Fuzzy clustering-based microaggregation to achieve probabilistic $k$-anonymity for data with constraints

Vicenç Torra\textsuperscript{a,b,*}
\textsuperscript{a}Hamilton Institute, Maynooth University, Maynooth, Ireland
\textsuperscript{b}School of Informatics, University of Skövde, Skövde, Sweden

Abstract. Microaggregation is an effective data-driven protection method that permits us to achieve a good trade-off between disclosure risk and information loss.

In this work we propose a method for microaggregation based on fuzzy $c$-means, that is appropriate when there are constraints (linear constraints) on the variables that describe the data. Our method leads to results that satisfy these constraints even when the data to be masked do not satisfy them.

Keywords: Microaggregation, edit constraints, $k$-Anonymity, clustering statistical disclosure control, data privacy

1. Introduction

Microaggregation is an effective masking method for statistical disclosure control. Given a dataset $X$ it permits to build a masked data set $X' = \rho(X)$ with a good compromise between disclosure risk and information loss. Roughly speaking, microaggregation builds small clusters and replaces the original values by the centers of the clusters. In order to avoid disclosure, all clusters must have at least a predefined number of records. Then, in order to have an acceptable information loss it is usual to define a partition of the variables and microaggregate each variable independently.

The literature on microaggregation is vast. Microaggregation was proposed in [4]. Then, [5] defined an heuristic method, and [30] defined an approach for categorical data. Hansen and Mukherjee [9] defined a polynomial algorithm for univariate microaggregation. [21] proved that the problem is NP for multivariate microaggregation. Since then, different methods have been proposed to improve the effectiveness. See e.g. [13, 17, 18, 22, 24, 26]. The problem of selecting a good partition of the variables is discussed in [19]. Microaggregation to achieving $k$-anonymity [27] is discussed in [7].

The transparency principle (see e.g., [35]) establishes that when data is published we need to include information on all processes applied to the data. This naturally includes how data has been protected. When users know this information, they can use it to compute statistics from the data with better accuracy. Nevertheless, intruders can use this information to attack the data. Transparency attacks for microaggregation have been studied in [20, 36, 37]. It has been proven that very effective attacks can be built for multivariate microaggregation.

In order to define a microaggregation method resistant to transparency attacks, fuzzy microaggregation was introduced in [6]. The method builds first a fuzzy partition where all clusters had at least $k$-elements and then data is replaced by cluster centers. As in fuzzy clustering elements can belong to different clusters,
assignment to a cluster center was done at random. In this way, given a record intruders do not know for sure which cluster has been used in the replacement. This increases intruders uncertainty and decreases disclosure risk. Another fuzzy clustering based approach was proposed in [12] in terms of an optimization problem in line with [8]. In recent papers [33, 34] a simpler algorithm for fuzzy microaggregation was introduced.

Edit constraints correspond to restrictions established on the metadata, or the schema, of the database. They establish how some variables relate to each other. For example, we may have three variables in the database net, tax, gross and a constraint that specifies net + tax = gross. It is usual that data is edited before publication so that it satisfies the edit constraints. Nevertheless, when data is protected by means of a masking method it is possible that the resulting masked file violates the constraints. In this case the file has to be edited again but it may be not so easy this time. Note that if the masked file is such that means are the same as in the original file, and we have added random noise that has caused some ages to be negative, replacing these negatives values by zero would change the mean. The combination of edit constraints and masking methods has been studied by Shlomo and De Waal [28, 29], by Torra et al. [2, 3, 31], and later by Kim et al. [10, 11].

In this paper we study the problem of defining a method for microaggregation that, following [34] is resistant to transparency attacks and at the same time is able to deal with edit constraints. The approach is based on fuzzy c-means. We introduce here two variations of this algorithm to deal with linear constraints. The algorithms are able to build a masked data set that satisfies the linear constraints even in the case that the original data does not satisfy them. Note that this is an important property, as we can then combine in a single step effectively data edition and data masking. The structure of the paper is as follows. In Section 2 we review clustering and fuzzy c-means. In Section 3, we introduce two methods for achieving constrained fuzzy c-means and use them to define constrained microaggregation. We study the properties of the approaches. In Section 4 we give an example. The paper finishes with some conclusions and lines for future research.

2. Preliminaries

This section is divided in three parts and describes three topics that are needed later. We begin with a review of fuzzy clustering, which is used in our approach. Then we review how fuzzy clustering is used in microaggregation. Finally, we make a short summary of edit constraints and, more particularly, on linear constraints.

2.1. Fuzzy clustering

Clustering is an approach in statistics and machine learning (more specifically, in unsupervised machine learning) to extract relevant structures and patterns from data. There is a large number of methods for clustering that depend on different assumptions on the underlying model of the data and the type of structure built from the data (i.e., dendrograms, partitions, fuzzy partitions, ...). In this paper, following [34] we will use a fuzzy clustering algorithm.

Fuzzy clustering algorithms [1, 14, 15] typically build a fuzzy partition from the data. Recall that in a fuzzy set, membership to the set is partial instead of Boolean. This is usually represented by a membership function over the reference set \( u : X \rightarrow [0, 1] \) where \( u(x) = 0 \) means no membership and \( u(x) = 1 \) means total membership. Then, fuzzy clustering builds partitions in which elements may belong to more than one cluster with non-null membership. We will use fuzzy partitions in the sense of Ruspini [1, 25] which can be interpreted in terms of probability distributions because memberships of an element to all clusters add to one (this is not necessarily the case in other types of fuzzy partitions).

