here is that with cell suppression, there would be a considerable disclosure risk due to incomplete coordination of cell suppression patterns across tables. Perturbation methods also have the advantage that they introduce ambiguity into the zero cells which helps to avoid attribute disclosure when (nearly) all members of a population group score on only one (sensitive) category of a variable.

In this paper, we investigate into basically three alternative methods. The software SAFE (c.f. Höhne, J. (2003a), Höhne, J. (2003b)) is in regular use at the State Statistical Institute Berlin-Brandenburg. SAFE is an implementation of an algorithm that yields a controlled cell frequency perturbation. When a microdata set has been protected by this method, any table which can be computed on the basis of this microdata set will not contain any small cells, e.g. cells with frequency counts one or two. These small frequencies are the main concern for disclosure risk in Census counts tables, since they give information on the uniqueness or rareness of certain attributes or attribute combinations of individuals. Because SAFE is a pre-tabular method, all tables computed from the perturbed microdata set protected by SAFE are fully consistent and additive.

In comparison to SAFE, we study two post-tabular perturbation methods which both are based on the use of microdata keys. This technique can ensure full, or at least approximate, consistency of perturbations across different tables. Across table consistency has two aspects: On one hand, inconsistencies may be irritating to users. More severe from the disclosure control point of view is that inconsistency may lead to disclosure risk. For example, an average taken over eventually inconsistently perturbed values of logically identical cells (taken from different tables) should not be an unbiased estimate of the original cell value.

Each of the two post-tabular methods involves two steps. The first step yields fully or approximately consistently perturbed, but non-additive tables. Non-additivity is a potential nuisance for users, and may also be a source of disclosure risk. Therefore, in a second step, table additivity should be restored. This can be achieved by statistical methods such as the iterative proportional fitting algorithm. In this paper we discuss using linear optimization techniques for this step.

In order to avoid a perception of disclosure risk, and to provide a "visible" kind of protection, we require both methods to provide, like SAFE, perturbed data without small cells (i.e. without counts of one and two). Note that we imagine a rather naïve use of the data here, keeping in mind that for researchers there will be other options of accessing the data.

This paper reports on findings of the first phase of the study when implementing the methodologies. It is organized in eight sections: In Section 2, we outline the methodological approach of SAFE. Technical issues of constructing suitable probability transition matrices for random perturbation methods are discussed in Sections 3 and 4. In Section 5, we suggest an optimization technique to restore table-additivity, e.g.

the CTA method of Castro, J., González, J.A. (2009). Some test results are presented in Section 6, and a measure of information loss on the cell level for SAFE results is proposed in Section 7. We conclude the paper with a brief summary Section 7.

2. Methodological background of SAFE

In this section we briefly describe the methodological approach of SAFE, as far as it is relevant for an application to protect tabulations of population Census counts data. Starting point for the method is a microdata file where all variables are recoded to give the highest degree of detail foreseen for any publication. Imagine a variable like age, where perhaps data are collected so that for each person age could be deduced down to the level of age in months, but publications should offer data at most by age in years. Then the variable would be recoded to the level of age in years. We also assume here that the data set consists of categorical variables only.

The basic idea of the method is to turn this data set (with, say, N variables at n_i $(i = 1,\ldots,N)$ categories) into a data set, in which either none of the records, or at least three records score on each of the $n_1 * n_2 * \cdots * n_N$ theoretical combinations of categories.

With respect to data quality, the method aims to preserve as far as possible cell counts in a pre-defined set of ‘controlled’ tables. For those tables, the method yields results that are in some sense ‘optimal’. If any other table is derived from the perturbed data set, it will be safe (i.e. it will not contain any ones or twos), but differences between original counts and those computed on basis of the perturbed data set can be much larger than they arise for the controlled tables. The experience is that the method is usually able to achieve a maximum deviation between 4 and 8 for a sensibly defined set of controlled tables.

2.1. The SAFE mathematical model

The algorithm computes a heuristic solution for the problem of minimizing the maximum absolute deviation between true and perturbed cell values in the controlled tables.

Instances are defined by the following parameters:

- A set of linear relations $Ay = a$ defining the table cells of the controlled tables as sums of cells of an elementary table consisting of all combinations of categories of all variables in the microdata set.
- Vector $a, a = (a_i,\, i \in I)$ denote the original frequencies presented in the controlled tables, and vector $y, y = (y_j,\, j = 1,\ldots,N)$ the entries (e.g. frequencies of category combinations) of the cells of the elementary table. In a valid solution, vector y does not contain any entries of 1 or 2.

- Vector w of weights associated to perturbations of the table cells of the controlled tables. For example, we may want to allow larger perturbations for larger cells, or avoid them for cells that are rated “highly important”.

The objective of the model is to minimize the maximum entry of vector $d = (d_i,\, i \in I)$, $d_i \in \mathbb{Z}$, denoting the deviations of original and perturbed cell counts in the controlled tables. With these definitions, broadly, the model is as follows:

Solve the problem

$$\begin{aligned} &\min_y \quad \max_{i \in I} (|d_i| + w_i) \\ &\textit{subject to} \quad Ay = a + d \\ &\qquad y_j \in \{0, 3, 4, 5, \ldots\} \quad j = 1, \ldots, N \end{aligned} \tag{1}$$

This statement of the problem resembles a huge non-linear integer optimization problem which is computationally intractable[1]. Therefore, an efficient heuristic algorithm has been developed that gives near optimal solutions at reasonable expense of computer resources.

