VI Congreso Galego de Estatística e Investigación de Operacións
Vigo 5–7 de Novembro de 2003

NEW ALGORITHMS FOR THE EDITING AND IMPUTATION PROBLEM

Jorge Riera Ledesma and Juan José Salazar González.

Departamento de Estadística, Investigación Opertativa y Computación
UNIVERSIDAD DE LA LAGUNA

RESUMEN

Data collected by statistical agencies may contain inconsistences caused by mistakes made during the acquiring, transcription and coding process. The optimization problem arising when an statistical agency must modify a microdata to guarantee that the records satisfy a set of rules, known as edits, is approached. Indeed, before using a collection of data records to infer statistical properties of some groups of responders, the agencies must check and possibly correct the consistence of the collected data. To this end, the edits must be tested on each record and whenever a record does not satisfied all the edits the agency must determine the fields in the record to be modified, as well as impute the new values. Among all the possible solutions, the statistical agency is interested in finding one concerning with the minimum number of fields to be modified, thus leading a combinatorial optimization problem known as *Editing-and-Imputation Problem*. An Integer Linear Programming model for the particular case in which the edits are linear constraints is proposed, and solving through cutting-plane approaches. The new proposals are compared to other previous published articles in literature and tested on benchmark instances. The overall performances of the new algorithms succeeded solving hard instances up to 100 variables and 50 edits about one minute of a PC.

Palabras e frases chave: Editing and Imputation, Integer Linear Programming.
Clasificación AMS: 90C11, 90C27.

1. INTRODUCTION

Data collected by statistical agencies may contain errors because questions have been misunderstood or mistakes have been made during the transcription and coding process. Therefore, since it may be impossible to get back to the original source, detection and correction of such errors becomes a necessary task before start data processing, in order to improve the integrity and quality of decision made on the basis of this information. The task of identifying records containing errors and the specific fields causing these errors is known as the *data editing* problem, and the task of changing these fields in order to correct the errors is known as *imputation*. Both are typically carried out by experts, which consumes a large amount of resources of the statistical agencies. Hence, producing automatic techniques which help these experts with such a complex work becomes an interesting goal.

More precisely, the microdata collected by the agencies consists of a set of records, each containing the answers to a set of queries. Every value in a record is known as *field*, and it contains either a discrete or a continuous value. Discrete values correspond to *categorical*

queries (e.g., Marital Status), while continuous numbers to *quantitative queries* (e.g., Age). A microdata may contain both data type.

To introduce the problem let us suppose to have n queries, indexed by a finite set $I := \{1, \ldots, n\}$. Each record a is a $(n \times 1)$ vector, say $a = [a_i : i \in I]$, whose component a_i is an entry in field i. Correctness of a record is given by a set E of consistence rules known as *edits*. This set is associated to a set $\mathcal{P}_E$ of all potential *valid records*. Given a set E of m edits, indexed by $J := \{1, \ldots, m\}$, a record a is said to be *consistent* if $a \in \mathcal{P}_E$. If microdata record a were inconsistent according to the edit set the aim of the data editing and imputation would be to modify the fewest possible items of data in order to make it consistent [3]. However, our goal in this paper shall be to minimize the weighted number of fields that would have to be changed into a^* through imputation to satisfy the set of edits. This problem is known as the *Minimum Weighted Fields to Impute* problem (MWFI) and it can be formulated using a 0-1 variable x_i and a variable y_i for all $i \in I$:

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

subject to

$$y \in (P)_E$$

$$x_i := \begin{cases} 1 & \text{if } a_i \neq y_i \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } i \in I \tag{1}$$

where the variable x_i $(i \in I)$ takes the value 1 if and only if the field i has to be modified to make y a valid record. Since the MFWI is not suitable to be solvable a mathematical programming approach, several articles in literature [3, 4, 5, 7, 8] have considered the following 0-1 integer Linear formulation (SCP):

$$\min \sum_{i \in I} w_i x_i \tag{2}$$

subject to

$$\sum_{i \in I_k \subseteq I} x_i \geq 1 \qquad \text{for all } k \in K \subseteq E \tag{3}$$

$$x_i \in \{0, 1\} \qquad \text{for all } i \in I, \tag{4}$$

where each constraint (3) is associated with an edit from the set $K \subseteq E$ of edits failed by a. The set $I_k \subseteq I$ associated to edit $k \in K$ represents the candidate set of fields to be modified in order to satisfy this edit. Hence the constraint (3) imposes that at least one of those fields must be modified if the current record a does not satisfy edit k.