Fuzzy c-means [1] is one of the most well known algorithms of this type. This method is defined in terms of an objective function to be minimized. It is a generalization of k-means resulting into a fuzzy partition instead of a crisp (standard) one. This is achieved by means of replacing the objective function of k-means by another one where memberships are involved, and including a parameter \( m \) that controls the fuzziness of the solution. Entropy-based fuzzy c-means is another fuzzy clustering method. Fuzziness is achieved by means of requiring membership functions to optimize the entropy. In this case, a parameter \( \lambda \) is used to control the degree of fuzziness.

The formulation of fuzzy c-means [1] follows. In this formulation \( x_k \) for \( k = 1, \ldots, N \) represent the records to be clustered, \( u_{ki} \) the membership degree of record \( k \) to the \( i \)th cluster, and \( v_i \) represents the cluster center of the \( i \)th cluster. Fuzzy c-means finds \( u_{ki} \) and \( v_i \) given \( x_k \) and the parameter \( m \) solving the following optimization problem.
\[
\begin{align*}
\min J(u, v) &= \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ki})^m ||x_k - v_i||^2 \\
\text{subject to} \ &\sum_{i=1}^{c} u_{ki} = 1 \text{ for all } k = 1, \ldots, N \\
&u_{ki} \geq 0 \text{ for all } i = 1, \ldots, c \\
&\text{and for all } k = 1, \ldots, N
\end{align*}
\]

In this definition, \( m \) is such that \( m \geq 1 \). When \( m = 1 \) the problem is equivalent to the one of \( k \)-means and solutions are crisp (i.e., elements \( x_k \) are only assigned to one cluster and \( u_{ki} \in \{0, 1\} \) instead of being in \([0,1]\)). In contrast when \( m \) is large (e.g. \( m > 2.5 \)) solutions tend to be extremely fuzzy and membership values \( u_{ki} \) tend to be equal to \( 1/c \).

Fuzzy \( c \)-means is usually solved using an iterative algorithm that interlaces two optimization problems. One that optimizes \( u \) given cluster centers \( v_i \), and another that optimizes \( v \) given memberships \( u_{ki} \). That is, the following algorithm is used to find the membership functions \( u_{ki} \) and the cluster centers \( v_i \):

1. Generate \( c \) initial values for cluster centers \( v = (v_1, \ldots, v_c) \)
2. Calculate \( \hat{u} = \arg \min_u J(u, V) \)
3. Calculate \( \hat{v} = \arg \min_v J(U, v) \)
4. Iterate the last two steps until convergence.

The expressions for computing \( \hat{u} \) and \( \hat{v} \) in steps (2) and (3) above are given in the next proposition. See e.g. [1] for details.

**Proposition 1.** The alternate optimization algorithm for the fuzzy \( c \)-means uses the following expressions for computing the new \( u_{ki} \) and \( v_i \).

- The solution of \( \hat{u} = \arg \min_u J(u, v) \) given \( v \) is:
  \[
  u_{ki} = \left( \sum_{i=1}^{c} \left( \frac{||x_k - v_j||^2}{||x_k - v_i||^2} \right) \right)^{-1}
  \]
- The solution of \( \hat{v} = \arg \min_v J(u, v) \) given \( u \) is:
  \[
  v_{ik} = \frac{\sum_{k=1}^{N} (u_{ki})^m x_k}{\sum_{k=1}^{N} (u_{ki})^m}
  \]

This iterative approach converges to a local optima. It can be proven that the local optima obtained when \( m \) is large satisfies the following properties.

**Proposition 2.** For large values of \( m \), the iterative algorithm for fuzzy \( c \)-means with the equations in Proposition 1 leads to

- memberships \( u_{ij} = 1/c \) for all objects \( x_j \in X \) and clusters \( j \), and
- cluster centers \( v_i = v_j = \bar{X} \) for all clusters \( j \)

There have been several extensions of this method as well as alternative approaches for defining fuzzy clustering. One of them is entropy-based fuzzy \( c \)-means (EFCM) [16], which is defined in a way similar to the one of fuzzy \( c \)-means. The function to be minimized is a regularized version of \( \sum \sum u_{ij}||x_k - v_{ij}||^2 \). Instead of adding \( m \) as in fuzzy \( c \)-means, a term based on the entropy is introduced in the optimization problem. This term is combined using a parameter \( \lambda \). This parameter \( \lambda \geq 0 \) is used to control the fuzziness of the solution. Formally, the entropy-based fuzzy \( c \)-means is defined in terms of the following optimization problem.

\[
J_{EFCM}(u, v) = \sum_{i=1}^{c} \sum_{j=1}^{N} [u_{ij}||x_k - v_{ij}||^2 + \lambda^{-1} u_{ki} \log u_{ki}]
\]

This objective function is subject to the constraints \( u_{ki} \in \{0, 1\} \) and \( \sum_{i=1}^{c} u_{ki} = 1 \) for all \( k \), as in the case of fuzzy \( c \)-means.