Beginning with the (infeasible) initial solution given by $d = 0$, i.e. where cell values are kept at their original value, a first feasible solution is obtained. This solution is optimized later on.

A first feasible solution

In addition to the above parameters, we define now

- Vector $b = (b_i,\, i \in I)$, $b_i \in B$ of bounds for maximum allowed deviations. In practice, B consists of two values only, one stating the maximum deviation to be allowed for cells defined by only one variable, the other one stating the maximum allowed deviation for the other cells, e.g. cells defined as cross-combination of categories of two or more variables,
- Vector $x = (x_j,\, j = 1, \ldots, N)$, $x_j \in \{0, 1\}$ is 1, if elementary table cell j is “unsafe”, e.g. if $y_j \in \{1, 2\}$ and 0 otherwise.

The problem to be solved is

$$\begin{aligned} &\min_y \quad \sum_{j=1,\ldots,N} x_j \\ &\textit{subject to} \quad |Ay - a| < w + b \\ &\quad y_j \in \{0, 1, 2, 3, \ldots\} \\ &x_j = 1;\ \textit{if}\, y_j \in \{1, 2\} \quad x_j = 0; if\ y_j \notin \{1, 2\} \end{aligned} \tag{2}$$

[1] Note, in our test setting which is still substantially smaller than the real setting will be, the size of vector y (and thus the number of columns of matrix A) is approximately $N = (2*4*7*8*111*10000) \sim 5*10^8$.

A feasible solution is obtained when the objective function is zero.

Minimizing the number of "unsafe" frequencies, using a heuristic, the algorithm step by step changes critical frequencies of 1 and 2 into uncritical frequencies 0,3,4,... If the process stagnates, the statement of the problem is modified automatically by increasing the vector of bounds b, e.g. $b = b + 1$.

Optimizing the solution

Once a feasible solution has been obtained, the method will seek to improve the solution by reducing the maximum allowed perturbation, e.g. b and eventually w. Usually, the number of cells where the deviation is identical or near-identical to the respective bound is relatively small. In the optimization step, after changing (decreasing) b or w, some of the constraints in model (2) will be violated. Accordingly, we define now

- Vector $z = (z_i, i \in I)$, $z_i \in \{0,1\}$ is 1, if for controlled tables cell i the bound constraint of model (2) is violated and 0 otherwise.

The algorithm derives a heuristic solution to

$$\begin{aligned} &\min_y \quad \sum_{i \in I} z_i \\ &\textit{subject to} \quad |Ay - a| - (w + b) < z \\ &\quad y_j \in \{0,3,4\ldots\} \end{aligned} \tag{3}$$

If a solution is obtained where $\sum_{i \in I} z_i = 0$, the constraints will be tightened further (e.g. decrease b or w), and model (3) will be solved again. This step is repeated until either an expected level of optimality (in the bounds) is reached, or further attempts seem to be rather unpromising.

3. Generating random noise for frequency tables

The Australian Bureau of Statistics Fraser, B., Wooton, J. (2006), Leaver, V. (2009) has developed a concept for a cell perturbation method. They propose that the random noise should have zero-mean and a fixed variance. An alternative cell perturbation method referred to as "Invariant Post-tabular SDL" method was suggested in Shlomo, N., Young, C. (2008). In the following two subsections we briefly outline the two alternative concepts and discuss the technical construction of suitable probability transition matrices for a random perturbation eliminating all small frequency counts.

3.1. How to create zero-mean/fixed variance cell perturbations?

Fraser, B., Wooton, J. (2006) propose to generate for each cell c with non-zero cell count i_c an independent integer value perturbation d_c satisfying the following two criteria:

(a) mean of zero

(b) fixed variance V for all cells c and all frequency counts i

A third criterion, in order to meet the requirement that perturbed cells do not have a count of one or two, would be

(c) $i_c + d_c \notin \{1, 2\}$ f.a. i_c, d_c

This means we look for a $L \times L$ transition matrix $\mathbf{P}$[2] containing conditional probabilities: $p_{ij} = p$ (perturbed cell value is $j \mid$ original cell value is $i)$ with the following properties:

(1) $p_i v_i = 0$

(2) $p_i (v_i)^2 = V$

(3) $p_{ij} = 0$ for $jin\{1,2\}$

(4) $\sum_j p_{ij} = 1$

(5) $p_{ij} = 0$; if $j < i - D$ or $j > i + D$,

(6) $p_{00} = 1$ and $p_{0j} = 0$ for $j > 0$, and of course

(7) $0 \leq p_{ij} \leq 1$

where p_i denote the ith row-vector of matrix $\boldsymbol{P}$ and v_i a column vector of the noise which is added, if an original value of i is turned into a value of j. I.e. the j^{th} entry of v_i is $(j - i)$. For example $v_i = (-1, 0, 1, 2, 3, \ldots, L-2)$. (1) is equivalent to (a) and expresses the requirement that the expected value of the noise should be zero. Similarly, (2) is equivalent to (b), expressing the requirement of a constant variance, and (3) relates to (c). (4) and (7) are of course necessary for any Transition matrix, (5) states a maximum allowed absolute perturbation of some pre-defined constant D and (6) states that zero frequencies must not change. Note for all rows after row $D+2$, condition (3) is always satisfied, when (5) holds. Hence we can facilitate the task of computing suitable transition probabilities by adding a symmetry requirement for all rows after row $D+2$:

(8) $p_{i,i-k} = p_{i,i+k}$ for $k = 1, \ldots, D$, if $i > D+2$

[2] As index j may take a value of zero (when a cell value is changed to zero), in the following we start counting matrix and vector indices at 0, enumerating rows and columns of the $L \times L$ matrix by $0, 1, 2, \ldots, L-1$.

