Computers & Operations Research 43 (2014) 61–67

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Enhanced controlled tabular adjustment

María-Salomé Hernández-García, Juan-José Salazar-González*

DEIOC, Universidad de La Laguna, 38271 Tenerife, Spain

ARTICLE INFO

Available online 29 August 2013

Keywords:
Statistical confidentiality
Tabular protection
Integer linear programming

ABSTRACT

Statistical agencies collect data from individuals and businesses, and deliver information to the society based on these data. A fundamental feature to consider when releasing information is the "protection" of sensitive values, since too many details could disseminate private information from respondents and therefore violate their rights. Another feature to consider when releasing information is the "utility" to a data user, as a scientist may need this information for research or a politician for making decisions. Clearly the more details there are in the output, the more useful it is, but it is also less protected. This paper discusses a new technique called Enhanced Controlled Tabular Adjustment (ECTA) to ensure that an output is both protected and useful. This technique has been motivated by another approach in the literature of the last decade, and both are compared and evaluated on a set of benchmark instances.

1. Introduction

Statistical agencies collect data from respondents, analyze this data, and release information to users. The released information is called *output*. In this process it is fundamental to maximize the utility of the output to the final data users, but also to maximize the protection of the information provided by each respondent. Therefore, in general, publishing data aims solving a two-criteria optimization problem. Since the two criteria are in conflict, this optimization problem is very complex.

A widely accepted paradigm is that protection has priority respect to utility. Based on this paradigm, a minimum level of protection is a priori decided and set inside the optimization problem through constraints. Then an output maximizing the utility to a data user is searched among all solutions with the required (acceptable) level of protection. The paradigm reduces the two-criteria problem into a single-criterion constrained problem, where it makes sense to find an optimal (or near-optimal) solution (the output to publish). The priority of protection over utility justifies why the area is called *Statistical Confidentiality*.

The proper definition of "utility" and "protection" of an output is a fundamental issue. There are different types of outputs, each type associated to a methodology. Some examples of methodologies to protect tabular data are cell suppression, controlled rounding, and controlled tabular adjustment (CTA). These methodologies replace the original table (with the true cell values) by another table where some cells induce a "range" of potential values. The true value belongs to the range of each cell, but for sensitive cells other values must also exist. The ranges of values guarantee uncertainty on the sensitive cells to a data user, thus protecting the sensitive information in the table. In most of the cases the ranges are not explicitly displayed in the output, but they may be computed by the data user from the output after it has being published. The user will solve two optimization problems to detect the extreme values defining the range of a cell in the output. These two mathematical problems are called *attacker problems* and the range of values is called *protected interval*. Before releasing an output, the statistical agency may desire to compute the protected interval of each sensitive cell. This procedure implies to solve all attacker problems and is called *auditing phase*. When the extreme values of all intervals satisfy the required levels of protection then the output is said to be *protected*. The required levels of protection for each sensitive cell are a-priori established by the statistical agency. The utility of an output is measured in general as a function on the difference between the extreme values of each protected interval. Clearly the larger this difference is, the more protected is the cell, but less useful will be the output to a data user. Following the above-mentioned paradigm, among all protected outputs, the statistical agency wishes to find one with maximum utility (or equivalently, with minimum loss of information). We refer the reader to (for example) the book of Duncan et al. [5,10] for further details.

In this paper we analyze CTA in this context and propose a variant called *Enhanced Controlled Tabular Adjustment* (ECTA). ECTA explores the space of tables within a kind of *Greedy Randomized Adaptive Search Procedure* (GRASP), introduced by Feo and Resende [6]. To this end it contains a random operator to better explore the feasible region of outputs. As a GRASP, the technique consists of iterations made up from successive constructions of greedy-randomized solutions. The protection is carefully considered with

* Corresponding author. Tel.: +34 922 318184; fax: +34 922 318170.
E-mail address: jjsalaza@ull.es (J.-J. Salazar-González).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.08.021

a built-in auditing phase. This auditing phase acts also as iterative improvement procedure to find protected tables from unprotected ones. Although this phase is the most time-consuming component of the ECTA approach, it guarantees that the output is protected. Part of the uncertainty to guarantee protection in ECTA is based on fixing some randomly selected cells to random values. The utility of an ECTA output is communicated to the data users through two parameters. One parameter α represents the maximum perturbation on a sensitive cell, and it depends directly on the protection level requirements defined by the statistical agency. Another parameter β represents the maximum perturbation on a non-sensitive cell, and it is determined by a mathematical model. On magnitude tables, all optimization problems are formulated by compact linear-programming models, so the implementation is simple and efficient in practice.

Section 2 describes a widely accepted concept of protection in Statistical Confidentiality, and sets up the mathematical notation that is used in the rest of this paper. CTA, introduced in the literature ten years ago, is summarized in Section 3. ECTA is motivated and proposed in Section 4. The overall algorithm follows the structure of a GRASP approach. Section 5 discusses computational results solving benchmark instances with our CTA and ECTA implementations. These results show the better performance of ECTA.

