



CASC PROJECT

Computational Aspects of Statistical Confidentiality

June 2002

Definition and implementation of aggregation procedures for SDC qualitative variables

Vicenç Torra

Institut d'Investigació en Intel·ligència Artificial - CSIC

e-mail: vtorra@iia.csic.es <http://www.iia.csic.es/~vtorra>

Deliverable 1.1-D7

Brief description of the microaggregation procedure

```
procedure microaggregation (M: matrix, GroupSize: int) is
  I:= Select Variables To Be Microaggregated
  for i=0 to |I| step GroupSize do
    TD := project M on variables (i..max(|I|, i+GroupSize)); // TD=Table with Data
    TD := Lexicographical ordering of WM;
    TD2 := Copy only different records from WM;
    HM := Frequency of records in TD2 in relation to TD;
    Define the appropriate number of clusters; // heuristic: See below
    Hard k-means of HM; // see below
    Compute microaggregation; // see below
  end for;
```

Heuristic about the number of clusters:

The heuristic has been defined considering the following aspects:

When the number of indistinguishable records is less than the constant K, as they cannot define a cluster by themselves, the larger is the number of elements to be located in some cluster.

When the number of indistinguishable clusters is larger than one, they can define a single cluster.

The number of clusters is never zero

This heuristic value is computed as follows:

```
count := 0;
for each record r in TD2 loop
  if (numberOfIndistinguishableRecords r in TD) < K) then {
    count = count + (numberOfIndistinguishableRecords r in TD); }
  else { count++; }
end for;
totalNumberOfClusters := (int)(count / constantK);
```

Computation of the clusters

The clusters are computed using k-modes, a clustering method for categorical data following the well-known method for numerical data: k-means. We have followed the description in [Huang et al., 1999].

To find the optimal clustering, the method repeats an iterative process consisting on two stages. Assuming that, initially, a set of clusters is already known, the stages are the following:

- 1) Representatives of the clusters are computed
- 2) Elements are assigned to the nearest clusters

Until a terminating condition is satisfied, these two stages are repeated.

To implement this method, the program includes the following elements:

- 1) A method to bootstrap the process:** This means defining the initial set of clusters. In our method we get the initial clusters assigning (at random) a record to each cluster. This also corresponds to assign representatives for the original clusters.
- 2) Determination of the nearest cluster for each record:** This corresponds to build a partition of the set of records. The determination of the nearest cluster of each record is based on the distance between the record and the representative. The distance is a summation of the distances between individual values, and where distance between values is defined according to the attribute type. Usual distance function is used. This is, (a) for nominal attributes, distance is 1 for different

values and 0 for equal; (b) for ordinal, distance is defined according to the position of the categories in the domain.

3) Computation of the representatives: This is achieved by means of a microaggregation (variable by variable) of the elements of the cluster. Several aggregation methods have been considered and implemented. Aggregation methods are based on the mode in the case of nominal scales and both the mode and the median in the case of ordinal scales. The methods are described in detail in [Domingo-Ferrer, Torra, 2002a and 2002b].

The implementation of these aggregation operators include the following:

- a) computation of the frequencies of each category
- b) convex transformation of the frequencies of a given variable (for ordinal variables only).

This is, the *frequency* freq of a category x is modified so that:

$$\text{freq}(x) = \min (\max_{y < x} \text{freq}(y), \max_{y > x} \text{freq}(y)).$$

- c) transformation of the frequencies (this is to obtain a parametric aggregation function).

4) Cardinality restrictions checking: To assure that all final micro-clusters have, at least, the desired cardinality, some elements are relocated.

Microaggregation

Microaggregation for data once has been clustered follow the same approach that for computing the representatives of the clusters (see (3) above).

References

- Domingo-Ferrer, J., Torra, V., (2002a), Median based aggregation operators for prototype construction in ordinal scales, submitted.
- Domingo-Ferrer, J., Torra, V., (2002b), Extending Microaggregation Procedures using Defuzzification Methods for Categorical Variables, submitted.
- Huang, Z., Ng, M. K., (1999), A Fuzzy k-Modes Algorithm for Clustering Categorical Data, IEEE Trans. on Fuzzy Systems, 7:4 446-452.

Brief description of the rank-swapping procedure

Implementation of the rank swapping procedure follows [Moore, 1996]

References

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

Implementation details

The code for categorical microaggregation and rank swapping procedures is embedded in our testing software for evaluating information loss. Three program files are included in the deliverable:

functions.h	headers
functions.cxx	functions
main.cxx	main program

For compilation we use:

```
gcc -c functions.cxx -o functions.o
gcc functions.o main.cxx -lm -o main
```

We have also included testing files for both methods.

examples

a file with two command lines to execute the main program for categorical microaggregation and rank swapping

data

folder with the data files for executing the examples

ahs93n1000.ori file to be masked with 1000 records

ahs93n.ori

original file used for computing information loss (not relevant here)

v001120

files describing the variables in the file to be masked (ahs93n1000.ori).

p010208R1p

example of parameter file for the rank swapping

p010208MTTT080305Fp

example of parameter file for the categorical microaggregation.