Obviously, each optimal solution for MWFI satisfies constraints (3), since they state that at least one of its fields must be changed for every failed edit. However, Fellegi and Holt [3] observed that not all solutions for SCP are feasible solutions for MWFI.

Our contribution in this paper is to describe new close related approaches with better performance in practice. Section presents a simple algorithm for categorical and continuous data. This algorithm follows a scheme similar to the cutting-plane algorithm proposed in [4], which differs on the set of generated cuts. Another algorithm close related to the one proposed in [5] for continuous data and linear edits, is also presented. The new algorithm has the advantage that the cuts can be generated also from non-integer solution of the set-covering problem, thus, only a linear program must be solved at each iteration. However, if

the final solution satisfying all the (explicit and implied) edits is non-integer, the approach applies a branch-and-bound scheme in order to achieve integrability of the variables. The underlying mathematical model is presented in Section 3.1 and the overall algorithm is described in Section 3.2. The paper finishes with an extensive computational analysis in Section 4.

2. GENERAL ALGORITHM

The here discussed scheme can be applied to both types of data, categorical and continuous, and general edits. It is similar to the cutting-plane algorithms described in Garfinkel, Kunnathur and Liepins [4, 5] and summarized as follows.

Step 0: Let $K \subseteq J$ be the set of edits not satisfied by the record a;

Step 1: Generate a candidate solution solving the SCP. Let x^* be the optimal integer solution;

Step 2: Check whether there exists a record y satisfying the edits with $y_i = a_i$ when $x_i^* = 0$ for $i \in I$. If such record does exist, then stop the procedure: x^* is the optimal solution of the problem. Otherwise, add the constraint $\sum_{i:x_i^*=0} x_i \geq 1$ to the set-covering problem, and go to Step 1.

This procedure iteratively strengthen the set-covering problem with additional constraints, one at each iteration. Each iteration checks whether the current solution x^* guarantees the existence of a valid record y such that $y_i = a_i$ when $x_i^* = 0$, generating a cutting-plane inequality in the negative case. Since the cutting-plane inequality is not satisfied by x^*, this procedure computes a different solution of the restricted set-covering problem in the next iteration. Each inequality imposes that a new field not currently present in the solution must be modified.

3. ALGORITHM FOR CONTINUOUS VARIABLES AND LINEAR EDITS

New algorithms for the specific scenario in which variables are continuous numbers and each edit is given for a linear inequality are described in this section. From now on, we shall assume that each field value a_i is a continuous number in a known interval $[lb_i, ub_i]$. Moreover, we shall also assume that the explicit edit set E is given by a collection of linear constraints, and therefore it can be represented by a linear system $My \leq b$, where M is a $[m \times n]$ matrix and b is a $[m \times 1]$ vector. Under these assumptions, the set of valid records $(P)_E$ is given by the points of the polyhedron

$$(P)_E := \{y : My \leq b, lb \leq y \leq ub\}.$$

The next section presents a new mixed integer linear model for MWFI which has the advantage of exploiting the bounds $[lb_i, ub_i]$ to link variables x_i and variables y_i in order to produce a set of constraints more compact than the SCP model.

3.1 MATHEMATICAL MODEL

A mathematical model for this problem can be written by associating the 0-1 binary variable x_i with certain continuous variable y_i denoting the value of the corrected record in the field $i \in I$ as follows.

$$\min \sum_{i \in I} w_i x_i, \tag{5}$$

subject to

$$\sum_{i \in I} m_{ij} y_i \leq b_j \qquad \text{for all } j \in J \tag{6}$$

$$a_i - (a_i - lb_i)x_i \leq y_i \leq a_i + (ub_i - a_i)x_i \qquad \text{for all } i \in I \tag{7}$$

$$x_i \in \{0, 1\} \qquad \text{for all } i \in I. \tag{8}$$

Constraints (6) ensure that the corrected record is in the valid set $(P)_E$, and constraints (7) guarantee that the field i is not modified unless $x_i = 1$.

The model (5)–(8) is a simple way of writing the non-linear model pointed in, e.g., Garfinkel Kunnathur and Liepins [5]. The key point for the new model is the assumption of having the external bounds lb_i and ub_i defining the interval $[lb_i, ub_i]$ of potential values y_i for each field $i \in I$, which is not a hard hypothesis. The next section illustrates this claim describing a branch-and-cut approach.

