WP. 24
ENGLISH ONLY

UNITED NATIONS STATISTICAL COMMISSION and ECONOMIC COMMISSION FOR EUROPE
CONFERENCE OF EUROPEAN STATISTICIANS

EUROPEAN COMMISSION
STATISTICAL OFFICE OF THE EUROPEAN COMMUNITIES (EUROSTAT)

Joint UNECE/Eurostat work session on statistical data confidentiality
(Bilbao, Spain, 2-4 December 2009)

Topic (iv): Tools and software improvements

ON OPEN SOURCE SOFTWARE FOR STATISTICAL DISCLOSURE LIMITATION

Invited Paper

Prepared by Juan José Salazar González, University of La Laguna, Spain

On open source software for Statistical Disclosure Limitation

Juan José Salazar González*

* Department of Statistics, Operations Research and Computer Science, University of La Laguna, 38271 Tenerife, Spain, e-mail: jjsalaza@ull.es

Abstract: Much effort has been done in the last years to design, analyze, implement and compare different methodologies to ensure confidentiality during data publication. The knowledge of these methods is public, but the practical implementations are subject to different license constraints. This paper summarizes some concepts on free and open source software, and gives some light on the convenience of using these schemes when building automatic tools in Statistical Disclosure Limitation. Our results are based on some computational experiments comparing different mathematical programming solvers when applying controlled rounding to tabular data.

1 Introduction

Statistical agencies must guarantee confidentiality of data respondent by using the best of modern technology. Today data snoopers may have powerful computers, thus statistical agencies need sophisticated computer programs to ensure that released information is protected against attackers. Recent research has designed different optimization techniques that ensure protection. Unfortunately the number of users of these techniques is quite small (mainly national and regional statistical agencies), thus there is not enough market to stimulate the development of several implementations, all competing for being the best. Still the problem of making available good implementations is very important and necessary in the public sector, which justifies the investment of public resources. Since the outcome from these investments should be automatic tools of interest for many statistical agencies, it makes sense that these tools should be open source codes and potentially free software. In this paper we analyze these related concepts, mention some tools that fit into these categories, and comments the impact on the today software in the Statistical Disclosure Limitation (SDL).

2 Basic concepts

2.1 Gratis versus Free

Free software is software that can be used, studied, and modified without restriction, and which can be copied and redistributed in modified or unmodified form either

without restriction, or with minimal restrictions only to ensure that further recipients can also do these things and that manufacturers of consumer-facing hardware allow user modifications to their hardware. The only requirement is that one must refer the authors when using a free software (it cannot be hidden behind another software).

Free software is available gratis (free of charge) in most cases. However, free software, which may or may not be distributed free of charge, is distinct from "*freeware*" which, by definition, does not require payment for use. The authors or copyright holders of freeware may retain all rights to the software; it is not necessarily permissible to reverse engineer, modify, or redistribute freeware. *Freeware* (from "free" and "software") is computer software that is available for use at no cost or for an optional fee. The opposite of Freeware is *Payware*. Freeware is also different from *shareware*; the latter obliges the user to pay after some trial period or to gain additional functionality.

The antonym of free software is "proprietary software" or "non-free software".

Since free software may be freely redistributed it is generally available at little or no cost. Free software business models are usually based on adding value such as applications, support, training, customization, integration, or certification. At the same time, some business models which work with proprietary software are not compatible with free software, such as those that depend on a user paying for a license in order to lawfully use a software product.

According to the *Free Software Foundation*, software is free software if people who receive a copy of the software have the following four freedoms:

- Freedom 0: The freedom to run the program for any purpose.
- Freedom 1: The freedom to study how the program works, and change it to make it do what you wish.
- Freedom 2: The freedom to redistribute copies so you can help your neighbour.
- Freedom 3: The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits.

Freedoms 1 and 3 require source code to be available because studying and modifying software without its source code is highly impractical.

