STATISTICAL DISCLOSURE CONTROL PROBLEMS FOR LINKED TABLES

Lucia Buzzigoli, Antonio Giusti
Dipartimento di Statistica “Giuseppe Parenti”, Università di Firenze.

Summary
The paper presents the problem of statistical disclosure control for linked tables in a general framework which underlines some important issues on the subject. In particular, after a brief review of the methods existing in the literature to manage this problem, a new algorithm is proposed, which shows interesting properties and possibilities of application. Finally, some links with traditional statistical aspects are shown.

Keywords: Statistical disclosure control, Confidentiality, Linked tables, Linear programming.

1. INTRODUCTION
The duty of national statistical agencies is to collect, process and release various kinds of statistical information both on individuals and on organisations.

At the same time the agencies are often required by law to protect the confidentiality of individual information: this means that the data which are released to the public must be compiled in such a way that the identification of specific respondents is no longer possible. Data are protected by means of particular statistical and mathematical procedures, which can be labelled as statistical disclosure control (SDC) methods.

Statistical agencies usually release cross-classified tables and microdata files. For a long time the primary data product has been in the form of aggregate information; however, nowadays, microdata files are getting more and more important, because researchers require more analytical data for their analysis. Although the difference between microdata and tabular data is not so great as it seems (individual files can be seen as a huge table deriving from \( n \) crossings, where \( n \) is the number of variables surveyed on each individual), traditionally the treatment of SDC methods for these two data products is different. During the past
ten years the research on these topics has been extensive and today is still in progress. A relevant literature was produced by official statisticians and academic researchers. Interesting reviews on this subject can be found in Federal Committee on Statistical Methodology (1994) and Willenborg and de Waal (1996). More recent contributions are contained in Eurostat (1998).

In our work we focus on a particular aspect of SDC for frequency count tabular data. In sect. 2 we describe the problem we are interested in and we briefly review some of the methods that have been proposed in the literature to protect linked tables. In sect. 3 we propose a general algorithm that could be useful to face some unsolved questions and in sect. 4 we relate it to some classical approaches from the statistical methodology procedures. We conclude with some brief remarks in sect. 5.

2. STATISTICAL DISCLOSURE CONTROL METHODS FOR 'LINKED' TABLES

Data collected by statistical agencies are originally in the form of set of records, each record containing information on \( n \) variables surveyed on individual respondents. These data are subsequently processed in order to produce the statistical information that the agencies have to release. As stated in the introduction, the released information is usually in the form of microdata files or tabulations and is protected by methods that modify the informative content of the data in order to avoid the identification of individuals.

For microdata files a first obvious operation is the deletion of direct identifiers (name, address, Social Security Number, etc.); many other SDC methods are available to avoid the indirect identification of individuals (global recoding, local suppression, perturbation, confidentiality edit, etc.; see the reviews cited in the introduction).

Tables of magnitudes and tables of counts represent the most usual data product released by national statistical agencies and, as they contain aggregate information, at first sight they seem to be free from identification problems. Actually, disclosure control problems arise for both kinds of tables, even if with different issues (Citteur and de Vries, 1994); in this work we limit our analysis to tables of counts.

Tables containing cells with low frequencies are generally considered at risk, because individuals with rare characteristics are usually more recognisable than others. Therefore most SDC methods for tables of counts have the purpose to avoid the release of tables with low frequency cells ('sensitive' cells).
The various methods can be classified into two different groups:

- methods that reduce the information contained in the table, which solve the problem of sensitive cells obscuring their contents (cell suppression methods; Cox, 1980, Geurts, 1992) or defining new categories of the involved variables in order to obtain cells with higher frequencies (recoding and grouping methods; Dalenius, 1988);

- methods that modify the original frequency counts of the table, using various mathematical criteria of different complexity: rounding methods (Fellegi, 1975; Cox, 1987), random perturbation (Evans and Zayatz, 1995), confidentiality edit (Griffin et al., 1989).

Actually, SDC methods for frequency tables involve further problems when, as usual, the agency releases groups of tables deriving from the same original data set.