EFCM is also solved using an iterative algorithm. The expressions for \( u \) and \( v \) are as follows (see [16] for details):

\[
\begin{align*}
 u_{ki} &= \frac{e^{-\lambda||x_k - v_i||^2}}{\sum_{j=1}^{c} e^{-\lambda||x_k - v_j||^2}} \\
 v_i &= \frac{\sum_{k=1}^{N} u_{ki} x_k}{\sum_{k=1}^{N} u_{ki}}
\end{align*}
\]

The parameter \( \lambda \) plays a role similar to \( m \) in fuzzy \( c \)-means. Here, the smaller the \( \lambda \), the fuzzier the solution.

**Proposition 3.** When \( \lambda \) tends to zero, the iterative algorithm for EFCM using Equations 3 and 4 leads to

- memberships \( u_{ij} = 1/c \) for all objects \( x_j \in X \) and clusters \( j \), and
- cluster centers \( v_i = v_j = \bar{X} \) for all clusters \( j \)

When \( \lambda \) tends to infinity, the second term becomes negligible and the algorithm yields to a crisp solution.

FCM and EFCM lead to different clusters for the same data. An important difference is about the membership values. For example, the centroids have a membership equal to one in the FCM while in the EFCM a lower membership is possible.
2.2. Fuzzy microaggregation for data privacy

Microaggregation is about building small clusters (with at least \( k \) records) and replace each record in the set by the cluster center. When the whole set is microaggregated at once (considering all the variables at the same time), the resulting file is such that there are sets of at least \( k \) indistinguishable records. Note that all records that belong to the same cluster are replaced by the same cluster center.

As processing the file in this way causes a high information loss, it is usual to consider a partition of the variables, and apply microaggregation to each subset of variables. Proceeding in this way, records that are indistinguishable with respect to one set of variables (because they belong to the same cluster for these variables) may be distinguishable with respect to another set (because they belong to different clusters for these other variables). This causes that the protected file has no longer sets of \( k \) indistinguishable records.

Nevertheless, this approach can be attacked effectively (see e.g. [20]). Any intruder can exploit the fact that records are usually masked replacing values by cluster centers that are near. Attacks are specially effective for optimal univariate microaggregation as in this case we know with certainty which clusters can be used for replacement.

To avoid this type of (transparency) attacks, we proposed in [34] an approach for microaggregation based on fuzzy microaggregation. Algorithm 1 reproduces this method. The approach is based on computing fuzzy clusters and then replacing values by cluster centers using the membership values as a probability distribution. In this way, intruders cannot know which clusters have been used. The algorithm uses two parameters \( m_1 \) and \( m_2 \) in relation to fuzzy \( c \)-means. The first one is to compute the fuzzy clusters, and the second one to build the probability distribution.

It can be proven that the larger the \( m_1 \), the larger the information loss. Similarly, the larger the \( m_2 \), the larger the information loss. In addition, we can also prove that for large values of \( m_2 \), and with \( c = |X|/k \), the expected size of all clusters is \( k \). Therefore, for large values of \( m_2 \) when microaggregation is applied to the whole file, we have probabilistic \( k \)-anonymity.

Algorithm 1. Fuzzy Microaggregation with parameters \( c, m_1 \), and \( m_2 \)

**Step 1:** Apply fuzzy \( c \)-means with a given \( c \) and a given \( m := m_1 \).

**Step 2:** For each \( x_j \) in \( X \), compute memberships \( u \) to all clusters in \( i = 1, \ldots, c \) for a given \( m_2 \). Use:

\[
u_{ij} = \left( \sum_{r=1}^{c} \left( \frac{||x_j - v_r||^2}{||x_j - v_r||^2} \right) \frac{1}{m_2^1} \right)^{-1}
\]

**Step 3:** For each \( x_j \) determine a random value \( \chi \) in \([0,1]\), and assign \( x_j \) to the \( i \)th cluster using the probability distribution \( u_{ij} \)

(i.e., the membership values computed in Step 2).

2.3. Constraints and linear constraints

A problem so far not much considered in the literature in data privacy is the one of data with constraints. Up to our knowledge, only the three groups mentioned in the introduction (Shlomo and De Waal, Kim et al., and ourselves) have considered this problem.

With data with constraints we refer to the fact that some databases include metadata expressing that some of the variables in the data sets should satisfy some constraints. Some of these constraints are enforced when data is introduced, and others are checked once all the data are available.

When data are processed for ensuring privacy, data is modified, and if such constraints are not taken into consideration, incompatible data are generated. The usual approach is to proceed in this way: (i) protect the dataset ignoring the constraints; then (ii) check the constraints, and finally (iii) if constraints are violated, data is modified again so that it is compliant with the constraints.

Such approach can lead to inconsistencies or larger information loss than necessary. Recall the example in the introduction. Protection with random noise is known to keep means. However, if variables are required to be positive, just transforming to zero all negative values can cause a relevant positive bias to the data. In addition, when data is first edited, and then protected we modify the original data twice. This adds extra noise to the data.

The results introduced in this paper are to edit and protect in a single step. In this way, the noise introduced into the data for data protection is used at the same time to solve the inconsistencies of the data with respect to the constraints.

