Discrete Models for Data Imputation [1]

Renato Bruni

Università di Roma "La Sapienza" - Dip. di Informatica e Sistemistica
Via Michelangelo Buonarroti 12, Roma, Italy, 00185.

Abstract

The paper is concerned with the problem of automatic detection and correction of inconsistent or out of range data in a general process of statistical data collecting. The proposed approach is able to deal with hierarchical data containing both qualitative and quantitative values. As customary, erroneous data records are detected by formulating a set of rules. Erroneous records should then be corrected, by modifying as less as possible the erroneous data, while causing minimum perturbation to the original frequency distributions of the data. Such process is called *imputation*. By encoding the rules with linear inequalities, we convert imputation problems into integer linear programming problems. The proposed procedure is tested on a real-world case of census. Results are extremely encouraging both from the computational and from the data quality point of view.

Key words: Data Correction; Information Reconstruction; Integer Programming

1 Introduction

When dealing with a large amount of collected information, a well-known relevant problem arises: perform the requested elaboration without being misled by erroneous data. Data correctness is a crucial aspect of data quality, and, in practical cases, it has always been a very computationally demanding problem. This paper is concerned with the problem of automatic detection and correction of inconsistent or out of range data in a general process of statistical data collecting. Examples of data collecting are cases of statistical investigations, marketing analysis, experimental measures, etc. Without

Email address: bruni@dis.uniroma1.it (Renato Bruni).

[1] Work developed during the research collaboration between the Italian Statistic Office (Istat) and the University of Roma "La Sapienza" on the data processing of the 2001 Census of Italian Population.

loss of generality, our attention will be focused on the problem of a census of population carried out by collecting questionnaires. Note, however, that the proposed methodology is general, in the sense that it can process any type of data, because it operates only at the formal level, and it is not influenced by the meaning of processed data. A census is a particularly relevant process and actually constitutes the most fundamental source of information about a country, and the processing of census data is in general a difficult task for an imputation procedure [18]. Errors, or, more precisely, inconsistencies between answers or out of range answers, can be due to the original compilation of the questionnaire, or introduced during any later phase of information conversion or processing.

As customary for structured information, data are organized into units called *records*. A record has the formal structure of a set of *fields*. Giving each field a *value*, we obtain a record instance, or, simply, a record [17]. In the case of a Census, each data unit (a *family*) is composed by more sub-units (*persons*). Data having such characteristic are called *hierarchical* data. The problem of *error detection* is generally approached by formulating a set of rules that the records must respect in order to be declared *correct*. A record not respecting all the rules is declared *erroneous*. In the field of database theory, rules are also called *integrity constraints* [17]. Integrity constraints are verified by correct records, and are generally checked before inserting a record into the database. In the field of statistics, rules are often called *edits* [7]. Edits express the error condition, being verified by erroneous records. In order to simplify our exposition, we consider here rules that are verified by correct questionnaires. Clearly, rules can easily be converted from one representation to the other.

Given an erroneous questionnaire, the problem of *error correction* is usually tackled by changing some of its values, obtaining a *corrected questionnaire* which satisfies the above rules and is as close as possible to the (unknown) *original questionnaire* (the one we would have if we had no errors). Such process is called *data imputation*. Many software systems deal with the problem of questionnaires correction, by using a variety of different edits encoding and solution algorithm (e.g. [1,5,10,14,15,20]). A very well-known approach to the problem, which implies the generation of all the rules logically implied by the initial set of rules, is due to Fellegi and Holt [7]. In practical case, however, such methods suffer from severe computational limitations [15,20]. Computational efficiency could sometimes be obtained only by sacrificing the data quality issue. Another serious drawback is that simultaneous processing of quantitative and qualitative fields is seldom allowed. A large monographic section on mathematical approaches to the problem is in [6]. Mathematical programming approaches for the case of data having only continuous fields have already been proposed, e.g. [16]. Recently, a declarative semantics for the imputation problem has been proposed in [8], as an attempt to give an unambiguous formalization of the meaning of imputation and of the behav-

ior of the various imputation systems. Another logic-based approach to the problem is in [9].