With (8), condition (1) is always satisfied because the negative and positive deviations balance each other. (2) simplifies into

(2a) $2 \sum_{j=1,\ldots,D} p_{ij}\, j^2 = V.$

For simplicity, in the following we therefore assume $L-1 = D+3$, applying the perturbation probabilities given by row $(D+3)$ of matrix $\boldsymbol{P}$ to all cell counts $\geq D+3$.

For every row (or cell count) i $(i = 1, \ldots, D+2)$ conditions (1) to (5) can be rewritten as system of three linear equations

(9) $\boldsymbol{A}_{iD}\, x = b,$

where $\boldsymbol{A}_{iD}$ is a $(3 \times (\min(i,D)+1+D-k))$ [3] coefficient matrix and $b = (1,0,V)'$. The elements of x correspond to the entries of row i in P which are not zero anyway by definition (because of (3) or (5)). The first row of $\boldsymbol{A}_{iD}$ corresponds to condition (4), the second row to (1) (e.g. unbiasedness) and the third row to (2) (fixed variance V).

Consider for example $\boldsymbol{A}_{13} = \left\{ \begin{array}{rrr} 1 & 1 & 1 \\ -1 & 2 & 3 \\ 1 & 4 & 9 \end{array} \right\}$. In this simple case, the coefficient matrix is invertible. The last row of the inverse is $(-1/2, -1/4, 1/4)$. Hence, in order for p_{13} to be positive, $(-1/2, -1/4, 1/4) \cdot b = (-1/2 + V/4)$ must be positive, and hence V must be at least 2. In this case (9) has a unique solution, depending on the choice of V only. If V is exactly 2, p_{13} is zero.

In general, $\boldsymbol{A}_{iD}$ has more columns than rows. So usually, there is no unique solution for (9). But we can use (9) to derive feasibility intervals for x (e.g. for the p_{ij}). A practical approach is to fix V to $2+\varepsilon$ with a small positive value for ε (increasing ε and hence the variance of the perturbation leads to an unnecessary loss of information). The system (9) can be further strengthened by additional constraints, for example to express desirable monotony properties like $p_{ij} \geq p_{i,j+1}$ for $j > i$, or to improve symmetry by bounding the difference between $p_{i,i-1}$ and $p_{i,i+1}$.

We have experimented with $D = 3$, 4 and 5. One of the findings was that for small D and i, the linear programming problem derived from (9) (eventually together with the additional constraints) gives quite small intervals for x. For larger D and i the intervals for x are wider. In those cases we first fixed a value (like 70 %) for the centre of the distribution, p_{ii}. Afterwards we fitted each tail of the distribution p_{ij}, $j > i$ and p_{ij}, $j < i$ to the tails of a normal distribution using a simple heuristic approach:

At first, provisionally fix one (say, the left-hand) tail of the distribution. This gives a target total probability and target total variance for the right-hand tail (through subtracting the corresponding left hand tail values from differencing one (V, resp.)).

[3] k is the number of elements in $\{1,2\} \cap [i-D;\, i+D]$.

Then approximate p_{ij} $(j > i)$ by $F_{k+0.5+i} - F_{k-0.5+i}$, where F_x denote the Normal distribution with zero expectation and suitable Variance σ^2 at x, and k denote the starting point of the distribution tail. The starting point k should be selected as to achieve that the approximate p_i, D_{+i} is about zero. See Gießing, S., Höhne, J., (2010) (Appendix), for further details, and how to obtain a suitable variance parameter.

The corrected approximate $p_{i,i+j}$ distribution can then be used to derive the target values for a corrected total probability and variance of the left-hand tail. Carry out the procedure described for the right hand tail for the left hand tail now. Finally, feed back the corrected approximate p_{ij} into the system (9) and (by minimizing or maximizing one of the variables) obtain a final distribution which meets the requirements of (9) with sufficient precision.

Table 1 in the appendix shows the final probability matrices for $D = 3$, 4 and 5, e.g. the design transition probabilities and compares them to the transition probabilities observed empirically for the cells of the set of controlled tables after protecting the data by SAFE. Obviously, the SAFE method results in much smaller probabilities that cell values change by less than three.