These tables contain only a part of the information included in the original data, but are obviously linked among each other. Sometimes the link is evident (for instance, the tables have some variables in common), sometimes it is less evident (for instance, the tables contain different variables, but there are logical or mathematical relations among some of them). The existence of links between sub-tables affects all the methods usually adopted to protect tables, in the sense that the information contained in the subtables can be used to get some information on the original data.

As a consequence, the controls on the single tables are not enough, because, even if they do not contain any sensitive cell, the variables in common allow the link of the tables increasing the set of information useful for disclosure purposes.

For example, suppose that an agency releases three 2–way tables, containing the variables gender, marital status and political opinion (which can be considered a sensitive variable):

<table>
<thead>
<tr>
<th></th>
<th>married</th>
<th>unmarried</th>
<th></th>
<th>opinion</th>
<th>opinion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
<td>b</td>
<td>c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>male</td>
<td>7</td>
<td>20</td>
<td></td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>female</td>
<td>16</td>
<td>4</td>
<td></td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>married</td>
<td>4</td>
<td>6</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unmarried</td>
<td></td>
<td></td>
<td></td>
<td>13</td>
<td>6</td>
</tr>
</tbody>
</table>

If we suppose that cells containing less than 3 counts are at risk, no sensitive cells are present in the released tables. But, considering that the three tables can be seen as the complete set of the 2–way marginals of a 3–way array, in this case it is possible to reconstruct the array contents, with any method we will see later:
<table>
<thead>
<tr>
<th></th>
<th>opinion a</th>
<th>opinion b</th>
<th>opinion c</th>
</tr>
</thead>
<tbody>
<tr>
<td>married</td>
<td>0</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>unmarried</td>
<td>9</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>opinion a</th>
<th>opinion b</th>
<th>opinion c</th>
</tr>
</thead>
<tbody>
<tr>
<td>married</td>
<td>4</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>unmarried</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The original – not released – array shows some sensitive information: for instance, all the unmarried females have ‘opinion a’, while no married male is of that political opinion. Moreover, there is also the possibility of a so-called ‘internal disclosure’, which happens when one of the surveyed individuals can get information on the other surveyed people: in our example, the only married male who is oriented for ‘opinion b’ knows the political opinion of all the other married men.

Note that the 3-way array corresponds to the following microdata file of 47 records containing information on three variables:

<table>
<thead>
<tr>
<th>record</th>
<th>gender</th>
<th>marital status</th>
<th>political opinion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>male</td>
<td>married</td>
<td>b</td>
</tr>
<tr>
<td>2</td>
<td>male</td>
<td>married</td>
<td>c</td>
</tr>
<tr>
<td>3</td>
<td>male</td>
<td>married</td>
<td>c</td>
</tr>
<tr>
<td>4</td>
<td>male</td>
<td>married</td>
<td>c</td>
</tr>
<tr>
<td>5</td>
<td>male</td>
<td>married</td>
<td>c</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>43</td>
<td>female</td>
<td>married</td>
<td>c</td>
</tr>
<tr>
<td>44</td>
<td>female</td>
<td>unmarried</td>
<td>a</td>
</tr>
<tr>
<td>45</td>
<td>female</td>
<td>unmarried</td>
<td>a</td>
</tr>
<tr>
<td>46</td>
<td>female</td>
<td>unmarried</td>
<td>a</td>
</tr>
<tr>
<td>47</td>
<td>female</td>
<td>unmarried</td>
<td>a</td>
</tr>
</tbody>
</table>

The joint analysis of linked tables permits to get the original file only in particular cases, but also when none of the cell values of the n-way array can be reconstructed, the analysis of linked tables permits in any case to find lower and upper bounds for each cell. For example, from these three 2-way tables, similar to those of the previous example:

```
<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>male</td>
<td>7</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>female</td>
<td>16</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>male</td>
<td>9</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>female</td>
<td>6</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>married</td>
<td>4</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>unmarried</td>
<td>11</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>
```

we cannot, without other information, precisely reconstruct the array contents, but only the following cell bounds:
If the difference between a lower and upper limit is very small, and their values are near to zero disclosure problems can arise. This means that, before releasing sets of tables obtained from the same survey, National Statistical Institutes must check how much information can be derived from them.

