Constrained Microaggregation: Adding Constraints for Data Editing

Vicenç Torra*

*IIIA - Artificial Intelligence Research Institute, CSIC - Spanish Council for Scientific Research,

Campus UAB s/n, 08193 Bellaterra (Catalonia, Spain)

E-mail: vtorra@iiia.csic.es

Abstract. Privacy preserving data mining and statistical disclosure control have introduced several methods for data perturbation that can be used for ensuring the privacy of data respondents. Such methods, as rank swapping and microaggregation, perturbate the data introducing some kind of noise. Nevertheless, it is usual that data are edited with care after collection to remove inconsistencies, and such perturbation might cause the introduction of new inconsistencies to them.

In this paper we study the development of methods for microaggregation that avoid the introduction of such inconsistencies. That is, methods that ensure the protected data to satisfy a set of given constraints.

Keywords. Privacy in statistical databases, privacy preserving data mining, statistical disclosure control, data edition, microaggregation.

1 Introduction

In recent years, several methods for ensuring data privacy have been developed. Data protection methods can be categorized into data-driven (or general purpose ones) and computation-driven (or specific purpose ones). That is, methods that can be used for publishing data independently of their posterior use, and methods developed for a particular data use (e.g., two parties want to mine association rules from their databases for market basket analysis). Computation-driven methods are usually based on cryptographic tools [30] (they follow the path of secure multi-party computation [31]) and data-driven ones on the perturbation of the data. That is, the latter methods consist of introducing some noise into the data so that the exact figures of a particular respondent are not disclosed. At present, a large number of perturbative methods exist. They are studied and compared in the areas of privacy preserving data mining and statistical disclosure control. Two of the most used and successful methods of the perturbative approach are rank swapping [16, 17] and microaggregation [7, 15, 19].

Data editing [14, 20] is a field of statistical disclosure control that is devoted to the analysis and correction of raw data for their improvement. The basic idea is that data should satisfy a set of requirements (or constraints) before their release. E.g. non negative values are not permitted for people's age. Data editing is typically applied to the original data and, in any case, before any perturbation takes place.

Perturbative methods have been developed so far without taking much into account the intrinstic properties of the data. That is, without taking into account the requirement or the constraints that the data elements have to satisfy. However, data after perturbation should maintain the constraints. E. g., it is not acceptable that a perturbative approach assigns negative values to the variable *age*, or that it assigns a value of 9000 miles to the variable *distance from household to work*. Naturally, it is always possible to edit again the data after data protection, but this can cause some additional difficulties. E.g., if the perturbated data maintains the mean of the original data, an assignment of a value equal to zero to all negative ages will affect the mean and, thus, the original mean is no longer maintained.

In this paper we study the case of microaggregation. We study how this method can be modified so that a set of requirements or constraints (edit constraints) are satisfied. We show that microaggregation can cope with some of the constraints in a natural manner.

In a few recent papers, the problem of combining protection mechanisms and editing constraints has been considered. See e.g. [24, 25]. In this paper we focus on microaggregation. As we will see here, this permits us to derive general rules on how microaggregation should be performed on real data to satisfy the constraints.

The structure of this paper is as follows. In Section 2 we review a few concepts that are needed in the rest of the paper. In particular, we will review a few elements of data editing and give an overview on some of the constraints that are into use. Although data editing is a field in statistical disclosure control, and although we will use the terminology in such field, the aspects outlined here are general to any database (e.g., data cleaning tasks and procedures are in common use and can be used for similar purposes). Section 2 also contains a short overview of microaggregation. In Section 3 we will present a few results about how microaggregation can take into account the (edit) constraints expressed on the data. The paper finishes with some conclusions and future work.

2 Preliminaries

In this section we review a few concepts that are needed in the rest of this paper. First, we will review some of the edit constraints in use. Then, we will review the main aspects of microaggregation, focusing on those that are relevant in this paper.

2.1 Edit Constraints

Data editing is the process of manipulating raw data to reduce the errors present and improve their quality. Several definitions exist each one strengthening different aspects of the process. For example, Pierzchala [20] distinguishes editing as either a validating procedure or a statistical process. Then, in the case of a validating procedure, edition is a "within-record action with the emphasis on detecting inconsistencies, impossibilities, and suspicious situations and correcting them. Examples of validation include: checking to see if the sum of parts adds up to the total, checking that the number of harvested acres is less than or equal to that of planted acres". In the case of editing as a statistical process, we have that "checks are based on a statistical analysis of respondent data". In this case, we might have, e.g., between-record checking. For example, we might be interested in the detection of outliers (through "edit limits generated from distributions of a subset of current records"). Time series have also been used for this purpose to detect inliers [8].

In addition to such edit constraints, there exists also macro-editing, which consists of checking aggregations of data. This is used when data is aggregated before its publica-

tion (e.g. to construct tabular data – following statistical disclosure control jargon – or data summaries). We will not consider this case here.

In any case, editing is usually formalized in terms of a set of edit constraints; that is, constraints that the data should satisfy. Once such constraints are written, it is possible to check the validity of the data by checking whether the data satisfy the constraints. Then, the data that violate the constraints can be corrected to solve the inconsistencies.

Several types of constraints can be found in the literature. We present a classification of a few of them below. Although the classification is biased towards our study, we consider it general enough for other purposes.

Constraints on the possible values. The values of a given variable are restricted to a predefined set. For example, the values should belong to a given interval, should be positive or bounded by a given value. This is the case of the variables *salary* and *age*.

$$\text{EC-PV: } age \in [0, 125]$$

Note that this constraint can be further generalized for subsets of variables. For example, constraining the values for a pair (v_1, v_2). Nevertheless, when such generalization takes place, it is possible to express the constraint in other terms. For example, when v_1 and v_2 correspond, respectively, to the probability of A and *not* A, we can require (v_1, v_2) to be in the set $\{(0, 1), (0.1, 0.9), \dots\}$ or, alternatively, use the contraint $v_1 + v_2 = 1$. Note that such latter expression for the constraint can also be classified as one of the constraints below (e.g. it can be either considered as a constraint of the type "one variable governs the possible values of another one", or as a "linear constraint").

One variable governs the possible values of another one. The values of a variable v_2 are constrained by the values of the first one. A simple case is the variable *sex* governing the variable *number of pregnacies*. It is clear that not all values are acceptable for *number of pregnacies* when *sex=male*. Formally,

$$\text{EC-GV1: If } sex{=}male \text{ THEN } number\ of\ pregnacies = 0$$

Another example is the constraint expressed in the following rule [25]:

$$\text{EC-GV2: IF } age < 17 \text{ THEN } gross\ income < mean\ income$$

In this case, three variables are taken into account *age*, *gross income* and *mean income*.

