Statistics Netherlands

Division Research and Development
Department of Statistical Methods

P.O.Box 4000
2270 JM Voorburg
The Netherlands

A heuristic approach to cell-suppression in hierarchical tables

Peter-Paul de Wolf

Remarks:
The views expressed in this paper are those of the author and do not necessarily reflect the policies of Statistics Netherlands.

Project number: *RSM-80750, LNM-22070*
BPA number: *1569-99-RSM*
Date: *22 March 1999*

A HEURISTIC APPROACH TO CELL-SUPPRESSION IN HIERARCHICAL TABLES

This paper describes a heuristic approach to find suppression patterns in tables that exhibit a hierarchical structure in at least one of the explanatory variables. The hierarchical structure implies that there exist (many) sub-totals, i.e., that (many) sub-tables can be constructed. These sub-tables should be protected in such a way that they cannot be used to undo the protection of any of the other tables. The proposed heuristic approach has a top-down structure: when a table of high level (sub-)totals is suppressed, its interior turns into the marginals of possibly several tables on a lower level. These lower level tables are then protected while keeping the marginals fixed.

Keywords: Statistical Disclosure Control, table suppression, hierarchical structure, top-down method.

1. Introduction

In statistical disclosure control (SDC), it is common practice to protect data published in tables, one table at a time. At Statistics Netherlands a software package called τ–ARGUS is developed to facilitate several SDC methods to protect tabular data. However, in practice different tables are often linked to each other, in the sense that certain cells can occur in these tables simultaneously. E.g., marginal cells in one table might well be the same cells that appear in the interior of another table. When dealing with such sets of tables, the used disclosure control methods should be consistent with each other: it should not be possible to undo the protection of a table using another -by itself safe- table.

In this paper, the disclosure control method of cell suppression is considered. Based on a dominance rule and a minimal frequency rule (see, e.g., Willenborg and de Waal (1996)) it is decided which cells of a certain table need to be suppressed. These suppressions are called *primary* suppressions. However, in order to eliminate the possibility to recalculate these suppressed cells (either exactly or up to a good approximation), additional cells need to be suppressed, which are called *secondary* suppressions. Whenever a variable has a hierarchical structure (e.g., regional variables, classification variables like NACE), secondary suppressions might imply even more secondary suppressions in related tables.

In Fischetti and Salazar-González (1998) a theoretical framework is presented that should be able to deal with hierarchical and generally linked tables. In that framework additional constraints to a linear programming problem are generated. The number of added constraints however, grows rapidly when dealing with hierarchical tables, since many dependencies exist between all possible (sub-)tables containing

many (sub-)totals. Hence the time needed to calculate a feasible solution might increase considerably as well. A heuristic approach will be proposed that deals with a large set of (sub-)tables in a particular order. In the next section the approach will be discussed in more detail and a simple example with artificial data will be used to illustrate the ideas. Section 3 presents results concerning the example introduced in Section 2. In Section 4 some additional remarks are made on the presented cell-suppression method. Appendix A contains all the sub-tables defined in the example of Section 2 that have to be considered when applying the method.

2. Hierarchical cell-suppression

The proposed heuristic approach in constructing a method to deal with cell suppression in hierarchical tables is a top-down approach. For the purpose of simplicity of the exposition, we will discuss the method in case of a two dimensional table in which both explanatory variables exhibit a hierarchical structure. The ideas are easily extended to the situation of higher dimensional tables, with an arbitrary number of hierarchical explanatory variables.

The example that will be used to illustrate the ideas consists of two fictitious hierarchical variables: a regional variable R (Region) and a classifying variable BC (Industrial Classification). The hierarchical structures are represented in Figures 1 and 2. The variable R consists of 4 levels: level 0 (R), level 1 (P1, P2, P3), level 2 (C21, C22, C31, C32) and level 3 (D211, D212). The variable BC consists of 3 levels: level 0 (BC), level 1 (I, A, O) and level 2 (LI, MI, SI, LA, SA).

