Mathematical Programming Models for Balancing Data Quality and Confidentiality in Tabular Data

James P. Kelly
OptTek Systems, Inc
Boulder, CO, U.S.A.
Kelly@OptTek.com

Rahul J. Patil
Leeds School of Business
University of Colorado, Boulder, CO
Rahul.Patil@colorado.edu

1. Mathematical Programming Model for Controlled Tabular Adjustment (CTA)

Statistical agencies use different methods to protect the confidentiality of tabular data. The most widely used method, complementary cell suppression, suppresses both primary (sensitive) and secondary (non-sensitive cells) to assure confidentiality. Despite its popularity, it suffers from severe limitations. Complementary cell suppression problem is an NP-hard problem, and thereby computationally difficult to solve. It generates tables with missing data and many end users find it difficult to analyse the resulting data from a statistical point of view. For example, users cannot easily estimate means or variance from this data. Finally, published data protected by cell suppression can be susceptible to possible disclosure.

Dandekar and Cox (2002) proposed controlled tabular adjustment (CTA) that overcomes many of the problems associated with traditional cell suppression and other perturbation methods. CTA assures confidentiality of the data by setting sensitive cells to either of their predefined protection limits and preserves table additivity by making the best mutual use of sensitive cell and non-sensitive cell adjustments. It can also control adjustment at the individual cell level. CTA has the ability of simultaneously satisfying different types of statistical goals. For example, Cox and Kelly (2003, 2004) demonstrate how CTA can be used to preserve univariate and bivariate properties of tabular data. The end result is that users receive clean, complete, and statistically accurate tabular data. Figure 1 shows a mixed integer-programming model for CTA.

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

Subject to:

For i = 1,…, n:

$M \left(y^+ - y^- \right) = 0$

$0 \le y_i^+ \le UB_i$

$0 \le y_i^- \le LB_i$

For i = 1,…, p:

$y_i^+ = LPROTECT_i * B_i$

$y_i^- = UPROTECT_i * (1 - B_i)$

where,

i = 1,…, p: denotes the p sensitive cells

i = p+1,…, n: denotes the n-p nonsensitive cells

B_i = binary (zero/one) variable denoting selection of the lower/upper limit for sensitive cell i = 1,…,p

$LPROTECT_i$ = lower deviation required to protect sensitive cell i = 1,…,p

$UPROTECT_i$ = upper deviation required to protect sensitive cell i = 1,…,p

y_i^+ = positive adjustment to cell value i

y_i^- = negative adjustment to cell value i
UB_i, LB_i = upper/lower cell bounds on change to cell i
c_i = cost per unit change in cell i

Figure 1. MILP for Optimal Controlled Tabular Adjustment (Cox 2000)

The objective function minimizes the cost due to cell deviations. Two linear cost functions are commonly used, usually defined over deviation variables $y_i^+ + y_i^-$. The first involves coefficients $c_i = 1$, corresponding to minimizing the distortion measure of total absolute adjustment, and the other $c_i = 1/(\text{cell value})$, corresponding to minimizing total percent absolute adjustment. CTA perturbs the sensitive cells until they are safe, i.e. sensitive cell values are sufficiently far from their original values. This creates inconsistency in the tabular system as sums are no longer maintained. The first constraint maintains tabular consistency. The second and third constraints are used to constrain the non-sensitive cell deviations. Usually, the upper bounds are computed using the estimated measurement errors for non-sensitive cells. The final two constraints ensure that the sensitive cells are set at their safe values. This is achieved by setting these cells at either their lower or upper protection limits. The protection limits for the cell include the minimum amount that must be added or subtracted from the true value to make the sensitive cells "safe." Cox (1980) and Willenborg and de Waal (2001) discuss protection limits theory in detail.

Unfortunately, computing an optimal solution may require a prohibitively long computation time. End users are interested in finding a good solution in a reasonable amount of time. Different heuristic methods such as an ordering heuristic have been proposed in the literature to overcome this computational problem. These heuristics have been shown to perform poorly with respect to the solution quality measure (Cox, Kelly, and Patil 2004). This motivated us to develop an algorithm that improves the computation aspects of the solution procedures.

In this paper, we propose a scheme that improves computational efficiency and solution quality. We describe a learning algorithm that combines heuristic search and math programming models to produce an efficient and effective technique for solving CTA problems.

