CASC PROJECT

Computational Aspects of Statistical Confidentiality

June 2002

Parameter and parameter files description of the main program

Vicenç Torra
Institut d'Investigació en Intelligència Artificial – CISC
Campus Universtat Autònoma de Barcelona

Deliverable No:1.1-D8-C

Parameter and parameter files description of `main` program

Vicenç Torra
Institut d'Investigació en Intel.ligència Artificial - CSIC
Campus UAB s/n, E-08193 Bellaterra (Catalunya, Spain)
e-mail: vtorra@iiia.csic.es, http://www.iiia.csic.es/˜vtorra

June 28, 2002

1 Introduction

In this document we describe the program `main`, its arguments and its output.

The structure of this document is as follows. First, a description of the arguments to be included in the command line when executing the program is given. Then, the file structure of the required files is detailed. In particular, the following ones are considered:

- The file describing the variables.
- The file selecting masking methods and stating their parameters.
- The file to declare Markov matrices for PRAM.
- The file to declare recoding schemes for Global recoding.

These descriptions are followed by the one of the output of the program.

2 Arguments to the program

The supplied program can be used for the three following purposes:

1. Given a file with categorical data, compute the corresponding masked file using one of the following masking methods):
   - Top coding
   - Bottom coding
   - Global recoding
   - PRAM
   - Rank swapping

- Microaggregation
- Adhoc local suppression (implemented for testing purposes)

These masking methods are described in [1].

2. Given an original file and a masked file, compute information loss measures. The following measures are considered:

   (a) Distances between original records and masked records (Distance Based Information Loss measure).

   (b) Distances between the contingency tables of the original file and the ones of the masked file (Contingency Table Based Information Loss measure and Average Contingency Table Based Information Loss measure). Contingency tables of both files and the table defined as the difference of both tables are also displayed.

   (c) Entropy.

   (d) Alternative measure based on probabilities.

   Information loss measures are described in [2].

3. Given a file, compute the corresponding masked file using one of the masking methods given above and then compute the information loss measures.

To do so, the program requires 10 parameters to be entered in the command line. These parameters are the following ones (see Table 1):

Arg. N.	Meaning (type)
1	File with the description of the variables (file name)
2	File with the parameters for the masking methods (file name)
3	File to estimate probabilities (file name)
4	Maximum dimension for contingency tables (an integer value)
5	File to mask (file name)
6	Microaggregation parameters (when needed)
7	Microaggregation parameters (when needed)
8	Variable file for μ-Argus (file name)
9	Input masked file (file name)
10	Output masked file (file name)

Table 1: Arguments of the program

The first two parameters correspond to the names of two files with the information about the variables and the masking methods to be applied to the files. The structures of these files are described in Section 3 and 4.

The third parameter correspond to the file used to estimate the probabilities for the entropy and for the alternative information loss measure also based on probabilities. This file can be either equal or different to the file we are masking (argument 5 in the command line).

The fourth parameter corresponds to the dimension K used for building contingency tables up to this dimension.

The fifth argument corresponds to the file the program is going to mask. When only information loss measures are computed but no masking is needed, this file corresponds to the original file.

The sixth and seventh arguments are used when any of the masking methods for the variables is microaggregation. As they are always parsed, use TTT000000 and 0.0 when not needed. The seventh parameter is the value α in $Q_\alpha = x^\alpha$ for WOW-microaggregation (see [5] for details). The sixth one is a combination of 9 characters. The first three ones are T or N and the last six ones are digits. The meaning is as follows ("*" means any other character):

char0 (either "T" or "*"): (T) when using Mode for both ordinal and nominal variables; (*) when Mode is only used for nominal (Median or Random Selection for ordinal)

char1 (either "T" or "*"): if char0=*, (T) random selection for ordinal variables (*) when selection for ordered is median

char2 (either "T" or "*"): (T) when Convex is used in Mode (*) Convex not used

char3-4: 2 digits for number of variables in microaggregation

char5-6: 2 digits for constant K

char7-8: 2 digits for maximum number of iterations in the adaptation of K-Modes algorithms

The eighth parameter corresponds to a file that is created by the program. This file is used when the masked file has to be masked again with local suppression. The file created by the program corresponds to the metadata file needed by μ-Argus [3] so that the masked file can be read by μ-Argus and local suppression can be applied to all those variables that `main` considers in the contingency tables. Only variables selected for contingency tables are used because μ-Argus bases suppression on contingency tables. Besides of this information and the information about the position of the variables, the file includes also the name of the suppressed variable.

The ninth argument corresponds to the file that will be compared with the original one (i.e. argument number 5) to measure information loss. When we want to measure the masked file generated by the program, arguments 9 and 10 should correspond to the same file.

The tenth argument is the name of a file to be created by the program and that contains the masked file. As it has just been said, it can be re-used (when argument 9 is equal to this argument) if we want to compute the information loss of this file.