A new automatic procedure for data imputation, capable of handling also hierarchical data, simultaneously operating on both qualitative and quantitative fields, and based on the use of a discrete mathematical model, is here presented. In an earlier paper, an imputation procedure for the case when all the rules are expressed by using propositional logic is already developed [4]. That would not suffice when dealing with rules containing also mathematical operators. The effectiveness of a discrete mathematical approach is also showed, for a similar problem, by the theory of Logical Analysis of Data ([2] among other papers). By encoding the rules into linear inequalities, as explained in Section 2, integer programming models of the imputation problem can be given. Note that, since a very precise syntax for writing the rules was developed, such encoding could be automatically performed. A sequence of integer programming problems, as described in Section 3, is therefore solved by means of a state-of-the-art integer programming solver (ILOG Cplex [2]). Moreover, due to the peculiar problem's structure, the efficient use of a separation routine for *set covering* problems was possible [13]. The proposed procedure is tested by executing the process of error detection and correction in the case of real world census data, as shown in Section 4. The practical behavior of the proposed procedure is evaluated both from the computational and from the data quality point of view. The latter analysis is carried out by means of recognized statistical indicators [11]. The overall software system developed for data imputation, called DIESIS (Data Imputation Editing System - Italian Software) is also described in [3].

2 Encoding Rules into Linear Inequalities

In Database theory, a *record schema* R is a set of fields $\{f_1, \ldots, f_h\}$. A *record instance* r is a set of values $\{v_1, \ldots, v_h\}$, one for each of the above fields. In the case of a Census, each record contains the answers given in one questionnaire by an entire household. A household consists in a set of individuals $I = \{1, \ldots, l\}$ living together in the same housing unit. We assume l predefined in our model, since data are generally subdivided into several data sets having the same number l of individuals per family. Such data sets are then processed independently. Census data are therefore called *hierarchical* data, i.e. data with records composed by more sub-units (the individuals). We generally consider for every individual the same set of fields $F = \{f_1, \ldots, f_m\}$. Considering all such fields for all such individuals, we have the following kind

[2] More informations available at `www.cplex.com`.

of record structure, that we will also call *questionnaire structure* Q.

$$Q = \{f_1^1, \ldots, f_m^1, \quad \ldots \quad f_1^l, \ldots, f_m^l\}$$

A *questionnaire instance* q, or, simply, a questionnaire, is therefore the following.

$$q = \{v_1^1, \ldots, v_m^1, \quad \ldots \quad v_1^l, \ldots, v_m^l\}$$

Example 2.1. In the case of a census, fields are for instance `age` or `marital status`, corresponding examples of values are `18` or `single`.

Each field f_j^i, with $i = 1 \ldots l$, $j = 1 \ldots m$, has a *domain* D_j^i, which is the set of every possible value for that field. Since we are dealing with errors, the domains include all values that can be found on questionnaires, even the erroneous ones. Fields are usually distinguished in *quantitative* and *qualitative* ones. A quantitative field is a field on whose values are applied (at least some) mathematical operators (e.g. $>$, $+$), hence such operators should be defined on its domain. Examples of quantitative fields are numbers (real or integer numbers, and we respectively speak of continuous or discrete fields), or even the elements of an ordered set. A qualitative field simply requires its domain to be a discrete set with finite number of elements. We are not interested here in considering fields ranging over domains having a non-finite number of non-ordered values. The proposed approach is able to deal with both qualitative and quantitative values.

Example 2.2. For the qualitative field `marital status`, answer can vary on a discrete set of possibilities in mutual exclusion, or, due to errors, be missing or not meaningful (`blank`).

$$D^i_{\texttt{marital status}} = \{\texttt{single}, \texttt{married}, \texttt{separate}, \texttt{divorced}, \texttt{widow}, \texttt{blank}\}$$

For the quantitative discrete field `age`, due to errors, the domain is

$$D^i_{\texttt{age}} = \mathbb{Z} \cup \{\texttt{blank}\}$$

A questionnaire instance q is declared correct if and only if it respects a set of rules $R = \{r_1, \ldots, r_p\}$. Each rule can be seen as a mathematical function r_s from the Cartesian product of all the domains (the questionnaire space) to the Boolean set {0,1}.

$$r_s : D_1^1 \times \ldots \times D_m^1 \times \quad \ldots \quad \times D_1^l \times \ldots \times D_m^l \to \{0, 1\}$$

Rules are such that q is a correct questionnaire if and only if $r_s(q) = 1$ for all $s = 1, \ldots, p$. Rules should be expressed according to some syntax. In our case, each rule is expressed as a disjunction ($\vee$) of conditions, also called propositions (p_v). Conditions can also be negated ($\neg p_v$). Therefore, rules have the structure of clauses (i.e. a disjunction of possibly negated propositions). By introducing, for each rule r_s, the set π_s of the indices of the positive conditions and the set ν_s of the indices of the negative conditions, r_s can be written as follows.