Therefore, in this context the main problem is that of finding an efficient method to calculate the lower and the upper bounds for the original values, starting from the released marginals. This is not a simple question, because the released set of tables can be numerous and composed by large dimension tables.

De Vries (1993) proposed linear programming (LP) as a general technique for solving the problem: it is well known that LP can evaluate the maximum and the minimum of a function subject to some constraints; the system of linear relationships between the various cells of the original array deriving from the marginal totals can be easily translated into a set of constraints, while the function to be maximised and – at the same time – to be minimised is the value contained in each single cell.

In practice, given a vector $y$ containing the elements of an $n$–way array arranged in lexicographical order (clearly the lexicographical order is an extension of the vec operator for $n$–way arrays $(n\geq 2)$), the LP method operates on each element of the array separately, solving two distinct problems: minimise $y_k$ and maximise $y_k (k=1,2,\ldots,q)$, subject to a finite number of linear constraints of the form $Ay=b$ and $Cy \geq 0$, where:

- $A$ is a $(0–1) r \times q$ matrix
- $r$ is the total number of elements in the given marginal tables;
- $q$ is the number of elements of the original $n$–way array;
- $y$ is a $q$–dimensional non–negative column vector;
- $C$ is the $q \times q$ identity matrix;
- $b$ is an $r$–dimensional column vector containing the marginal values.

This kind of problem is solved by the simplex algorithm (Brickman, 1989), as it is common practice for LP problems. Unfortunately, although LP is extremely powerful, it is not particularly efficient in this field of application, because the procedure must be applied $2\times q$ times. Therefore, when the dimensions of the array increase, it can become very time consuming.
Moreover, other relevant questions are still open (de Vries, 1993; sect.6): firstly, at the moment it is not known when some cell values can be exactly identified (some conclusions are given from an algebraic point of view, but they cannot be easily translated into the LP frame); secondly, it is not sure that LP gives integer solutions.

The computational problems affecting LP are an obstacle to the development of interactive packages to check tables. In the last few years, many efforts have been made by researchers to find new methods to calculate lower and upper bounds, which are alternative to LP. All procedures are intended to drastically reduce the computation time.

An interesting method, which can be useful to partly overcome the computational problems of LP, is that relating to the graph theory, which has been extensively used in SDC literature, especially for applications involving cell suppression in two-dimensional tables (Carvalho et al., 1994; Cox, 1980 and 1995). Chowdury et al. (1995) and Roehrig et al. (1998) propose network models as a practical tool of disclosure auditing for linked tables. The method can be applied only to 3-way tables, because it "takes advantages of the structure that results from the interlocked nature of the three-dimensional disclosure problem" (Roehrig et al., 1998; sect. 5). Therefore, even if the procedure seems promising, for the time being it can be applied in a limited number of real situations.

3. THE SHUTTLE ALGORITHM

In order to get a more general methodology, Buzzigoli and Giusti (1996) proposed an alternative algorithm deriving the solution from an iterative procedure that – at each step – finds the lower (or upper) bounds, using the marginal distributions and the array formed by the upper (or lower) bounds computed in the previous step.

In order to formalise the algorithm, we introduce the following notation. The original data set can be considered as an n-way array of non-negative integer values with generic element $x_{i_1,i_2,...,i_n}$ representing the frequency of the records which present the modality $i_1$ for the first variable, the modality $i_2$ for the second variable, and so on.

Given the n-way array, a finite number of marginals is consequently defined, the dimensions of which go from n–1 to 0:

- the (n–1)–dimensional marginals: $x_{+,i_2,...,i_n}$
- the (n–2)–dimensional marginals: $x_{+,...,+}$
- and so on.

$x_{+,i_2,...,i_n}$

$x_{+,+,...,+}$
the \((n-2)\)-dimensional marginals:
\[ x_{+,i_1,i_2,...,i_n}, x_{+,i_1,i_2,...,i_n}, ..., x_{+,i_1,i_2,...,i_n}, \]
\[ x_{i_1,i_2,...,i_n}, ..., x_{i_1,i_2,...,i_n}, \]

- ...

the unidimensional marginals:
\[ x_{+,i_1,i_2,...,i_n}, x_{+,i_1,i_2,...,i_n}, ..., x_{+,i_1,i_2,...,i_n}; \]

- the scalar representing the general total:
\[ x_{+,i_1,i_2,...}; \]

Each marginal represents the projection of the original points on the subspace defined by the \(n-m\) variables (with \(m \leq n\)) which are not included in the marginal. For instance:

\[ x_{+,i_1,i_2,i_3,...,i_n} = \sum_{i_1,i_2,i_3,...,i_n} x_{i_1,i_2,i_3,...,i_n} \]
\[ x_{+,+,+,i_3,...,i_n} = \sum_{i_1,i_2,i_3,...,i_n} x_{i_1,i_2,i_3,...,i_n} \]

etc..