Constraints on the variables are known by edit constraints in the field of data privacy (in particular, in the subfield of statistical disclosure control). There are different types of constraints. E.g., constraints on the possible values, linear constraints, one variable
that governs the values of another (e.g., sex= male, implies number of pregnancies=0). See [23, 28, 31]
for details on the classification of the constraints. In this paper, we will consider linear constraints. Note
that some other types of equality constraints can be transformed into linear ones. Naturally, we have a
linear constraint when a variable can be expressed as a linear combination of a set of other variables.
For example, the following relation between family income, person income, and other persons income
should hold:

EC-LC: person income + other person income = family income

In general, linear constraints can be expressed in its more general form as \( V = \sum \alpha_i V_i + A \), for some values \( \alpha_i \), variables \( V_i \), the dependent variable \( V \) and a constant \( A \). In this paper, we will rewrite them, equivalently, as \( \sum \alpha_i V_i = A \), or, in a vector form with \( v = (V_1, \ldots, V_l) \) and \( \alpha = (\alpha_1, \ldots, \alpha_t) \), as \( \alpha \cdot v = A \). Here \( \cdot \) represents the inner product.

3. Constrained fuzzy clustering for data masking

As our goal is that the masked data satisfies the linear constraints (independently of whether the original
data satisfies them or not), we investigate in this section clustering algorithms that lead to cluster centers
that satisfy linear constraints.

We use again \( x_k \) for \( k = 1, \ldots, N \) to represent the set of elements to be clustered. Each element \( x_k \)
belongs to a \( t \) dimensional space (i.e., \( x_k \in \mathbb{R}^t \)). Elements \( x \) will be clustered in \( c \) different clusters. Then,
we need to consider cluster centers \( u_i \) for \( i = 1, \ldots, c \) with \( v_i \) also represented in a \( t \) dimensional space. For
both \( x_k \) and \( v_i \) we will use here a second subindex to express its \( s \)th component. In other words, \( x_{kis} \)
represents the \( s \)th component of the \( k \)th object and \( v_{is} \) represents the \( s \)th component of the \( i \)th cluster center.

As stated above, linear constraints on the cluster centers are represented in terms of a vector \( \alpha =
(\alpha_1, \ldots, \alpha_t) \) and the inner product \( \cdot \), by means of the equation \( \alpha \cdot v_i = A \) for all clusters \( i = 1, \ldots, c \). Of
course, this means that, \( \sum_{s=1}^{t} \alpha_s v_{is} = A \).

3.1. Constrained FCM

We start considering the formalization of fuzzy \( c \)-means when we require the cluster centers to satisfy
linear constraints. Therefore, using the notation above, the constrained fuzzy \( c \)-means with linear constraints
corresponds to the following problem.

\[
\min CF(u, v) = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ki})^m ||x_k - v_i||^2
\]
subject to \( \sum_{i=1}^{c} u_{ki} = 1 \) for all \( k = 1, \ldots, N \)
\( \alpha \cdot v_i = A \) for all \( i = 1, \ldots, c \)
\( u_{ki} \geq 0 \) for all \( i = 1, \ldots, c \)
and for all \( k = 1, \ldots, N \).

Note that this problem is the same we saw before for the fuzzy \( c \)-means adding linear constraints for
each cluster center. To solve this optimization problem we can apply the alternative algorithm problem
using new expressions for \( u \) and \( v \). The following proposition gives these expressions.

**Proposition 4.** The alternate optimization algorithm for the fuzzy \( c \)-means with linear constraints will use
the following expressions for computing \( u_{ki} \) and \( v_{is} \):

- The solution of \( \hat{u} = \arg \min_u CF(u, v) \) given \( v \) is:
  \[
u_{ki} = \left( \left( \sum_{i=1}^{c} ||x_k - v_i||^2 \right)^\frac{1}{m-1} \right)^{-1}
\]

- The solution of \( \hat{v} = \arg \min_v CF(u, v) \) given \( u \) is:
  \[
v_{is} = \frac{\sum_{k=1}^{N} (u_{ki})^m x_{kis} - \sum_{r=1}^{c} \left( \sum_{s=1}^{t} \alpha_s v_{rjs} - A \right) \sum_{k=1}^{N} (u_{ki})^m \alpha_s x_{rjs}}{\sum_{k=1}^{N} (u_{ki})^m \alpha_s}
\]

We now present the proof of this proposition. We considered a simpler version of this problem in [32].

**Proof.** The solution of this problem is based on the Lagrange multipliers. We consider two sets of
Lagrange multipliers. One set are for the constraints \( \sum_{i=1}^{c} u_{ki} = 1 \) and the other set for the constraints
\( \alpha \cdot v_i = A \). These multipliers are called, respectively, \( \lambda_k \) and \( \lambda_i \). Then, we have that the expression to be
minimized is:

\[
L = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ki})^m ||x_k - v_i||^2 \\
+ \sum_{k=1}^{N} \lambda_k (\sum_{i=1}^{c} u_{ki} - 1) \\
+ \sum_{i=1}^{c} v_i (\alpha v_i - A)
\]

In order to compute an expression for \( u_{ki} \), let us now consider the derivative of \( L \) with respect to \( u_{ki} \). We obtain
\[
\frac{\partial L}{\partial h_{ki}} = m(u_{ki})^{m-1} ||x_k - v_i||^2 + \lambda_k.
\]