$$\bigvee_{v \in \pi_s} p_v \vee \bigvee_{v \in \nu_s} \neg p_v \tag{1}$$

Since all rules must be respected, a conjunction ($\wedge$) of conditions is simply expressed using a set of different rules, each made of a single condition. As known, all other logic relations between conditions (implication $\Rightarrow$, etc.) can be expressed by using only the above operators ($\vee$, $\wedge$, $\neg$). Differently from the case of propositional logic, conditions have an internal structure. We need to distinguish between two different structures. A condition involving values of a single field is here called a *logical condition*. A condition involving mathematical operations between values of fields is here called *mathematical condition*.

Example 2.3. A logical condition is, for instance, $(\texttt{age} < 14)$, or $(\texttt{marital status} = \texttt{married})$. A mathematical condition is, for instance: $(\texttt{age - years married} \geq 14)$.

We call *logical rules* the rules expressed only with logical conditions, *mathematical rules* the rules expressed only with mathematical conditions, and *logic-mathematical rules* the rules expressed using both type of conditions.

A special case of logical rules are the ones delimitating the *feasible domain* $\mathring{D}_j^i \subseteq D_j^i$ of every field. Very often, in fact, some values of the domain are not acceptable, regardless of values of all other fields. They are called *out-of-range* values. By removing the out-of-range values from a domain D_j^i we have the feasible domain $\mathring{D}_j^i$.

Example 2.4. A logical rule expressing that all people declaring to be married should be at least 14 years old is:

$$\neg(\texttt{marital status} = \texttt{married}) \vee \neg(\texttt{age} < 14)$$

Rules delimitating the feasible domain for the field `age` are for instance:

$$(\texttt{age} \geq 0),\ (\texttt{age} \leq 110),\ \neg(\texttt{age} = \texttt{blank})$$

One can observe that, depending on the rules, some values (e.g. `age 32` or `33`) appear to have essentially the same effect on the correctness of a questionnaire. Formally, we say that two values ${v'}_j^i$ and ${v''}_j^i$ are *equivalent* from the rules' point of view when, for every couple of questionnaires $q' = \{v_1^1, \ldots, {v'}_j^i, \ldots, v_m^l\}$ and $q'' = \{v_1^1, \ldots, {v''}_j^i, \ldots, v_m^l\}$ having all values identical except for field f_j^i , q' and q'' are either both correct or both erroneous:

$$r_s(q') = r_s(q'') \quad \text{for all} \quad s = 1, \ldots, p$$

A key point is that we can always partition each domain D_j^i into n_j subsets

$$D_j^i = S_{j1}^i \cup \ldots \cup S_{jn_j}^i$$

in such a way that all values belonging to the same S_{jk}^i are equivalent from the logical point of view (i.e. considering all and only the rules containing logical conditions). Such partition is obtained as follows. The values of each domain D_j^i explicitly appearing in the logical conditions are called *breakpoints*, or *cut-points*, for D_j^i. They represent logical *watersheds* among the values of D_j^i. Their set will be denoted by B_j^i. Domain D_j^i can now be *cut* in correspondence of each breakpoint in order to obtain subsets (which are intervals for continuous fields, sequences of values for discrete fields, sets of values for qualitative fields). By furthermore merging possibly equivalent subsets, which are detected by using again the set of rules, we obtain the above mentioned partition.

A subset for the `out-of-range` values is always present. Moreover, the value for some field can be the missing value. Such value is described as *blank*, and, depending on the field, can belong or not to the feasible domain. If the blank answer belongs to the feasible domain (such as the case of `years married`, which should be blank for unmarried people), the subset `blank` is also present. Otherwise, it belongs to the `out-of-range` subset.