The example given above (from [20]) that the number of harvested acres should be less than or equal to that of planted acres is another example of this type of constraint. That is,

$$\text{EC-GV3: } harvested\ acres \leq planted\ acres$$

Linear constraints. Some numerical variables satisfy some linear constraints. That is, a variable can be expressed as a linear combination of a set of other variables. For example, in a data set about economical data the following equation involving variables *net*, *tax* and *gross* should hold:

$$\text{EC-LC1: } net + tax = gross$$

Non-linear constraints. In this case some relationships between variables hold, but they are not linear. For example, the following equation should be satisfied for the variables *applicable VAT Rate*, *price exc. VAT* and *retail price*:

EC-NLC1: *price exc. VAT* · (*1.00* + *applicable VAT Rate*) = *retail price*

Also, the following should hold with respect to *wage sum*, *hours paid for*, and *wage rate* (from [13]):

EC-NLC2: *wage sum* = *hours paid for* · *wage rate*

Other types of constraints. Other classes of constraints might be considered. For example, constraints on non-numerical variables (ordinal or categorical variables).

In the case of considering data editing in combination with a perturbative approach, we can consider an additional constraint.

Values are restricted to exist in the domain. In this case, not only the values should belong to a predefined set (as in the constraints on the possible values), but the values should really exist in the domain. For example, although we can consider that the variable *age* might have a range of $[0, 120]$, we require that the published data contains only records with ages really existing in the population or the sample.

In the case of considering this constraint with a perturbative approach, it means that if the original data includes only records about young people (e.g., *age* < 30), we cannot introduce error in the data and cause a record to have its *age* = 50.

Although this constraint is specially appropriate when perturbative methods are under consideration, it is also applicable when data editing is to be used on a file constrained by the data in another file [5, 23] (e.g. when linked files are edited). For example, the edition of a file with data from a school typically should be in agreement with the population data from the same town or neighborhood.

We give below an example that illustrates some edit constraints.

Example 1. Table 1 represents a file with data from 12 individuals that satisfies some edit constraints. The file is described in terms of variables $V_1, \ldots, V_7$, which stand for *Expenditure at 16%*, *Expenditure at 7%*, *Total Expenditure*, *Hours paid for*, *Wage rate*, *Wage sum*, *Total hours*.

The data is required to satisfy three edit constraints. First, the variables V_1, V_2, and V_3 are such that $V_3 = \alpha_1 V_1 + \alpha_2 V_2$. So, V_1, V_2, and V_3 define a linear constraint. Then, variables V_4, V_5, and V_6 satisfy the multiplicative constraint $V_6 = V_4 * V_5$, and, finally, V_4 and V_7 satisfy the inequality $V_4 \leq V_7$.

2.2 Microaggregation: An Overview

Microaggregation is a perturbative method. From an operational point of view, microaggregation applies first (i) a clustering algorithm to a set of data obtaining a set of clusters. Formally, the algorithm determines a partition of the original data. Then, microaggregation proceeds by calculating (ii) a cluster representative for each cluster (formally, a cluster

Exp 16%	*Exp 7%*	*Total*	*Hours paid for*	*Wage rate*	*Wage sum*	*Total hours*
V_1	V_2	V_3	V_4	V_5	V_6	V_7
15	23	42.01	23	50	1150	37
12	43	59.93	28	70	1960	37
64	229	319.27	12	84	1008	25
12	45	62.07	29	73	2117	30
28	39	74.21	9	30	270	40
71	102	191.5	10	63	630	20
23	64	95.16	9	74	666	10
25	102	138.14	72	30	2160	80
48	230	301.78	26	30	780	35
32	50	90.62	6	45	270	15
90	200	318.4	8	45	360	15
16	100	125.56	34	55	1870	45

Table 1: A data file with three edit constraint.

representative for each partition element). Finally, (iii) each original datum is replaced by the corresponding cluster representative.

Microaggregation is said to satisfy the requirements of privacy because each cluster is required to contain at least k records (where k is a parameter of the method). Then, each cluster representative aggregates (or summarizes) the information of at least k records. Due to this, in some sense, a microaggregated file satisfies k-anonymity [21, 27, 26].

In microaggregation, the parameter k also controls the level of perturbation. The larger the k, the larger the perturbation. This is so because for larger k the *degree* of summarization is larger. Naturally, the maximal summarization is when k corresponds to the size of the file and the protected file is useless.

It is said that microaggregation satisfies, in some sense, k-anonymity. Nevertheless, this is only partially true. There exist a few variations of microaggregation when a file contains several variables. E.g. univariate microaggregation and multivariate microaggregation. The former applies microaggregation to each variable in an independent manner. In this case, k-anonymity is not ensured because it might happen that all records in the protected file are different. Nevertheless, data is still perturbated because cluster representatives are computed for each variable and data is replaced by such cluster representatives. Multivariate microaggregation applies the clustering process to sets of variables. In this case, when all the variables are microaggregated together, k-anonymity will be satisfied. In this case, however, the perturbation suffered in the file might be large. Microaggregation for k-anonymity was studied in [12].

At present there exist several algorithms for microaggregation. See, for example [9, 18, 28] for details. The example below illustrates the application of microaggregation.

Example 2. Let us consider the data in Example 1. Table 2 illustrates the application of PCA microaggregation to this data file. In this example, we have applied microaggregation partitioning the data set into two sets of variables. One defined with variables $\{V_1, V_2, V_3\}$ and the other with variables $\{V_4, V_5, V_6, V_7\}$. In this example we can observe that the resulting file does not satisfy the multiplicative edit constraints for variables V_4, V_5, and V_6.

Computations have been done using the sdcMicro package [28] in **R**. The following commands were used (`zz` corresponds to the original file, and `zzNew` is the protected one):

Exp 16%	*Exp 7%*	*Total*	*Hours paid for*	*Wage rate*	*Wage sum*	*Total hours*
V_1	V_2	V_3	V_4	V_5	V_6	V_7
13.000	37.00	54.670	26.0000	51.000	1349.00	34.000
13.000	37.00	54.670	44.6667	51.667	1996.67	54.000
67.333	219.67	313.150	10.3333	59.000	636.00	28.333
13.000	37.00	54.670	26.0000	51.000	1349.00	34.000
27.667	51.00	86.663	10.3333	59.000	636.00	28.333
37.333	101.33	151.733	10.3333	59.000	636.00	28.333
27.667	51.00	86.663	7.6667	54.667	432.00	13.333
37.333	101.33	151.733	44.6667	51.667	1996.67	54.000
67.333	219.67	313.150	26.0000	51.000	1349.00	34.000
27.667	51.00	86.663	7.6667	54.667	432.00	13.333
67.333	219.67	313.150	7.6667	54.667	432.00	13.333
37.333	101.33	151.733	44.6667	51.667	1996.67	54.000

Table 2: The protected data file using PCA microaggregation.

```
res <- microaggregation(zz[,1:3], aggr=3, method="pca", measure="mean")
zzNew[,1:3] <- res$blowxm
res <- microaggregation(zz[,4:7], aggr=3, method="pca", measure="mean")
zzNew[,4:7] <- res$blowxm
```

As microaggregation does not satisfy k-anonymity (see e.g. the protected data file of Example 1) it is of interest the study of its disclosure risk, as well as of its information loss. Information loss and disclosure risk are two measures that give information about the extent in which the perturbated data is useful for the same purposes than the original one, and about the extent in which the perturbated data can lead to disclosure of sensitive information. In [10, 11], a large number of data protection methods were analysed with respect to information loss and disclosure risk. Microaggregation and rank swapping were ranked about the best of them.