According to all this, to compute the masking of a file ahs93n100.ori (we save it as ahs93n100.msk) with variables described in v001120 and with the masking methods described in p001120 we run the program in the following way:

```
main v001120 p001120 ahs93n.ori 2 ahs93n100.ori TTT000000 0.0 ahs93n100.rda
ahs93n100.msk ahs93n100.msk
```

If needed, ahs93100.rda will be the file to be used in μ-Argus. Moreover, I use here ahs93n.ori as the original file to compute the probabilities for the entropy and the alternative definition for information loss. This latter file contains all the records supplied by the Data Extraction System [4]. Instead, ahs93100 only contains 100 records.

3 File describing the variables

The file describing the variables has the following information for each variable:

1. The name of the variable. This field is a string.

2. The initial position of the variable in the file. This field is an integer.

3. The length (in number of characters) of that variable in the file. This field is an integer.

4. Description whether this is a numerical variable (it has a zero value), a non-ordered categorical one (it has a value of 1), or an ordered categorical one (the value in this case is 2). This field is an integer.

5. The number of categories. This field is an integer. The program expects to read as much as categories as the number indicates. This is so except for the case that the number of categories is less than one. In this latter case only a string is read (and ignored). This string is to be used to give information to the user about the variable.

6. As much as categories as the previous field indicates. Categories are considered as strings.

In Table 2 it is displayed the variable file v001120.

4 File describing the masking parameters

.

The file describing the masking methods has the following information for each variable:

1. A string corresponding to the variable name. The variables in this file have to be the same and in the same order that the ones in the file with the description of the variables. The order in which variables appear is irrellevant, but the same order has to be kept in both files.

Name	I P	Len	Type	N C	Categories
AGE	6	2	0	3	91 99 –
BUILT	8	2	2	25	01 02 03 04 05 06 07 08 09 80 01 82 83 84 85 86 87 88 89 90 91 92 93 99 –
DEGREE	10	1	2	8	1 2 3 4 5 6 9 -
GRADE1	11	2	2	21	00 01 10 11 12 02 21 22 23 24 25 26 03 04 05 06 07 08 09 99 –
METRO	13	1	1	9	1 2 3 4 5 6 7 9 -
SCH	14	1	1	6	1 2 3 8 9 -
SHP	15	1	1	6	1 2 3 8 9 -
SMSA	16	4	0	-1	4.digits
TRAN1	20	2	1	12	01 04 05 06 07 08 09 10 11 12 99 -
WEIGHT	22	7	0	0	Two.implied.decimal.places
WFUEL	29	2	1	10	01 02 03 04 05 06 07 08 09 –
WHYMOVE	31	2	1	18	01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 98 99 –
WHYTOH	33	2	1	13	00 01 02 03 04 05 06 07 08 09 98 99 –
WHYTON	35	2	1	13	00 01 02 03 04 05 06 07 08 09 98 99 –

Table 2: Example of the data file: `v001120` (I P is the Initial Position in the file, Len is the number of characters used, and N C is the number of categories).

2. An integer stating if the variable is going to be used for information loss measures. Selected variables are used to build contingency tables, to measure distance based information loss and to compute entropy. Also, these variables are marked in the file generated for the μ-Argus. When the integer is zero, the variable will not be used for information loss measures. Instead, when the integer is one, the variable will be used.

3. A character stating the masking method to be applied to this variable. The following characters are recognized (description of the methods, except for categorical microaggregation, in [1]):

   - P: the masking method is PRAM.
   - G: the masking method is Global Recoding.
   - T: the masking method is Top coding.
   - B: the masking method is Bottom coding.
   - R: the masking method is Rank Swapping.
   - M: the masking method is Microaggregation.
   - S: the masking method is Local Suppression. This method is implemented in an adhoc way and it has been used for testing. It generates suppressed variables at random with a certain probability.

4. A parameter is considered for each of the masking methods but microaggregation. The type of the parameter depends on the method. They are the following ones:

   - PRAM method requires a string corresponding to a file with the Markov matrix. The structure of this file is described in Section 5.
   - Global Recoding requires a string corresponding to a file with all the recodings. File structure is given in Section 6.
   - Top and Bottom coding require an integer corresponding to the number of categories to be recoded.
   - Local suppression needs a float corresponding to the probability of suppressing a category.
   - Rank swapping requires a float in $(0, 1]$ corresponding to relative positions where swapping is allowed.

   An example of this file is given in Table 3

V. name	Cont. Tablee	Masking Method	Parameter
AGE	0	S	0.2
BUILT	1	T	3
DEGREE	1	B	2
GRADE1	1	T	5
METRO	0	-	
SCH	0	-	
SHP	1	S	0.8
SMSA	0	-	
TRAN1	1	G	tran1.hie.01
WEIGHT	0	-	
WFUEL	0	M	
WHYMOVE	0	-	
WHYTOH	0	-	
WHYTON	1	P	whyton.pra.01

Table 3: Example of the parameters file: `p001120`

5 File for PRAM: Markov matrix

.

