SEVENTH FRAMEWORK PROGRAMME

Project N°: 262608

DwB
Data without Boundaries

ACRONYM: Data without Boundaries

DELIVERABLE D11.1 – Part A

(Exploratory Report on the Future of SDC-Software Tools in General and the ECTA Method in More Details)

WORK PACKAGE 11

(Improved Methodologies for Managing Risks of Access to Detailed OS Data)

REPORTING PERIOD:	From: Month 1	To: Month 18
PROJECT START DATE:	1st May 2011	DURATION: 48 Months
DATE OF ISSUE OF DELIVERABLE:	27 April 2012	VERSION: 1.0
DOCUMENT PREPARED BY:	Partner 20	ULL

Combination of CP & CSA project funded by the European Community
Under the programme "FP7 - SP4 Capacities"
Priority 1.1.3: European Social Science Data Archives and remote access to Official Statistics

EXPLORATORY REPORT ON THE FUTURE OF SDC-SOFTWARE TOOLS IN GENERAL AND THE ECTA METHOD IN MORE DETAILS

Mª Salomé Hernández García mshergar@ull.edu.es
Juan José Salazar González jjsalaza@ull.es
University of La Laguna, Tenerife, Spain

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 262608 (DwB - Data without Boundaries).

TABLE OF CONTENTS

I. INTRODUCTION

1. Statistical agencies collect input data from individuals and deliver output information to the society based on these data. A fundamental measure of an output is the "utility" to a user, like a scientific that will use this output for research or a politician that will use this output for making decisions. Clearly more details are in the output, more useful it is. Another measure of an output is the "protection", since too many details could disseminate privacy information from individuals, and therefore violate their rights. The statistical office aims at publishing output information maximizing utility and maximizing protection, but clearly this two-criteria optimization problem is difficult to be approached, not only because the complexity of a proper definition of utility and of privacy, but also because the two criteria are in conflict.

2. A widely accepted paradigm is that protection has priority respect to utility. This means that a minimum level of protection is a-priori decided, and then an output maximizing the utility is searched among all the output with the minimum level of protection. This paradigm reduces the two-criterion problem to a single-criterion constrained problem, where it makes sense to find an optimal solution (the output to publish).

3. Still there is the issue of properly defining "utility" and "protection" of an output. To this end practitioners have proposed in the literature several methodologies. Some examples are cell suppression, controlled rounding, and controlled tabular adjustment (CTA). All the methodologies replace the original table with the true cell values by another table where some cells induce a "range" of potential values (being the true value in the range). This is the way of create uncertainty to a user, and hence protect data. In most of the cases this range of values is not explicitly given in the output, but it will be anyway computed by the user after the output has been published. The user will solve two optimization problems for each sensitive cell to detect the extreme values defining its range. The two problems for each sensitive cell are called "attacker problems". Before releasing a given output, the statistical agency may be interested in checking these ranges by solving all attacker problems in the so-called "auditing phase". When the extreme values of all ranges satisfy the required level of protection then the output is said to be "protected". The utility is measure as a function of the difference between the extreme values of each range. Clearly larger this difference is, more protected is the true cell value, but less useful will be the output to a user. Following the above mentioned paradigm, among all protected outputs the statistical agency wish to find one with maximum utility. The area is known as *Statistical Confidentiality*, and we refer the reader to (for example) the book of Duncan, Elliot and Salazar (2011) for details.

4. In this paper we analyze CTA in this context and propose a variant called ECTA. More precisely, CTA is summarized in Section II. Our alternative variant is motivated and proposed in Section III. Section IV gives some computational results using a free-and-open-source code (SCIP, Section V), which is able to solve very large tables.

II. CONTROLLED TABULAR ADJUSTMENT

5. CTA is a technique proposed by Cox and Dandekar (2002) as alternative to the classical cell suppression methodology. The motivation for creating this methodology is based on the fact that the best technique to apply cell suppression requires solving many subproblems through a sophisticated mathematical programming framework (see Fischetti and Salazar (2000)) and in practice it is difficult to find optimal outputs even for medium-size tables. Instead CTA can be formulated through a compact model that can be also easily implemented.

6. CTA consists of publishing another table obtained by changing some values with a perturbation that is obtained after solving a Mixed Integer Linear Programming (MILP) model. We now summarize some details.