3.2 BRANCH-AND-CUT ALGORITHM

We now describe a procedure to implicitly enumerative the solutions of the MWFI by using the model (5)–(8). An immediate way of proceeding is to apply a general-purpose software for MILP models. Since the model (5)–(8) is an MILP model, one can alternatively apply Benders' Decomposition (see, e.g, [9]) to solve it. Briefly, this method consists in iteratively solving a master problem defined only by the binary variables and whose constrains are enlarged at each iteration by solving a subproblem defined by the continuous variables. See, e.g., Shrijver [9] for details. Let us now describe the procedure in detail.

Suppose we are given with an array $x^* = [x_i : i \in I]$. For simplicity, we will assume that $x_i^* \in \{0, 1\}$, even if we will observe that this integrality requirement can be relaxed and the procedure be also applied with a minor modification. We are interested in checking if the polyhedron

$$P(x^*) := \left\{ y : \sum_{i \in I} m_{ij} y_i \leq b_j, j \in J \; ; \; a_i - (a_i - lb_i)x_i^* \leq y_i \leq a_i + (ub_i - a_i)x_i^*, i \in I \right\}$$

is empty or not. If yes, then the pattern x^* is a feasible solution for the MWFI. Otherwise, there is something wrong with x^* and we are interested in deriving a linear inequality cutting off this infeasible solution but not any feasible solution for the MWFI. Having a procedure to generated such inequality from x^* means having a cutting-plane procedure to find an optimal solution of the x^*. If x_i^* is not an integer number, then the same mechanism applied with the only difference that the procedure should continue by fixing the variables with non-integer values to either zero or one, thus entering in a branching phase. Therefore, the kernel is the procedure to generate a cut from a non-feasible x^*, which is called *separation problem*. For the MWFI it is quite immediate to solve the separation problem by applying Farkas' Lemma (see, e.g., [9]) on $P(x^*)$. In fact, this polyhedron is non-empty if and only if

$$\sum_{j \in J} \alpha_j b_j + \sum_{i \in I} \beta_i (a_i + (ub_i - a_i)x_i^*) - \sum_{i \in I} \gamma_i (a_i - (a_i - lb_i)x_i^*) \geq 0 \tag{9}$$

for all direction of the cone:

$$C := \{(\alpha, \beta, \gamma) : M^T \alpha + \beta - \gamma = 0, \alpha \geq 0, \beta \geq 0, \gamma \geq 0\}.$$

By simple operations on (9) we get a valid inequality of a feasible pattern x as follows:

$$\sum_{i \in I} [\beta_i (ub_i - a_i) + \gamma_i (a_i - lb_i)] x_i \geq \alpha^T (Ma - b). \tag{10}$$

Because $\beta_i, \gamma_i, ub_i - a_i, a_i - lb_i$ are non-negative numbers and because x_i must be 0 or 1, then it is possible to strength these constraints by rounding down a left-hand side coefficient to the right-hand side whenever it is bigger. As reported in the next section, our computational experiments proved that this strengthening is very effective for the success of the approach.

Observe that in the particular case of small values of lb_i and large values of ub_i, the strengthening operation produces the following set covering constraints:

$$\sum_{i \in I'} x_i \geq 1$$

where I' is the set of field indices with a dual variable β_i or γ_i at a positive value, which is equivalent to the field indices with non-zero value in the array $M^T\alpha$. This is a quite interesting observation since the obtained inequalities coincide with the inequalities generated by the method proposed by Garfinkel, Kunnathur and Liepins [5]. Still, an improvement of the here-proposed approach is that it can be also applied when x_i^* are non-integer solutions, which make the approach quite suitable to work in a branch-and-cut framework.

4. PRELIMINARY COMPUTATIONAL EXPERIMENTS

To measure the effectiveness of our proposal compared with other previous works, we have implemented several algorithms, using the general software CPLEX 8.1 as a framework for the mathematical models. In particular, we have considered the following algorithms:

Algorithm 1: It is a classical branch-and-bound solver for the mixed integer linear model (5)–(7), where the bound is computed by solving the linear relaxation.

Algorithm 2: It is the branch-and-cut algorithm described in Section 3.4.

Since there is not a benchmark collection of test-bed instances in the literature, we have conducted our experiments on three families of instances. The first family are randomly generated instances described in Ragsdale and McKeown [8]. This instances are generated as follows. The number of fields is $|I| = n = 100$, the number of edits is $|J| = m = 50$, the weights are all identical ($w_i = 1$ for all $i \in I$), the record values are uniformly generated in the interval [-100,+100], thus $lb_i = -100$ and $ub_i = 100$ for all $i \in I$, the right-hand side b_j where generated in the interval [0,1000], the elements m_{ij} are zero with probability 0.2, and the non-zero values are generated in [1,20] with probability 0.3 and in [-20,-1] with probability 0.7.