The structure of the file to be used for PRAM methods is simple: an integer corresponding to the number of probabilities in the Markov matrix and then as much as floats as the integer indicate. The program checks if the integer corresponds to the expected value (it should be the square of the number of categories) and stops if it is not correct.

An example of this file is given in Table 4 for the categorical variable `WHYTON`.

00	01	02	03	04	05	06	07	08	09	98	99	–
169												
0.8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0
0.01	0.8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0
0.8	0.01	0.8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0
0.8	0.01	0.01	0.8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0
0.8	0.01	0.01	0.01	0.8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0
0.8	0.01	0.01	0.01	0.01	0.8	0.01	0.01	0.01	0.01	0.01	0.01	0.0
0.8	0.01	0.01	0.01	0.01	0.01	0.8	0.01	0.01	0.01	0.01	0.01	0.0
0.8	0.01	0.01	0.01	0.01	0.01	0.01	0.8	0.01	0.01	0.01	0.01	0.0
0.8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.8	0.01	0.01	0.01	0.0
0.8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.8	0.01	0.01	0.0
0.8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.8	0.01	0.0
0.8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.8	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 4: Example of the Markov matrix file for PRAM: `whyton.pra.01`

6 File for Global recoding: Recodings

.

The file to be used for Global recoding has the following parameters:

1. An integer corresponding to the number of recoding rows that the file have. Each recoding row is explained below.

2. An integer corresponding to the number of recoding rows that recode categories in the initial file. This number can be smaller than the previous one. The recoding scheme can be a complete hierarchy in the sense of the Non-overlapping hierarchical categorical variables. This integer is not used, but corresponds to the number of categories that μ-Argus recodify.

3. An integer corresponding to the number of probabilities. My definition of global recoding uses a probability of changing a value to the recoded one. It should be one probability for each category (this is not checked by the program).

4. As much as floats as the previous integer describes. These values correspond to the probabilities for initial categories. Recoding will use this probabilities and they are also used for computing conditional probabilities when information loss is measured using probability based measures.

5. A recoding scheme. Recoding scheme has the following structure:

   (a) A string corresponding to the name of a new category. Let us call `cat` to this category. This name should not have appeared before.

(b) A float that corresponds to the probability that the `cat` changes to another value. This value is not used because the category does not belong to the initial file.

(c) An integer corresponding to the number of categories recoded with `cat`.

(d) As much as strings as the previous integer indicates. All strings should have appeared before (either in the variables file or in a previous line of this file).

An example of this file is given in Table 5 for the categorical variable `TRAN1`. Here *pu* stands for $public - transport$, *pr* for *private*, *oo* for *others* and *to* for the *total* (all categories are of this kind).

4	3							
9								
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
pu	1.0	2	04	05				
pr	1.0	4	01	06	07	08		
oo	1.0	3	09	10	99			
to	1.0	3	pu	pr	oo			

Table 5: Example of a Global recoding scheme. File `tran1.hie.01`

7 Output of the progam

At present, the program displays a lot of data. So, it is convenient to redirect the output to a file.

The last written line of the program gives the rellevant information about information loss measures. In particular it displays (see [2] for details on the meaning of the measures):

1. Number of cells in contingency tables
2. Distance, CTBIL and ACTBIL
3. EBILRF, ILRF, EBILMF, ILMF (using probabilities computed from the masking methods)
4. EBILRF, ILRF, EBILMF, ILMF (using probabilities computed comparing the original file and the masked file)

References

[1] Domingo-Ferrer, J., Torra, V., (2001), Disclosure Control Methods and Information Loss for Microdata, 91-110, in Confidentiality, Disclosure, and

Data Access: Theory and Practical Applications for Statistical Agencies, P. Doyle, J. I. Lane, J. J. M. Theeuwes, L. M. Zayatz (Eds.), Elsevier.

[2] Domingo-Ferrer, J., Torra, V., (2001), A Quantitative Comparison of Disclosure Control Methods for Microdata, 111-133, in Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, P. Doyle, J. I. Lane, J. J. M. Theeuwes, L. M. Zayatz (Eds.), Elsevier.

[3] Hundepool, A., Willenborg, L., Wessels, A., Van Gemerden, L., Tiourine, S., Hurkens, C., (1998), μ-Argus 3.0 User's Manual, Statistics Netherlands.

[4] U. S. Bureau of the Census (2000), The Data Extraction System, http://www.census.gov/DES/www/welcome.html

[5] Domingo-Ferrer, J., Torra, V., (2002), Median based aggregation operators for prototype construction in ordinal scales, submitted.

[6] Domingo-Ferrer, J., Torra, V., (2002), Extending Microaggregation Procedures using Defuzzification Methods for Categorical Variables, in press.