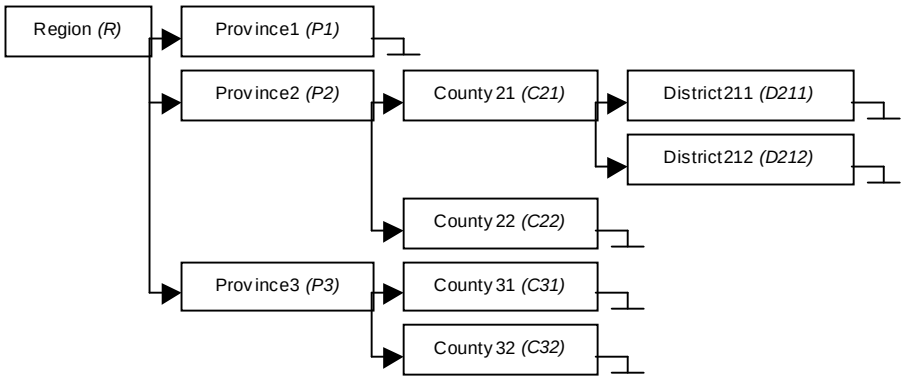

Figure 1: Hierarchical structure of regional variable R

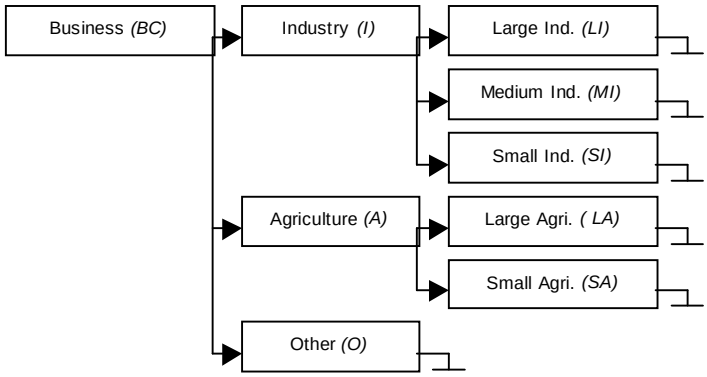

Figure 2: Hierarchical structure of classification variable BC

The first step is to determine the primary unsafe cells in the base-table consisting of all the cells that appear when crossing the two hierarchical variables. This way all cells, representing a (sub-)total or not, are checked for primary suppression. The base-table of the example is represented in Figure 3, in which the cells that primarily will be suppressed are marked by a boldface value in a darkly shaded cell and an empty cell is denoted by a -.

	BC	I				A			O
			LI	MI	SI		LA	SA	
R	300	125	41	44	40	93	51	42	82
P1	77	31	8	12	11	26	13	13	20
P2	128	52	21	18	13	44	24	20	32
C21	77	25	11	9	**5**	31	18	13	21
D211	35	16	4	7	**5**	10	6	**4**	**9**
D212	42	9	7	**2**	-	21	12	9	12
C22	51	27	10	**9**	8	13	6	7	11
P3	95	42	12	14	16	23	14	9	30
C31	45	27	8	9	10	**5**	**2**	**3**	13
C32	50	15	4	5	6	18	12	6	17

Figure 3: Base-table of R × BC (darkly shaded means primary unsafe)

Knowing all primary unsafe cells, the secondary cell suppressions have to be found in such a way, that each (sub-)table of the base-table is protected and that the different tables cannot be combined to undo the protection of any of the other (sub-)tables.

There are (at least) two possible approaches to this problem: a bottom-up and a top-down approach. Each approach attempts to protect a large set of (sub-)tables in a sequential way. In the bottom-up approach one would move upwards in the hierarchy of the explanatory variables and calculate the secondary suppressions using the (fixed) suppression pattern of a lower level table as its interior. In the example, one of the lowest level tables to consider is (D211, D212) × (LA, SA). Since the cell (D211, SA) is primarily unsafe, some additional suppressions have to be found. One

possible suppression pattern is given by suppressing the interior, while keeping the marginals publishable. See Figure 4.

	LA	SA	A
D211	6	**4**	10
D212	12	9	21
C21	18	13	31

⇒

	LA	SA	A
D211	6	**4**	10
D212	12	9	21
C21	18	13	31

Figure 4: Low level table, = primary, = secondary suppression

However, the higher level table (D211, D212) × (I, A, O) has a primary unsafe cell (D211, O) and needs secondary suppressions as well. This could lead to a (secondary) suppression of cell (D211, A). See Figure 5.

	I	A	O	BC
D211	16	10	**9**	35
D212	9	21	12	42
C21	25	31	21	77

⇒

	I	A	O	BC
D211	16	10	**9**	35
D212	9	21	12	42
C21	25	31	21	77

Figure 5: Higher level table, = primary, = secondary suppression

Unfortunately, one of the secondarily suppressed cells is also a marginal cell of the previously considered table, hence backtracking is needed: the table of Figure 4 has to be considered again, using the information of the suppression pattern of the table in Figure 5. In other words, the tables can not be dealt with independently of each other. Moreover, using backtracking, it not at all clear whether the method will converge.