In our opinion, microaggregation is specially suited when data has to satisfy constraints. The fact that microaggregation has to be computed for sets of variables, permits us to divide the set of variables according to the constraints, and then apply the microaggregation to each set separately. Then, as we will show latter, we can force the microaggregation to satisfy the given constraints. We detail in Section 3 how microaggregation can deal with the constraints described in Section 2.1. In Section 3, we will also revisit Example 1 to show how the results can be applied so that the protected file satisfies the constraints.

3 Microaggregation and the Edit Constraints

In this section we will show that microaggregation can cope easily with some of the constraints listed above. We will consider in successive sections the constraints included in Section 2.1, although not necessarily in the same order.

In the rest of this section we will use the following notation. We consider a data file with n records $x_1, \ldots, x_n$ that take values over a set of variables $V_1, \ldots, V_m$. We express the value for record x_i in variable V_j by $x_{i,j}$.

V	V_1	$\cdots$	V_K
x_1	$x_{1,1}$	$\ldots$	$x_{1,K}$
$\vdots$	$\vdots$		$\vdots$
x_N	$x_{N,1}$	$\ldots$	$x_{N,K}$
$\mathbb{C}(x_1,\ldots,x_N)$	$\mathbb{C}(x_{1,1},\ldots,x_{N,1})$	$\ldots$	$\mathbb{C}(x_{1,K},\ldots,x_{N,K})$

Table 3: Representation of the data to be microaggregated.

3.1 Linear Constraints

We will start our analysis with the linear constraints, presenting a few results that support our claim that microaggregation can be used effectively with such edit constraints. Although in general, we might have any variable j expressed in terms of a linear combination of a set of k variables $V_{K_1},\ldots,V_{K_k}$ (with $V_{K_i} \in \{V_1,\ldots,V_m\}$), for the sake of simplicity we will use a variable V expressed in terms of a linear combination of variables $V_1,\ldots,V_K$. This is illustrated in Table 3 where the first variable (column) is pressumed to be a linear combination of the other variables (column) in the same table. Naturally, $V \in \{V_1,\ldots,V_m\}$ and also $V_i \in \{V_1,\ldots,V_m\}$.

The suitability of microaggregation for dealing with linear constraints is based on the assumption that all the variables in the linear model are microaggregated together. That is, multivariate microaggregation is applied to a set of variables that includes $\{V,V_1,\ldots,V_K\}$. The results included in this section are based on this assumption. In addition, we will also pressume that the steps (i), (ii), and (iii) described in Section 2.2 for microaggregation can be separated and, thus, that we know the partition obtained by (i). Under this premise, the goal is to obtain in (iii) a cluster representative for each cluster (or partition element) that satisfies the linear constraint.

Table 3 represents the data of a single cluster. For the sake of simplicity, we only include here the variables with linear dependencies. Thus, the cluster is defined in terms of variables $V,V_1,\ldots,V_K$ and N records (the whole file has n records, instead). We also assume here that V is the dependent variable and that the linear constraint is of the form $V = \sum_{i=1}^{K} \alpha_i V_i$, for some values α_i and variables V_i. Naturally, we pressume here that the original data also satisfy the constraints (that is, the data were already edited). That is, the linear equation holds on the original data ($x_j = \sum_{i=1}^{K} \alpha_i x_{j,i}$ for all j).

Now we consider which are the methods suitable for computing the cluster representatives for this particular cluster. To do so, we need some notation first. We will assume that the cluster representative is a function of the data in the cluster. That is, the representative of the data in Table 3 is a function of such data. More specifically, we pressume that the representative of the variable V in that table is a function of the values of the records for V. Denoting the function by $\mathbb{C}$, this means that the value for the representative of V is $\mathbb{C}(x_1,\ldots,x_N)$. Similarly, the representative for variable V_i is $\mathbb{C}(x_{1,i},\ldots,x_{N,i})$. The last row of Table 3 includes these representatives.

Note that it would be possible to consider other functions not only depending on the values of the records in the cluster. E.g. we could consider the values in other clusters, or consider the values for other variables $V_j \neq V_i$ when computing the values for V_i. Due to this restriction on the function $\mathbb{C}$, $\mathbb{C}$ satisfies Arrow's condition of independence of irrelevant alternatives [3].

Under these definitions, when the editing rule requires linear dependence, it means that

the cluster center should satisfy the following equation:

$$\mathbb{C}(x_1, \ldots, x_N) = \sum_{i=1}^{K} \alpha_i \mathbb{C}(x_{1,i}, \ldots, x_{N,i})$$

In this equation, the values x_j on the left hand side can be expressed in terms of the $x_{j,i}$ on the right hand side. This is so because we have assumed that the original data is already edited and satisfy the edit constraints. Formally, we have that $x_j = \sum_{i=1}^{N} \alpha_i x_{j,i}$ for all j in $\{1, \ldots, N\}$. Including this information in the previous equation, we have that the following equality should hold:

$$\mathbb{C}(\sum_{i=1}^{K} \alpha_i x_{1,i}, \ldots, \sum_{i=1}^{K} \alpha_i x_{N,i}) = \sum_{i=1}^{K} \alpha_i \mathbb{C}(x_{1,i}, \ldots, x_{N,i})$$

In addition to this equality, we will require the function $\mathbb{C}$ to satisfy reflexivity. That is, $\mathbb{C}(x, \ldots, x) = x$. We require this property because it means that, when all the elements in the cluster have the same value, the function returns this very value. This condition seems natural in the context of microaggregation. Taking all this into account, we can establish and proof the following proposition that gives shape to the function $\mathbb{C}$. The proof, included in the appendix, is based on functional equations [1, 2] (see also [6, 29]).

Proposition 3. *Let $\mathbb{C}$ be a function satisfying*

$$\mathbb{C}(\sum_{i=1}^{K} \alpha_i x_{1,i}, \ldots, \sum_{i=1}^{K} \alpha_i x_{N,i}) = \sum_{i=1}^{K} \alpha_i \mathbb{C}(x_{1,i}, \ldots, x_{N,i}) \tag{1}$$

for given values $\alpha_1, \ldots, \alpha_K$ ($\alpha_i \neq 0$) and arbitrary values $x_{i,j}$ for $1 \leq i \leq N$ and $1 \leq j \leq K$, and reflexivity

$$\mathbb{C}(x, \ldots, x) = x$$

Then, the most general solution for $\mathbb{C}$ is a function of the form

$$\mathbb{C}(x_1, \ldots, x_N) = \sum_{i=1}^{N} \kappa_i x_i \tag{2}$$

for κ_i such that $\sum_{i=1}^{N} \kappa_i = 1$ but otherwise arbitrary.

This proposition characterizes the function $\mathbb{C}$. That is, this proposition implies that the only functions that can be applied in microaggregation, when clusters are known, are the ones described in Equation 2.

In the proposition above we have considered that the equation is only required for some particular α_i. Now we consider the case that the equation is valid for all arbitrary α_1, ..., α_K. The next proposition shows that this consideration does not change the set of possible functions $\mathbb{C}$.