Finally, our last family of instances is a set of artificial instances supplied by US Census consisting of 10,994 records with 17 fields and two set of edits. The first set of 136 ratio edits is related to bound of the variables ratio as follows.

$$lb_j \leq \frac{a_i}{a_k} \leq ub_j \qquad \text{for } i, k \in I, j \in J \text{ and } i < j,$$

in which $ub_j - lb_j$ take values between 10^{-1} and 10^7. The second set corresponds to two balancing edits which have the form $a_i + a_k = a_l$ for $i, k, l \in I$.

References

[1] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems", *Numerische Mathematik* 4 (1962) 238–252.

Table 1: Average results on 10694 records

			Algorithm 1		*Algorithm* 2			
	#	*Obj.*	# *Nodes*	*Sec.*	(13)	*Cliq.*	# *Nodes*	*Sec.*
[1, 15]	223	1.75	0.229	0.008	0.135	0.000	0.000	0.004
[16, 30]	5384	2.91	0.673	0.016	0.297	0.004	0.001	0.005
[31, 45]	4281	4.10	1.342	0.026	0.308	0.008	0.000	0.005
[46, 60]	1018	5.28	3.949	0.037	0.315	0.006	0.000	0.006
[61, 75]	87	6.20	8.034	0.046	0.287	0.126	0.000	0.005

Table 2: Average results on five instances for each group

			Algorithm 1			*Algorithm* 2		
		Obj.	# *Nodes*	*Sec.*	*Limit*	(13)	# *Nodes*	*Sec.*
	[1, 10]	6.4	926.4	6.2	5/5	1189.4	631.2	6.7
	[11, 20]	9.2	6297.4	27.7	5/5	1641.4	661.6	9.4
$[-10^3, 10^3]$	[21, 30]	10.0	9822.2	47.1	5/5	17816.0	4438.0	115.6
	[31, 40]	12.0	8752.8	53.1	5/5	14348.6	4170.2	88.1
	[41, 50]	12.4	6336.0	44.6	5/5	17275.0	3511.2	120.9
	[1, 10]	3.6	1389.2	17.7	5/5	138.8	122.6	1.0
	[11, 20]	5.0	25644.8	187.4	5/5	2254.6	1159.6	13.8
$[-10^4, 10^4]$	[21, 30]	5.6	149298.5	1595.9	4/5	13319.8	3199.4	92.0
	[31, 40]	5.8	93463.0	1011.3	4/5	19524.0	4104.2	127.2
	[41, 50]	6.8	232025.0	3063.8	4/5	87244.0	15334.6	1452.4
	[1, 10]	3.6	6501.8	54.0	5/5	100.4	67.8	0.6
	[11, 20]	4.8	96563.5	1366.0	4/5	1741.4	1930.4	14.4
$[-10^5, 10^5]$	[21, 30]	6.4	-	-	0/5	2210.2	2536.2	17.0
	[31, 40]	6.6	-	-	0/5	3868.6	4063.8	31.0
	[41, 50]	7.6	-	-	0/5	7054.0	7621.0	57.2

[2] A. Caprara, M. Fischetti, "Branch and cut algorithms". In M. Dell'Amico, F. Maffioli, S. Martello (eds), *Annotated Bibliographies in Combinatorial Optimization*, Wiley, Chichester, 45–63, 1997.

[3] I.P. Fellegi, D. Holt, "A systematic approach to automatic edit and imputation", *Journal of the American Statistical Association* 71 (1976) 17–35.

[4] R.S. Garfinkel, A.S. Kunnathur, G.E. Liepins, "Optimal imputation of errorneous data: continuous data, linear constraints", *Operations Research* 34 (1986) 744–751.

[5] R.S. Garfinkel, A.S. Kunnathur, G.E. Liepins, "Error location for errorneous data: continuous data, linear constraints", *SIAM Journal on Scientific and Statistical Computing* 9 (1988) 922–931.

[6] G. E. Liepins, "A rigorous, systematic approach to automatic data editing and its statistical basis" ORNL/TM -7126, 1980.

[7] P.G. McKeown, "A mathematical programming approach to editing of continuous survey data", *SIAM Journal on Scientific and Statistical Computing* 5 (1984) 785–797.

[8] C.T. Ragsdale, P.G. McKeown, "On solving the continuous data editing problem", *Computers & Operations Research* 23 (1996) 263–273.

[9] Schrijver, A. *Theory of linear and integer programming*, John Wiley & Sons (1986).