Present and future research on controlled tabular adjustment

Jordi Castro
jordi.castro@upc.edu

José Antonio González
jose.a.gonzalez@upc.edu

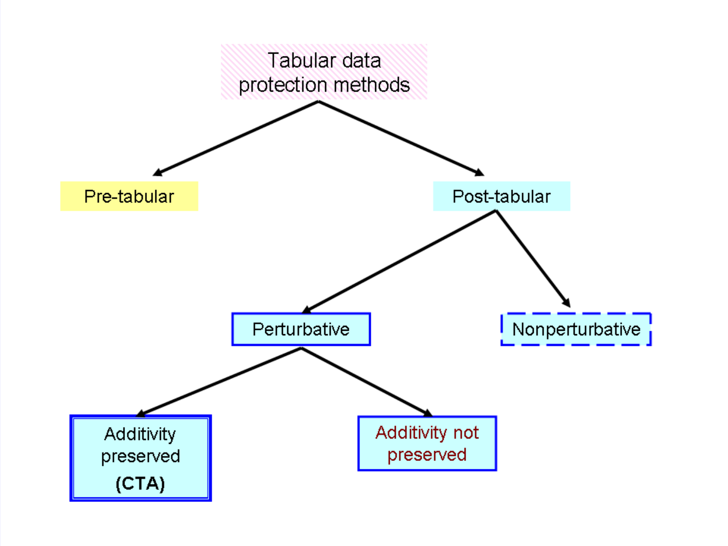
Dept. of Statistics and Operations Research
Group of Numerical Optimization and Modelling
Universitat Politècnica de Catalunya
Barcelona

This work has been supported by grants MTM2009-08747 of the Spanish Ministry of Science and Innovation, SGR-2009-1122 of the Government of Catalonia, and the project DwB INFRA-2010-262608 of the EU FP7

Contents

Contents

Broad classification



CTA features

CTA is a tool based on mathematical optimization:

- flexible with user requirements (additivity, subtotals, cell perturbations, etc.);
- applicable to any type of table;
- *customizable*:
  - L_1, L_2 or other distances,
  - accuracy limit,
  - time limit,
  - different solvers (CPLEX, Xpress, free solvers as CBC, GLPK...).

However, finding an optimal solution may not be an easy task.

Contents

Parameters for the MILP CTA model

- Set of cells $a_i, i = 1, \ldots, n$.
- Set $\mathcal{S} = \{i_1, i_2, \ldots, i_s\} \subseteq \{1, \ldots, n\}$ of indices of sensitive cells.
- Linear relations $A\, a = b$.
- Lower and upper protection level for each sensitive cell $i \in \mathcal{S}$: lpl_i and upl_i.
- Lower and upper bound for each cell: l_{a_i} and u_{a_i}.
- Cell weights w_i for cost of adjustment of each cell.

Aim of CTA

Find released values x_i such that:

- Remain near a_i (distance considered: absolute value).
- Satisfy the linear relations $A\,x = b$
- Satisfy the bounds: $l_{a_i} \leq x_i \leq u_{a_i}$
- Satisfy the protection levels: **either** $x_i \geq a_i + upl_i$ **or** $x_i \leq a_i - lpl_i$.

The optimization problem is:

$$
\begin{array}{rl}
\min\limits_{x} & ||x-a||_L \\
\text{subject to} & Ax = b \\
 & l_x \leq x \leq u_x \\
 & x_i \leq a_i - lpl_i \text{ or } x_i \geq a_i + upl_i \quad i \in \mathcal{S}.
\end{array}
$$

The MILP CTA model

- Defining *cell deviations* as: $z_i = x_i - a_i$,
- and introducing binary variables for sensitive cells: $y_i, i \in \mathcal{S}$ (e.g., when $y_i = 1$, the protection sense is *up*: $x_i \geq a_i + upl_i$; when $y_i = 0$, the protection sense is *down*: $x_i \leq a_i - lpl_i$).

The MILP model is:

$$\begin{array}{ll} \min\limits_{z^+,z^-,y} & \sum\limits_{i=1}^{n} w_i(z_i^+ + z_i^-) \\ \text{subject to} & A(z^+ - z^-) = 0 \\ & 0 \leq z^+ \leq u_z, \quad 0 \leq z^- \leq -l_z \\ & y \in \{0,1\}^s \\ & \left.\begin{array}{lll} upl_i\, y_i & \leq z_i^+ & \leq u_{z_i} y_i \\ lpl_i(1-y_i) & \leq z_i^- & \leq -l_{z_i}(1-y_i) \end{array}\right\} i \in \mathcal{S} \end{array}$$

Contents

Approach

- ▷ The Block Coordinate Descent (BCD) strategy is based on the solution of a sequence of CTA subproblems, where some sensitive cells are free while the remaining ones have a fixed protection sense.
- ▷ At each iteration, the MILP solution affects only a reduced set of binary variables. Once solved the subproblem, these variables remain fixed at their new state and another set is optimized.
- ▷ Caution: convergence to an optimum is not guaranteed, but satisfactory behaviour in practice.