Proposition 4. *The general solution for $\mathbb{C}$ when $\mathbb{C}$ satisfies reflexivity and Equation 1 for arbitrary $\alpha_1, \ldots, \alpha_K \neq 0$ is*

$$\mathbb{C}(x_1, \ldots, x_N) = \sum_{i=1}^{N} \kappa_i x_i$$

for κ_i such that $\sum_{i=1}^{N} \kappa_i = 1$ but otherwise arbitrary.

Proof. Now the equations have to hold for all α_i, therefore they also hold for particular α_i. E.g. it should hold for $\alpha_i = 1$ for all i. Then, applying Proposition 3 for such α_i we obtain the same solution above. As this result satisfies reflexivity and Equation 1, the proposition is proven. □

As stated above, in the step (i) of the microaggregation process, the application of the clustering method onto a data set results into a partition of the set. It is important to underline that the number of elements in each partition element is not (usually) known before hand, and that it might be different from one partition element to another. Note that although k is the minimum number of records possible, and most algorithms require the total number to be less than $2k$, nothing is said about the exact number of records in each cluster. Due to this, although the propositions establish that $\mathbb{C}$ is a linear combination of the data with respect to κ, it is difficult to define *a priori* which are the appropriate values of κ. Note that when each cluster has a different number of elements, we need a different number of κ_i in each cluster. In addition, it is clear that the order in which the records appear in the file (e.g. the order of the records in Table 3) should not be relevant in the final value of the cluster representative.

One way to solve these difficulties is to require the function $\mathbb{C}$ to be symmetric. This means that there are no *distinguished* records in the set. With a symmetric $\mathbb{C}$, any permutation of the data will lead to the same result. When we add this constraint to Propositions 3 and 4, we obtain that κ_i should be constant. This is stated in the next proposition.

Proposition 5. *When $\mathbb{C}$ is required to satisfy reflexivity, Equation 1 and symmetry, i.e.*

$$\mathbb{C}(x_1, \ldots, x_N) = \mathbb{C}(x_{\pi(1)}, \ldots, x_{\pi(N)})$$

for an arbitrary permutation π, then the most general solution for $\mathbb{C}$ is

$$\mathbb{C}(x_1, \ldots, x_N) = (1/N) \sum_{i=1}^{N} x_i \tag{3}$$

Note that this function $\mathbb{C}$ is the one used in Example 2 when computing the cluster representatives. As it can be observed in the protected data file in Table 2, variables V_1, V_2, and V_3 satisfy the linear constraints. Propositions 3, 4, and 5 prove that other functions different from the weighted mean would not be suitable for this purpose as the protected data set would not satisfy the linear constraints.

Another way to solve the difficulties described above is to define weights κ_i that depend on the record x. So, two records x_1 and x_2 such that $x_1 = x_2$ should lead to the same κ. I.e.,

$$\kappa_i(x_1) = \kappa_i(x_2) \text{ for all } x_1, x_2 \in X \text{ such that } x_1 = x_2.$$

However, note that in this case, the κ_i should be the same for all variables. The procedure to compute the centroid in the fuzzy c-means [4] and in most fuzzy clustering algorithms correspond to this latter approach. The procedure in fuzzy c-means is as follows: given records $x_1, \ldots, x_N$ with memberships to the cluster equal to $\mu_1, \ldots, \mu_N$, define $\kappa_i = \frac{(\mu_i)^m}{\sum_{k=1}^{n}(\mu_k)^m}$ and then use the function $\mathbb{C}$ of Propositions 3 and 4. As this construction satisfies the propositions, linear edit constraints will be satisfied. We do not go into details of this approach. For details on fuzzy clustering see e.g. [4].

3.2 Nonlinear Constraints

In the case of nonlinear constraints we can follow the same approach used above based on functional equations. In this case, the equations will be different, and thus the resulting functions $\mathbb{C}$ will be also different. We will consider below one case corresponding to multiplicative variables. Formally, we consider variables $V, V_1, \ldots, V_K$ satisfying $V = \prod_{i=1}^{K} V_i^{\alpha_i}$ for some $\alpha_i \neq 0$. Using the same analysis used in Section 3.1, we can conclude that the cluster representatives will satisfy:

$$\mathbb{C}(x_1, \ldots, x_N) = \prod_{i=1}^{K} \mathbb{C}(x_{1,i}, \ldots, x_{N,i})^{\alpha_i}$$

and, naturally, if the original data also satisfy this constraint (i.e., $x_j = \prod_{i=1}^{N} x_{j,i}^{\alpha_i}$), we have

$$\mathbb{C}(\prod_{i=1}^{K} x_{1,i}^{\alpha_i}, \ldots, \prod_{i=1}^{K} x_{N,i}^{\alpha_i}) = \prod_{i=1}^{K} \mathbb{C}(x_{1,i}, \ldots, x_{N,i})^{\alpha_i}$$

Note that the following example (presented in Section 2.1)

EC3-NLC1: *price exc. VAT* · (*1.00 + applicable VAT Rate*) = *retail price*

can be rewritten as a constraint of this type if we define a new variable $VATplus1 = 1.00 + applicableVATRate$, and we fix $\alpha_i = 1$. That is,

price exc. VAT · *VATplus1* = *retail price*

In the case of multiplicative variables, the following result can be established. The proof of this proposition is in the Appendix.

Proposition 6. *Let $\mathbb{C}$ be a function satisfying*

$$\mathbb{C}(\prod_{i=1}^{K} x_{1,i}^{\alpha_i}, \ldots, \prod_{i=1}^{K} x_{N,i}^{\alpha_i}) = \prod_{i=1}^{K} \mathbb{C}(x_{1,i}, \ldots, x_{N,i})^{\alpha_i} \tag{4}$$

for given values $\alpha_1, \ldots, \alpha_K$ ($\alpha_i \neq 0$) and arbitrary values $x_{i,j}$ for $1 \leq i \leq N$ and $1 \leq j \leq K$, and reflexivity

$$\mathbb{C}(x, \ldots, x) = x$$

Then, the most general solution for $\mathbb{C}$ is a function of the form

$$\mathbb{C}(x_1, \ldots, x_N) = \prod_{i=1}^{N} x_i^{\kappa_i}$$

for κ_i such that $\sum_{i=1}^{N} \kappa_i = 1$ but otherwise arbitrary.

In the case of multiplicative values, we can also prove two propositions that are the counterparts of Propositions 4 and 5. We state them below.

Proposition 7. *The general solution for $\mathbb{C}$ when $\mathbb{C}$ satisfies reflexivity and Equation 4 for arbitrary $\alpha_1, \ldots, \alpha_K$ is*

$$\mathbb{C}(x_1, \ldots, x_N) = \prod_{i=1}^{N} x_i^{\kappa_i}$$

for κ_i such that $\sum_{i=1}^{N} \kappa_i = 1$ but otherwise arbitrary.

