Enhanced Controlled Tabular Adjustment

M. Salomé Hernández García , Juan-José Salazar-González

mshergar@ull.es , jjsalaza@ull.es

DEIOC, Universidad de La Laguna, 38271 Tenerife

.

.

.

Research funded by MTM2009-14039-C06-01 and DwB

Basic concepts

- We are given with
  - a table is an array $a = [a_i : i \in I]$ satisfying:

$$\sum_{i \in I} m_{ij} y_i = b_j \qquad \text{for all } j \in J$$

$$lb_i \leq y_i \leq ub_i \qquad \text{for all } i \in I.$$

  - A subset of sensitive cells $P \subset I$ with $[LPL_p, UPL_p]$ for each $p \in P$.

Basic concepts

- We are given with
  - a table is an array $a = [a_i : i \in I]$ satisfying:

$$\sum_{i \in I} m_{ij} y_i = b_j \qquad \text{for all } j \in J$$

$$lb_i \leq y_i \leq ub_i \qquad \text{for all } i \in I.$$

  - A subset of sensitive cells $P \subset I$ with $[LPL_p, UPL_p]$ for each $p \in P$.
- Find "something" such that a data user will see the original table and other tables as possible tables such that:
  - LPL_p and UPL_p should be possible values in a table.
  - The loss of information due to the existence of other tables is minimized.

Cell Suppression

- “something” is a table with some missing values (called suppressions).

Cell Suppression

- “something” is a table with some missing values (called suppressions).
- The optimization problem can be solved by using Integer Linear Programming, and it is NP-hard.

Cell Suppression

- "something" is a table with some missing values (called suppressions).
- The optimization problem can be solved by using Integer Linear Programming, and it is NP-hard.
- A model using only one binary variable for each cell requires a cutting-plane generation technique.

Cell Suppression

- “something” is a table with some missing values (called suppressions).
- The optimization problem can be solved by using Integer Linear Programming, and it is NP-hard.
- A model using only one binary variable for each cell requires a cutting-plane generation technique.
- The solutions from this model are always protected (thus auditing is unnecessary).

Cell Suppression

- “something” is a table with some missing values (called suppressions).
- The optimization problem can be solved by using Integer Linear Programming, and it is NP-hard.
- A model using only one binary variable for each cell requires a cutting-plane generation technique.
- The solutions from this model are always protected (thus auditing is unnecessary).
- Alternative: 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.

Cell Suppression: mathematical model

$$\min \sum_{i \in I \setminus P} w_i x_i$$

Subject to:

$$\underline{y}_p \leq lpl_p \quad , \quad \overline{y}_p \geq upl_p \quad , \quad \overline{y}_p - \underline{y}_p \geq spl_p \qquad \forall p \in P$$

$$x_i \in \{0, 1\} \qquad \forall i \in I \setminus P$$

where

$\underline{y}_p := \min y_p$ and $\overline{y}_p := \max y_p$

$\sum_{i \in I} m_{ij} y_i = b_j$ for all $j \in J$

$a_i - (a_i - lb_i)x_i \leq y_i \leq a_i + (ub_i - a_i)x_i$ for all $i \in I$

Controlled Tabular Adjustment

- “something” is a table with perturbed values.

Controlled Tabular Adjustment

- “something” is a table with perturbed values.
- The optimization problem can be solved by using Integer Linear Programming, and it is NP-hard. However, a model uses only one binary variable for each sensitive cell and not cutting-plane generation technique is needed.

Controlled Tabular Adjustment

- “something” is a table with perturbed values.
- The optimization problem can be solved by using Integer Linear Programming, and it is NP-hard. However, a model uses only one binary variable for each sensitive cell and not cutting-plane generation technique is needed.
- The solutions from this model may be unprotected (thus auditing is necessary).

Controlled Tabular Adjustment

- "something" is a table with perturbed values.
- The optimization problem can be solved by using Integer Linear Programming, and it is NP-hard. However, a model uses only one binary variable for each sensitive cell and not cutting-plane generation technique is needed.
- The solutions from this model may be unprotected (thus auditing is necessary).
-

$$\min \sum_{i \in I} w_i |y_i - a_i|$$

subject to:

$$\sum_{i \in I} m_{ij} y_i = b_j \qquad \text{for all } j \in J$$

$$lb_i \leq y_i \leq ub_i \qquad \text{for all } i \in I$$

$$y_p \leq lpl_i \;\text{ or }\; y_p \geq upl_i \qquad \text{for all } p \in P$$

which one has smaller loss of information?

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

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

which one has smaller loss of information?

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

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

Enhanced ControlledTabularAdjustment

- Motivation:
  - Utility: How useful is a perturbed data from the user point of view?
  - Protection: How can we ensure both lower AND upper protection levels on each sensitive cell?

Enhanced ControlledTabularAdjustment

- Motivation:
  - Utility: How useful is a perturbed data from the user point of view?
  - Protection: How can we ensure both lower AND upper protection levels on each sensitive cell?
- Complexity: CTA, like CS, requires solving an Integer Linear Programming model, thus their optimization problems are both NP-hard.

Enhanced ControlledTabularAdjustment

- Motivation:
  - Utility: How useful is a perturbed data from the user point of view?
  - Protection: How can we ensure both lower AND upper protection levels on each sensitive cell?
- Complexity: CTA, like CS, requires solving an Integer Linear Programming model, thus their optimization problems are both NP-hard.
- Instead, ECTA consists in solving a model without binary variables, with only 1 variable y_i for cell $i \in I$, and with only 1 additional variable β that will be also published.

Enhanced ControlledTabularAdjustment

- Select k sensitive cells in a random way.
- Fix each of these cells to a random value ξ_p in $[lpl_p, ulp_p]$.
- Fix the other sensitive cells to their original value $\xi_p := a_p$.
- Solve the LP model: $\min \beta$

$$\begin{aligned} \text{subject to} \quad \textstyle\sum_{i \in I} m_{ij} y_i = b_j & \qquad \text{for all } j \in J \\ lb_i \le y_i \le ub_i & \qquad \text{for all } i \in I \\ y_p = \xi_p & \qquad \text{for all } p \in P \\ \left(1 - \tfrac{\beta}{2}\right) a_i \le y_i \le \left(1 + \tfrac{\beta}{2}\right) a_i & \qquad \text{for all } i \in I \setminus P \end{aligned}$$

- If the problem is infeasible or the solution is unprotected then restart with a larger k.

ECTA: computational results

- we have implemented CTA and ECTA using a free-and-open-source optimizer: SCIP.
- we considered 2 instances from the CSPLIB where this solver needs more than 1 hour: Hier16 , Ninenew
- ECTA found protected solutions in a few minutes
- Details: ($k+ = 5$)

	Hier16.csp	Ninenew.csp
$\lvert I \rvert$	3564	6546
$\lvert P \rvert$	224	858
$\lvert J \rvert$	5484	7340
Number of ECTA models	30	10
Total ECTA time	197 seconds	459 secs
Time for finding (β, y)	51 secs	72 secs
Time for auditing (β, y)	143 secs	382 secs
Protected solutions	1 sol	1 sol
Non-protected solutions	29 sols	9 sols
Infeasible solutions	0 sols	0 sols