Modernizing Cell Suppression Software at the U.S. Census Bureau

Paul B. Massell

Center for Disclosure Avoidance Research, U.S. Census Bureau,

Washington, D.C. 20233; paul.b.massell@census.gov

Abstract[1]

An Economic Census of the United States is conducted every five years by the Economic Directorate of the U.S. Census Bureau. The main data products are additive magnitude data tables that typically involve NAICS (North American Industry Classification System) categories as rows and geographic entities as columns (some tables have a 3rd dimension). 'Sensitive' cells are those cells in the table that cannot be published 'as is' due to confidentiality concerns with respect to the companies whose values contribute to the cell value. The p% rule is used for determining which cells are sensitive and how much protection each such cell requires. First, the sensitive cells are suppressed. Then a cell suppression program is run against a file with information about each cell including an identifier for each establishment that contributes to the cell value, its associated company, and the contributed value. This program calls an optimization routine for each sensitive cell in order to find the optimal set of additional cells that must be suppressed in order to find the set of cells with the minimum total value, that, when suppressed, lead to a protection of the sensitive cells at the company level. We discuss a number of complex aspects of the software and how each of these was modernized.

Key Words: Cell Suppression, Sensitive Cells, Protection at the Company Level

[1] This report is released to inform interested parties of ongoing research and to encourage discussion of work in progress. The views expressed on statistical issues are those of the authors and not necessarily those of the U.S. Census Bureau.

1. Introduction

Assume that cell suppression is being used to protect the confidentiality of the contributors to the cell values in a magnitude data table.

The following are requirements for our cell suppression that are beyond those typically found with cell suppression for magnitude data tables.

1. A company consists of one or more establishments, typically at different locations throughout some region. For statistical purposes, the key fact is that we have data for each establishment for a given company. These establishment values form the contributions to cell values. Thus a given establishment contributes to at most one cell of a table, but if there are k establishments for that company, those k establishment values may contribute to as many as k different interior cells in the table. The basic type of protection done in cell suppression involves protecting only the individual contributions to cell values; i.e., it does not involve protecting sums of associated values. For economic data this is called 'protection at the establishment level'. The type of protection we are required to do is more complicated; it involves the basic type of protection described above, plus protection of the sums of establishment values associated with each company. This type of sum-level protection is often called, for economic data, 'protection at the company level'. 'Protection' of these data involves suppressing carefully chosen cell values, so that estimation of the cell contributions (or sums of such associated with a given company) cannot be made better than some accuracy threshold.
2. Protection of linked Tables. 'Linked tables' here refers mainly to tables generated from the same microdata file that have some cells in common, i.e. tables that overlap (e.g., tables that have a column in common). Tables can also be linked via additive relations (e.g., [Table1] = 3*[Table2] + 5*[Table3].)
3. Use of an audit program that can model linked tables to determine if full protection of sensitive cells has been achieved, when theory related to the mathematical program does not guarantee full protection of establishment or company values, e.g., when backtracking is used. The audit program should, ideally, test protection of contributions rather than just cell values.

1.1 Algorithm Improvements being Developed to Meet those Requirements

1. We have defined a new structure called a 'supercell' which will improve the protection of sensitive cells in certain situations. A 'supercell' is defined as the union of all interior primaries, along with the set of all secondaries, which exist in an additive constraint. The secondaries (may) exist because we assume that all primaries have been protected. (Sometimes the geometric term 'shaft' is used in place of 'additive constraint'; 'interior' means we are excluding primaries or secondaries that may exist for the marginal (i.e., sum cell) of the constraint). In a simple 2-dimensional ('2d') table, a 'supercell' would be the cell-union of all primaries and secondaries in a given row or in a given column. A 'cell-union'

takes into account the contributions to each cell and sums establishment contributions into company sums. The cell-union thus consists of company sums which may be tested for sensitivity in the same way we apply the p% rule to an ordinary cell (except that for the latter, the company contributions are typically coming from a single establishment). The team realized we needed some secondary type of protection to protect cell-unions that were not being (fully) protected by the traditional single primary protection.

2. We are developing the capability for the program to read in two or more linked tables and protect the resulting table group as a single entity with one pass through the primaries contained in them. This reduces the amount of backtracking and in some cases, may even eliminate it. By 'backtracking ', we mean the aspect of the optimization process that often requires more than one 'visit' to a table to protect its sensitive cells. These revisits are needed because the protection need of some of the sensitive cells is affected by the protection need of those same cells in other tables. The goal of backtracking is to ensure that each sensitive cell has consistent protection in all tables in which it appears.