Example 2.5. Consider domain $D_{\texttt{age}}^i$, together with an hypothetic set of rules R (including those of Example 2.4.) such that the set of obtained breakpoints is

$$B_{\texttt{age}}^i = \{\texttt{0},\ \texttt{14},\ \texttt{18},\ \texttt{26},\ \texttt{110},\ \texttt{blank}\}$$

From R, values below `0` or above `110` are `out-of-range`, and the blank answer does not belong to the feasible domain, hence belongs to the `out-of-range` subset. Therefore, by using again R for deciding whether each breakpoint is the upper value of a subset or the lower one of the next subset, we have the

following subsets.

$$S^i_{\texttt{age 1}} = \{\texttt{0}, \ldots, \texttt{13}\},\ \texttt{S}^{\texttt{i}}_{\texttt{age 2}} = \{\texttt{14}, \ldots, \texttt{17}\},$$
$$S^i_{\texttt{age 3}} = \{\texttt{18}, \ldots, \texttt{25}\},\ \texttt{S}^{\texttt{i}}_{\texttt{age 4}} = \{\texttt{26}, \ldots, \texttt{110}\}$$
$$S^i_{\texttt{age 5}} = \{\ldots, -1\} \cup \{\texttt{111}, \ldots\} \cup \{\texttt{blank}\}$$

Now, the variables of our mathematical model can be defined. They are a set of $l \times m$ integer variables $z^i_j \in \{0, \ldots, U\}$, one for each domain D^i_j, a set of $l(n_1 + \ldots + n_m)$ binary variables $x^i_{jk} \in \{0, 1\}$, one for each subset S^i_{jk}, and a set of $l(n_1 + \ldots + n_m)$ binary variables $\bar{x}^i_{jk} \in \{0, 1\}$, which are the complements of the x^i_{jk}. We represent each value v^i_j of the questionnaire with an integer variable z^i_j, by defining a mapping φ^i_j (a different mapping for each field) between values of the domain and integer numbers between 0 and an upper value U. U is the same for all domains, and such that no elements of any feasible domain maps to U.

$$\begin{aligned} \varphi^i_j : D^i_j &\rightarrow \{0, \ldots, U\} \\ v^i_j &\mapsto \quad z^i_j \end{aligned}$$

Mapping for integer domains is straightforward. We approximate real domains with rational domains and then map them on the set of integer positive numbers. Qualitative domains also are mapped on the set of integer numbers by choosing an ordering. The integer variables are therefore:

$$z^i_j = \varphi^i_j(v^i_j)$$

Note that, in the case of the considered application, values were wanted to be integer. However, variables z^i_j are not structurally bounded to be integer. All the `out-of-range` values map to the greater number used U. The `blank` value, when belonging to the feasible domain, is encoded with the integer value η^i_j immediately consecutive to the greatest value of the encoding of the rest of the feasible domain $\mathring{D}^i_j \setminus \texttt{blank}$. Note that $\eta^i_j < U$ is always required.

The membership of a value v^i_j to the subset S^i_{jk} is encoded by using the binary variables x^i_{jk}.

$$x^i_{jk} = \begin{cases} 1 & \text{when} \quad v^i_j \in S^i_{jk} \\ 0 & \text{when} \quad v^i_j \notin S^i_{jk} \end{cases}$$

Finally, the complementary binary variables $\bar{x}^i_{jk}$ are bound the former ones by the following so-called *coupling constraints*.

$$\bar{x}^i_{jk} + x^i_{jk} = 1$$

The presence of the complementary variables is motivated by algorithmic issues (see Section 4). Integer and binary variables are linked by using a set of linear inequalities called *bridge constraints*. They impose that, when z_j^i has a value such that v_j^i belongs to subset S_{jk}^i, the corresponding x_{jk}^i is 1 and all others binary variables $\{x_{j1}^i \dots x_{jk-1}^i, x_{jk+1}^i \dots x_{jn_j}^i\}$ are 0.

By using the above variables all the above mentioned rules can be expressed. Logic conditions p_v are expressed by using the binary variables x_{jk}^i or $\bar{x}_{jk}^i$, mathematical conditions p_v are expressed by using the integer variables z_j^i. Rules involving more than one individual (called interpersonal rules) are expressed by using the opportune variables for the different individuals. By doing so, each logical rule r_s having the structure (1) of a clause can be written as the following linear inequality

$$\sum_{i,j,k \in \pi_s} x_{jk}^i \; + \sum_{i,j,k \in \nu_s} \bar{x}_{jk}^i \; \geq 1$$

Moreover, with a commonly used slight abuse of notation, let $x, \bar{x}$ and z be the vectors respectively made of all the components $x_{jk}^i, \bar{x}_{jk}^i$ and z_j^i, $i = 1, \dots, l$, $j = 1, \dots, m$, $k = 1, \dots, n_j$. By introducing the incidence vectors a_s^π and a_s^ν respectively of the set of the positive conditions π_s and of set of the negative conditions ν_s, each logical rule can be expressed with the following vectorial notation.

$$a_s^\pi x + a_s^\nu \bar{x} \geq 1$$

The only difference when mathematical conditions are present is that they do not correspond to binary variables but to operations between the integer variables. We limit mathematical rules to those which are linear or linearizable. In particular, we allow rules composed by a division or a multiplication of two variables. For a digression on linearizable inequalities, see for instance [19]. Occasionally, further binary variables are introduced, for instance to encode disjunctions of mathematical conditions. Note, moreover, that a very precise syntax for rules was developed. Therefore, the encoding into linear inequalities could be performed by means of an automatic procedure.