Exp 16%	*Exp 7%*	*Total*	*Hours paid for*	*Wage rate*	*Wage sum*	*Total hours*
V_1	V_2	V_3	V_4	V_5	V_6	V_7
13.000	37.00	54.670	25.8841	47.84149	1238.3339	33.869
13.000	37.00	54.670	40.9251	48.69982	1993.0452	51.070
67.333	219.67	313.150	10.2598	54.14774	555.5480	27.144
13.000	37.00	54.670	25.8841	47.84149	1238.3339	33.869
27.667	51.00	86.663	10.2598	54.14774	555.5480	27.144
37.333	101.33	151.733	10.2598	54.14774	555.5480	27.144
27.667	51.00	86.663	7.5595	53.11521	401.5258	13.104
37.333	101.33	151.733	40.9251	48.69982	1993.0452	51.070
67.333	219.67	313.150	25.8841	47.84149	1238.3339	33.869
27.667	51.00	86.663	7.5595	53.11521	401.5258	13.104
67.333	219.67	313.150	7.5595	53.11521	401.5258	13.104
37.333	101.33	151.733	40.9251	48.69982	1993.0452	51.070

Table 4: The protected data file using PCA with constrained microaggregation.

Proposition 8. *When $\mathbb{C}$ is required to satisfy reflexivity, Equation 4 and symmetry, i.e.*

$$\mathbb{C}(x_1, \ldots, x_N) = \mathbb{C}(x_{\pi(1)}, \ldots, x_{\pi(N)})$$

for an arbitrary permutation π, then the most general solution for $\mathbb{C}$ is

$$\mathbb{C}(x_1, \ldots, x_N) = \prod_{i=1}^{N} x_i^{1/N} \tag{5}$$

On the light of this proposition, we reconsider now Example 2.

Example 9. Let us reconsider the data in Example 1 with the edit constraints. Then, the results of Proposition 5 and 8 imply that we should use Equation 3 for variables V_1, V_2, and V_3 involved in the linear constraints, and Equation 5 for variables V_4, V_5, and V_6 involved in the multiplicative constraints. Table 4 gives the results using these functions.

3.3 Constraints on the Possible Values.

When microaggregated data has to satisfy constraints on the values, and such constraints correspond to lower and upper boundary conditions, we can add an additional constraint in terms of the records in the set. We require the cluster representative to be in the interval defined between the minimum and the maximum of the elements in the cluster. Formally,

$$\min_i x_i \leq \mathbb{C}(x_1, \ldots, x_N) \leq \max_i x_i$$

This equation is known as internality. Note that if the constraint is that $x_i \in [a, b]$ for some a and b, it is clear that for edited data, we have $x_i \in [a, b]$, and, thus, this constraint implies that $\mathbb{C}(x_1, \ldots, x_N) \in [a, b]$. The following result establishes which functions $\mathbb{C}$ satisfy reflexivity, Equation 1, as well as internality.

Proposition 10. *When $\mathbb{C}$ is required to satisfy reflexivity, Equation 1, and internality, i.e.*

$$\min_i x_i \leq \mathbb{C}(x_1, \ldots, x_N) \leq \max_i x_i;$$

then, the most general solution for $\mathbb{C}$ is

$$\mathbb{C}(x_1, \ldots, x_N) = \sum_{i=1}^{N} \kappa_i x_i$$

for κ_i such that $\sum_{i=1}^{N} \kappa_i = 1$ and $\kappa_i \geq 0$ but otherwise arbitrary.

Proof. Using Proposition 3 (or 4), we have that $\mathbb{C}(x_1, \ldots, x_N) = \sum_{i=1}^{N} \kappa_i x_i$ with $\sum_{i=1}^{N} \kappa_i = 1$. Now, as $\mathbb{C}(1, 0, \ldots, 0) = \kappa_1$ should be between $[0, 1]$, it is clear that $\kappa_i \geq 0$. As the same applies to all other κ_i, the proposition is proven. □

This proposition is a *refinement* of Propositions 3 and 4. We consider below a refinement of Propositions 6 and 7. We do not consider any refinement of Propositions 5 and 8 because it can be easily seen that the functions $\mathbb{C}$ in such propositions already satisfy internality.

Proposition 11. *When $\mathbb{C}$ is required to satisfy reflexivity, Equation 4, and internality, i.e.*

$$\min_i x_i \leq \mathbb{C}(x_1, \ldots, x_N) \leq \max_i x_i;$$

then, the most general solution for $\mathbb{C}$ is

$$\mathbb{C}(x_1, \ldots, x_N) = \prod_{i=1}^{N} x_i^{\kappa_i}$$

for κ_i such that $\sum_{i=1}^{N} \kappa_i = 1$ and $\kappa_i \geq 0$ but otherwise arbitrary.

3.4 One Variable Governs the Possible Values of Another One

Microaggregation cannot deal easily with this type of constraints. They should be considered in a case by case basis. We analyze below each of the edit rules presented in Section 2.1 and we describe how to apply microaggregation so that the protected file satisfies such rules.

In any case, note that, in general any monotonic function $\mathbb{C}$ permits us to generate a protected file with $V_1 < V_2$ for variables V_1 and V_2 if in the original file it also holds $V_1 < V_2$. In fact, $x_{i,j} \leq x_{i,k}$ for all i and $j \neq k$ implies $\mathbb{C}(x_{1,j}, \ldots, x_{N,j}) \leq \mathbb{C}(x_{1,k}, \ldots, x_{N,k})$, corresponds to the monotonicity of $\mathbb{C}$. Note also that the functions in Propositions 5, 8, 10, and 11 are monotonic. In fact, the properties of monotonicity and reflexivity imply internality. Due to this, we might consider the replacement of internality by the properties of monotonicity in such propositions. The resulting functions $\mathbb{C}$ would be the same in this alternative case.

When all this is taken into account, it is clear that the constraint

EC-GV3: *harvested acres* $\leq$ *planted acres*,

is satisfied if we use a monotonic function $\mathbb{C}$.

In relation to Example 2, note that both functions $\mathbb{C}$ in Equations 3 and 5 are monotonic. So, the constraint $V_4 \leq V_7$ is satisfied in Examples 2 and 9.

In relation to the other two constraints

EC-GV1: If *sex=male* THEN *number of pregnacies = 0*

and

EC-GV2: IF *age* < 17 THEN *gross income* $<$ *mean income*,

the simplest way to ensure that the protected file satisfies them is to partition the data set into subsets, according to the antecedent in the rule, and then applying microaggregation separately to each subset (using monotonic functions $\mathbb{C}$. In the first rule, this is to partition the data set into two sets one with the records satisfying *sex=male* and the other with the records satisfying *sex=female*, and then to microaggregate the two sets separately. In the same way, for the second rule, we will partition the data into one set with the records with *age* < 17 and another with the records with *age* ≥ 17. Then, we will microaggregate these two sets independently. If we use multivariate microaggregation with a set of variables including the variables *gross income* and *mean income*, and we use a monotonic $\mathbb{C}$, the constraint will be satisfied.

This approach of partitioning the file is similar to the one presented in [25].

3.5 Values are Restricted to Exist in the Domain.

The functions $\mathbb{C}$ determined above do not ensure, in general, that the cluster representative is one of the data in the cluster. The only single case is in Proposition 3 and 4 when $\kappa_i = 1$ for a particular i and $\kappa_j = 0$ for all $j \neq i$. If we add this constraint to the conditions in the other propositions, the problem is overconstrained and no function $\mathbb{C}$ exists.

