Using Tau-Argus and sdcTable to Conduct Secondary Cell Suppression for Linked Tables

Amang Sukasih, Donsig Jang, David Edson
Mathematica Policy Research, 600 Maryland Ave., S.W., Suite 550, Washington, DC 20024

Abstract

When a data cell in a table is suppressed by dropping its value based on a primary cell suppression rule, the value of that cell can still be determined if the table, subtable, or linked tables provide totals, marginal totals, or subtotals. Secondary cell suppression is therefore needed to avoid such disclosures. Two software packages are available to assist researchers with secondary cell suppression: Tau-Argus (Statistics Netherland 2009) and R-statistical package sdcTable (Meindl 2010). Just recently, these two software programs have included the option to perform protection for linked tables. We explored the capabilities of the two programs in performing linked-table suppression, identifying the strengths and limitations of each and comparing the results.

Key Words: tabular data, sensitive cells, statistical disclosure limitation

1. Disclosure Limitation in Tabular Data

When data collected from a sample survey are disseminated either in the form of tabular data or public use microdata, the data producer often needs to protect the confidentiality of the respondents who provided the information. Confidential information may include identity of the respondents as well as information about them. The Federal Committee on Statistical Methodology, in their Statistical Policy Working Paper 22 (2005), summarized three types of data disclosure discussed in Duncan et al. (1993, pp. 23-24) as follows:

> "**Disclosure** relates to inappropriate attribution of information to a data subject, whether an individual or an organization. Disclosure occurs when a data subject is identified from a released file (**identity disclosure**), sensitive information about a data subject is revealed through the released file (**attribute disclosure**), or the released data make it possible to determine the value of some characteristic of an individual more accurately than otherwise would have been possible (**inferential disclosure**)."

To avoid such disclosure, data producer develop rules and procedures to protect confidentiality, and implement these rules to their tables or microdata files prior to publishing the tables or releasing the data. Confidentiality protection rules may vary from data to data and from agency to agency—or even from table to table within the same data source.

This paper focuses only on disclosure limitation for tabular data. Some statistical disclosure limitation (SDL) techniques that include specifications to identify sensitive

cells and how to protect those sensitive cells will be discussed, as well as SDL techniques for linked tables and availability of software to perform SDL in the linked tables.

1.1 Identifying Sensitive Cells

A cell in a tabular data can present count/frequency data that represent the number of respondents who fall into this cell (for example the number of people with a certain disease), or magnitude data, which aggregate values of a particular variable from all respondents in that particular cell (for examples total assets or mean income). Throughout this paper, a cell that potentially discloses confidential information and therefore needs to be protected is called a sensitive cell. In tabular data, conditions where confidential information risks disclosure include small cells, cells with high contribution from only a few cases, and cells where external information is available that could be used to disclose confidential information. In understanding such disclosure risks and identifying sensitive cells, readers need to consider a range of intruder scenarios that lead to disclosure. For example, a small cell with fewer than three to five respondents may be a disclosure risk, as the identity of those respondents may be easily discovered—especially when the respondents within the cell correspond to rare cases or the extreme cases in a skew distribution. In another scenario, a respondent or coalition of respondents who belong to this cell could become the intruder who can disclose the identities of other respondents within the cell. Hundepool et al. (2010) provide many examples and illustrations of intruder scenarios. The following are systematic techniques that may be used to identify the sensitive cells.

a. Threshold rule or minimum frequency rule

The cell is considered sensitive if the cell frequency is less than a pre-specified threshold value, say n. A most common value of n is either 3 or 5. The choice of n often depends on the type of reporting unit, as well as the sensitivity of the information presented. For examples, if the reporting unit is by business entity rather than by individual person, a larger n may be required. Similarly, a count of people by certain type of disease may require a larger n than that one by level of education.

b. (n, k) rule or dominance rule

The cell is considered sensitive if the sum of n largest contributions in the cell exceeds k percent of total value for that cell. For example, in an (n = 3, k = 80) rule, if the cell total value is $10,000 and the top three respondents individually reported $3,200, $3,000, and $2,170, which sum to $8,370, then this cell is sensitive: the sum is greater than 80 percent of the total cell value. Note that this rule does not depend on the cell size, but rather is based on domination or concentration of the respondent contributions.

c. p percent rule

The cell is considered sensitive if p percent of the largest contribution is larger than or equal to the cell total value minus the two largest contributions. Similar to the dominance rule, this rule does not depend on the cell size. It is based on an intruder scenario in which the second largest respondent can use the cell value to estimate the contribution of the largest respondent. Using the example cell from the previous rule, if we implement an 80 percent rule, 80 percent of the largest contribution (0.8 ×