Making this expression equal to zero, and assuming that there is no \(x_k = v_i\), we can obtain the following expression for \(u_{kj}\) (for convenience we use \(j\) instead of \(i\)).

\[
u_{kj} = \left( \frac{-\lambda_k}{m ||x_k - v_j||^2} \right)^{\frac{1}{m-1}}.
\]

In order to eliminate \(\lambda_k\) in this expression, let us consider the constraint \(\sum_{i=1}^c u_{ki} = 1\). Replacing \(u_{kj}\) by Equation 6 we obtain:

\[
1 = \sum_{i=1}^c u_{ki} = \sum_{i=1}^c \left( \frac{-\lambda_k}{m ||x_k - v_i||^2} \right)^{\frac{1}{m-1}}
= (-\lambda_k)^{\frac{1}{m-1}} \sum_{i=1}^c \left( \frac{1}{m ||x_k - v_i||^2} \right)^{\frac{1}{m-1}}
= \left( \sum_{i=1}^c \left( \frac{||x_k - v_i||^2}{||x_k - v_i||^2} \right)^{\frac{1}{m-1}} \right)^{-1}.
\]

Now, if we compute the quotient between Equation 6 and 7 we obtain:

\[
u_{kj} = \left( \frac{-\lambda_k}{m ||x_k - v_j||^2} \right)^{\frac{1}{m-1}} \left( \sum_{i=1}^c \left( \frac{1}{m ||x_k - v_i||^2} \right)^{\frac{1}{m-1}} \right)^{-1}.
\]

This solution is a valid solution because it satisfies \(u_{kj} \geq 0\). As the objective function is convex, this is the unique solution of the problem.

We now consider the derivative of \(L\) with respect to \(v_k\) in order to compute an expression for the later. As \(v_k\) is a vector, we consider one of its components: \(v_{is}\).

We decompose \(L\) above in \(L_1 + L_2 + L_3\) and make explicit \(v_{is}\) in these components:

\[
L_1 = \sum_{i=1}^c \sum_{k=1}^N (u_{ki})^m ||x_k - v_i||^2
= \sum_{i=1}^c \sum_{k=1}^N \sum_{s=1}^t (u_{ki})^m (x_{ks} - v_{is})^2
L_2 = \sum_{k=1}^N \lambda_k \left( \sum_{i=1}^c u_{ki} - 1 \right)
\]

and

\[
L_3 = \sum_{i=1}^c v_i (\alpha v_i - A) = \sum_{i=1}^c v_i \sum_{s=1}^t \alpha_s v_{is} - \sum_{i=1}^c v_i A
= \sum_{k=1}^N (u_{ki})^m 2(x_{ks} - v_{is})(-1) + 0 + v_i \alpha_s.
\]

It is easy to see that the derivative of \(L_1\) with respect to \(v_{is}\) is \(\sum_{k=1}^N (u_{ki})^m 2(x_{ks} - v_{is})(-1)\), the derivative of \(L_2\) with respect to \(v_{is}\) is zero, and the one of \(L_3\) with respect to \(v_{is}\) is \(v_i \alpha_s\). Therefore, it follows that

\[
\frac{\partial L}{\partial v_{is}} = \sum_{k=1}^N (u_{ki})^m 2(x_{ks} - v_{is})(-1) + v_i \alpha_s.
\]

If we assign this expression to zero we obtain,

\[
0 = \sum_{k=1}^N (u_{ki})^m 2(x_{ks} - v_{is})(-1) + v_i \alpha_s.
\]

Therefore, we can say that \(v_{is}\) should be as follows.

\[
v_{is} = \frac{\sum_{k=1}^N (u_{ki})^m 2 x_{ks} - v_i \alpha_s}{\sum_{k=1}^N (u_{ki})^m 2}.
\]

We study the case of \(\alpha_s \neq 0\) (otherwise the variable \(s\) is not in the linear constraint). In this case, let us consider the equation \(A = \alpha \cdot v_i\), which corresponds to

\[
A = \sum_{s=1}^t \alpha_s v_{is} = \sum_{s=1}^t \alpha_s \frac{\sum_{k=1}^N (u_{ki})^m 2 x_{ks} - v_i \alpha_s}{\sum_{k=1}^N (u_{ki})^m 2}
\]

If the \(i\)th cluster contains at least one element with some non-null membership, we can obtain by algebraic transformation the following expression for \(v_i\):

\[
v_i = \frac{\sum_{s=1}^t \alpha_s (\sum_{k=1}^N (u_{ki})^m 2 x_{ks}) - 2 \sum_{k=1}^N (u_{ki})^m A}{\sum_{s=1}^t \alpha_s \alpha_s}.
\]

We use this expression for \(v_i\) in Expression 9, and then with further algebraic manipulation we obtain Equation 5 in the proposition. □

Note that for variables \(v_s\) not involved in the linear constraint (i.e., variables \(v_s\) such that \(\alpha_s = 0\), \(v_{is}\) will be computed in the same way as for the standard fuzzy \(c\)-means. This is so because Equation 5 reduces to \(v_{is} = \sum_{k=1}^N (u_{ki})^m x_{ks} / \sum_{k=1}^N (u_{ki})^m\) (Equation 1). Because of this result, we do not need to consider separately those variables in a database that satisfy linear constraints and those that do not take part of such type of constraint.