Example 2.6. Consider the following logical rule for all the individuals.

$$\neg(\texttt{marital status} = \texttt{married}) \vee \neg(\texttt{age} < 14)$$

By substituting the logical variables, we have the logic formula $\bar{x}^i_{\texttt{marital status \{married\}}} \vee \bar{x}^i_{\texttt{age \{0..13\}}}$, $i = 1, \dots, l$. This becomes the following linear inequalities:

$$\bar{x}^i_{\texttt{marital status \{married\}}} + \bar{x}^i_{\texttt{age \{0..13\}}} \geq 1 \quad i = 1, \dots, l$$

Consider now the following logic-mathematical rule for all the individuals.

$$\neg(\texttt{marital status} = \texttt{married}) \vee (\texttt{age} - \texttt{years married} \geq 14)$$

By substituting the logical and integer variables, we have $\bar{x}^i_{\texttt{marital status \{married\}}} \vee (z^i_{\texttt{age}} - z^i_{\texttt{years married}} \geq 14)$, $i = 1, \ldots, l$. This becomes the following linear inequalities:

$$U\bar{x}^i_{\texttt{marital status \{married\}}} + z^i_{\texttt{age}} - z^i_{\texttt{years married}} \geq 14 \quad i = 1, \ldots, l$$

Finally, the following interpersonal mathematical rule between individual 1 and 2

$$\texttt{age (of 1)} - \texttt{age (of 2)} \geq 14$$

becomes the linear inequality

$$z^1_{\texttt{age}} - z^2_{\texttt{age}} \geq 14$$

Evidently, rules involving more than one record cannot be directly expressed by means of the above variables. However, quite often, this problem can be solved as follows. In the case when the inter-record rule involves fields which are obtained from the whole data set, such as a mean value, this can be considered constant and introduced as an additional field in each record for which the rule should be valid. In the case when such constant assumption cannot be done, on the contrary, an *augmented* record containing all the data that should be imputed together should be generated, although this may clearly increase computational times.

Altogether, from the set of rules, a set of linear inequalities is obtained (to which the coupling constraints and the bridge constraints are added). From the set of answers to a questionnaire, values for the introduced variables are given. By construction, all and only the variable assignments corresponding to correct questionnaires satisfy all the linear inequalities, hence the linear system

$$\begin{cases} A^\pi x + A^\nu \bar{x} & \geq 1 \\ B^\pi x + B^\nu \bar{x} + Bz & \geq b \\ x + \bar{x} & = 1 \\ z^i_j \in \{0, \ldots, U\}, & \ i = 1 \ldots l, \ j = 1 \ldots m \\ x^i_{jk}, \bar{x}^i_{jk} \in \{0, 1\}, & \ i = 1 \ldots l, \ j = 1 \ldots m, \ k = 1 \ldots n_j \end{cases} \tag{2}$$

The coefficient matrices A^π, A^ν are given by encoding the logical rules, B^π, B^ν, B and b are given by encoding mathematical and logic-mathematical rules, and

also by any other additional constraints such as the bridge constraints. Briefly, even if slightly improperly, a questionnaire q must satisfy (2) to be correct.

3 Modeling the Problems

After a phase of *rules validation*, were the system (2) is checked to be feasible and to have more than one solution, detection of erroneous questionnaires q^e trivially becomes the problem of testing if the variable assignment corresponding to a questionnaire instance q satisfies (2).