The tables released on a particular survey are nothing more than a set of marginals of the \(n\)-way array representing the original data.

The algorithm calculates lower and upper bounds of the elements of an \(n\)-way array with generic element \( x_{i_1,i_2,...,i_n} \), starting from the complete set of its \((n-1)\)-way marginal distributions:

\[ x_{+,i_1,i_2,...,i_n}, x_{+,+,+,i_1,i_2,...,i_n}, ..., x_{i_1,i_2,...}; \]

The procedure is based on the following logic:

- the upper bound of the generic element cannot be greater than the lowest difference between each of its marginals and the sum of the lower bounds of the other elements along the same dimension (row, column, etc.); to start the procedure the lower bounds are all set to zero;
- the lower bound of the generic element cannot be less than the highest positive difference between each of its marginals and the sum of the upper bounds of the other elements along the same dimension (row, column, etc.); if no difference is positive the lower bound remains zero;
- if some of the lower bounds are greater than zero the previously computed upper bounds could obviously change; revised upper bounds imply the revision of the previously computed lower bounds and so on.
In order to formalise this logic, we consider two $n$-way arrays $X^{U_i}$ and $X^{L_i}$, which are the arrays containing, respectively, the upper and the lower bounds of the elements of $X$ computed at the $i$-th iteration. Their generic elements are $x^{U_i}_{i_1,i_2,\ldots,i_n}$ and $x^{L_i}_{i_1,i_2,\ldots,i_n}$.

The algorithm is based on two steps which are repeated alternatively: each step uses the matrix $X$ and either of the matrices $X^{L_i}$ or $X^{U_i}$, to calculate, respectively, the upper or the lower bound of each element of the original array. The lower bounds can be used to revise upper bounds (repetition of the first step) and the revised upper bounds to recalculate new lower bounds (repetition of the second step), and so on.

The procedure is therefore iterative and can be implemented starting from a lower limit equal to 0 and an upper limit equal to the general total of counts $x_{++\ldots}=N$.

The algorithm was named “shuttle” because it alternatively jumps from one array to the other calculating $X^{U_i}$ at every odd step and $X^{L_i}$ at every even step.

In summary, the algorithm can be described as follows:

1st step:

$$x^{L_s}_{i_1,i_2,\ldots,i_n} = \max \left( x^{L_{s-1}}_{i_1,i_2,\ldots,i_n}, x_{++\ldots} + \sum_{i \neq i_1} x^{L_{s-1}}_{i,i_2,\ldots,i_n}, \ldots, x^{L_{s-1}}_{i_1,i_2,\ldots,i_n} \right)$$

2nd step:

$$x^{U_s}_{i_1,i_2,\ldots,i_n} = \min \left( x^{U_{s-1}}_{i_1,i_2,\ldots,i_n}, x_{++\ldots} - \sum_{i \neq i_1} x^{U_{s-1}}_{i,i_2,\ldots,i_n}, \ldots, x^{U_{s-1}}_{i_1,i_2,\ldots,i_n} \right)$$

where $x^{L_0}_{i_1,i_2,\ldots,i_n} = 0$ and $x^{U_0}_{i_1,i_2,\ldots,i_n} = x_{++\ldots}$.

It obviously stops when $X^{U_s} = X^{U_{s-1}}$ or $X^{L_s} = X^{L_{s-1}}$.