7. Let us consider a table with *n* cells, among which some are marginal values. This table can be seen as a solution of a linear system of equations and inequalities. Let us denote this solution by the vector *a* with *n* numbers, and the linear system on the variables *x* by

$$Ax = b$$
$$LB \leq x \leq UB$$

where *A* is a matrix with *n* columns and *m* rows, *b* is a vector con *m* numbers, and *LB* and *UB* are vectors with *n* numbers. The matrix A and vector b describe the algebraic structure of the table (e.g., *k*-dimensional, hierarchical, linked…). Let *i*={1,…,*n*}. In most cases b_j=0 for all *j*=1,…,*m*. The vectors *LB* and *UB* represent a-priori known bounds on the cells (for example LB_i=0 and UB_i=10000 for each *i* =1,...,*n*).

8. Suppose that the statistical agency has detected a set of cells that need protection. These cells are called *sensitive*. Let us denote this subset of *I* by *P*. Suppose also that the statistical agency has fixed the upper and lower protection levels to guarantee protection as defined in Section I. Let us denote these levels by l_i and u_i for each *i* in *P*.

9. CTA consists in publishing a vector *v* instead of the vector *a* (of true values), where $v_i = a_i + y_i^+ - y_i^-$ for all *i*=1,…,*n* and where y_i^+ and y_i^- are the values of two set of mathematical variables defined by the following MILP model:

$$Min \sum_{i=1}^{n} c_i (y_i^+ + y_i^-) \qquad (1)$$

Subject to:

$$A(y^+ - y^-) = 0 \qquad (2)$$

$$0 \leq y_i^+ \leq UB_i - a_i \qquad i = 1, \ldots, n \qquad (3)$$

$$0 \leq y_i^- \leq a_i - LB_i \qquad i = 1, \ldots, n \qquad (4)$$

$$y_j^+ \geq u_j x_j \qquad j = 1, \ldots, p \qquad (5)$$

$$y_j^- \geq l_j (1 - x_j) \qquad j = 1, \ldots, p \qquad (6)$$

$$x \in \{0,1\} \qquad j = 1, \ldots, p \qquad (7)$$

10. In addition to the continuous variables y_i^+ and y_i^- for each cell *i* in *I*, there is also a binary variable x_i for each sensitive cell *i* in *P*. The variables y_i^+ and y_i^- represent the perturbation in the output respect to the true value, while x_i decides if a sensitive value must be perturbed over the upper protection level or bellow the lower protection level. The vector *c* represents weights per unit of perturbation on cells, and is defined by the statistical agency to possible encourage perturbing some cells before than others.

11. The objective function (1) is a weighted function that minimizes the perturbation. Equations (2) imply that the perturbation should conform an additive table *v*. Inequalities (3) and (4) enforce the a-priori bounds on the cell values. Inequalities (5) and (6) guarantee that the perturbation on each sensitive cell satisfies *one* protection level, either the upper *or* the lower. Constraints (7) allow the mathematical model to select which protection level will be guarantee.

12. The MIPL model in CTA can be seen as a linearization of the non-linear model:

$$Min\sum_{i=1}^{n} c_i |v_i - a_i|$$

Subject to:

$$Av = b$$

$$LB \leq v \leq UB$$

$$v_j \leq l_j \ \ or \ \ v \geq u_j \qquad j = 1, ..., p$$

13. CTA was original proposed by Cox and Dandekar (2002), and deeply analyzed later in Cox, Kelly, and Patil (2005). An excellent research with optimal and near-optimal approaches to solve the MILP model is given in Glover, Cox, Kelly and Patil (2008). Castro and Giessing (2006) discuss their experience applying CTA to real-world tables. Although CTA was originally proposed as a simpler technique than cell suppression, in practice the MILP model in CTA is far from trivial to be solved (see e.g. González and Castro (2011)).

III. ENHANCED CONTROLLED TABULAR ADJUSTMENT

14. CTA uses the objective function that minimize a distance between the output table *v* and the original table *a*. Therefore one could a-priori think that it maximizes the "utility" of the data. However, an important observation is that a user does not have *a*, hence the user cannot compute the objective value that CTA minimizes. For example, even if the distance between *v* and *a* is very small after the optimization problem was solved, a user will see *v* and will only know that these values are a result of a perturbed technique on the original table. Therefore, the "utility" of the output is different (and larger) to the user than to the statistical agency.

15. Further observations regard the "protection" issue. On one side, CTA requires the existence of a table *v* that must show a value outside the required protection range for all sensitive cells in parallel, i.e., at the same time. This differs from the meaning of protection given in Section I, where a single table valid for all sensitive cells is not required. Instead, it is required that there should exist a table for each sensitive cell, and therefore these tables may not necessary be the same for all sensitive cells. On another side, the requirement of an upper *and* lower protection levels given in Section I has been replaced by upper *or* lower protection level in CTA.