3.2. Combination of invariance and a “no-small-cells” requirement?

The idea of the “Invariant Post-tabular SDL” method Shlomo, N., Young, C. (2008) is to preserve the frequency distribution of the cell counts. But in our setting we require the frequency of perturbed small counts (ones and twos) to be zero. So for the small counts these are aims that clearly exclude each other. A possible way out would be to relax the goal of invariance. E.g. only seek to preserve the frequency distribution of cell counts above three and the total frequency of all cell counts below four. This can be achieved as follows:

As shown in Shlomo, N., Young, C. (2008), an invariant matrix $\boldsymbol{R}$ is obtained by multiplying some pre-defined initial transition matrix $\boldsymbol{P}$ (for an example see Shlomo, N., Young, C. (2008)) with a suitable matrix $\boldsymbol{Q}$. $\boldsymbol{Q}$ is obtained by transposing matrix $\boldsymbol{P}$, multiplying each column j by the relative frequency of count j and then normalizing its rows so that the sum of each row equals one. Finally the diagonal elements of this matrix are increased by the following transformation $\boldsymbol{R*} = \alpha R + (1-\alpha) I$, where I is the identity matrix of the appropriate size.

Gießing, S., Höhne, J., (2010) explain how to adapt this procedure to the “no-small-cells” requirement. In a first stage, an invariant matrix $\boldsymbol{R*}$ is computed such that the first row gives the joint transition probabilities of all counts under four, and the first column gives the probabilities for changing a given count into a count smaller than four. The procedure to obtain $\boldsymbol{R*}$ is the same as in Shlomo, N., Young, C. (2008), except that here we use a vector of relative frequencies, where the entries corresponding to the ones, twos and threes are added up to one joint entry v_{1-3}. We also replace the first row of the initial transition matrix by a column vector where all entries except for the first two are zero. See Gießing, S., Höhne, J., (2010) for details of how to compute the first two entries of this vector, and on how to compute separate transition probabilities for counts under four. Finally, we replace the first line of ***R**** by the separate transition probabilities for counts under four (and attach three columns of zeros to the other lines). This way we get a transition matrix ***R*****, which is almost invariant, except that for counts under four only their total frequency is preserved. For illustration, in the following we present an example using real data of a table of the last West German census of 1987.

Example 1:
For a census table with frequencies $(V_1, V_2, V_3, V_4, V_5, \ldots) = (96, 32, 20, 16, 15, \ldots)$ observed for counts (1,2,3,4,5,...), we computed an initial invariant matrix ***R**** (with $D = 2$). Table 2 shows the first four rows and six columns of the matrix of expected frequencies obtained from $(V_{1-3}, V_4, V_5, V_6, V_7, \ldots) \cdot \boldsymbol{R*}$

Table 2: *Expected frequencies $n_{i,j}$ of counts of i perturbed into counts of j.*

	0-3	4	5	6	7	8
0-3	144.62403	2.9690406	0.4069246	0	0	0
4	2.9690406	11.81552	1.0893785	0.1260606	0	0
5	0.4069246	1.0893785	12.211631	1.196397	0.0956693	
6	0	0.1260606	1.196397	10.676843	0.9239783	0.0767213

Table 3 below shows the first six rows and six columns of the matrix of expected frequencies computed as $(V_1, V_2, V_3, V_4, V_5, \ldots) \cdot \boldsymbol{R**}$. The sum of the first two column totals in Table 3 (regarding $j = 0.3$) is 148, e.g. the total observed frequency of the counts under 4 $(= 96 + 32 + 20)$ is exactly preserved.

Table 3: *Expected frequencies $n_{i,j}$ of counts of i perturbed into counts of j for example 1.*

	0	3	4	5	6
1	60.531974	35.468026			
2	9.510658	22.489342			
3	4.6240348	12	2.9690406	0.4069246	0
4		2.9690406	11.81552	1.0893785	0.1260606
5		0.4069246	1.0893785	12.211631	1.196397
6		0	0.1260606	1.196397	10.676843

Note that apart from this introductive example, we did not carry out further testing of this method. The concept of preserving frequencies for each individual cell count is not too convincing when the expected use of the data is a rather naïve one[4]. An exception would be a situation where the frequencies for individual cell counts are a statistic of interest for the user. Such an application is outlined in Section 7.

[4] Note that there will be options for researchers to access the original data via research data centres.

4. Selection of random noise

The random mechanism proposed in Fraser, B., Wooton, J., (2006) can be implemented very easily: For our experiments, we used the SAS random number generator which produces pseudo random numbers distributed uniformly over $[0;2^{31}-1]$. We assign such a random key to each record in the microdata file. When computing the tables, also the random keys are aggregated. The result is then transformed back into a random number on this interval by applying the modulo function, e.g. $\mathrm{mod}_{2^{31}-1}$. If the same group of respondents is aggregated into a cell, the resulting random key will always be the same. Cells which are logically identical thus have identical random keys.

Then we simply use a transition matrix computed to give zero-mean / fixed variance noise (as explained in 3.1), compute cumulated probabilities (for each row) and multiply the resulting matrix by $2^{31}-1$. Denoting the entries of this matrix by $\boldsymbol{M}_{ij}$ we change a cell count of i of some cell c into j, if the random key of cell c is between $\boldsymbol{M}_{i,j-1}$ and $\boldsymbol{M}_{ij}$. This will guarantee that the expected values of the perturbed counts are identical to the original counts (unbiasedness) and lead to consistently perturbed data. However, for a given table, the mean perturbation of cells of a given frequency count i is not necessarily zero. This mean will depend on the actual distribution of the corresponding record keys. For the data of example 1 above the observed difference between a true cell count and the mean of the corresponding perturbed counts varies between -0.82 and 0.78.

See Section 4.1 of Gießing, S., Höhne, J., (2010) for some special issues regarding an appropriate selection procedure in the context of the invariant post tabular method.