$3,200 = $2,560) is less than the cell total value minus the two largest contributions ($10,000 – $3,200 – $3,000 = $3,800). Thus, this cell is not sensitive under this rule.

d. *p*/*q* percent rule

If the intruder has a prior knowledge of the total value from contributions outside the two largest contributions to within *q* percent, and if the total cell value minus the two largest contributions minus that value of prior knowledge is less than *p* percent of the largest contribution, then the cell is considered sensitive. In this situation, the intruder scenario is that the second largest contributor with prior knowledge about all other smaller contributions, or at least able to estimate them within *q* percent, will be able to estimate the value of the largest contribution to within *p* percent. For example, for the cell value $10,000 with the largest value $3,200 and the second largest value $3,000, suppose the second largest respondent has an estimate of the aggregate smaller values to be $3,040 (this is actually 80 percent of the true value $ 3,800, which is $10,000 – $3,200 – $3,000). Using this estimate, he can subtract his contribution and the estimate of other contributions from the cell value to estimate the largest contribution; that is, $10,000 – $3,000 – $3,040 = $3,960, which overestimates the true value by ($3,960 – $3,200)/$3,200 = 24 percent. In this particular cell example, if one implements a *p*=80/*q*=80 percent rule, since total cell value minus the two largest contributions minus 80 percent of other contributions is $10,000 – $3,200 – $3,000 – (0.8 × $3,800) = $760, which is less than 80 percent of the largest contribution (0.8 × $3,200 = $2,560) then this cell is sensitive.

1.2 Protecting Sensitive Cells

Once the table format has been fixed and the cell values tabulated (with no further table redesign, recoding of categories, or collapsing of cells), and the sensitive cells have been identified, the table can be protected by implementing the SDL techniques, including perturbation, cell suppression, or control tabular adjustment.

a. Perturbation

In these techniques, the true cell values are protected by either rounding (up or down) the cell values to a specific base, or perturbing the cell values by adding or multiplying with some chosen value. The goal of protection is that the cell can still be published but the intruder no longer finds the true value in the published table. This paper will not discuss methods in this group. Readers can see Hundepool et al. (2010) and Federal Committee on Statistical Methodology (2005) for more details.

b. Cell suppression

In this method, sensitive cells are simply dropped/suppressed (not published) to protect confidentiality. Cells that are identified as sensitive based on the sensitivity rules discussed previously and then dropped are called the primary cells. However, simply dropping the values of the sensitive cells will not completely protect them when marginal totals of these cells are published, because an intruder may recalculate the dropped values by way of simple subtraction. Therefore, to completely protect sensitive cells, one or more nonsensitive cells (called secondary or complementary cells) must be suppressed as well. The most common way is that for each primary

suppressed cell there should be at least one secondary suppressed cell in the same row and one secondary suppressed cell in the same column. Note, however, that for each primary cell suppressed, there are many possible choices of secondary cells. Also, it may still be possible for the intruder to compute a range (sensitivity interval) in which the suppressed cells lie. This is motivation to find secondary cells that maximize disclosure limitation and minimize information loss. The method to address this objective becomes more complicated and involves linear programming (LP) problems. Two common methods for secondary cell suppression are discussed below.

Hypercube Method

For an *n*-dimensional table with hierarchical structure, this method subdivides the table into a set of *n*-dimensional subtables without substructure. For each of these simple tables without hierarchical structure, if we consider secondary cell suppression where in each row and in each column there has to be exactly one secondary suppressed cell, nevertheless, there are still many possible patterns of secondary suppressed cells. The SDL task is then to check whether the sensitivity interval is wide enough and to calculate the loss of information for each pattern of secondary cell suppression.

Successively, for each primary suppression in the current subtable, all possible hypercubes with this cell as one of the corner points are constructed. A cell in a simple *n*-dimensional table without substructure cannot be disclosed exactly if the cell is contained in a pattern of suppressed, nonzero cells, forming the corner of a hypercube. By solving LP problems, the suppression can choose a secondary cell suppression pattern that optimizes sensitivity interval and loss of information constraints. A heuristic approach that does not need LP optimization can be used; the computation can done by generating all candidates of *n*-dimensional hypercubes and selecting the one with minimum loss of information. Willenborg and de Waal (1996) provide details information on how the hypercube method for secondary cell suppression works.