As a simple example to show the use of the algorithm, suppose to have the
following marginals of a 2x2x2 array:

\[
\begin{array}{cc|cc|cc}
\text{row totals} & \text{column totals} & \text{layer totals} \\
13 & 13 & 14 & 12 & 11 & 14 \\
12 & 11 & 16 & 7 & 19 & 5 \\
\end{array}
\]

The shuttle algorithm gets the solution in three steps.
In the first step we obtain the array \( X^{U_i} \):

\[
\begin{array}{cc|cc|cc}
\text{row totals} & \text{column totals} & \text{layer totals} \\
11 & 12 & 13 & 5 & 11 & 7 \\
13 & 5 & 11 & 5 & 11 & 5 \\
\end{array}
\]

The second step produces the following \( X^{L_i} \) array:

\[
\begin{array}{cc|cc|cc}
\text{row totals} & \text{column totals} & \text{layer totals} \\
1 & 7 & 8 & 0 & 5 & 2 \\
1 & 7 & 8 & 0 & 6 & 0 \\
\end{array}
\]

which contains some elements different from zero. This implies that \( X^{U_i} \) must be revised.

The third step produces the final estimates \( X^{U_2} \) for the upper bounds:

\[
\begin{array}{cc|cc|cc}
\text{row totals} & \text{column totals} & \text{layer totals} \\
6 & 12 & 13 & 5 & 10 & 7 \\
13 & 5 & 11 & 5 & 10 & 7 \\
\end{array}
\]

It can be shown that the algorithm always converges in a finite number of steps (Buzzigoli and Giusti, 1998; sect. 3). Buzzigoli and Giusti (1998; sect. 4) proved that, for specific cases of relevance, the dimensions of the original array bound the number of steps needed to reach a stop. In the case of a unidimensional array (i.e. a vector), the lower bounds are all equal to \( x^{L_i} = x^{L_0} = 0 \forall i \) and the upper bounds are all equal to the total of counts \( x^{U_i} = x^{U_0} = x_{+ \forall i} \). For an \( m \times n \) array the shuttle algorithm needs almost two steps to get to the solution: the first step to find \( X^{U_i} \), the second one to find \( X^{L_i} \). The repetition of the first step is not necessary, because
\( X^{U_2} = X^{U_1} \).

For 3-way arrays the algorithm surely reaches the stop after 3 steps in the 2×2×2 case only. For a general \( l \times m \times n \) array this is not the case. For instance, for the 2×6×6 array with the following marginals:

<table>
<thead>
<tr>
<th>row totals</th>
<th>column totals</th>
<th>layer totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 2 2 2 2 1</td>
<td>1 2 2 2 2 2</td>
<td>1 0 0 0 0 0</td>
</tr>
<tr>
<td>1 2 2 2 2 0</td>
<td>0 2 2 2 2 1</td>
<td>2 2 0 0 0 0</td>
</tr>
</tbody>
</table>

the algorithm stops after seven steps, and, in particular, identifies exactly the original array:

| 1 0 0 0 0 0 |
| 1 1 0 0 0 0 |
| 0 1 1 0 0 0 |
| 0 0 1 1 0 0 |
| 0 0 0 1 1 0 |
| 0 0 0 0 1 1 |

because the final lower and upper bounds coincide.

From a computational point of view the shuttle algorithm has some evident advantages. First of all, it is very easy to implement with a matrix language that allows the definition of \( n \)-way arrays, even if the use of a traditional compiled programming language is more effective. Secondly, it is very fast. Thirdly, it has limited storage and memory requirements: the storage occupation for data, results and working data, is mostly represented by the marginal distributions (given data) and by \( n+3 \) arrays of the same dimensions of the unknown array (where \( n \) is the
number of the dimensions of the original array, two arrays represent the results, the other \( n+1 \) arrays are working data). Finally, it necessarily gives integer solutions, unlike LP methodology.

The algorithm could be useful also in other problems of statistical disclosure control, as in the cell suppression problem, where it is necessary to check the possibility of calculating the value of the suppressed cells starting from the released information. For this purpose, the algorithm must be initialised with

\[
x_{i_1,i_2,\ldots,i_n}^L = x_{i_1,i_2,\ldots,i_n}^U = x_{i_1,i_2,\ldots,i_n}^0,
\]

for the cells that are not suppressed.