5. How to restore table-additivity?

Non-additivity is a potential nuisance for users, and may also be source of some disclosure risk. As simple example, assume random noise with a maximum perturbation of two has been applied. Assume two cells with original count one are perturbed to count three, and the original total of two is perturbed to zero. Users are informed on the maximum perturbation. Hence they know that both inner cells must have original count one at least. But if any of them were greater then one, the original total would be at least three and could not have turned into a perturbed value of zero.

This kind of disclosure risk typically arises, when all inner cells are all perturbed in the same direction, each with the maximum possible perturbation, and the total cell is perturbed in the other direction, also with the maximum possible deviation. With perturbations based on transition matrices like the ones discussed in Section 3 with usually small probabilities on the tails these events will be relatively rare. However, we should also bear in mind, that this is only the simplest kind of attack. A systematic analysis based on linear optimization techniques and taking into account the aggregate structure of a perturbed non-additive multidimensional table with a published maximum perturbation might eventually break other perturbation patterns as well.

Restoring table additivity, as suggested in Fraser, B., Wooton, J. (2006) and Shlomo, N., Young, C., (2008) is considered there an integral part of the method. Leaver, V. (2009) and Shlomo, N., Young, C., (2008) point out that restoring additivity can be achieved by iterative methods. As an alternative, we suggest to consider a linear programming based method like Controlled Tabular Adjustement (see f.i. Dandekar, R.H., Cox, L. (2002), Castro, J. (2006)).

For a first experiment, we use the CTA implementation of Castro, J., González, J.A. (2009)[5]. The algorithm restores additivity to a table, minimizing an overall distance to the table provided as input. The distance function implemented is a weighted sum of absolute per-cell-distances. Weights are provided by the user of the software. The user can define for each cell upper and lower bounds on the deviations, and can define a set of cells labeled as '*sensitive cells*'. Sensitive cells are forced to change their values. For each sensitive cell, the user defines a '*protection interval*'. The adjusted cell value is not allowed to take a value within the protection interval.

Computational complexity of the problem depends strongly on the number of sensitive cells. In a first experiment, we therefore use a two stage approach: in a first CTA run, we only restore additivity to the table. Although in this step we assign cell weights which will avoid to some extent that the algorithm adjusts cell counts of zero[6] or three, we will usually get an adjusted table with some small cell counts (e.g. ones and twos). In a refinement run, we define these ones and twos as sensitive, and define the corresponding protection interval as the interval (0;3). At the same time, for all cells with counts greater or equal to three we defined a lower bound of at least three. For all cells with zero count, the upper bound is zero. This way, however, we run a certain risk of defining an infeasible problem, especially if we define at the same time rather narrow bounds for the non-sensitive cells. See Section 6 for a test result.

Because the adjustment cannot simultaneously take into account all tables ever to be released[7], it introduces inconsistencies in the perturbation. Identical cells, even if they received the same perturbation by the random process, may become adjusted to different values. This fact leads to some risk that some perturbations might be undone, if intruders run an LP-based analysis taking into account the aggregate structure across several tables. But this is not such an easy task, on one hand, and on the other hand, it may not be very successful, because it may happen that only original frequencies can be broken that do not cause disclosure risk.

Of course one might consider using the adjustment methodology without previous random perturbation, only to 'remove' cells with small counts from the table. But as long as this does not – unlike the SAFE method – yield a fully consistent data base, there is then a risk that by averaging cell values over a number of tables a user can recover the

[5] See Castro (2011) for an extension of the methodology.

[6] Note that we do not allow original zero cell counts to be adjusted.

[7] (This would be a problem similar to the on solved by SAFE, c.f. 2 (in particular in size) and too huge for today computational resources).

original data. With a previous random perturbation, such an approach will only recover the underlying perturbed table, as pointed out in Leaver, V., (2009).

6. Some test results

Table 1 in the appendix shows the probability matrices we computed for the zero mean/fixed variance noise approach when the maximum allowed deviations are $D = 3$, 4 and 5 respectively, and compares them to the transition probabilities observed empirically for the cells of the set of controlled tables after protecting the data by SAFE. Obviously, the SAFE method results in much smaller probabilities that cell values change by less than three.

For all counts after $D+3$, in our implementation of the stochastic noise, transition probabilities are defined identical to those obtained for $D+3$. Figure 1a below shows the empirical SAFE probabilities for counts i to change by d for counts between 9 and 16 in the set of controlled tables, compared to the transition probabilities of the stochastic noise obtained for $D = 5$. Figure 1b shows those probabilities for counts grouped into count size classes observed for cells that are not in the controlled tables. For our experiment we defined as control tables only tables defined by cross-combination of at most 3 variables. The results presented in Figure 1b on the other hand relate to cells defined by cross-combination of 4 variables.

As can be seen in Figure 1a, the SAFE probabilities become approximately normal when the cell count increases. It is also very clear that the SAFE perturbation is stronger than that of our stochastic noise implementation: Compare f.i. the probability of no change (i.e. at $d = 0$) which is about 70 % for the stochastic noise, but between 13 % and 26 % for SAFE. However, the difference matters mainly for the small perturbations, and hence will matter more for smaller counts.