This algorithm will produce a set of clusters that satisfy the linear constraints. We can also prove that when the data satisfies these constraints, the expressions for \(v_{is}\) above reduce to the ones of standard fuzzy \(c\)-means (Equation 1).
These two properties are established in the following proposition.

**Proposition 5.** The solution of Proposition 4 is such that

1. When \( \alpha_s = 0 \) (i.e., variable \( v_s \) is not involved in the linear constraint), Equation 5 reduces to Equation 1 for all \( i = 1, \ldots, c \).
2. When data satisfy linear constraints, Equation 5 reduces to Equation 1 for all \( i = 1, \ldots, c \) and all \( s = 1, \ldots, t \).

**Proof.** To prove 1 it is enough to replace \( \alpha_s \) by zero in Equation 5. To prove 2 observe that when data satisfy the linear constraints, then for all \( k = 1, \ldots, N \) it holds

\[
\sum_{r=1}^{t} \alpha_r x_{kr} - A,
\]

so, Equation 5 reduces to Equation 1.

\[ \square \]

3.2. Constrained EFMC

We focus now on entropy-based fuzzy c-means. We will obtain similar results. Considering the linear constraints for all cluster centers, we define the following optimization problem.

\[
\min CE(u, v) = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ki} ||x_k - v_i||^2 + \lambda^{-1} \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ki} \log u_{ki}
\]

subject to

- \( \alpha \cdot v_i = A \) for all \( i = 1, \ldots, c \)
- \( u_{ki} \geq 0 \) for all \( i = 1, \ldots, c \) and for all \( k = 1, \ldots, N \)

For this problem, the following result is obtained.

**Proposition 6.** The alternate optimization algorithm for this optimization problem leads to the following two expressions.

\[
u_{ki} = \frac{e^{-\lambda ||x_k - v_i||^2}}{\sum_{j=1}^{c} e^{-\lambda ||x_k - v_j||^2}} \tag{10}
\]

\[
v_{is} = \frac{\sum_{k=1}^{N} u_{ki} x_{ks} - \alpha_s \sum_{r=1}^{t} \left( \sum_{k=1}^{N} u_{ki} (\alpha_r x_{kr} - A) \right)}{\sum_{k=1}^{N} u_{ki}} \tag{11}
\]

**Proof.** The proof of this proposition follows the same schema as the case of the fuzzy c-means. That is, we use the Lagrange multipliers to define an objective function that includes subexpressions for each constraint.

\[
L = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ki} ||x_k - v_i||^2 + \lambda^{-1} \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ki} \log u_{ki}
\]

\[
+ \sum_{k=1}^{c} \sum_{i=1}^{c} \lambda_k \left( \sum_{k=1}^{N} u_{ki} - 1 \right)
\]

\[
+ \sum_{i=1}^{c} v_i (\alpha \cdot v_i - A)
\]

Derivating \( L \) with respect to \( u_{ki} \) and making it equal to zero, we can later obtain the following expression for \( u_{ki} \):

\[
u_{ki} = e^{-1-\lambda \lambda_k - \lambda ||x_k - v_i||^2}.
\]

Then, as \( \sum_{j} u_{kj} = 1 \), we can compute

\[
u_{ki} = \frac{\sum_{j} u_{kj}}{\sum_{j=1}^{c} e^{-1-\lambda \lambda_k - \lambda ||x_k - v_j||^2}} = \frac{e^{-\lambda ||x_k - v_i||^2}}{\sum_{j=1}^{c} e^{-\lambda ||x_k - v_j||^2}}
\]

which corresponds to the expression for \( u_{ki} \).

The proof of the expression for \( v_{is} \) starts derivating \( L \) with respect to \( v_{is} \). Then, when this derivative is zero we can obtain the following expression for \( v_{is} \):

\[
v_{is} = \frac{2 \sum_{k} u_{ki} x_{ks} - v_i \alpha_s}{2 \sum_{k} u_{ki}}.
\]

On the other hand, replacing this expression for \( v_{is} \) in the equation \( \sum_{s=1}^{t} \alpha_s v_{is} = A \) we obtain:

\[
\sum_{s=1}^{t} \alpha_s \frac{2 \sum_{k} u_{ki} x_{ks} - v_i \alpha_s}{2 \sum_{k} u_{ki}} = A
\]

from which we get the following expression for \( v_i \):

\[
v_i = \frac{2 \sum_{r=1}^{t} \alpha_r \sum_{k} u_{ki} x_{kr} - 2 A \sum_{k} u_{ki}}{2 \sum_{r=1}^{t} \alpha_r \alpha_r}
\]

Replacing this expression for \( v_i \) in \( v_{is} \) we have proven Equation 11.

\[ \square \]
Note that the expression for $u_{kl}$ is the same we had before when no linear constraints were considered. In contrast, the expression for $v_{iis}$ is different as it includes terms corresponding to the constraints.

The next proposition establishes that the solution given in Proposition 6 corresponds to the standard solution in EFCM when data already satisfies the constraints.