2. Learning Algorithm

Parametric image processes create a strategic image of part of a problem to generate information about problem characteristics. Such processes have been used successfully in a network flow context (Glover et al. 2003), and are the basis for a class of mixed integer programming procedures proposed in Glover (2003). The basic idea is to introduce parameters that penalize violations of integer feasibility, and to drive selected subsets of variables in preferred directions (e.g., toward 0 or 1).

We represent the objective function as Minimize $x_o = cx$, where x is set of binary variables used to protect sensitive cells. We refer to "1" direction as (UP) and "0" direction as (DN) direction in our framework. These are called goal conditions (denoted as x_j') because we do not seek to enforce (UP) and (DN) directions by imposing them as constraints in the manner of customary branch and bound method but rather indirectly by incorporating them into the objective function of the linear programming relaxation. Let N^+ and N^- denote selected subsets of N whose union is denoted by N' whose elements contain UP and DN goal conditions respectively. Let x' denote the associated goal imposed solution vector. Let M denote a very large positive number used to impose the goal conditions.

$$(1) \quad (LP') \text{ Minimize } x_o' = \sum_{j \in N^-} (c_j + M) x_j + \sum_{j \in N^+} (c_j - M) x_j + \sum_{j \in N \diagup (N^+ + N^-)} c_j x_j$$

(LP') targets imposed down and up goal conditions by using incentive mechanism driven by the penalty M. Binary variables included in subset N^- are induced to go in the DN direction and

binary variables in subset N^+ are induced to go in UP direction. Remaining variables are free to select their own favorable directions. If a variable indeed favors a particular direction, then it will achieve its targeted goal otherwise it will show some resistance to its imposed goal. The method estimates the resistances using experimental design techniques and uses the results to reassign the varaibles to N^+ and N^-. The solution of the resulting problem produces the protected table. Space limitations prohibit us from providing a detailed algorithmic description of the learning process, however, interested readers can obtain it from the authors.

3. Results

We implemented the learning algorithm using C++, Concert Technology, and CPLEX. Figure 2 shows the performance of our proposed learning method compared to several other variable fixing heuristics. It was extremely time consuming to run larger problem instances to optimality using the CPLEX. A efficient alternative was needed to compute a better lower bound essential for measuring the optimality gap. Cox, Kelly, and Patil (2004) proposed a set partitioning-based computationally efficient method for generating a tight lower bound on the objective in the CTA context. We used the lower bound as a proxy for an optimum value for computing the optimality gap for larger instances. Lower bounds were reliable in the sense that they were consistently very close to the optimal values for those problems where an optimal solution could be verified by running CPLEX. In Figure 2, the "Learning Method (optimal)" curve identifies the optimality gap with respect to the known optimal value, and the "Learning Method (lower bound)" curve identifies the optimality gap with respect to the lower bound.

We found our learning method to yield significant improvements in reducing the optimality gap across the entire 2-dimensional test set as demonstrated by Figure 2. Optimality gap values obtained by the methods used in the other papers degraded considerably for the larger problem instances. For example, for the 25x25 table, the mean gap increased to 117.6% compared to the overall mean gap of 70 %. In contrast, the learning method consistently generated high quality solutions irrespective of the problem size, giving an overall mean gap of 6 % and a gap of 5.72% for the 25x25 problem. This was possible because of the high predictive accuracy of our algorithm. Reliable predictions helped in fixing high ranking variables to their true directions therby reduced the problem size considerably without hampering solution quality. We define *prediction accuracy* to be the percentage of variables that are correctly assigned their optimal values, from a selected set of the "top" (highest scoring) variables identified .The prediction accuracy of our method was very high - for a 14x14 problem which contains 69 variables, was 85.5% for the top 10% of the problem variables (6 correct decisions out of 7 fixed variables).