4. SOME LINKS TO STATISTICAL ASPECTS

The subject we presented in the preceding section has interesting links with traditional statistical aspects. The problem of finding lower and upper bounds for the cells of a contingency table given a set of subtables can be converted into the problem of finding the probability distribution corresponding to some given marginal distributions (Fréchet, 1951). In the past, relevant contributions appeared on this subject from the Italian school (Bonferroni, 1936; Rizzi, 1957; Dall’Aglio, 1960 e 1961). Recently, the interest in this topic led to new developments, regarding, in particular, the case of multi–dimensional marginals (just a few references are: Dall’Aglio et al., 1990; Rüschendorf, 1991; Genest et al., 1995).

Many of these contributions focus on the problem of the existence of the multi–dimensional probability distribution, once a certain number of marginal distributions is given. In our context this is not a problem, because the various marginals are surely compatible, as they derive from the same original array. In this sense our problem is simpler, and focuses on computational issues.

Fienberg (1998) proposes an original statistical reading of linked tables disclosure problems, presenting \( m \)-dimensional Fréchet and Bonferroni bounds for \( k \)-way tables (recently analysed by Rüschendorf, 1996 and Galambos and Simonelli, 1996). The author reinterprets the Fréchet and Bonferroni equations in terms of counts instead of probabilities, and shows some interesting results that are connected to our analysis.

In particular, he uses a ‘combined’ solution mixing Fréchet and Bonferroni equations, in order to get sharp bounds. In the case of a 3–way table, given all the 2–way marginals, this combination leads to the following solution (Fienberg, 1998; sect.6):

\[
x_{ijk}^U = \min(x_{ij+}^+, x_{i+k}^+, x_{+jk}^+, x_{ijk} + \bar{x}_{ijk})
\]  (1)
\[ x_{ijk}^L = \max(x_{+++} - S_{1[ijk]} + S_{2[ijk]} - \min(\bar{x}_{ij^+}, \bar{x}_{i+k^+}, \bar{x}_{+jk^+}), 0) \]  

where:

\[ S_{1[ijk]} = x_{i++} + x_{+j} + x_{+++} \]
\[ S_{2[ijk]} = x_{ij^+} + x_{i+k} + x_{+jk} \]

\( \bar{x}_{ij} \) is the diagonally complementary count opposite \( x_{ijk} \) in the 2³ table formed by collapsing all the remaining categories for each of the three variables in the table into a single complementary category,

\( \bar{x}_{ij^+}, \bar{x}_{i+k^+}, \bar{x}_{+jk} \) are the 2-way marginals of the diagonal complementary element.

It is very easy to find the same result with the shuttle algorithm; without loss of generality, from now on we refer to the element \( x_{111} \).

In the 2×2×2 case, we have (Buzcigoli and Giusti, 1998; sect. 4.4):

\[ x_{111}^U = \min(x_{11}, x_{11}^+, x_{+11}, x_{+11}^+, x_{+11}^+, x_{222}) \]
\[ x_{111}^L = \max(0, x_{111} - x_{222}, x_{111} - x_{11}, x_{11} - x_{22}) \]

If we have an \( l \times m \times n \) table, we can collapse it with respect to \( x_{111} \) obtaining the 2×2×2 matrix \( X_{222} \):

\[
\begin{array}{cc}
X_{111} & \sum_{k \neq 1} X_{11k} \\
\sum_{j \neq 1} X_{1j} & \sum_{j \neq 1} \sum_{k \neq 1} X_{1jk}
\end{array}
\]

\[
\begin{array}{cc}
\sum_{i \neq 1} X_{i1} & \sum_{i \neq 1} \sum_{k \neq 1} X_{i1k} \\
\sum_{i \neq 1} \sum_{j \neq 1} X_{ij} & \sum_{i \neq 1} \sum_{j \neq 1} \sum_{k \neq 1} X_{ijk}
\end{array}
\]

Therefore, applying the above expressions to the collapsed table we get:

\[ 222 X_{111}^U = \min(x_{11^+}, x_{11^+}, x_{++1}, X_{11}^+, X_{111}^+, \sum_{i \neq 1} \sum_{j \neq 1} \sum_{k \neq 1} X_{ijk}) \]  