Having said that, in general, and when no other constraints are under consideration, the median is an appropriate operator for $\mathbb{C}$ as it satisfies this constraint. In fact, the median has already been used for microaggregation in [22]. Other operators might also be used in this setting. For example, order statistics and boolean max-min functions [29].

The median (as well as the order statistics) are monotonic functions. Due to this, they could also be applied in the case of constraints where one variable governs the possible values of another one. This monotonicity makes the median also suitable for constraints on the possible values.

4 Conclusions and Future Work

In this paper we have described the edit constraints, and discussed how microaggregation can cope with them. We have presented a few propositions that establish which are the functions to be used when protected data should satisfy the edit constraints.

Other types of constraints and the analysis of information loss for these types of functions $\mathbb{C}$ are left for future work. In relation to information loss and disclosure risk, once functions $\mathbb{C}$ are mathematically characterized, the interest relies on the selection of the parameters κ. Further work in this direction is needed.

5 Acknowledgements

Partial support by the Spanish MEC (projects eAEGIS – TSI2007-65406-C03-02, and ARES – CONSOLIDER INGENIO 2010 CSD2007-00004) is acknowledged.

The research described in this paper was partially carried out while the author was at the University of Tsukuba. Discussions with Junko Yamazoe (formerly at NSTAC - National Statistics Center, Japan) are acknowledged.

Appendix: Proofs

This appendix includes the proof of the two main propositions in this article: Propositions 3 and 6.

Proposition 3. *Let $\mathbb{C}$ be a function satisfying*

$$\mathbb{C}(\sum_{i=1}^{K}\alpha_i x_{1,i},\ldots,\sum_{i=1}^{K}\alpha_i x_{N,i}) = \sum_{i=1}^{K}\alpha_i \mathbb{C}(x_{1,i},\ldots,x_{N,i}) \tag{6}$$

for given values $\alpha_1,\ldots,\alpha_K$ ($\alpha_i \neq 0$) and arbitrary values $x_{i,j}$ for $1 \leq i \leq N$ and $1 \leq j \leq K$, and reflexivity

$$\mathbb{C}(x,\ldots,x) = x$$

Then, the most general solution for $\mathbb{C}$ is a function of the form

$$\mathbb{C}(x_1,\ldots,x_N) = \sum_{i=1}^{N}\kappa_i x_i$$

for κ_i such that $\sum_{i=1}^{N}\kappa_i = 1$ but otherwise arbitrary.

Proof. To prove this proposition, we will apply a standard technique used in solving functional equations [6, 29] that consists of considering several instantiations of Equation 6.

First, if the equation is valid for all $x_{i,j}$ it should also be valid for $x_{1,1} = x$ and $x_{i,j} = 0$ for all $(i,j) \neq (1,1)$. In this case, Equation 6 corresponds to

$$\mathbb{C}(\alpha_1 x_{1,1},0,\ldots,0) = \alpha_1\mathbb{C}(x_{1,1},0,\ldots,0) + \sum_{i=2}^{K}\alpha_i\mathbb{C}(0,\ldots,0),$$

that, due to the reflexivity of $\mathbb{C}$ (i.e, $\mathbb{C}(0,\ldots,0) = 0$), is equivalent to:

$$\mathbb{C}(\alpha_1 x_{1,1},0,\ldots,0) = \alpha_1\mathbb{C}(x_{1,1},0,\ldots,0)$$

As this equation is true for all $x_{1,1}$, it is also true when $x_{1,1}$ is defined as $x_{1,1} = x'_{1,1}/\alpha_1$ for the particular value α_i of the edit constraint, and an arbitrary $x'_{1,1}$. Therefore, we have

$$\mathbb{C}(\alpha_1 x'_{1,1}/\alpha_1,0,\ldots,0) = \alpha_1\mathbb{C}(x'_{1,1}/\alpha_1,0,\ldots,0).$$

This equation, through simplification and a change of variable, corresponds to:

$$\mathbb{C}(x,0,\ldots,0) = \alpha_1\mathbb{C}(x/\alpha_1,0,\ldots,0). \tag{7}$$

Note that this equation is true for all x because $x'_{1,1}$ is an arbitrary value.

In relation to this latter equation it should also be said that the selection of $x_{1,1}$ was arbitrary and, therefore, any other $x_{i,j}$ could be selected. As a consequence, this equation holds for all α_i in the set of $\{\alpha_1,\ldots,\alpha_K\}$ and for all components (or arguments) of $\mathbb{C}$. In a similar

way, defining $x_{i,1} = x_i$ and $x_{i,j} = 0$ for all $j > 1$ we obtain the following equation using a similar development

$$\mathbb{C}(x_1, \ldots, x_N) = \alpha_1 \mathbb{C}(x_1/\alpha_1, \ldots, x_N/\alpha_1), \tag{8}$$

As for Equation 7, this equation is also true for all α_i in $\{\alpha_1, \ldots, \alpha_K\}$.

Now, let us consider again Equation 6. As it is valid for all $x_{i,j}$ it is also valid for $x_{1,1} = x/\alpha_1$, $x_{1,2} = y/\alpha_2$ and $x_{i,j} = 0$ for all other (i, j) such that $(i, j) \neq (1, 1)$ and $(i, j) \neq (1, 2)$. In this case, Equation 6 leads to

$$\mathbb{C}(\alpha_1 x/\alpha_1 + \alpha_2 y/\alpha_2, 0, \ldots, 0) = \alpha_1 \mathbb{C}(x/\alpha_1, 0, \ldots, 0) + \alpha_2 \mathbb{C}(y/\alpha_2, 0, \ldots, 0).$$

That, of course, is equivalent to,

$$\mathbb{C}(x + y, 0, \ldots, 0) = \alpha_1 \mathbb{C}(x/\alpha_1, 0, \ldots, 0) + \alpha_2 \mathbb{C}(y/\alpha_2, 0, \ldots, 0)$$

that, using Equation 7, can be rewritten as:

$$\mathbb{C}(x + y, 0, \ldots, 0) = \mathbb{C}(x, 0, \ldots, 0) + \mathbb{C}(y, 0, \ldots, 0).$$

Now, denoting $\mathbb{C}(x, 0, \ldots, 0)$ by $\phi(x)$, we find that this equation corresponds to a Cauchy equation. That is, $\phi(x+y) = \phi(x)+\phi(y)$. This Cauchy equation is well known in functional equations and its solution is the function $\phi(x) = \kappa x$, for an arbitrary constant κ (see e.g. [1, 2, 29]).

The selection of $x_{1,1}$ and $x_{1,2}$ was arbitrary and any $x_{i,1}$ and $x_{i,2}$ could be selected as well. In this case we would have obtained the same equation with the values x and y in the ith component (instead of being in the first component as above). Therefore, in general, we obtain

$$\mathbb{C}(0, \ldots, 0, x + y, 0, \ldots, 0) = \mathbb{C}(0, \ldots, 0, x, 0, \ldots, 0) + \mathbb{C}(0, \ldots, 0, y, 0, \ldots, 0).$$

Let us define $\phi_i(x)$ the function equal to $\mathbb{C}(0, \ldots, 0, x, 0, \ldots, 0)$ when x is located in the ith component (and zero for all the other components). Then, the equation above corresponds to a Cauchy equation for ϕ_i. That is, $\phi_i(x + y) = \phi_i(x) + \phi_i(y)$. Naturally, we have for each equation the same solution given above. Nevertheless, as independent equations are formulated, the κ are not necessarily the same for all i. Thus, in general, κ is different for each ϕ_i. If we use κ_i as the constant for ϕ_i, we have $\phi_i(x) = \kappa_i x$.