Thus, free software means that computer users have the freedom to cooperate with whom they choose, and to control the software they use. To summarize this into a remark distinguishing *libre* (freedom) software from *gratis* (zero price) software, Richard Stallman said: "Free software is a matter of liberty, not price. To understand the concept, you should think of 'free' as in 'free speech', not as in 'free beer'". *Gratis* versus *libre* is the distinction between "for zero price" (*gratis*) and "freedom" (*libre*).

2.2 Open source

Open source is an approach to the design, development, and distribution of software, offering practical accessibility to software's source code. The term “open source” software is used by some people to mean more or less the same category as free software. It is not exactly the same class of software. The differences in extension of the category are small but important. All free software is open source, but not all open source software is free. Indeed, some companies may have open source to allow users checking what is inside and what exactly the software does, but they do not allow free modifications or redistributions of the software.

A clear advantage of open source is that, as users can analyze and trace the source code. Many more people with no commercial constraints can inspect the code and find bugs and loopholes than a corporation would find practicable.

2.3 Licenses

All free software licenses must grant people all the freedoms discussed above. However, unless the applications' licenses are compatible, combining programs by mixing source code or directly linking binaries is problematic, because of license technicalities.

The *Free Software Foundation* categorizes licenses in the following ways:

- *Public domain software* – the copyright has expired, the work was not copyrighted or the author has released the software onto the public domain. Since public-domain software lacks copyright protection, it may be freely incorporated into any work, whether proprietary or free.
- *Permissive licenses,* also called BSD-style because they are applied to much of the software distributed with the BSD operating systems. The author retains copyright solely to disclaim warranty and require proper attribution of modified works, and permits redistribution and any modification, even proprietary ones.
- *Copyleft licenses*, the GNU General Public License being the most prominent. The author retains copyright and permits redistribution and modification provided all such redistribution is licensed under the same license. Additions and modifications by others must also be licensed under the same "copyleft" license whenever they are distributed with part of the original licensed product.

3 Software to Statistical Disclosure Limitation

Protecting data is (mainly) a two criteria optimization problem. On one hand, the statistical agency must publish information with the maximum data utility. On

another hand the statistical agency aims to publish information with the minimum risk of disclosure. Since it is not always clear how to define what an optimal solution is when one has two simultaneous optimization two criterions, one approach to avoid this discussion is to redefine the problem as a one-criterion constrained problem. This means optimizing one criterion while the other is required to be under control. In SDL is widely accepted that the criterion to be under control should be the disclosure risk. Among all potential publications satisfying some protection level requirements (i.e., limited disclosure risk) an *optimal solution* is a publication with maximum utility (or minimum information loss). In the literature there are many methodologies in SDL following this framework. For example, for protecting tabular data, some methodologies following this framework are cell suppression, cell perturbation, interval publication, and controlled rounding. See e.g. Salazar (2008) for details.

A common feature of these methodologies is that their underlying optimization problems can be formulated as mathematical programs (*model*), thus an optimal (or near-optimal) publication may be found by applying a general-purpose mathematical programming solver. Alternatively one may also try to find a good publication by using common-sense, without applying a solver to a program. While on some simple methodologies and some simple data this alternative approach may works, in general sophisticated methods (like controlled rounding) and complex data (like a 3-dimensional table) requires applying a mathematical programming solver to a model. That is the reason why software like tau-ARGUS needs a mathematical programming solver.

3.1 Tau-ARGUS

The piece of code writing the mathematical programming model and interacting with mathematical programming software (called "optimizer") has been implemented at University of La Laguna using a Standard C compiler. The input of this peace is the original table, the required protection levels, some parameters defining the loss of information, etc., and the output is an optimal (or near-optimal) protected table (the publication). To solve the model, this piece of code requires a Mathematical Programming solver (provided as a Callable library of routines called API for "application program interface").

3.2 Commercial mathematical programming solvers.

In the current implementation tau-ARGUS only allows that the solver should be one of the following two options:

- Cplex under IBM (http://www.ilog.com/products/cplex/)
- Xpress under FICO (http://www.fico.com/xpress/)

This means that although tau-ARGUS is freeware (no free software!), a user needs to pay for a license to either IBM or to FICO for running the optimizer under tau-ARGUS.