The basic idea behind the top-down approach is to start with the highest levels of the variables and calculate the secondary suppressions for the resulting table. In theory the first table to protect is thus given by a crossing of level 0 of variable R with level 0 of variable BC, i.e., the grand total. The interior of the protected table is then transported to the marginals of the tables that appear when crossing lower levels of the two variables. These marginals are then ‘fixed’ in the calculation of the secondary suppressions of that lower level table. This procedure is then repeated until the tables that are constructed by crossing the lowest levels of the two variables are dealt with.

Using the top-down approach, the problems that were stated for the bottom-up approach are circumvented: a suppression pattern at a higher level only introduces restrictions in the marginals of lower level tables. Calculating secondary suppressions in the interior while keeping the marginals fixed, is then independent between the tables on that lower level, i.e., no backtracking is needed. Moreover, added primary suppressions in the interior of a lower level table are dealt with at that same level: secondary suppressions can only occur in the same interior, since the marginals are kept fixed.

The introduction of suppressions in the marginals of a table requires additional (secondary) suppressions in its interior. Most models that are used to find suppression

patterns can only deal with either safe cells or primary unsafe cells. Secondary suppressions in the marginals hence need to be considered as primary unsafe cells when applying such models in the present situation. Usually, rather large safety-ranges (see e.g., Willenborg and de Waal (1996)) are imposed on primary unsafe cells, which may be too restrictive in the case of secondary suppressions. Actually, the suppression pattern of the higher level table that produced the secondary suppression for the marginal of a lower level table produces a certain actual safety-range for that secondary suppression. Ideally, that actual safety-range should be used when considering the secondary suppression as a primary unsafe cell. The actual safety-range can be calculated by solving two Linear Programming (LP) problems (maximising and minimising the cell-value under the restrictions imposed by the additivity of the table), which will cost additional computing time. In practice it might be more convenient to manually impose rather small safety-ranges on these cells, in order to eliminate the possibility that the value can be recalculated exactly, while being less restrictive as in the case of 'real' primary unsafe cells.

Obviously, all possible (sub)tables should be dealt with in a particular order, such that the marginals of the table under consideration have been protected as the interior of a previously considered table. This can be assured by defining certain classes of tables. First define a group by a crossing of levels of the explanatory variables. The classes of tables mentioned before then constitute the groups in which the numbers of the levels add to a constant value. Figure 6 contains all the classes and groups that can be defined in the example.

Class	Groups
0	00
1	01, 10
2	02, 20, 11
3	12, 21, 30
4	22, 31
5	32

Figure 6: The classes defined by crossing R with BC

E.g., group 31 is a crossing of level 3 of variable R and level 1 of variable BC. Such a group thus may consist of several tables: group 31 consists of the table (D211, D212) × (I, A, O) whereas group 21 consists of the tables (C21, C22) × (I, A, O) and (C31, C32) × (I, A, O). Note that the latter two tables need to be dealt with separately. Class 4 consists of group 22 (2 + 2 = 4) and group 31 (3 + 1 = 4).

Defined in this way, marginals of the tables in class i have been dealt with as the interior of tables in a class j with $j < i$. As a result, each table in class i can be protected independently of the other tables in that particular class, whenever the tables in classes j with $j < i$ have been dealt with.

The number of tables in a group is determined by the number of parent-categories the variables have one level up in the hierarchy. A parent-category is defined as a category that has one or more sub-categories. E.g., group 22 has four tables, since variable R has two parent-categories at level 1 (categories P2 and P3) and variable

BC has two parent-categories at level 1 (categories I and A) and thus $2 \times 2 = 4$ tables can be constructed.

To illustrate things, in Appendix A all tables are given that had to be checked in case of the example, ordered by the corresponding classes and groups as given in Figure 6.

3. Examples of output

A prototype C++ 32 bits Windows console program has been written, to perform the proposed heuristic approach to cell-suppression in hierarchical tables. Figure 7 contains the results of running that program on the example of Section 2. The unprotected table was given in Figure 3. The darkly shaded cells in Figure 7 are primary suppressions, the lightly shaded cells are secondary suppressions.