Modular/HiTaS

This technique is also a heuristic approach that implements LP optimization to choose secondary cells. This technique breaks down the hierarchical table into several non-hierarchical tables, protects them using LP-solver, and then composes a protected table from the smaller tables. Detailed information on how this method works can be found in Hundepool et al. (2011) and de Wolf (2002).

1.3 Cell Suppression in Linked tables

Linked tables are defined as two or more tables presenting data on the same response variable and sharing cell(s) from the same category(ies) of at least one explanatory variable. Linked tables can occur across published tables for a particular year, or across years for a particular table for longitudinal data. SDL through cell suppression for linked tables requires an extra rule that if the common cells are suppressed in one table, then they must be suppressed in the other table(s) as well. This adds a level of complication

when the goal is to find optimum protection. De Wolf and Giessing (2009) present several techniques that implement the modular optimization approach, as follows.

a. Complete modular approach

In this technique, first a cover table is created, constructed by crossing all categorical variables used in all linked tables. Then the modular approach discussed earlier is used to protect the complete cover table. This technique may lead to oversuppression, since the modular approach must also protect individual simple subtables even though some of them may not actually be published.

b. Adapted modular approach

This technique also implements the modular approach on a cover table; however, it only considers those subtables that are also subtables of at least one of the specified linked tables. It disregards the others; that is, any simple subtable that is not a subtable of any of the linked tables is skipped.

c. Linked subtables modular approach

This technique is a more complex approach dealing directly with linked subtables at the same time.

d. Traditional approach

This technique utilizes iterative backtracking procedure that uses suppression results from one table in a previous iteration as the conditional input (suppression status) to suppress the other table in the next iteration. Below is the procedure described in de Wolf and Giessing (2009) for two linked tables T1 and T2:

1. Protect table T1 on its own.
2. Each cell in T2 that is also present in T1 will get the status (that is, suppressed or not suppressed) of the cell in the protected table T1.
3. Table T2, with the additional suppressions carried over in step 2, is protected on its own.
4. Each cell in T1 that is also present in T2 will get the status of the cell in the protected table T2.
5. Repeat steps 1–4 until no changes occur in protecting table T1 or in protecting T2.

The following figure illustrates this process in two linked tables, T1 and T2.

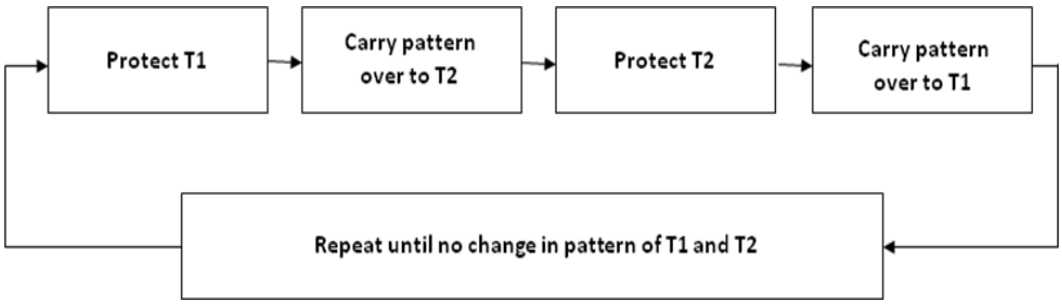

Figure 1: Traditional approach to protect two linked tables using backtracking procedures

2. Software for Secondary Cell Suppression: Tau-Argus and sdcTable

The development of SDL approaches has been driven primarily by the confidentiality requirements for statistics produced by government statistical agencies. Hence the development of software for performing the SDL has been based on individual agency needs. Typically, the software is limited to the specific agency problem and solution in hand, is run on a specific agency platform, or has limited documentation. The software is proprietary to the agency and often limited to use by a specific group within the agency; sometimes the only ones who know how to use it are the programmers themselves. When the software is used outside the agency, it is usually used only by other governmental agencies, under interagency agreement. Very little SDL software is available for public use; however, this paper discusses two SDL software programs that are currently available. About a decade ago a consortium of statistical agencies from several European countries developed an SDL software program named Tau-Argus that has now become available for public use. Another publicly available software program, sdcTable, has been developed by Bernhard Meindl of Statistics Austria as a package for *R* statistical software.

2.1 Tau-Argus

Tau-Argus is a freeware that can be downloaded from the following link: http://neon.vb.cbs.nl/casc/tau.htm. Anco Hundepool of Statistics Netherland maintains and updates this software, adding in results from methodological research by people within the consortium. The most recent version available from the link is version 3.5.0 build 6 (September 2, 2011).