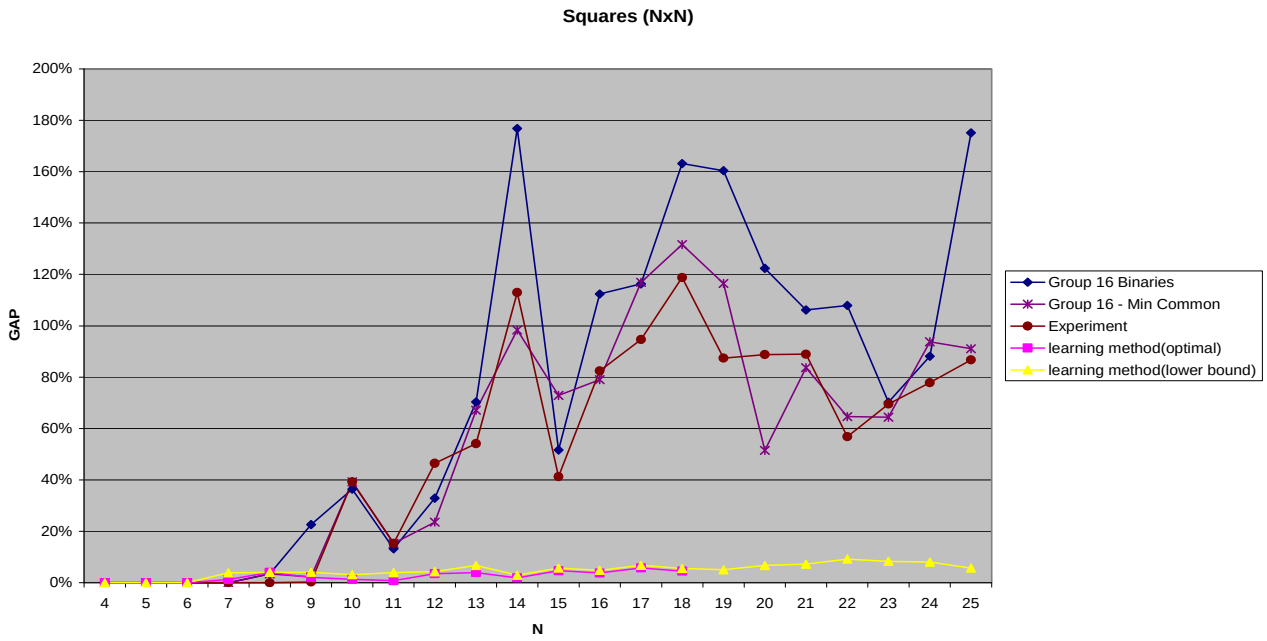

Figure 2: Performance of Learning Method on Optimality Gap

REFERENCES

Cox, L.H. (1980). Suppression Methodology and Statistical Disclosure Control. Journal of the American Statistical Association. 75,377-385.

Cox, L.H (2000). Discussion (on Session 49: Statistical Disclosure Control forEstablishment Data).ICES II: The Second International Conference on Establishment Surveys-Survey methods for businesses, farms and institutions,Alexandria, VA: American Statistical Association, 904-907.

Cox, L.H., and Kelly, J. P. (2003). Balancing Data Quality and Confidentiality for Tabular Data. Proceedings of the UNECE/EUROSTAT Work Session on Statistical Data Confidentiality, Luxembourg, 7-9 April, 2003, Monographs of Official Statistics, Luxembourg: Eurostat.

Cox, L.H., and Kelly, J.P. (2004). Preserving Quality and Confidentiality for Multivariate Tabular Data. Proceedings of Privacy in Statistical Databases 2004 (PSD 2004), Barcelona, 9-11 June, 2004, Lecture Notes in Computer Science**,** New York: Springer Verlag.

Cox, L.H., Kelly, J.P., and Patil, R.J. (2004). Computational Aspects of Controlled Tabular Adjustment: Algorithm and Analysis. Accepted to Proceedings of INFORMS Computing Society Conference 2005.

Dandekar, R.A., and Cox, L. H. (2002). Synthetic Tabular Data-An Alternative to Complementary Cell Suppression. manuscript.

Glover, F., Amini, M., and Kochenberger, G. (2003).Parametric Ghost Image Processes for Fixed Charge Problems: A Study of Transportation Networks.to appear in *Journal of Heuristics*.

Willenborg, L., and de Waal, T. D. (2001).Elements of Statistical Disclosure Control. Lecture Notes in Statistics. 155, Springer, New York

RÉSUMÉ

JAMES P. KELLY *is CEO of OptTek Systems, Inc. and is responsible for all aspects of the business. He was an Associate Professor in the College of Business at the University of Colorado. He holds bachelors and masters degrees in engineering and a Ph.D. in mathematics. Dr. Kelly has authored or co-authored numerous published articles and books in the fields of optimization, computer science and artificial intelligence. He has published over a dozen papers in the area of data confidentiality. His interests focus on the use of state-of-the-art optimization techniques for*

providing competitive advantages in engineering and business. His email address is <kelly@OptTek.com>.

***RAHUL PATIL** is a research associate at OptTek Systems while currently pursuing a Ph.D. at the University of Colorado at Boulder in Operations Research. He holds bachelors and masters degrees in engineering. Mr. Patil has more than 3 years of work experience in industry. His email address is <rpatil@Colorado.edu>.*