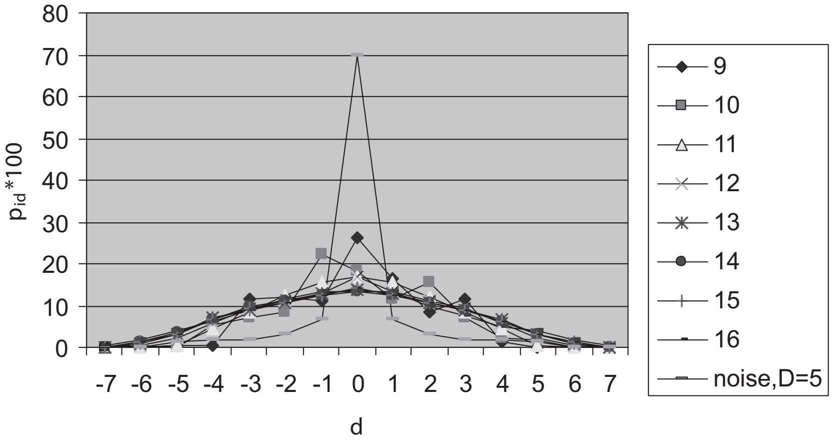

Figure 1a: *SAFE vs. stochastic noise transition probabilities.*

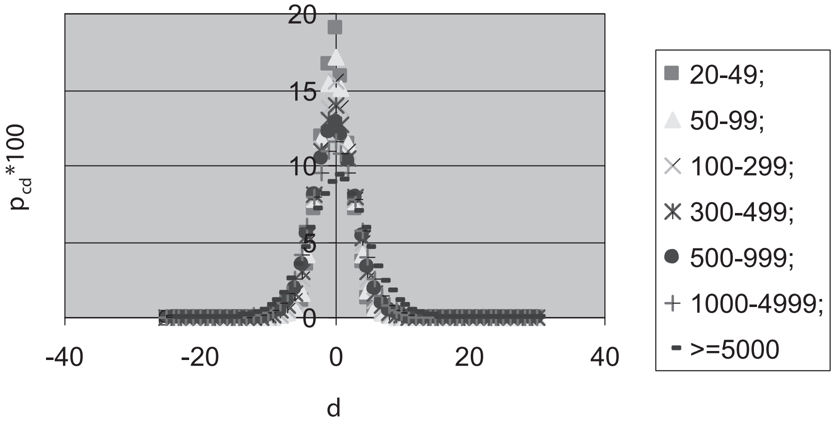

Figure 1b: *SAFE, transition probabilities for banded counts in non-control tables. probabilities.*

Figure 1b shows that also for cells that are not contained in the controlled tables, the deviations resulting from SAFE are still normally distributed, but the tails of the distribution are longer. While we got a maximum deviation between true and perturbed count of 7 for controlled tables cells, deviations of up to 30 occurred in the set of 4-dimensional cells, as can be seen in table 4 presenting the maximum observed deviations for the cell count size classes of Figure 1b.

Table 4: *SAFE, maximum observed deviations D for non-control table by cell count size class.*

Counts	20-49	50-99	100-299	300-499	500-999	1000-4999	≥ 5000
D	16	19	25	21	30	24	26

Considering that table additivity is a very important issue, it makes sense to compare SAFE transition probabilities not only to the design transition probabilities of stochastic noise, but also to the noisy tables after restoring additivity. We have applied the approach of Section 5 for restoring table additivity using CTA to a 3-dimensional test table. The table has been perturbed using the design transition probabilities displayed in Table 1 (appendix). For this instance we obtained adjusted tables where the maximum perturbation of cell counts is identical before and after the adjustment. This is certainly encouraging, but it seems unlikely that it is a general result. Table 5 compares the noisy

Table 5: *Distribution of 22670 non-zero test table cells by absolute deviations to true cell values.*

Abs. Dev.	SAFE	noiseD3 adj.	noiseD3	noiseD4 adj.	noiseD4	noiseD5 adj.	noiseD5
0	12.88	23.17	29.44	29.44	38.99	30.62	40.18
1	44.62	44.05	40.23	39.87	34.08	40.05	34.63
2	27.68	23.31	22.31	21.01	19.20	20.41	18.48
3	11.09	9.47	8.01	5.86	4.31	5.07	3.33
4	3.16			3.82	3.42	2.38	1.92
5	0.48					1.47	1.46
6	0.10						

tables before and after restoring additivity to those computed with SAFE protected data. It presents the frequency distribution of the 22670 non-zero cells of the example by absolute deviations between true and perturbed values.

For this example, we observed a mean-deviation between true and perturbed values of 1.49 for SAFE. For stochastic noise at $D = 3$, 4 and 5 we got mean deviations of 1.09, 0.99 and 0.97, resp., and after restoring additivity 1.19 ($D = 3$), 1.15 ($D = 4$) and 1.13 ($D = 5$). Obviously, in this example, even after restoring additivity, stochastic noise outperforms SAFE. On the other hand, the experiment also shows that – at least when we use the methodology of Section 5 not allowing that new small cells appear in the adjusted tables – restoring additivity tends to increase deviations (for example the mean deviation for $D = 5$-noise from 0.97 to 1.13)[8]. It has to be expected that this effect increases with increasing size of the tables where additivity has to be restored. The computationally expensive second CTA step[9] required between about 6 and 24 minutes. As it is intended that table generation for the Census results should be an OnLine process, this is certainly too long. Even, if this issue could be solved, before such an approach could be put into practice, a lot of experimentation would be necessary, for example to determine “sustainable” parameters for the initial random perturbation in the sense that the adjustment process can preserve to some extent the properties of the random perturbation (like f.i. the maximum perturbation).