(3)

\[ 222 X_{111}^L = \max(0, x_{111} - \sum_{i \neq 1} \sum_{k \neq 1} X_{i1k}, X_{111} - \sum_{j \neq 1} \sum_{k \neq 1} X_{1jk}, X_{111} - \sum_{i \neq 1} \sum_{j \neq 1} X_{ij}) \]  

(4)

When \( i = j = k = 1 \) the expression (1) is obviously equal to the expression (3), while some simple algebra is needed to show that expressions (2) and (4) coincide. For \( x_{111} \), expression (2) becomes:

\[ x_{111}^L = \max(x_{+++} - S_{1[11]} + S_{2[11]} - \min(\bar{x}_{11}, \bar{x}_{1+}, \bar{x}_{++}), 0) \]
where:

\[ x_{+++} - S_{1[11]} + S_{2[11]} = x_{111} + \bar{x}_{111} = x_{111} + \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} x_{ijk} \]

\[ \bar{x}_{11+} = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} x_{ijk} \]

\[ \bar{x}_{+1+} = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} x_{ijk} \]

\[ \bar{x}_{+++} = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} x_{ijk} \]

It follows that if, for example, \( \min (\bar{x}_{11+}, \bar{x}_{1+1}, \bar{x}_{+11}) = \bar{x}_{11+} \) than:

\[ x_{+++} - S_{1[11]} + S_{2[11]} - \bar{x}_{11+} = x_{111} + \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} x_{ijk} - \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} x_{ijk} = x_{111} - \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ijl} \]

that is the last part of expression (4). Analogous proofs are possible when the minimum is \( \bar{x}_{1+1} \) or \( \bar{x}_{+11} \).

Repeating the procedure for the other elements of the original 3-way array, different from \( x_{111} \), it is possible to find all the Fienberg’s lower and upper bounds.

It can be shown that applying the shuttle algorithm to collapsed tables of higher dimensions it is possible to reproduce the alternating sequence of signs implied by the Bonferroni inequalities (Fienberg, 1998; sect. 5).

The two approaches, although independently developed and deriving from different theoretical foundations, reach the same results. The approach proposed by Fienberg gets explicit expressions for the bounds, using interesting links with the probabilistic framework of the problem. The shuttle algorithm gets a simple iterative procedure to find the bounds, and therefore can also be seen as a useful tool to implement the automated calculus of the combined Fréchet–Bonferroni bounds in a rapid and efficient way, filling a gap in this field of application.

5. CONCLUDING REMARKS

The problem of disclosure auditing in a set of tables linked over some variables is not yet completely solved. Linear programming may be computationally burdensome when the dimensions of the released tables become large. On the other hand, the national statistical institutes work with large tables and arrays: efficient heuristics are therefore needed, which can determine the lower and upper bounds of the unknown original array starting from the released tables within a reasonable time and with a reasonable computational effort.
The various proposals in the literature are not satisfactory, as they usually refer to limited dimension cases, which are not of great interest for official producers of statistics.

The new algorithm we presented seems to have interesting properties that could be exploited in this as well as in other fields of application. Many issues involve this kind of computation; moreover other problems, currently solved with different methodologies, could be brought back to this algebraic definition (i.e. to calculate the lower and upper bounds of the elements of an array given its marginals).

The algorithm is very efficient and its speed increases with the increasing of array dimensions; for example, with a 9×9×9 array the computing time, with a ‘Pentium 200’ processor, is less than one second. It is very easy to implement on a personal computer and has a low requirement of memory space.

Finally, its use is interesting not only for practical applications, but also for theoretical developments, because it seems to have relevant links with some probabilistic and statistical issues.

REFERENCES


PROBLEMI DI CONTROLLO DELLA RISERVATEZZA PER ‘TABELLE COLLEGATE’

Riassunto

Il lavoro presenta, in un contesto generale, il problema statistico del controllo della riservatezza nel caso di ‘tabelle collegate’ e sottolinea alcuni importanti aspetti di questa tematica. In particolare, dopo una breve rassegna dei metodi esistenti nella letteratura per trattare questo problema, viene proposto l’utilizzo di un nuovo algoritmo, che mostra interessanti proprietà e possibilità di applicazione. Successivamente sono presentati alcuni collegamenti a tradizionali aspetti statistici.