16. These two drawbacks on CTA have motivated us to introduce what we call *Enhanced Controlled Tabular Adjustment* (ECTA). Keeping the main scheme of CTA, modifications are introduced both in the way of modeling the "utility" and the "protection" in the output information. Other modifications have also been inserted to simplify the computational complexity of the approach in practice.

17. To track the utility aspect, we assume that a table *v* will be released together with some parameters. These parameters will inform the user on the maximum perturbation that has been applied on each cell. To simplify the exposition in this paper, we consider only two parameters, α and β. Parameter α is the maximum perturbation that has been applied on sensitive cells, and is defined a-priori by the statistical agency (e.g. α =0.3). It allows enforcing upper and lower protection levels on each sensitive cell. Parameter β is a mathematical unknown value that will be computed by a mathematical model, and it represents the maximum perturbation that has been applied on non-sensitive cells. Since both α and β will be released together with *v* then the user receives also a measure of the utility of the data. The mathematical model minimizes β subject to guarantee a table that is protected according to the definition given in Section I.

18. More precisely, ECTA solves a sequence of linear programs (i.e., no binary variables). In each iteration each sensitive cell is randomly fixed to a value ξ_i in the interval [(1- α /2)a_i , (1+ α /2)a_i]. If this interval is not in [LB_i , UB_i] then it is translated to be inside, e.g. [LB_i , LB_i+ α]. Then the following linear program is solved:

$$Min\ \beta$$

Subject to:

$$\begin{array}{ll} Av = 0 & \\ LB \leq v \leq UB & \\ v = \xi_i & \textit{for sensitive cells} \\ (1-\frac{\beta}{2})a_i \leq v_i \leq (1+\frac{\beta}{2})a_i & \textit{for non-sensitive cells} \end{array}$$

19. This model is solved T times, where T is a parameter fixed by the statistical agency (e.g. T=10). Each time differs in the realization of ξ_i. Some models may result infeasible. Even when a model is feasible, its solution *v* may be a non-protected output. For that reason the auditing phase is applied on each optimal solution. Among the feasible and protected tables, ECTA proposes to publish the best one (i.e. with the minimum β value).

20. When all the T models are infeasible or produce non-protected optimal tables, then some sensitive cells are fixed to its original value (i.e. ξ_i is replaced by a_i in the above model for a subset of *P*) and the whole process is repeated. In the worst scenario, all sensitive cells will be fixed to their original value. If the model is still infeasible when all the sensitive cell have been fixed to their original value, then this proves that the desired protection imposed by the statistical agency is impossible, which means that α must be reduced and the full process repeated.

21. For selecting the subset of sensitive cells to be fixed, we propose to randomly choose K among the ones that detected the non-protection of a table. If no one exists, then the K sensitive cells are randomly chosen among all the sensitive cells no previously fixed. For example, select K=1 when |P| is small, i.e., fix one sensitive cell at a time.

22. A variant of the above approach would let v_i variable in [(1- α /2) a_i , (1+ α /2) a_i], instead of randomly fixed.

23. Remember that a solution *v* from the above model may be protected or not. For that reason, we do always need to check protection by applying the auditing phase. The two attacker problems to be solved for each sensitive cell *k* in *P* are:

$$Min\ /\ Max \quad x_k$$

Subject to:

$$\begin{array}{ll} Ax = b & \\ LB \leq x \leq UB & \\ (1-\alpha)a_j \leq x_j \leq (1+\alpha)a_j & \textit{for sensitive cells j in P} \\ (1-\beta)a_i \leq x_i \leq (1+\beta)a_i & \textit{for non-sensitive cells i in I\textbackslash P} \end{array}$$

where α is the pre-specified parameter by the statistical agency to the sensitive cells, and β and *v* are the optimal solution computed by ECTA linear program. As mentioned, there is no guarantee that *v* is protected with respect to α and β. However, clearly increasing β increases the protection issue, while reduces the utility of *v* to a user. In some cases it could be more convenient to increase β instead of fixing extra sensitive cells and solver another sequence of T problems.

24. After of some tests and of to find, in some cases, non-protected solutions for α and β obtained, we decided to make a reauditing phase where the value of β is doubled.