Tau-Argus runs in Windows platform only. It is relatively easy to use, with a menu-driven user interface. Its features include the capability to accept either a microdata file or a tabular data file as input. When the input data are in the form of a microdata file, Tau-Argus can perform tabulation while protecting the table at the same time. It also has the capability to recode categories on the fly. In performing SDL, Tau-Argus can also put an individual cost to each cell, so that the measure of quality-protected table or loss of information can be based on this cost. A useful feature, especially when implementing the backtracking procedure, is the ability to keep track of suppression history and to use pre-assigned cell suppression status as the input in performing SDL. All sensitivity rules

and SDL methods discussed earlier in Section 1, including suppression for linked tables, are available in this software.

Though Tau-Argus is a powerful software program for SDL, it has a number of limitations or disadvantages, including:

- It is not flexible in terms of precision of data values involved. Small discrepancies, for example between the marginal sum of individual cell values and their marginal total value, may not be tolerable. Rounding error will cause the software to stop the calculation with an error message.
- The error messages are not always intuitive. There is no help menu or documentation explaining the meaning of error messages or suggesting solutions.
- User support may be obtained only from the author and current maintainer, Anco Hundepool.
- Tau-Argus has a quite complete manual (a pdf file) containing more than 100 pages (the most current is version 3.5); however, this pdf file is not searchable.
- To be able to implement SDL methods that utilize LP solution such as modular and optimal techniques, Tau-Argus requires an external commercial LP-solver, which is not inexpensive.
- Tau-Argus has a common drawback of proprietary software in that there is no way to check, control, modify, or adapt it to the user's needs. For example, the current version of linked tables suppression can only be done by a way of a cover table that is a complete cross-classification of sub-linked tables.

2.2 sdcTable

sdcTable is an SDL package for the statistical software *R* (http://www.r-project.org/). *R* and its packages, including sdcTable, are freeware. One of the advantages of sdcTable is that it is open source software: users can access the source codes of each statistical method. That way, users can study what is going on inside each process, as well as modify the codes to meet any special needs. sdcTable is available for Windows, Linux, and Mac operating system. To be able to use it, however, the user needs knowledge of *R* programming.

Compared to Tau-Argus, sdcTable is relatively new; the development has been underway for only the past 3 or 4 years. For primary cell suppression, the following options are available: threshold method, dominance rule, or p-percent rule. For secondary cell suppression, it only provides Hypercube and HiTaS/modular approaches. One advantage of sdcTable is that LP-solver is also available for free as an *R* package.

The most recent version (0.6.4, April 4, 2011) can perform suppression for linked tables based on Hypercube or HiTaS/modular approach. The technique, however, is limited to the traditional approach using iterative backtracking procedure. The manual is available, but it is written in a very basic *R* documentation style that may not be easily understood by non-users of *R*. User support may be obtained from the *R* community (such as user groups, forums, and so on).

Ichim and Franconi (2009) discuss the sustainability of the SDC software tools. In their work they compared Tau-Argus and sdcTable in terms of software development, documentation, user friendliness, and other features in a table, reproduced below,

Table 1: Features of the available software tools in the current situation

Feature	Tau-Argus	sdcTable
Possibility to check/control/modify/adapt	NO	YES
Coordinated development	YES	NO
Predictable results	YES	NO
Development agenda	YES	NO
SDC people involved	YES	NO
Documentation	YES	YES/NO
Help	NO	YES/NO
Modular architecture	YES	YES
Extensibility	YES	YES
Platform dependent	YES	NO
Unique maintainer	YES	YES
Personalisation	NO	YES
User-friendly	YES/NO	NO
Programming skills required	NO	YES
Free	YES/NO	YES
Open source	NO	YES
Designed for official statistics	YES	NO/YES
Mirror sites	NO	YES
Consortium	YES	NO
Test reports	YES	NO

2.3 Comparison of Linked Table Suppression

a. Input Data

Both Tau-Argus and sdcTable accept either a microdata file or tabular data file as input. In Tau-Argus, when input data are in a microdata file, the user needs to provide a metadata file that specifies each explanatory variable; identifies all response variables, the sample weight variable, and the external file for the code list file; and, if the explanatory variable is hierarchical, includes a file that specifies the structure of hierarchy. When the input file consists of tabular data, it should contain respondent frequency and magnitude data for each cell. In addition, if sensitivity rules based on domination or concentration of the respondent contributions are to be performed, it should contain the largest three contributions in each cell. In Tau-Argus, common cells in linked tables must be a subset of categories in a common variable, code list and hierarchy for the cover table must be present in one of linked tables, and code list and hierarchy in linked tables must be a subset of those in the cover table.