3.3 Free/libre/open source solvers

A good modification in tau-ARGUS would be to have a third option of mathematical programming solver being freeware (i.e. gratis). Currently there are some possibilities:

- lp_solve http://lpsolve.sourceforge.net/
  It is a linear (integer) programming solver based on the revised simplex method and the Branch-and-bound method for the integers. It contains full source, examples and manuals. It can also be called as a library from different languages like C, VB, .NET, Delphi, Excel, Java, ... It is currently licensed under the GNU lesser general public license (LGPL).
- GLPK (GNU Linear Programming Kid) http://www.gnu.org/software/glpk/
  GLPK is currently licensed under the GNU General Public License (GPL). GLPK is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation. GLPK is not licensed under the Lesser General Public License (LGPL) as distinct from other free LP codes such as lp_solve. The most significant implication is that code that is linked to the GLPK library must be released under the GPL, whereas with the LGPL, code linked to the library does not have to be released under the same license.
- COIN-OR (COmputational INfrastructure for Operations Research) - open source for the operations research community http://www.coin-or.org/
  Most current COIN-OR projects use the Common Public License (CPL). The author of the CPL is IBM. CPL has been recently superseded by the Eclipse Public License (EPL). This license covers nine topics of concern. Chief among these is the requirement that a license not restrict any party from selling or giving away the software. Further, the Program must include source code, must allow distribution in source code as well as compiled form, and must allow modifications and derived works. Find more information at opensource.org.

3.4 Other solvers in Mathematical Programming

The following list is not exhaustive, but gives a good idea on the large number of solvers that one can choose to solve a mathematical programming model:

ABACUS - A Branch-And-CUt System - ABACUS is a software system which provides a framework for the implementation of branch-and-bound algorithms using linear programming relaxations that can be complemented with the dynamic generation of cutting planes or columns (branch-and-cut, branch-and-price, branch-and-cut-and-price).

AIMMS - Advanced modelling environment for building optimization-based decision support applications and advanced planning systems.

AMPL – Modelling language and system for formulating, solving and analyzing large-scale optimization (mathematical programming) problems.

CBC - The COIN Branch and Cut solver is an open-source mixed-integer program (MIP) solver written in C++.

CPLEX- Large-Scale Programming Software for Optimization. - The CPLEX division of ILOG provides large-scale mathematical programming software and services for resource optimization.

GAMS - The General Algebraic Modelling System (GAMS) is a high-level modeling system for mathematical programming problems.

GIPALS: Linear Programming Tool - Linear programming software for industrial size constrained optimization based on Interior-Point method.

GLPK (GNU Linear Programming Kit) - A package is intended for solving large-scale linear programming (LP), mixed integer programming (MIP), and other related problems. It is a set of routines written in ANSI C and organized in the form of a callable library. GLPK supports the GNU MathProg language, which is a subset of the AMPL language.

HOPDM - Package for solving large-scale linear, convex quadratic and convex nonlinear programming problems. The code is an implementation of the infeasible primal-dual interior point method, and compares favourably with commercial LP, QP and NLP packages.

LINDO Optimization Modelling Tools - Software for linear, integer and nonlinear optimization. LINDO supplies large scale solvers with links to Excel and database applications.

LIPSOL is a Matlab-based package for solving linear programs by interior-Point methods. LIPSOL is free software and comes with no warranty.

Linear Programs Solvers - A free software package that solves linear programming models by the simplex and/or the push-and-pull methods.

LLamasoft - Modelling software that combines simulation and optimization together in one program, from the creator of Supply Chain Guru.

MINTO - Mixed INTeger Optimizer - is a software system that solves mixed-integer linear programs by a branch-and-bound algorithm with linear programming relaxations. It also provides automatic constraint classification, preprocessing, primal heuristics and constraint generation.