**Proposition 7.** When the original data satisfy the linear constraints $\alpha \cdot v_1 = A$, the alternate optimization algorithm for entropy-based FCM reduces to the one of standard fuzzy c-means.

### 3.3. Constrained microaggregation

The application of these algorithms to microaggregation consists in replacing in Algorithm 1 the fuzzy c-means in Step 1 by one of the two clustering methods discussed in this paper (constrained FCM and constrained EFCM). They will lead to solutions that satisfy the linear constraints. Algorithm 1 will have two parameters $m_1$ and $m_2$ for the constrained FCM, and two parameters $\lambda_1$ and $\lambda_2$ for the constrained EFCM.

**Proposition 8.** Constrained FCM applied according to Algorithm 1 satisfies the following properties:

- The larger the $m_1$, the larger the information loss. This follows from Proposition 2. Clusters will collide, cluster centers will be $v_i = \bar{X}$, and all protected data will be $v_i$.
- The larger the $m_2$, the larger the information loss. Proposition 2 shows that in this case all memberships tend to be $1/c$. Therefore, any cluster center can be used to replace a given value.
- The smaller the number of clusters $c$, the larger the information loss. Naturally, when $c = 1$, all data are replaced by a single cluster center. Minimum loss can be achieved when $c = |X|$ and each data has its own cluster (and with $m_1$ and $m_2$ near to one).
- For large $m_2$, and $c = |X|/k$ for a given $k$, the expected size of all clusters is $k$. I.e., we have probabilistic $k$-anonymity. This follows from the uniform distribution with membership/probability $1/c$, and the number of clusters.

These properties show that with an appropriate selection of $m_1$ and $m_2$ information loss ranges from no information loss (i.e., $c = |X|, m_1 = m_2 = 1$ with the masked data being equal to the original one) to maximum loss (i.e., $c = 1$). We can also obtain data that probabilistically satisfies $k$-anonymity (i.e., $c = |X|/k$ and $m_2$ large). In addition to these properties, the resulting file satisfies given linear constrains (edit linear constraints) even in the case that the original file does not satisfy them.

We have detailed the properties for the case of the constrained FCM. These properties can also be inferred for constrained EFCM. In that case, values of $\lambda_1$ and $\lambda_2$ near to zero play the role of large $m_1$ and $m_2$.

### 4. Experiments

For illustration, we have applied our approach to a small data set. We have considered the example in [31] where a linear constraint involving three variables was considered. The data is reproduced in Tables 1 and 2. Variables $v_1$, $v_2$ and $v_3$ stand for Expenditure at 16%, Expenditure at 7%, and Total Expenditure. Then, variables $v_1$, $v_2$ and $v_3$ define the following linear constraint: $V_3 = \alpha_1 V_1 + \alpha_2 V_2$. That is, $v_3 = 1.16v_1 + 1.07v_2$.

<table>
<thead>
<tr>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>23</td>
<td>42.01</td>
</tr>
<tr>
<td>12</td>
<td>43</td>
<td>59.93</td>
</tr>
<tr>
<td>64</td>
<td>229</td>
<td>319.27</td>
</tr>
<tr>
<td>12</td>
<td>45</td>
<td>62.07</td>
</tr>
<tr>
<td>28</td>
<td>39</td>
<td>74.21</td>
</tr>
<tr>
<td>71</td>
<td>102</td>
<td>191.50</td>
</tr>
<tr>
<td>23</td>
<td>64</td>
<td>95.16</td>
</tr>
<tr>
<td>25</td>
<td>102</td>
<td>138.14</td>
</tr>
<tr>
<td>48</td>
<td>230</td>
<td>301.78</td>
</tr>
<tr>
<td>32</td>
<td>50</td>
<td>90.62</td>
</tr>
<tr>
<td>90</td>
<td>200</td>
<td>318.40</td>
</tr>
<tr>
<td>13</td>
<td>100</td>
<td>125.56</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.91695</td>
<td>26.67021</td>
<td>41.37696</td>
</tr>
<tr>
<td>15.48220</td>
<td>42.61481</td>
<td>60.60212</td>
</tr>
<tr>
<td>65.86964</td>
<td>228.47892</td>
<td>318.70371</td>
</tr>
<tr>
<td>12.97750</td>
<td>45.84617</td>
<td>60.80475</td>
</tr>
<tr>
<td>25.93508</td>
<td>38.55444</td>
<td>75.96227</td>
</tr>
<tr>
<td>72.14286</td>
<td>103.34332</td>
<td>191.54478</td>
</tr>
<tr>
<td>24.43550</td>
<td>65.84895</td>
<td>96.49401</td>
</tr>
<tr>
<td>24.56774</td>
<td>101.54299</td>
<td>137.49281</td>
</tr>
<tr>
<td>24.43550</td>
<td>65.84895</td>
<td>96.49401</td>
</tr>
<tr>
<td>91.86226</td>
<td>197.98087</td>
<td>318.96431</td>
</tr>
<tr>
<td>11.64466</td>
<td>100.13359</td>
<td>127.12980</td>
</tr>
</tbody>
</table>
Table 3
Centroids obtained for the original and noisy data