Now, we have an expression for $\mathbb{C}(0, \ldots, 0, x + y, 0, \ldots, 0) = \phi_i(x) = \kappa_i x$. Nevertheless, the expression for $\mathbb{C}(x_1, \ldots, x_N)$ is still not ready. To obtain such expression and finish the proof we need to consider again $\mathbb{C}$. In this case, the goal is to decompose $\mathbb{C}(x_1, \ldots, x_N)$ into its different components. In particular, we need to show that

$$\mathbb{C}(x_1, \ldots, x_N) = \mathbb{C}(x_1, 0, \ldots, 0) + \mathbb{C}(0, x_2, 0, \ldots, 0) + \cdots + \mathbb{C}(0, \ldots, 0, x_N)$$

holds.

Defining $x_{1,1} = x_1/\alpha_1$, $x_{i,1} = 0$ for $i > 1$, $x_{1,2} = 0$ and $x_{i,2} = x_i/\alpha_2$ for $i > 1$, and $x_{i,j} = 0$ for all i and all $j > 2$, we can decompose the first component of $\mathbb{C}(x_1, \ldots, x_N)$. Formally, we can rewrite Equation 6 obtaining

$$\mathbb{C}(\alpha_1 x_1/\alpha_1, \alpha_2 x_2/\alpha_2, \ldots, \alpha_2 x_N/\alpha_2) =$$

$$\alpha_1 \mathbb{C}(x_1/\alpha_1, 0, \ldots, 0) + \alpha_2 \mathbb{C}(0, x_2/\alpha_2, \ldots, x_N/\alpha_2)$$

This equation is rewritten using Equations 7 and 8 and simplifying its left hand side into:

$$\mathbb{C}(x_1, x_2, \ldots, x_N) = \mathbb{C}(x_1, 0, \ldots, 0) + \mathbb{C}(0, x_2, \ldots, x_N)$$

Then, by induction, we obtain the desired decomposition of $\mathbb{C}$. That is,

$$\mathbb{C}(x_1, \ldots, x_N) = \mathbb{C}(x_1, 0, \ldots, 0) + \mathbb{C}(0, x_2, 0, \ldots, 0) + \cdots + \mathbb{C}(0, \ldots, 0, x_N)$$

Now, using the equivalence $\mathbb{C}(0, \ldots, 0, x, 0, \ldots, 0) = \kappa_i x$ proven above, it follows that the function $\mathbb{C}$ corresponds to

$$\mathbb{C}(x_1, \ldots, x_N) = \sum_{i=1}^{K} \kappa_i x_i. \tag{9}$$

Finally, we consider the requirement of reflexivity, $\mathbb{C}(x, \ldots, x) = x$. Formally, we have

$$x = \mathbb{C}(x, \ldots, x) = \sum_{i=1}^{K} \kappa_i x$$

So, $\sum_{i=1}^{K} \kappa_i = 1$.

Taking all into account, we have that $\mathbb{C}$ is of the form $\mathbb{C}(x_1, \ldots, x_N) = \sum_{i=1}^{K} \kappa_i x_i$ with $\sum_{i=1}^{K} \kappa_i = 1$, but otherwise arbitrary constants κ_i.

As this solution satisfies the conditions, this is the general solution of the problem and the proposition is proven. □

Proposition 6. *Let $\mathbb{C}$ be a function satisfying*

$$\mathbb{C}(\prod_{i=1}^{K} x_{1,i}^{\alpha_i}, \ldots, \prod_{i=1}^{K} x_{N,i}^{\alpha_i}) = \prod_{i=1}^{K} \mathbb{C}(x_{1,i}, \ldots, x_{N,i})^{\alpha_i} \tag{10}$$

for given values $\alpha_1, \ldots, \alpha_K$ ($\alpha_i \neq 0$) and arbitrary values $x_{i,j}$ for $1 \leq i \leq N$ and $1 \leq j \leq K$, and reflexivity

$$\mathbb{C}(x, \ldots, x) = x$$

Then, the most general solution for $\mathbb{C}$ is a function of the form

$$\mathbb{C}(x_1, \ldots, x_N) = \prod_{i=1}^{N} x_i^{\kappa_i}$$

for κ_i such that $\sum_{i=1}^{N} \kappa_i = 1$ but otherwise arbitrary.

Proof. To prove this proposition, we start considering Equation 10. Naturally, this equation holds for all $x_{r,s}$. Defining $z_{r,s} = \log x_{r,s}$, we can rewrite the equation in terms of $z_{r,s}$. We obtain

$$\mathbb{C}(\prod_{i=1}^{K} (e^{z_{1,i}})^{\alpha_i}, \ldots, \prod_{i=1}^{K} (e^{z_{N,i}})^{\alpha_i}) = \prod_{i=1}^{K} \mathbb{C}(e^{z_{1,i}}, \ldots, e^{z_{N,i}})^{\alpha_i}.$$

This equation is equivalent to

$$\mathbb{C}(\prod_{i=1}^{K}(e^{\alpha_i z_{1,i}}),\ldots,\prod_{i=1}^{K}(e^{\alpha_i z_{N,i}})) = \prod_{i=1}^{K}\mathbb{C}(e^{z_{1,i}},\ldots,e^{z_{N,i}})^{\alpha_i},$$

which can be rewritten as

$$\mathbb{C}(e^{\sum_{i=1}^{K}\alpha_i z_{1,i}},\ldots,e^{\sum_{i=1}^{K}\alpha_i z_{N,i}}) = \prod_{i=1}^{K}\mathbb{C}(e^{z_{1,i}},\ldots,e^{z_{N,i}})^{\alpha_i}.$$

Applying $\log$ to both sides of the equation, we obtain:

$$\log\mathbb{C}(e^{\sum_{i=1}^{K}\alpha_i z_{1,i}},\ldots,e^{\sum_{i=1}^{K}\alpha_i z_{N,i}}) = \log\prod_{i=1}^{K}\mathbb{C}(e^{z_{1,i}},\ldots,e^{z_{N,i}})^{\alpha_i},$$

which corresponds to

$$\log\mathbb{C}(e^{\sum_{i=1}^{K}\alpha_i z_{1,i}},\ldots,e^{\sum_{i=1}^{K}\alpha_i z_{N,i}}) = \sum_{i=1}^{K}\log\mathbb{C}(e^{z_{1,i}},\ldots,e^{z_{N,i}})^{\alpha_i}.$$

So, as $\log a^b = b\log a$, it is equivalent to

$$\log\mathbb{C}(e^{\sum_{i=1}^{K}\alpha_i z_{1,i}},\ldots,e^{\sum_{i=1}^{K}\alpha_i z_{N,i}}) = \sum_{i=1}^{K}\alpha_i\log\mathbb{C}(e^{z_{1,i}},\ldots,e^{z_{N,i}}).$$