Microsoft Solver Foundation. It is freeware system that runs under Microsoft Excel to solve mathematical programming models. It uses operations integrated with C# and Microsoft Visual Studio 2008.

MOSEK - Large scale optimization software. It solves linear, quadratic, general convex and mixed integer optimization problems. Details of products, trial downloads, licensing information, and documentation.

.NET OPTIMIZATION - TOMNET - The TOMNET Optimization Platform provides a standardized environment for general operations research development for the Microsoft .NET Framework. Well-known optimization solvers, such as SNOPT and MINOS are fully integrated.

Nonlinear Optimization - Nonlinear finite and infinite dimensional optimization, identification, methods and demo software by Prof. V. K. Tolstykh, Donetsk National University.

OPBDP - A Davis-Putnam Based Enumeration Algorithm for Linear Pseudo-Boolean Optimization

Optimization Tools for MATLAB, LabVIEW and .NET (C# C++ and more) - For fast and robust large-scale optimization in Matlab, LabVIEW, AMPL and .NET with packages including CPLEX, Xpress, MINOS, SNOPT.

SCIP - Solving Constraint Integer Programs. SCIP is implemented as C callable library and provides C++ wrapper classes for user plugins.

SolvOpt - Matlab, C, and Fortran codes to minimize nonlinear, possibly non-smooth objective functions and solve nonlinear minimization problems, taking into account constraints by the method of exact penalization.

SoPlex - Sequential object-oriented simplex class library, free to download for research purposes for members of non-commercial and academic institutions.

SYMPHONY - Mixed-integer linear programming solver framework. By default, SYMPHONY reads both the MPS file format and AMPL files. The SYMPHONY source code is available under the Common Public License.

XPRESS – FICO offers a software suite for modelling and optimization, information about product components, overview of services, and a client area.

4 Case study: Rounding a table

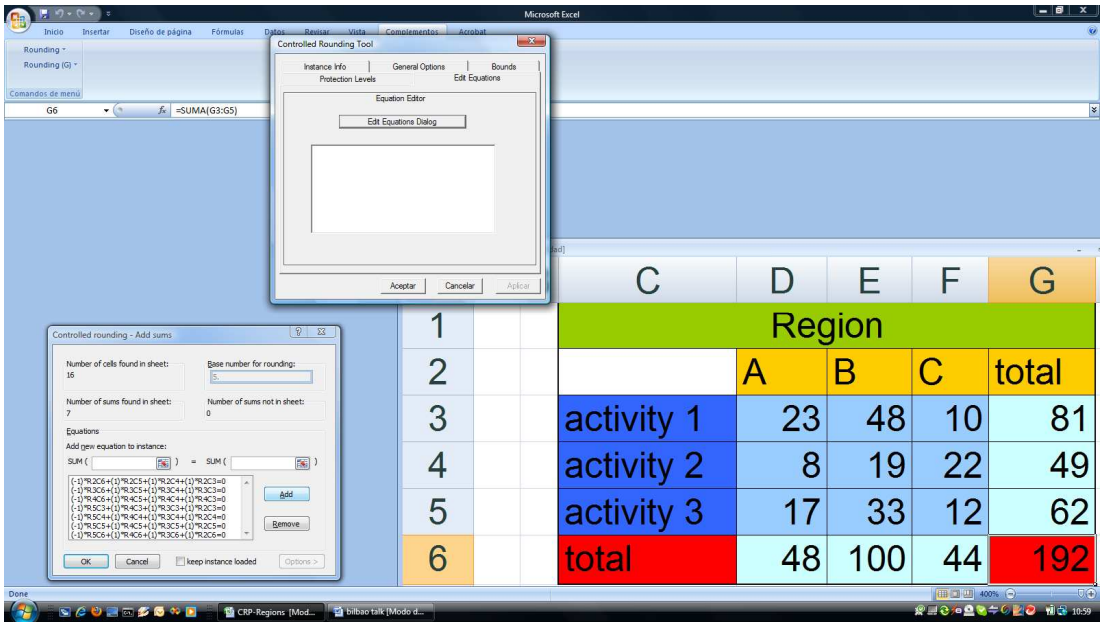
Controlled Rounding is a technique for tabular data where confidential details are eliminated by perturbing cell values. The permutation consists of replacing the original value of each (internal or marginal) cell by a multiple of the so-called base number (e.g., b=5). It is fundamental that the modified table should verify the same equations as the original table. The loss of information of a cell is defined as the distance between the original and the modified values of this cell. In the so-called zero-restricted version of the controlled rounding technique the loss of information of each cell must be smaller than the cell value. This restriction implies that when a cell