<table>
<thead>
<tr>
<th>Data</th>
<th>Cluster</th>
<th>Center</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original data</td>
<td>v_1</td>
<td>13.44075, 37.16236, 55.35500</td>
</tr>
<tr>
<td></td>
<td>v_2</td>
<td>67.32890, 219.64071, 313.11708</td>
</tr>
<tr>
<td></td>
<td>v_3</td>
<td>27.59963, 52.64698, 88.34783</td>
</tr>
<tr>
<td></td>
<td>v_4</td>
<td>37.11288, 101.71213, 151.88292</td>
</tr>
</tbody>
</table>

N(0, 0.5) no const.
| v_1     | 14.77033, 37.46462, 55.53745 |
| v_2     | 67.66607, 219.68178, 313.78662 |
| v_3     | 28.25678, 52.81140, 88.66240 |
| v_4     | 37.37631, 101.05854, 153.25147 |

N(0, 0.5) w/t const.
| v_1     | 14.21093, 36.94862, 56.03790 |
| v_2     | 67.74399, 219.75566, 313.71945 |
| v_3     | 28.04952, 52.62022, 88.84108 |
| v_4     | 37.96198, 101.59877, 152.74659 |

N(0, 1.0) no const.
| v_1     | 13.35356, 37.43389, 55.51220 |
| v_2     | 68.10518, 219.87425, 313.35174 |
| v_3     | 28.32082, 52.62483, 87.69683 |
| v_4     | 36.05573, 102.03179, 152.27787 |

N(0, 1.0) w/t const.
| v_1     | 13.34286, 37.42402, 55.52142 |
| v_2     | 67.80086, 219.59354, 313.61408 |
| v_3     | 27.83433, 52.17608, 88.11623 |
| v_4     | 36.48085, 102.42393, 151.91139 |

N(0, 1.5) no const.
| v_1     | 15.38410, 38.46154, 54.85374 |
| v_2     | 68.56108, 217.71406, 313.44706 |
| v_3     | 26.27168, 52.15607, 88.14111 |
| v_4     | 36.43006, 101.70231, 152.30233 |

N(0, 1.5) w/t const.
| v_1     | 14.00637, 37.19070, 56.04144 |
| v_2     | 68.88084, 218.00900, 313.17140 |
| v_3     | 27.11313, 52.93224, 88.08872 |
| v_4     | 36.83617, 102.07690, 151.95224 |

Table 4
Distance between original centroids and centroids of noisy data.

<table>
<thead>
<tr>
<th>Noise</th>
<th>d_1</th>
<th>d_2</th>
<th>d_3</th>
<th>d_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>N(0, 0.5)</td>
<td>1.3756566</td>
<td>0.75078064</td>
<td>0.7468825</td>
<td>1.5393139</td>
</tr>
<tr>
<td>N(0, 0.5)</td>
<td>1.095613</td>
<td>0.74021727</td>
<td>0.6681398</td>
<td>1.2164527</td>
</tr>
<tr>
<td>N(0, 1.0)</td>
<td>0.3251812</td>
<td>0.6870117</td>
<td>0.57486784</td>
<td>0.95232975</td>
</tr>
<tr>
<td>N(0, 1.5)</td>
<td>2.3907604</td>
<td>2.3106573</td>
<td>1.4905896</td>
<td>0.8013973</td>
</tr>
<tr>
<td>N(1, 1.5)</td>
<td>0.8899032</td>
<td>2.25254</td>
<td>0.620389</td>
<td>0.4630664</td>
</tr>
</tbody>
</table>

Table 3 display the cluster centers we have obtained for the original data set without noise, and the ones obtained with noise. For these three last data sets we include both the cluster centers obtained by the standard fuzzy c-means and by the new method.

Comparison between the cluster centers of the noisy data with the ones of the original data show that the cluster centers are more similar to the original ones when constraints are considered. This is shown in Table 4. This table shows the differences, measured using the Euclidean distance, between the 4 cluster centers obtained using a particular data set and the clustering algorithm with respect to the cluster center using fuzzy c-means with the original (no-noise) data. It can be seen that for each of the noisy data, the nearest cluster is the one where constraints are considered (second row for each of the noisy data).

These results permits us to show on the one hand the suitability of the method for combining in the same step the edit constraints and the data protection method, and on the other hand, that our approach leads to protected data with less information loss. This is so because the cluster centers with constrained FCM are more similar to the original cluster centers than the cluster centers using standard FCM.

5. Conclusions and future work

We have presented two variations of fuzzy c-means to deal with linear constraints and shown their use for data masking. We have given the solution of the optimization problem when linear constraints are considered on the data. The solution leads to cluster centers that satisfy the constraints even when the data does not. This permits to combine in the same step data editing and data masking. This approach is one of the first to combine these two steps. We have also given an example that shows how the method can be applied and that for noisy data we obtain cluster centers that are nearer to the ones without noise.

We have also discussed the effects of the parameters in information loss. Further work is about the appropriate selection of the parameters of the method, and comparison with other approaches as e.g. [10, 11]. That is, m_1 and m_2 for constrained FCM and λ_1 and λ_2 in constrained EFCM. We have seen however, that with an appropriate selection we can range from no information loss (and, thus, maximum disclosure risk) to maximum loss (and, thus, no risk).
Acknowledgments

Partial support of the project Swedish Research Council (Vetenskapsrådet) (grant number VR 2016-03346) is acknowledged.

References