Finding a feasible starting point

We need an initial, feasible assignment for sensitive cells (up or down).

The SAT method is an approach which has proven to be successful in many instances:

- For each constraint with at least one sensitive cell, detect any combination of these leading to infeasibility.
- Collect all the infeasible combinations and look for a join assignation of the binary variables such that every combination is feasible.

SAT. An example

Every one of these combinations produces an infeasible problem:

$$\begin{array}{lll} (1) & y_1 = 1, y_2 = 0, y_3 = 1, y_4 = 1 & \Rightarrow y_1 \cap \neg y_2 \cap y_3 \cap y_4 \\ (2) & y_3 = 1, y_2 = 1, y_4 = 1 & \Rightarrow y_3 \cap y_2 \cap y_4 \\ (3) & y_5 = 1, y_2 = 0, y_1 = 1 & \Rightarrow y_5 \cap \neg y_2 \cap y_1 \end{array}$$

Therefore, we need to make TRUE (SATisfiable) the logical negation:

[NOT (1)] AND [NOT (2)] AND [NOT (3)]

For instance: $(\neg y_1 \cap \neg y_3)$, i.e., $y_1 = 0$ and $y_3 = 0$. The other variables can take any value.

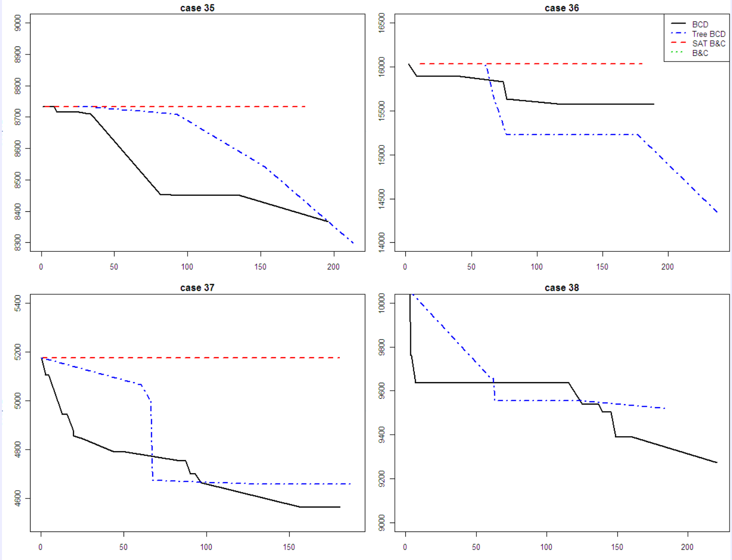
Solvers for the Satisfiability problem are very efficient.

Branch and cut vs Heuristics: some results

Dimensions of instances

instance	n	s	m	N. coef.	cont.	bin.	constr.
case 35	499298	55527	20747	1007124	998596	55527	242855
case 36	1200439	107743	45638	2417196	2400878	107743	476610
case 37	296004	42652	10904	597057	592008	42652	181512
case 38	572373	81359	18873	1152345	1144746	81359	344309

Results



Contents

Requirements

Features of on-line tabular data servers should include:

- *Consistency on input*: if a cell in different tables, always sensitive or nonsensitive.
- *Consistency on output*: same protection sense for a cell in different tables.
- *Efficiency*: quick solution.
- *Reliability*: a solution always provided

Proposal

A possible CTA-like approach:

- *First stage*: compute parameters of the CTA model and protection senses of sensitive cells.

Pros:
- ⋆ sensitivity rules satisfy consistency on input.
- ⋆ fixed protection senses satisfy consistency on output.

Cons: risk of bad assignment of senses, making problem infeasible.

- *Second stage*: solution of CTA problem (without binary variables);

Pros: Linear problem guarantees efficiency.

Cons/Opp. "Soft constraints" to deal with possible infeasibilities. This guarantees reliability.

Contents

Conclusions

- The use of heuristics to solve CTA problems is highly advisable. BCD + SAT achieves good solutions in a reasonable amount of time.
- On-line data servers provide new challenges: keeping tables consistency and reliability with fast delivery of results.
- Future CTA versions implemented for on-line servers may solve continuous (fast) problems, at the expense of considering “soft-constraints”.
- The above tasks will be included to the RCTA package in the DwB project

Thanks for your attention!