7. Data utility – a cell level measure of information loss

Probably, many users of census counts data do not use them for complex statistical analyses, but are merely interested in learning simple facts, like ‘how many people with properties X live in area Y?’. When those counts are perturbed, they should be informed how reliable each individual cell is. This is especially important, if a perturbation method may produce fairly large perturbations, although only for a very small portion of the cells, which can f.i. be the case for SAFE for cells which do not belong to the set of controlled tables.

A simple information loss measure on the cell level could be given by publishing along with the perturbed counts the absolute value of the perturbation. However, this may be too much information, leading to disclosure risk. Instead, one might publish the absolute value of a perturbed version of the perturbation.

Usually, to inform about data utility, one publishes information on the perturbation on the table level, like the frequency distribution of the noise (c.f. Table 5). Therefore, when perturbing the perturbations, it makes sense seeking to preserve these frequencies. E.g. use an invariant matrix of transition probabilities for perturbing the perturbations

[8] Note that these findings may not apply to all additivity methods.

[9] The first step which only restores additivity to the table takes just a few seconds for this instance.

of the original counts in a table. Generating such a transition matrix is a straightforward application of Shlomo, N., Young, C. (2008). The only difference is that, unlike the original counts which are positive numbers, the perturbations take values between $-D$ and D. Table 6 shows the results of an application to table Region x Age x Country_of_Birth[10]. The observed frequencies of the perturbed SAFE-deviations (n_d*) match the frequencies of the unperturbed SAFE-deviations (n_d) nearly exactly.

Table 6: *Number of cells of a test table by deviation of the SAFE protected results: true frequencies* (n_d) *vs. frequencies after invariant perturbation of observed deviations* (n_d*).

Cells with negative deviation *d*				**Cells with positive deviation *d***			
d	n_d	n_d*	$n_d - n_d*$	***d***	n_d	n_d*	$n_d - n_d*$
−13	1	0	1	13	0	0	0
−12	5	5	0	12	7	6	1
−11	30	31	−1	11	44	45	−1
−10	110	110	0	10	108	108	0
−9	310	309	1	9	372	370	2
−8	836	837	−1	8	878	879	−1
−7	1872	1871	1	7	2141	2141	0
−6	8203	8204	−1	6	9230	9231	−1
−5	34859	34859	0	5	37674	37675	−1
−4	162116	162115	1	4	170659	170657	2
−3	369234	369234	0	3	393652	393654	−2
−2	622462	622464	−2	2	778735	778735	0
−1	1226831	1226831	0	1	783760	783758	2
0	739905	739905	0	0	739905	739905	0

8. Summary and final remarks

In preparation for a comparative study of several perturbation methods for census tabular frequency data, in this paper we have raised some practical issues regarding the implementation of two alternative approaches explained in literature. In particular, this paper has discussed in some detail how to construct zero-mean/fixed variance transition matrices required to implement the methodology of Fraser, B., Wooton, J. (2006). We also discuss an extension of an idea of an invariant transition matrix suggested in Shlomo, N., Young, C. (2008) to a situation where the perturbation procedure should eliminate small cells.

As pointed out in Fraser, B., Wooton, J. (2006) and Shlomo, N., Young, C. (2008), additivity is not preserved by the post-tabular random perturbation method, but can be restored afterwards – however, at the expense of between tables consistency. We have

[10] Note that variable Country_of_Birth has been defined here to involve one category which defines an extra-subtotal not contained in the set of cells defined by the set of controlled tables. Therefore, SAFE perturbs some cells of this table by more than the control-tables maximum of 7.

outlined and tested on a small instance an approach based on linear optimization, e.g. CTA methodology.

Leaving a larger scale empirical comparison of the post-tabular methods discussed in the paper with the pre-tabular perturbation method SAFE outlined in Section 2 for the future, the paper provides evidence that the post-tabular methods as implemented here tend to result in smaller changes to the data than SAFE. On the other hand, as a pre-tabular method, SAFE preserves additivity and consistency, is easier to implement in a flexible OnLine table generation environment, and is able to keep the maximum deviations in a set of pre-specified tables acceptably small. These are important properties and may be worth "less optimal" performance regarding data quality to some degree. While the perturbation caused by SAFE tends to be stronger than those caused by a non-additive post-tabular approach, the paper shows that they tend to be normally distributed, e.g. large deviations are relatively unlikely, also for cells that are not contained in the set of pre-specified, controlled tables.

References

Castro, J. (2006). Minimum-distance controlled perturbation methods for large-scale tabular data protection. *European Journal of Operational Research*, 171, 39–52.

Castro, J. and González J.A. (2009). A Package for L1 Controlled Tabular Adjustment, paper presented at the *Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality (Bilbao, 2-4 December 2009)* available at http://www.unece.org/stats/documents/2009.12.confidentiality.htm

Castro, J. (2011). Extending controlled tabular adjustment for non-additive tabular data with negative protection levels. *Statistics and Operations Research Transactions*, 35.