2. Background

Let us consider a statistical table with n cells, among which some are marginal values (i.e. values obtained by adding other cell values). The values in the cells determine a vector a which is the solution of a linear system of equations $My = b$. The vectors a and b have n and m values, respectively; thus, the matrix M has n columns and m rows. The matrix M and vector b describe the algebraic structure of the table (e.g., k-dimensional, hierarchical, linked, etc.). In most of the cases $b = 0$ and each row of M has one coefficient equals to -1 while the others are 0 or 1 (i.e. a row defines a marginal cell). The set of cells is denoted by I and the set of equations by J.

The cell values of a table may be floating-point or integer numbers. The first type of tables are called *magnitude tables*, and are typically generated by adding a categorical feature of a microdata. The second type of tables are called *frequency* or *contingency tables*, and are typically obtained by counting the number of responders in each cell. We assume in this article that a is a vector of floating-point numbers, although the methodologies can be extended to the case where a is a vector or integer numbers by adding integrality conditions to some linear programs. Indeed, Section 5 concludes with some numerical experiments on contingency tables.

As usual when describing a methodology for protecting a table, we assume that the statistical agency has determined a priory the set of sensitive cells that need protection. To this end the statistical agency may has applied a common-sense rule like the so-called *dominance rule* (see e.g. [5]). Let P be the subset of I defining the sensitive cells.

In general, the statistical agency is interested in protecting the sensitive cells against different potential attackers. These attackers may be individuals that contributed to the microdata from which the vector a was computed. They can also represent coalitions of individuals. Let K be the set of attackers. Each attacker has a priori bounds on each cell value. Let us call lb_i^k and ub_i^k the bounds defining the worst-case estimation that the attacker k knows on cell i. This means that, before releasing information on a, the attacker k knows that the true value a_i belongs to the interval $[lb_i^k, ub_i^k]$. Then, using the released information (unprecise data from a), the attacker k will compute the so-called protected interval for cell i, which is a subinterval of $[lb_i^k, ub_i^k]$. The attacker will solve two optimization problems to determine the extreme values of the protected interval. One problem minimizes y_i and the other maximizes y_i. The feasible region of both problems is the same and is defined by the released information. Let us call $[\underline{y}_i^k, \overline{y}_i^k]$ the protected interval that will be calculated by the attacker k on cell i. As said, $lb_i^k \le \underline{y}_i^k \le a_i \le \overline{y}_i^k \le ub_i^k$.

The statistical agency sets up three non-negative parameters UPL_i^k, LPL_i^k and SPL_i^k for each sensitive cell i and each attacker k. The released information is said to protect the sensitive information in the original table, or simply it is *a protected output*, when $\underline{y}_i^k$ and $\overline{y}_i^k$ satisfy the following conditions:

$$\overline{y}_i^k \ge a_i + UPL_i^k \tag{1}$$

$$\underline{y}_i^k \le a_i - LPL_i^k \tag{2}$$

$$\overline{y}_i^k - \underline{y}_i^k \ge SPL_i^k \tag{3}$$

for each sensitive cell $i \in P$ and each attacker $k \in K$. Intuitively, these three parameters intend to guarantee protection level requirements on the range of values that the attacker will see on each sensitive cell.

The next section describes a methodology to protect a table against one attacker. As a usual notation in the literature, when $K = \{1\}$ we write lb_i, ub_i, LPL_i, UPL_i and SPL_i instead of lb_i^1, ub_i^1, LPL_i^1, UPL_i^1 and SPL_i^1, respectively.

3. Controlled tabular adjustment

There are several types of outputs to protect a table. Each output has a format determined by a methodology. *Cell Suppression* (CS) is the oldest and most used methodology. It consists of publishing another table where some cells contain the original values and other cells contain missing values. Tables (a) and (b) in Fig. 1 are two examples of potential outputs using CS to protect a 2-dimensional table.

A natural mathematical formulation for CS and other methodologies is in Bilevel Programming: a master problem (first level) looks for an output minimizing the loss of information, while a subproblem (second level) solves the attacker problems to check the conditions (1)–(3). It is a very special case of bilevel optimization because the first-level problem only uses the optimal objective value of the second-level problems, and not their optimal solutions. Although under some assumptions the second-level optimization problem can be eliminated with the use of a decomposition technique (e.g. Dantzig–Wolfe or Benders' Decompositions), the number of

(a)

34566	–	–	92525
53453	66345	43563	163361
145343	–	–	243131
233362	113315	152340	499017

(b)

–	–	–	92525
–	–	–	163361
–	–	–	243131
233362	113315	152340	499017

(c)

34566	3425	54534	92525
53453	66345	43563	163361
145343	43545	54243	243131
233362	113315	152340	499017

(d)

66477	5730	20318	92525
89552	53242	20567	163361
77333	54343	111455	243131
233362	113315	152340	499017

Fig. 1. Examples of potential output from CS and from CTA.