			BC	I				A			O
					LI	MI	SI		LA	SA	
R			300	125	41	44	40	93	51	42	82
	P1		77	31	8	12	11	26	13	13	20
	P2		128	52	21	18	13	44	24	20	32
		C21	77	25	11	9	**5**	31	18	13	21
		D211	35	16	4	7	**5**	10	6	**4**	**9**
		D212	42	9	7	**2**	-	21	12	9	12
		C22	51	27	10	**9**	8	13	6	7	11
	P3		95	42	12	14	16	23	14	9	30
		C31	45	27	8	9	10	**5**	**2**	**3**	13
		C32	50	15	4	5	6	18	12	6	17

Figure 7: Base-table of R × BC after applying the method

To illustrate the size of the problem and the associated running time to complete the cell-suppression of a hierarchical table, consider the following more realistic and much more complex example.
Consider a data file with 9004 records containing four variables: three hierarchical explanatory variables and one response variable. The explanatory variables are:

1. A regional variable (690 codes, including the (sub-)totals) with a hierarchical structure of 7 levels (level 0 up to level 6)
2. A classification variable (105 codes, including the (sub-)totals) with a hierarchical structure of 5 levels (level 0 up to level 4)
3. A business-size variable (15 codes, including the (sub-)totals) with a hierarchical structure of 3 levels (level 0 up to level 2)

Note that this implies a 3 dimensional base-table with $690 \times 105 \times 15 = 1{,}086{,}750$ cells. For this case, the program had to check 11,135 (sub-)tables of which 2,700 needed to be considered more carefully, because of primary and/or secondary sup-

pressions. To complete all this, the program needed about 3 ¾ hours on a Pentium 200 MHz Windows95 PC with 64 Mb RAM.

4. Remarks and future developments

In the present paper, a heuristic approach is given to find a suppression pattern in a hierarchical table. Whether or not the resulting pattern is acceptable still needs further evaluation. Moreover, it would be interesting to compare the results with results that would be generated by more sophisticated methods like the one proposed in Fischetti e.a. (1998).

Another interesting research topic would be to investigate to what extent the proposed heuristic approach for hierarchical tables can be generalised to the situation of generally linked tables.

The actual implementation of the method includes some interesting future developments as well. First of all it is desirable to incorporate this approach into the more user friendly software package τ–ARGUS. Moreover, the computation and use of the actual safety-ranges for both primary and secondary suppressions should be included. Finally, it could be interesting to investigate the effect of parallel computing on the overall running time. Since the structure of the method is such that several tables can be dealt with simultaneously, it seems rather straightforward to implement parallel computing.

References

Willenborg, L. and T. de Waal (1996). *Statistical Disclosure Control in practice*. Lecture Notes in Statistics 111, Springer-Verlag, New York.

Fischetti, M and J.J. Salazar-González (1998). Models and Algorithms for Optimizing Cell Suppression in Tabular Data with Linear Constraints. Technical Paper, University of La Laguna, Tenerife.

Appendix A Tables to be checked in example of Section 2

This appendix contains all tables that had to be checked for secondary suppression in case of the example of Section 2. The base-table (only needed for determining the primary cell-suppression) is shown in Figure 3 of Section 2 itself. The group names and class numbers are explained in Section 2. Note that the table of group 00 is already dealt with in the base-table.

In each table, the primary suppressions are marked by a dark shade and the secondary suppressions by a lighter shade.

Class 0

Group 00

	BC
R	300

Class 1

Group 10

	BC
P1	77
P2	128
P3	95
R	300

Group 01

	I	A	O	BC
R	125	93	82	300

Class 2

Group 20

	BC
C21	77
C22	51
P2	128

	BC
C31	45
C32	50
P3	95

Group 02

	LI	MI	SI	I
R	41	44	40	125

	LA	SA	A
R	51	42	93

Group 11

	I	A	O	BC
P1	31	26	20	77
P2	52	44	32	128
P3	42	23	30	95
R	125	93	82	300

Class 3

Group 21

	I	A	O	BC
C21	25	31	21	77
C22	27	13	11	51
P2	52	44	32	128

	I	A	O	BC
C31	27	5	13	45
C32	15	18	17	50
P3	42	23	30	95

Group 12

	LI	MI	SI	I
P1	8	12	11	31
P2	21	18	13	52
P3	12	14	16	42
R	41	44	40	125

	LA	SA	A
P1	13	13	26
P2	24	20	44
P3	14	9	23
R	51	42	93

Group 30

	BC
D211	35
D212	42
C21	77

Class 4

Group22

	LI	MI	SI	I
C21	11	9	5	25
C22	10	9	8	27
P2	21	18	13	52

	LA	SA	A
C21	18	13	31
C22	6	7	13
P2	24	20	44

	LI	MI	SI	I
C31	8	9	10	27
C32	4	5	6	15
P3	12	14	16	42

	LA	SA	A
C31	2	3	5
C32	12	6	18
P3	14	9	23

Group 31

	I	A	O	BC
D211	16	10	9	35
D212	9	21	12	42
C21	25	31	21	77

Class 5

Group 32

	LI	MI	SI	I
D211	4	7	5	16
D212	7	2	-	9
C21	11	9	5	25

	LA	SA	A
D211	6	4	10
D212	12	9	21
C21	18	13	31