Dandekar, R.H. and Cox, L. (2002). Synthetic Tabular Data – an Alternative to Complementary Cell Suppression, unpublished manuscript.

Fraser, B. and Wooton, J. (2006). A proposed method for confidentialising tabular output to protect against differencing, in *Monographs of Official Statistics. Work session on Statistical Data Confidentiality*, Eurostat-Office for Official Publications of the European Communities, Luxembourg, 2006, 299–302.

Giessing, S. andf Höhne, J. (2010). Eliminating small cells from census counts tables: some considerations on transition probabilities. In J. Domingo-Ferrer and E. Magkos, eds., *Privacy in Statistical Databases*, 52–56. New York: Springer-Verlag. LNCS 6344.

Höhne, J. (2003a). SAFE – ein Verfahren zur Geheimhaltung und Anonymisierung Statistischer Einzelangaben, in *Berliner Statistik-Statistische Monatsschrift 3/2003*.

Höhne, J. (2003b), SAFE – a method for statistical disclosure limitation of microdata, paper presented at the *Joint ECE/Eurostat Worksession on Statistical Confidentiality* in Luxembourg, December 2007, available at www.unece.org/stats/documents/2003/04/confidentiality/wp.37.e.pdf

Leaver, V. (2009). Implementing a method for automatically protecting user-defined Census tables, paper presented at the *Joint ECE/Eurostat Worksession on Statistical Confidentiality* in Bilbao, December 2009, available at http://www.unece.org/stats/documents/2009.12.confidentiality.htm

Shlomo, N. and Young, C. (2008). Invariant post-tabular protection of census frequency counts. In J. Domingo-Ferrer and Y. Saygin, eds., *Privacy in Statistical Databases*, 77–89. New York: Springer-Verlag. LNCS 5262.

Appendix

Table 1: *Zero mean, Variance $2+\varepsilon$ probability transition matrices for maximum perturbations D of 3, 4 and 5 vs. empirically observed transition probabilities for SAFE.*

	0	3	4	5	6	7	8	9	10	11	12	13
							Random Noise, *D* = 3					
1	0.667	0.332	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.334	0.666	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.125	0.687	0.063	0.063	0.063	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.601	0.099	0.100	0.100	0.100	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.167	0.167	0.416	0.083	0.083	0.083	0.000	0.000	0.000	0.000	0.000
6	0.000	0.072	0.072	0.072	0.571	0.072	0.072	0.072	0.000	0.000	0.000	0.000
7	0.000	0.000	0.072	0.072	0.072	0.571	0.072	0.072	0.072	0.000	0.000	0.000
8	0.000	0.000	0.000	0.072	0.072	0.072	0.571	0.072	0.072	0.072	0.000	0.000
							Random Noise, *D* = 4					
1	0.667	0.333	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.334	0.666	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.120	0.700	0.082	0.045	0.027	0.026	0.000	0.000	0.000	0.000	0.000	0.000
4	0.064	0.076	0.700	0.068	0.037	0.029	0.026	0.000	0.000	0.000	0.000	0.000
5	0.000	0.143	0.143	0.542	0.043	0.043	0.043	0.043	0.000	0.000	0.000	0.000
6	0.000	0.063	0.063	0.063	0.662	0.038	0.038	0.038	0.038	0.000	0.000	0.000
7	0.000	0.032	0.033	0.034	0.050	0.700	0.050	0.034	0.033	0.032	0.000	0.000
8	0.000	0.000	0.032	0.033	0.034	0.050	0.700	0.050	0.034	0.033	0.032	0.000
							Random Noise, *D* = 5					
1	0.667	0.333	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.334	0.666	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.119	0.700	0.082	0.050	0.028	0.014	0.007	0.000	0.000	0.000	0.000	0.000
4	0.062	0.076	0.704	0.075	0.037	0.020	0.014	0.012	0.000	0.000	0.000	0.000
5	0.025	0.068	0.068	0.700	0.059	0.027	0.019	0.018	0.017	0.000	0.000	0.000
6	0.000	0.057	0.057	0.057	0.700	0.041	0.023	0.021	0.021	0.021	0.000	0.000
7	0.000	0.025	0.035	0.035	0.062	0.700	0.060	0.028	0.020	0.018	0.018	0.000
8	0.000	0.015	0.016	0.019	0.032	0.068	0.700	0.068	0.032	0.019	0.016	0.015
							SAFE					
1	0.680	0.288	0.031	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.408	0.472	0.073	0.006	0.040	0.001	0.000	0.000	0.000	0.000	0.000	0.000
3	0.208	0.514	0.101	0.015	0.153	0.008	0.000	0.001	0.000	0.000	0.000	0.000
4	0.077	0.440	0.122	0.026	0.262	0.058	0.002	0.013	0.000	0.000	0.000	0.000
5	0.022	0.294	0.112	0.046	0.337	0.111	0.023	0.053	0.002	0.000	0.000	0.000
6	0.004	0.157	0.085	0.051	0.347	0.154	0.052	0.136	0.010	0.000	0.002	0.000
7	0.000	0.037	0.070	0.044	0.294	0.182	0.087	0.198	0.071	0.004	0.013	0.000
8	0.000	0.009	0.015	0.035	0.203	0.164	0.119	0.244	0.123	0.044	0.042	0.002