25. While CTA requires solving an NP-hard problem, ECTA requires solving a number of polynomially solvable problems. This number is typically T, but it could go in the worst case to T times

the number of sensitive cells in the table. In all situations the computational complexity of the ECTA approach is manageable. The next section shows some computational experiments.

IV. COMPUTATIONAL RESULTS

26. To evaluate the behavior in practice of the approach proposed in Section III, it has been implemented in standard C++ using the free-and-open-source software SCIP to solve the mathematical programs. The code is fully portable to computers with different operating systems, including Linux, Mac and Microsoft Windows. This section shows some results obtained by running the code on a computer with an Inter Core(TM)2 Duo CPU at 3,34 GHz and Microsoft Windows Vista. The instances for our experiments are taken from the public-available CSPLIB (webpages.ull.es/users/casc).

27. We found two instances in which the CTA has had to be interrupted without obtaining feasible solution. In contrast, the ECTA results are as follows:

	Instance 1 (Hier16.csp)	**Instance 2 (Ninenew.csp)**
Number of cells	3564	6546
Number of sensitive cells |P|	224	858
Number of sums (rows in A)	5484	7340
Number of ECTA models	10	10
Total ECTA time	292 seconds	6336 seconds
Time for finding (*β*,*v*)	18 seconds	80 seconds
Time for auditing v	274 seconds	6256 seconds
Protected solutions	2 solutions	1 solution
Non-protected solutions	8 solutions	9 solutions
Infeasible solutions	0 solutions	0 solutions

28. On these two instances ECTA was not able to produce a protecting table by solving T linear programs. It required two iterations for solving Instance 1 and three iterations for solving instance 2. Still the total time is quite reasonable (a few minutes). But the more important aspect is that the generated table *v* is companied with parameters α and β that help a use to measure the utility of the *v*, while protect the sensitive cells.

29. Others results:

				ECTA					**CTA**	
Name	**No. cell**	**No. sensitive cell**	**No. sums**	**Time finding**	**Time auditing/ reauditing**	**Protected solutions**	**Non-protected solutions**	**Infeasible solutions**	**Time**	**Solutions**
Hier13.csp	2020	112	3313	5 sec.	109 sec.	3 sols.	7 sols.	0 sols.	Stop 3600 sec.	0 sol.
hier13x7x7d.csp	637	50	525	0 sec.	13 sec.	2 sols.	18 sols.	0 sols.	0 sec.	1 sol.
hier13x13x7d.csp	1183	75	1443	2 sec.	26 sec.	3 sols.	7 sols.	0 sols.	0 sec.	1 sol.
hier13x13x13a.csp	2197	108	3549	4 sec.	109 sec.	4 sols.	6 sols.	0 sols.	Stop 3600 sec.	0 sol.
hier13x13x13b.csp	2197	108	3549	4 sec.	83 sec.	3 sols.	7 sols.	0 sols.	Stop 3600 sec.	0 sol.
hier13x13x13c.csp	2197	108	3549	1 sec.	82 sec.	3 sols.	7 sols.	0 sols.	Stop 3600 sec.	0 sol.
hier16x16x16a.csp	4096	224	5376	78 sec.	432 sec.	3 sols.	57 sols.	0 sols.	Stop 3600 sec.	0 sol.
hier16x16x16c.csp	4096	224	5376	78 sec	266 sec.	1 sol.	59 sols.	0 sols.	Stop 3600 sec.	0 sol.

hier16x16x16d.csp	4096	224	5376	79 sec.	301 sec.	2 sols.	58 sols.	0 sols.	Stop 3600 sec.	0 sol.
hier16x16x16e.csp	4096	224	5376	79 sec.	319 sec.	2 sols.	58 sols.	0 sols.	1 sec.	1 sol.
Nine5d.csp	10733	1661	17295	117 sec.	11685 sec.	1 sol.	19 sols.	0 sols.	Stop 12000 sec.	0 sol.
table5.csp	4991	517	2464	17 sec.	944 sec.	1 sol.	14 sols.	15 sols.	Stop 3600 sec.	0 sol.
table7.csp	623	17	230	2 sec.	7 sec.	1 sol.	23 sols.	56 sols.	0 sec.	1 sol.
table8.csp	1270	3	72	2 sec.	0 sec.	3 sol.	7 sols.	0 sols.	1 sec.	1 sol.
targus.csp	162	13	63	0 sec.	2 sec.	2 sols.	8 sols.	0 sols.	1 sec.	1 sol.