1.2 Software Procedures Being Used To Direct Algorithm Development

1. We are documenting weaknesses in the current production program. We will continually revisit the list of these needs to ensure the new program eventually overcomes all these weaknesses.
2. We are developing 'functional specifications' for each major feature of the new program (e.g., data preparation, optimization model, solution strategy, and backtracking). This will help with modifications of the program in the future. A programmer will be able to modify most aspects of the program even without a good understanding the optimization process details.
3. We are developing easier input for users running the program and more informative output. The easier input may be interactive and allow for users to specify various quantities, such as the value of 'p' used for the sensitivity rule and the associated uncertainty intervals that create protection 'flow'.
4. The new software will be written in a modern language such as C++ that is object oriented and allows for dynamic allocation of data structures. This will allow for creation of complex data structures and easier use of direct access files. One direct access file serves as the database for all cells in the microdata (e.g., associated with an economic census sector). During the course of the program, the protection status of each cell is updated for each cell that was part of the suppression pattern for the previously protected target primary. This means when a cell becomes used as a secondary and a certain amount of 'protection flow' passes through it, the suppression flag (a 'C' for a secondary) and the 'flow' must be stored in the database.
5. We are considering use of a mathematical modeling language (e.g., AMPL, SAS OPTMODEL, CPLEX OPL) to invoke a linear programming solver (e.g., Cplex, Gurobi) to solve the optimization problem involved in finding an optimal 'flow'.

This is a much simpler way of describing a linear program compared to the solver specific methods which involve many software details (e.g., calls to subroutine for each subtask). This way of describing models simplifies the code, thus reducing the chance of a programming error and making the code easier to understand to those with knowledge of optimization models but little programming background.

2. Company Level Protection

2.1 Protection of Sums of Related Contributions in Different Cells

Why is the protection of economic data so difficult ? We often say we are protecting sensitive **cells**, but the real quantity we are trying to protect is not a cell value, but the underlying **contributions** to cell values. Then there is the issue of 'company protection' described above, which is easy to state but not so easy to implement.

In the early 1980's, Larry Cox, then a researcher at the Census Bureau, developed a notion of the 'capacity' of a cell X to protect a target primary suppression P, whose main purpose was to provide company level protection. For each cell X, in the table, we form the 'cell-union' X and P and compare the protection need of that union with the protection requirement for P alone. The reduction in the protection required of the cell-union compared to the requirement for P alone was then called the 'capacity of X to protect P'. The computation of this quantity is a bit complicated since at least five cases need to be considered. But even with this number of cases, the protection measure is not complete because there is an inherent limitation when viewing protection as a property of only two cells at a time. This was the main motivation for the following idea.

2.2 Supercells: A Way to Improve the Current Approach to Protection

Supercells can prevent under-suppression when there are primaries P1 and P2 in some additive constraint (e.g., a row or column) each of which is protected by various other suppressions in the constraint but the cell-union U[P1;P2] (abbr. 'U12') is not protected. Supercells measure the 'additional' protection needed by the union of all the primaries in an additive constraint. When we protect a supercell, all data in contained in the P's of the supercell, is protected at the company level. More generally, supercells prevent under-suppression that might occur when 2 or more P's exist in a (one-dimensional) additive relation; i.e. a 'shaft'. {See Appendix SC for an example of this type of protection}. The Census Bureau version of this idea was developed by Jim Fagan and other members of the R&M team while doing requirements analysis for the new program. Statistics Canada has developed a similar idea.

Supercells also can be useful in preventing a type of over-suppression when 3 or more P's in some constraint protect each other as a group but not as pairs. E.g., if there are 3

P's in a row, each with a contribution from a single company and the companies are distinct. Say, the contributions are 100, 60, and 30. Then the cell union of P2 and P3 protect P1, but neither P2 nor P3 alone protects P1. A similar situation holds for the other contributions.

2.3 Remark on Conceptual Tradeoffs

Consider the two concepts discussed above that are ways of using company contribution data rather than just cell values to improve the quality of protection. They are 'supercells' and 'capacity'.

The notion of supercell that the Census Bureau has developed is not as complicated as that implemented by Statistics Canada. However, their more complicated definition of this allows them to use a simpler definition of capacity.