value a is a multiple of the base number b then it should not be modified, while otherwise the modified value must be the closest multiple of the base number (either up to $\lceil a \rceil$ or down to $\lfloor a \rfloor$). Since there are tables for which a modified table of the zero-restricted version does not exist, there exists also the k-restricted version of the controlled rounding technique where, for each table, the distance between the original and the modified value must be smaller than k times the base number. In all cases, no matter if k is zero or positive, an optimal solution is a modified table where the sum of the loss of information of all cells is minimum.

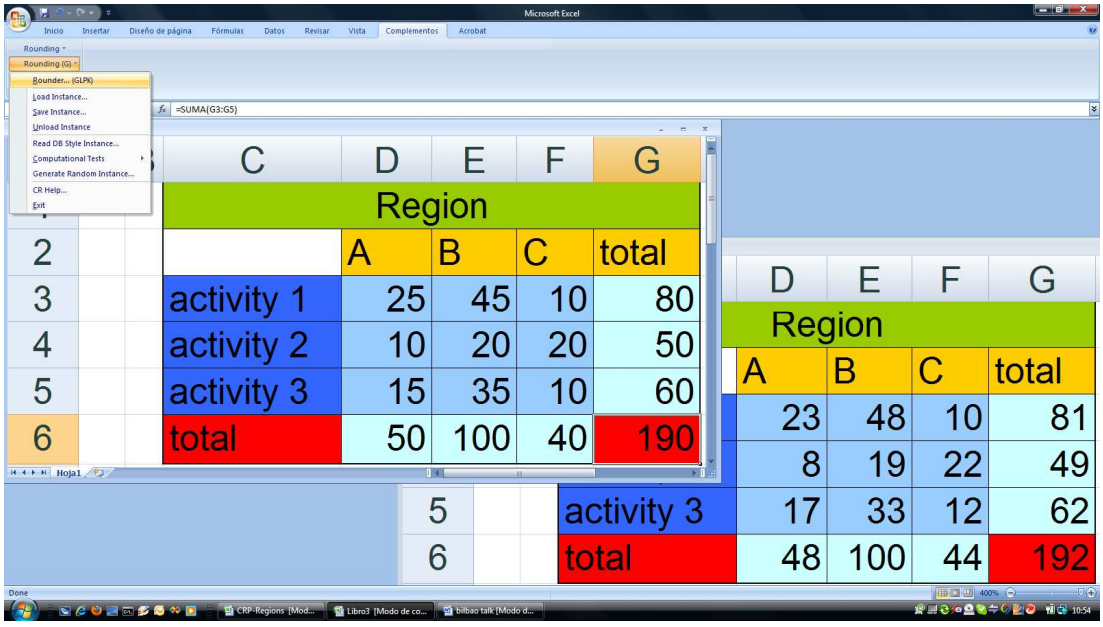
Random rounding is a standard technique where a cell value a (not multiple of b) is rounded up to $\lceil a \rceil$ with probability $(a - \lfloor a \rfloor)/b$, or down to $\lfloor a \rfloor$ otherwise. Clearly positive aspects of using random rounding instead of controlled rounding is that the modified table is simple to be computed and it is also unbiased. The major disadvantage is that either the marginal cells are rounded with the same criterion (and then the modified table may not satisfy the equations that a user expect) or are recomputed from the modified internal cells (and then the modified table may show marginal values which are too far from the original values).