sdcTable has been changing the way it handle input data and data parameters from version to version. In the most recent version, the user needs to specify a variable in common in both files. In sdcTable, common cells may come from different variables as long as the common cells represent the same response variable and the same specific category of explanation variable; the user specifies the common cells. For

example, two variables, OLD_CODING in the first table and NEW_CODING in the second table, represent two different coding systems. Each uses three-digit numbers but aggregation up to two-digit numbers represents common categories.

b. Method

Tau-Argus provides adapted modular approach (optimization using HiTaS) to suppress sensitive cells in linked tables. Linked table suppression using Hypercube is also available. Tau-Argus can handle a set of linked tables with more than two tables; however a limitation is that SDL for linked tables can only be done if the cover table from a set of linked tables is no larger than four-dimensional.

sdcTable provides optimization suppression for linked tables using either the Hypercube or HiTaS approach. Iterative backtracking procedure is used with a stopping criterion to stop the iteration when all common cells have the same suppression status. The current release of sdcTable can only handle a set of two linked tables.

Hence, the different results between Tau-Argus and sdcTable for linked table suppression are due to differential implementation of the approaches used in the two software programs. Nevertheless, one may use sdcTable with the complete modular approach to protect the cover table, and carefully evaluate the results to see if this approach produces oversuppressed tables.

c. Output

When we used Tau-Argus on our sample of two linked tables (one with two explanatory variables and the other with three explanatory variables, with one common variable), both modular and Hypercube methods resulted in the same cell suppression pattern for the first table, and resulted in different cell suppression patterns in the second, more detailed table.

When we performed protection for linked tables using sdcTable on our sample linked tables, the iteration did not converge, so we did not get results. When we looked at the source codes, we noticed that there seemed to be a bug inside the protectLinkedTables function. When we modified the code for this apparent bug, sdcTable resulted in oversuppression in the second table and extreme oversuppression in the first table.

3. Conclusion

Software packages for linked table protection that implement optimization technique are now publicly available, namely Tau-Argus and sdcTable. The methods in these two software packages continue to be developed. Currently Tau-Argus provides a reasonable practical tool if users have access to a commercial LP solver. With a careful evaluation of the results from a complete modular approach, sdcTable may prove to be a useful tool as well.

References

de Wolf, P. P., and S. Giessing. 2009. Adjusting the τ-argus Modular Approach to Deal with Linked Tables. In *Data & Knowledge Engineering*, 68:11, November, 2009.

de Wolf, P. P. 2002. HiTaS: A heuristic approach to cell suppression in hierarchical tables. In *Inference Control in Statistical Databases*, J. Domingo-Ferrer (editor). Springer-Verlag Berlin Heidelberg, 74–82.

Duncan, G.T., T.B. Jabine, and V.A. de Wolf. 1993. Private Lives and Public Policies: Confidentiality and Accessibility of Government Statistics. Committee on National Statistics and the Social Science Research Council, National Academy Press, Washington, D.C.

Federal Committee on Statistical Methodology. 2005. Statistical Policy Working Paper 22 (second version 2005). Report on Statistical Disclosure Limitation Methodology, Statistical and Science Policy, Office of Information and Regulatory Affairs, Office of Management and Budget.

Giessing, S., and D. Repsilber. 2002. Tools and Strategies to Protect Multiple Tables with the GHQUAR Cell Suppression Engine, in *Inference Control in Statistical Databases*, J. Domingo-Ferrer (editor), Springer Lecture Notes in Computer Science, v. 2316.

Hundepool, A., A. Van de Wetering, R. Ramaswamy, P. P. de Wolf, S. Giessing, M. Fischetti, J.-J. Salazar, J. Castro, and P. Lowthian. April 2011. τ-ARGUS User's Manual, Version 3.5.

Hundepool, A., J. Domingo-Ferrer, L. Franconi, S. Giessing, R. Lenz, J. Naylor, E. S. Nordholt, G. Seri, and P. P. de Wolf. January 2010. *Handbook on Statistical Disclosure Control*. A Network Excellence in the European Statistical System in the Field of Statistical Disclosure Control (ESSNet SDC).

Meindl, B. 2011. Package 'sdcTable,' version 0.6.4, April 4, 2011.

Repsilber, D. 1999. "Das Quaderverfahren." In *Forum der Bundesstatistik*, Band 31/1999.

Willenborg, L., and T. De Waal. 1996. *Statistical Disclosure Control in Practice*. Springer Lecture Notes in Statistics. v. 111.