Now, defining $\mathbb{C}'(x_1,\ldots,x_N) = \log\mathbb{C}(e^{x_1},\ldots,e^{x_N})$, we can rewrite the previous equation into the following one.

$$\mathbb{C}'(\sum_{i=1}^{K}\alpha_i z_{1,i},\ldots,\sum_{i=1}^{K}\alpha_i z_{N,i}) = \sum_{i=1}^{K}\alpha_i\mathbb{C}'(z_{1,i},\ldots,z_{N,i}).$$

Now, as this expression corresponds to Equation 1, Proposition 3 applies. Therefore, the solution for $\mathbb{C}'$ is $\mathbb{C}'(x_1,\ldots,x_N) = \sum_{i=1}^{K}\kappa_i x_i$ with $\sum_{i=1}^{K}\kappa_i = 1$ but otherwise κ arbitrary.
Now, from $\mathbb{C}'(x_1,\ldots,x_N) = \log\mathbb{C}(e^{x_1},\ldots,e^{x_N})$ with $y_i = e^{x_i}$ we have that

$$e^{\mathbb{C}'(\log y_1,\ldots,\log y_N)} = \mathbb{C}(y_1,\ldots,y_N).$$

So, as $\mathbb{C}'(x_1,\ldots,x_N) = \sum_{i=1}^{K}\kappa_i x_i$, the result corresponds to a function $\mathbb{C}$ as follows:

$$\mathbb{C}(y_1,\ldots,y_N) = e^{\sum_{i=1}^{K}\kappa_i\log y_i} = \prod_{i=1}^{K}x_i^{\kappa_i}$$

As this expression for $\mathbb{C}$ satisfies the conditions, this is the general solution of the problem and the proposition is proven. □

References

[1] Aczél, J. (1966) Lectures on Functional Equations and their Applications, Academic Press.

[2] Aczél, J. (1987) A Short Course on Functional Equations, D. Reidel Publishing Company (Kluwer Academic Publishers Group).

[3] Arrow, K. J. (1951) Social Choice and Individual Values, Wiley (2nd edition 1963).

[4] Bezdek, J. C. (1981) Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[5] Blum, O. (2005) Evaluation of editing and imputations supported by administrative records, Conference of European Statisticians, WP7.

[6] Castillo, E., Ruiz-Cobo, M. R. (1992) Functional Equations and Modelling in Science and Engineering, Marcel Dekker, Inc.

[7] Defays, D., Nanopoulos, P. (1993) Panels of enterprises and confidentiality: The small aggregates method, Proc. of the 1992 Symposium on Design and Analysis of Longitudinal Surveys, Statistics Canada, 195-204.

[8] Dinh, K. T. (1987) Application of Spectral Analysis to Editing a Large Data Base, Journal of Official Statistics 3:4 431-438.

[9] Domingo-Ferrer, J., Mateo-Sanz, J. M. (2002) Practical data-oriented microaggregation for statistical disclosure control, IEEE Trans. on Knowledge and Data Engineering 14:1 189-201.

[10] Domingo-Ferrer, J., Torra, V. (2001) Disclosure Control Methods and Information Loss for Microdata, in P. Doyle, J. I. Lane, J. J. M. Theeuwes, L. Zayatz (eds.) Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, Elsevier Science, 91-110.

[11] Domingo-Ferrer, J., Torra, V. (2001) A quantitative comparison of disclosure control methods for microdata, in P. Doyle, J. I. Lane, J. J. M. Theeuwes, L. Zayatz (eds.) Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies, North-Holland, 111-134.

[12] Domingo-Ferrer, J., Torra, V. (2005) Ordinal, Continuous and Heterogeneous k-Anonymity Through Microaggregation, Data Mining and Knowledge Discovery 11:2 195-212.

[13] Gasemyr, S. (2005) Editing and imputation for the creation of a linked micro file from base registers and other administrative data, Conference of European Statisticians, WP8.

[14] Granquist, L. (1997) The new view on editing, Int. Statistical Review 65:3 381-387.

[15] Hansen, S., Mukherjee, S. (2003) A Polynomial Algorithm for Optimal Univariate Microaggregation, IEEE Trans. on Knowledge and Data Engineering 15:4 1043-1044.

[16] Moore, R., (1996), Controlled data swapping techniques for masking public use microdata sets, U. S. Bureau of the Census (unpublished manuscript).

[17] Nin, J., Herranz, J., Torra, V. (2008) Rethinking Rank Swapping to Decrease Disclosure Risk, Data and Knowledge Engineering, 64:1 346-364.

[18] Nin, J., Herranz, J., Torra, V. (2008) On the Disclosure Risk of Multivariate Microaggregation, Data and Knowledge Engineering 67 399-412.

[19] Oganian, A., Domingo-Ferrer, J. (2000) On the Complexity of Optimal Microaggregation for Statistical Disclosure Control, Statistical J. United Nations Economic Commission for Europe, volume:18, number:4, 345-354.

[20] Pierzchala, M. (1994) A review of the state of the art in automated data editing and imputation, in Statistical Data Editing, Vol. 1, Conference of European Statisticians Statistical Standards and Studies N. 44, United Nations Statistical Commission and Economic Commission for Europe, 10-40.

[21] Samarati, P., Sweeney, L. (1998) Protecting privacy when disclosing information: k-anonymity

and its enforcement through generalization and suppression, SRI Intl. Tech. Rep.

[22] Sande, G. (2002) Exact and approximate methods for data directed microaggregation in one or more dimensions, Int. J. of Unc., Fuzz. and Knowledge Based Systems 10:5 459-476.

[23] Shlomo, N. (2006) Making use of alternate data sources, in Statistical Data Editing, Vol. 3: Impact on data quality, United Nations Statistical Commission and Economic Commission for Europe, 301.

[24] Shlomo, N., De Waal, T. (2005) Preserving edits when perturbing microdata for statistical disclosure control, Conference of European Statisticians, WP11.

[25] Shlomo, N., De Waal, T. (2008), Protection of micro-data subject to edit constraints against statistical disclosure, Journal of Official Statistics 24:2 229-253.

[26] Sweeney, L. (2002) Achieving k-anonymity privacy protection using generalization and suppression, Int. J. of Unc., Fuzz. and Knowledge Based Systems 10:5 571-588.

[27] Sweeney, L. (2002) k-anonymity: a model for protecting privacy, Int. J. of Unc., Fuzz. and Knowledge Based Systems 10:5 557-570.

[28] Templ, M. (2008) Statistical Disclosure Control for Microdata Using the R-Package sdcMicro, Transactions on Data Privacy 1:2 67 - 85.

[29] Torra, V., Narukawa, Y. (2007) Modeling decisions: information fusion and aggregation operators, Springer.

[30] Vaidya, J., Clifton, C., Zhu, M. (2006) Privacy Preserving Data Mining, Springer.

[31] Yao, A. C. (1982) Protocols for Secure Computations, Proc. of 23rd IEEE Symposium on Foundations of Computer Science, Chicago, Illinois, 160-164.