V. SCIP

30. SCIP, Solving Constraint Integer Programs, is currently one of the fastest non-commercial *mixed integer programming (MIP) solvers*. It is also a framework for *Constraint Integer Programming* and *branch-cut-and-price*. It allows total control of the solution process and the access of detailed information down to the guts of the solver.

31. SCIP can use different solver:

CLP: Coin-or linear programming. Clp is an open-source linear programming solver written in C++. https://projects.coin-or.org/Clp

SoPlex: Sequential object-oriented simPlex. SoPlex is a Linear Programming (LP) solver based on the revised simplex algorithm. It features pre-processing techniques, exploits sparsity, and offers primal and dual solving routines. It can be used as a standalone solver reading MPS or LP format files as well as embedded into other programs via a C++ class library. http:/soplex.zib.de/

Cplex: High-performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

32. Our project has used SCIP 2.0.1 and CLP 1.13.3, compiled with MinGW (Minimalist GNU for Windows, is a minimalist development environment for native Microsoft Windows applications. http://www.mingw.org/).

VI. CONCLUSION

33. This paper proposes ECTA, which is a variant of the CTA methodology to protect tables. ECTA aims to publish a table together of a measure of its utility. To simplify notation, although more sophisticated measures are possible, this paper consider two parameters α and β. The first is a-priori fixed and will inform a user about the maximum perturbation that was applied on a sensitive cell. The second parameter is computed by ECTA and represents the maximum perturbation on other cells. To maximize the utility of the released table, this second parameter is minimized. In addition ECTA guarantee protection in the released table by solving the auditing phase during the process.

34. A preliminary version of the algorithm has been implemented using a free-and-open-source code. Our experiments show that ECTA is able to provide a protected table in a few minutes on tabular data where CTA needs more than one hour to solve its integer model. Therefore ECTA is a promising technique to protect tabular data that could merit further investigations.

VII. ACKNOWLEDGMENT

35. This work have been supported by the FP7-INFRASTRUCTURES-2010-1 project “DwB-Data without Boundaries”, number 262608. This support is appreciated.

VIII. BIBLIOGRAPHY

Castro, J. and Giessing, S. (2006). Testing variants of minimum distance controlled tabular adjustment. In Monographs of Official Statistics. Work session on Statistical Data Confidentiality, Eurostat-Office for Official Publications of the European Communities, Luxembourg, 2006, 333–343. ISBN 92-79-01108-1.

Cox, L.H., Kelly, J.P., and Patil, R.J. (2004). "Balancing Quality and Confidentiality for Multivariate Tabular Data," in: Privacy in Statistical Databases, Lecture Notes in Computer Science 3050 (J. Domingo-Ferrer and V. Torra, eds.), Berlin: Springer-Verlag, 2004, 87-98.

Cox, L.H., Kelly, J.P., and Patil, R.J. (2005). "Computational Aspects of Controlled Tabular Adjustment: Algorithm and Analysis," The Next Wave in Computer, Optimization and Decision Technologies (B. Golden, S. Raghavan and E. Wasil, eds.), Boston: Kluwer, 45-59.

Cox, L.H. and Dandekar, R.A. "Disclosure Limitation Method for Tabular Data That Preserves Accuracy and Ease-of-Use" in: Proceedings of the 2002 FCSM Statistical Policy Conference, Washington, DC: Office of Management and Budget, 2004, 15-30.

Danderkar, R.A. and Cox, L.H. (2002). "Synthetic Tabular Data-An Alternative to Complementary Cell Suppression", manuscript. Energy Information Administration, U.S. Department of Energy.

Duncan, G., Elliot, M. and Salazar, J.J. (2011), "Statistical Confidentiality: Principles and Practice", Springer.

Fischetti, M. and Salazar, J.J. (2000). "Solving the Cell Suppression Problem on Tabular Data with Linear Constraints," Management Science 47, 1008-1026.

Glover, F., Cox, L.H., Kelly, J.P. and Patil R. (2008). "Exact, heuristic and metaheuristic methods for confidentiality protection by controlled tabular adjustment", International Journal of Operations Research Vol. 5, No. 2, 117-128.

González, J. A. and Castro, J. (2011). "A heuristic block coordinate descent approach for controlled tabular adjustment". Computers & Operations Research, 38, 1826–1835.

Cox, L.H., Orelien J.G. and Shah, B.V. (2006). "A Method for Preserving Statistical Distributions Subject to Controlled Tabular Adjustment", Lecture Notes in Computer Science, 2006, Volume 4302, 1- 11.