3. Linked Tables

3.1 Ways to Protect Linked Tables

There is a problem that arises frequently when tables overlap (i.e., have cells in common). We need to ensure that a given sensitive cell that appears in 2 or more tables is fully protected in all these tables.

Approach 1. Backtracking

Problem: The current implementation involves complex programming, which, despite its complexity, may not fully protect all suppressions that appear in the intersection of tables. In addition, backtracking often adds considerable run time because suppressions in certain intersections may need to be 're-protected' several times.

Approach 2. Table Groups

The idea is to combine linked (i.e., overlapping) tables into a smaller set of combined tables that we call 'table groups' that create a partition of the full set of tables to be published from a given microdata file. Such a formation is always theoretically achievable, but may not be achievable in practice if one or more of the table groups is so large that it takes a very long time to process, even with a fast LP solver. We are currently planning that the grouping of tables into table groups will be done by analysts who are familiar with the data. We suspect that this table group approach will reduce the amount of backtracking in most (large multiple tables) suppression runs, but will not eliminate it. Backtracking will still be needed when excessive run-times are encountered for a large complex table group, necessitating its decomposition into smaller table groups.

3.2 Comparing the Protection Efficiency of Backtracking and Table Groups

We claim the table group approach will fully protect all the primaries that are contained in the table group because the linear programming model generated by a table group is not fundamentally different than that for a single table. However, if run times are excessive for this approach, and we must continue to use backtracking, it would be informative to compare the protection given to each primary in a given table group by each method. It may turn out that backtracking is actually doing a good job of protection in practice, even though it is hard to prove it.

4. Deciding How to Measure Information Loss

4.1 Using a Two Step Optimization Process if there are Two Quantities to Minimize

In the current production suppression program, it is clear from the code and the documentation that there are two quantities that the program is trying to minimize. One is the total **value** of the cells not yet suppressed but being considered for suppression. {This is one measure of information loss}. The other is the total **number** of cells being considered for suppression. The way the current program attempts to minimize both quantities is by doing **two calls to the solver**. The first call uses a cost function that is the best approximation to the total value that can be expressed with the LP variables (viz., the variables that measure 'protection flow'). A perfect expression of that total value cost function requires the use of binary variables; but such variables are not available in LP models. Then, one does a second to the solver using a cost function that gives a much lower weight to cells that can support the entire protection need (because their capacity exceeds the protection need of the primary being protected). This second optimization is done on a reduced set of cells, viz., those secondary suppressions which were found in the first optimization. We are current experimenting with this two step approach to minimizing two quantities.

4.2 Exploring use of more Complex Cost Functions and Multiple Step Optimization

The cost function is one of the easiest components of a linear programming model to modify. Thus if analysts or other table users decide in the future that it would be best to minimize quantities other than total value or total numbers of cells suppressed, this could easily be implemented. Likewise, three or more quantities could be minimized using a multiple step optimization scheme.

5. Special Data Types

5.1 Special Data Types for Economic Data

1. Rounded data.
   Economic data is commonly rounded. The result is typically that the table is not perfectly additive. Fortunately, in the LP models we are using, the actual tabular data is not required to be additive; being nearly additive is sufficient. Only the perturbations to the cell values, as represented by the protection flows, are required to be additive. Another issue is how to extend the standard p% rule to include the extra uncertainty that rounding provides. There are various ways to model the protection provided by rounding. Currently we are using a rounding protection formula based on the fixed interval version of the p% rule, which is the version of the p% rule that the Census Bureau typically uses.

2. Negative Values
   Magnitude data which may be either positive or negative, doesn't seem, at first analysis, to pose a challenge for the protection methods discussed here. However, it turns out there is no simple way to handle it. The first problem that arises is how to extend the standard p% rule, to the case of negative values, in a reasonable way. In order to do that, one needs to develop a data user knowledge model; i.e., reasonable assumptions about what users know about contributions to the table {e.g., do they know the sign of a contributed value, even when they do not know the magnitude (i.e. absolute value) of the contribution ? }. Work on this topic is continuing.

6. Conclusions

Research on most of the topics discussed above is ongoing. By the end of 2011, decisions will be made on the best way to address the various challenges to meet Census Bureau economic census data needs. These decisions will be based on testing with a variety of datasets, e.g., data from the previous economic censuses.