In the implementation of the techniques in Salazar (2005) the controlled rounded has been extended not only to consider the k-restricted version, but also to ensure an unbiased modified tables. Then controlled rounding technique shares the same advantages of random rounding and avoids the disadvantage. Still the main drawback of implementing a controlled rounding procedure is the requirement of a mathematical programming solver to be sure that it will work on all tables not matter the structure. To this end, we have implemented the controlled rounding using three options for the mathematical programming solver: Cplex, Xpress or GLPK. The implementation has been compiled to be an add-in usable through Microsoft Excel (Salazar and Schoch (2004)). The decision of this embedding is just because this spreadsheet software is widely used through statistical agencies along the World for tabulating data, and it is much more simple to be used than tau-ARGUS. Of course, tau-ARGUS offers extra options that Excel does not offer in a direct way, like building a table from a microdata. However, these extra options are not relevant when one simply wants to protect a given table, which is what our piece of code does.

The next picture is the computer screen when the editor of equations is called. The basic way of adding equations (i.e. the table structure) in Microsoft Excel is by adding a formula to each marginal cell. However in some cases it may be interesting to have more than one equation defining a marginal cell. For suppose a table with 5 cells $\{i,j,k,p,q\}$ where the value in k is the sum of the values in i and j, and also the sum of the values p and q, i.e., $a_i+a_j=a_k=a_p+a_q$. Within Excel only one equation can be inserted in cell k, while with our new equation editor one can also insert other equations if the user needs them.



The next picture shows also the modified table after the user has inserted the base number *b*=5, so the optimizer (linked to GLPK) has been executed.



Based on our experiments, comparing XPRESS 2009, Cplex 12.1 and GLPK 4.4, the three versions have similar performances on small tables (e.g., they all solve a *10x10x10* table in about 2 or 3 seconds). GLPK becomes less competitive for tables with medium and large size (e.g., solving *20x20x20* or *10x10x10x10* tables).

5 Conclusion

Many SDL techniques imply solving optimization problems, at least when the most data useful publication is aimed. For tabular data many of these optimization problems have been modelled as mathematical programs. Although when applied to simple tables (like 2-dimensional) some specific algorithm (like network optimization) may avoid using a general purpose mathematical programming solver, a SDL techniques aiming to be applied to general tables cannot avoid using a general solver. That is the reason why tau-ARGUS currently needs a commercial license.

Today there are several non-commercial options to replace these commercial licenses. Of course, there is a cost (at least in time) to do this replacement, and in addition there are several questions regarding the efficiency of the new options when solving the inherent programs of the SDL techniques (like cell suppression). There are even issues related to the licenses, etc. However, it is worth to try it: if we succeed, tau-ARGUS would be free!

The ideal situation would be to have free and open source software, no matter if it is free or not. The reason is that then:

- You can check inside and be sure that the software implements what you think.
- You can change the source to make it work as you wish.
- You can compile and link the software within your own data producing system.
- You can benefit from the effort done by other organizations similar to yours.
- You do not need to pay for adapting and using the software.

It the support to do the implementation comes from public money, then free and open source should be the only way to ensure the success and the best use of the software.

References

[1] http://en.wikipedia.org/wiki/

[2] http://neon.vb.cbs.nl/casc/tau.htm

[3] Robert Fourer, "Eighth in a series of linear programming surveys highlights recent trends in profession's most popular software", OR/MS Today - June 2005

[4] J.J. Salazar González, M. Schoch. "A new tool for applying Controlled Rounding to a Statistical Table in Microsoft Excel" In "Privacy in Statistical Databases" (edited by J. Domingo-Ferrer) Springer Lecture Notes in Computer Science 3050 (2004) 44-57

[5] J.J. Salazar González, "Controlled Rounding and Cell Perturbation: Statistical Disclosure Limitation Methods for Tabular Data", Mathematical Programming 105 (2005) 583-603

[6] J.J. Salazar González, "Statistical Confidentiality: Optimization Techniques to Protect Tables", Computers & Operations Research 35 (2008) 1638-1651.