Acknowledgements

The Research and Methodology Group (a subset of the Cell Suppression Modernization Team within the Economic Directorate of the U.S. Census Bureau) was the source of most of the ideas discussed in this paper. Many people on that team made contributions to the topics discussed herein.

References

Cox, Lawrence H., (1980) Suppression methodology and statistical disclosure control. J. Am. Stat. Assoc. 75, pp. 377-385.

Cox, Lawrence H., (1995) Network models for complementary cell suppression, J. Am. Soc. Assoc. 90, pp. 1453-1462.

Cox, Lawrence H., (2005) Quality Preserving Controlled Tabular Adjustment: A Method for Resolving Confidentiality and Data Quality Issues for Tabular Data, Statistics Canada Symposium
http://www.statcan.gc.ca/pub/11-522-x/2005001/4199010-eng.pdf

Duncan, George T., Mark Elliot, Juan-Jose Salazar-Gonzalez (2011). Statistical Confidentiality: Principles and Practice, Springer.

Jewett, Robert; 'Disclosure Analysis for the 1992 Economic Census' http://www.census.gov/srd/sdc/Jewett.disc.econ.1992.pdf

Massell, Paul B., An Overview of Uncertainty Creation to Protect Statistical Data, Amer. Stat. Assoc. Proceedings of JSM-2009.
http://www.amstat.org/sections/srms/proceedings/y2009/Files/303711.pdf

Federal Committee on Statistical Methodology; Working Paper #22 : Report on Statistical Disclosure Limitation http://www.fcsm.gov/working-papers/spwp22.html

Tau Argus: software for statistical tabular protection and European work on SDC; http://neon.vb.cbs.nl/casc/..%5Ccasc%5Cindex.htm

Research and Methodology Group documents; Numerous papers on the topics of this paper, though mainly for internal use, some can be sent upon request.

Appendix SC: Supercells

Let SC = a (potential) supercell. A (potential) supercell is the 'symmetric cell-union' of all interior P's and C's in some additive relation (or constraint): (where P is a primary and C is a secondary; 'interior' means a cell other than the sum cell (i.e., the marginal).
Notation: SC = Union[P1, P2, P3, ..., PK] = U[P1, P2, P3, ..., PK]
Below, as we define the (potential) SC we are defining the notion of a 'symmetric cell union'. The 'symmetric' refers to the fact that the order of P's and C's in the list does not affect the result. Our goal is to view a (potential) SC like an ordinary cell in that it has

i. a value ii. a top company id iii. a top comp value
iv. a 2nd comp id v. a 2nd comp value

value(SC) = value(SC) = sum over k=1... K of value(Pk)
for each 1st or 2nd comp id in each Pk, find the total contributions for that comp id over all Pk's. Thus we need to construct a list of all companies that appear in any of Pk's.

This will be a list of at most 2*K companies, each with a value found by summing over all the Pk's that has a contribution for the given company.
Sort the list in descending order by value of the comp.
Then one will have a top comp id and value for the union, and 2nd comp id and value.
In this way, one constructs the 5 quantities for SC listed above.

SC-under: Case where supercells can be used to prevent under-suppressions

Let A, B, C, ... represent distinct companies. Assume the following cells lie in some additive constraint (e.g., a row) in some table.

Cell X1 , contributions: [A=100, B=50]

Cell X2, contributions: [A= 120, B=70]

Cell X3, contributions: [A=80, C=20]

Note that cells X1, X2, and X3 are primaries, since they have only 2 contributors. Assume p=10, as used in the p% protection rule. {prot need = (p/100)*(top comp val) – remainder} Note that based on the p% rule : (i) X3 protects X1 and (ii) X3 protects X2. But X3 does not protect the cell union, denoted U12, of X1 and X2, where U12 has contributions: [A=220, B=120].

In the current production program, we test whether individual cells are protected but we do not test whether cell unions of 2 (or more) cells are protected. We plan to do so in the future.

One way to ensure that unions are protected (at least unions in a shaft) is to create supercells and to protect them.

Define a supercell to be all the primaries in an additive constraint (e.g., a row). So, for this example, create supercell SC = {X1;X2; X3}

Then the supercell SC has contributions: [A=300, B=120, C=20]

It has protection need = (.1)*300 – 20 = 10. We can search for cells to provide that protection. When we have found them, then all the cell unions formed from the primaries in the row will be protected; viz., U12, U13, U23, and U123. Thus all the X's and their unions are fully protected with the help of supercells. There is no remaining under-suppression of these cells.