When detected an *erroneous questionnaire* q^e, the *imputation* process consists in changing some of its values, obtaining a *corrected questionnaire* q^c which satisfies the system (2) and is as close as possible to the (unknown) *original questionnaire* q^o (the one we would have if we had no errors). In order to reach this purpose, two general principles should be followed during the imputation process: to apply the minimum changes to erroneous data, and to modify as less as possible the original frequency distribution of the data [7]. Generally, a cost for changing each value of q^e is given, based on the reliability of the field, according to a previous statistical analysis of the data which cannot be described here. It is assumed that, when error is something unintentional, the erroneous fields are the minimum-cost set of fields that, if changed, can restore consistency. Questionnaire q^e corresponds to a variable assignment. In particular, we have a set of $l(n_1 + \ldots + n_m)$ binary values e^i_{jk} and a set of $l \times m$ integer values g^i_j. We have a cost $c^i_{jk} \in \mathbb{R}_+$ for changing each e^i_{jk}, and a cost $\tilde{c}^i_j \in \mathbb{R}_+$ for changing each g^i_j

$$\{c^1_{\ 1\ 1}, \ldots, c^1_{\ 1\ n_1}, \ldots, c^1_{\ m\ 1}, \ldots, c^1_{\ m\ n_m} \ \ldots \ c^l_{\ 1\ 1}, \ldots, c^l_{\ 1\ n_1}, \ldots, c^l_{\ m\ 1}, \ldots, c^l_{\ m\ n_m}\}$$
$$\{\tilde{c}^1_1, \ldots, \tilde{c}^1_m \ \ldots \ \tilde{c}^l_1, \ldots, \tilde{c}^l_m\}$$

The questionnaire q^c that we want to obtain corresponds to the values of the variables (x^i_{jk}, $\bar{x}^i_{jk}$, and z^i_j) at the optimal solution of the integer linear programming problems described below.

The problem of *error localization* is to find a set V of fields of minimum total cost such that q^c can be obtained from q^e by changing (only and all) the values of V. Imputation of actual values of V can then be performed in a deterministic or probabilistic way. This causes the minimum changes to erroneous data, but has little respect for the original frequency distributions.

A *donor questionnaire* q^d is a correct questionnaire which should be as close as possible to q^o. Questionnaire q^d corresponds to a variable assignment. In particular, we have a set of binary values d^i_{jk} and a set of integer values f^i_j.

Donors are selected according to an opportune distance function specified by the user.

$$\delta : (q^e, q^d) \rightarrow \mathbb{R}_+$$

The problem of *imputation through a donor* is to find a set W of fields of minimum total cost such that q^c can be obtained from q^e by copying from the donor q^d (only and all) the values of W. This is generally recognized to cause low alteration of the original frequency distributions, although changes caused to erroneous data may be not minimum. We are interested in solving both of the above problems, and in choosing for each questionnaire the solution having the best quality.

Let us introduce $l(n_1 + \ldots + n_m)$ binary variables $y^i_{jk} \in \{0, 1\}$ representing the changes we introduce in e^i_{jk}.

$$y^i_{jk} = \begin{cases} 1 & \text{if we change } e^i_{jk} \\ 0 & \text{if we keep } e^i_{jk} \end{cases}$$

Furthermore, only in the case of imputation through a donor, let us introduce $l \times m$ binary variables $w^i_j \in \{0, 1\}$ representing the changes we introduce in g^i_j.

$$w^i_j = \begin{cases} 1 & \text{if we change } g^i_j \\ 0 & \text{if we keep } g^i_j \end{cases}$$

The minimization of the total cost of the changes can be expressed with the following objective function (where the terms $\tilde{c}^i_j w^i_j$ appear only in the case of imputation through a donor).

$$\min_{y^i_{jk}, w^i_j \in \{0,1\}} \sum_{i=1}^{l} \sum_{j=1}^{m} \sum_{k=1}^{n_j} c^i_{jk} y^i_{jk} + \sum_{i=1}^{l} \sum_{j=1}^{m} \tilde{c}^i_j w^i_j \tag{3}$$

However, the constraints (2) are expressed by means of variables x^i_{jk}, $\bar{x}^i_{jk}$, and z^i_j. A key issue is that there is a relation between variables in (2) and variables in (3). In the case of error localization, this depends on the values of e^i_{jk}, as follows:

$$y^i_{jk} = \begin{cases} x^i_{jk} \ \ (= 1 - \bar{x}^i_{jk}) & \text{if } \ e^i_{jk} = 0 \\ 1 - x^i_{jk} \ \ (= \bar{x}^i_{jk}) & \text{if } \ e^i_{jk} = 1 \end{cases}$$

In fact, when $e^i_{jk} = 0$, to keep it unchanged means to put $x^i_{jk} = 0$. Since we do not change, $y^i_{jk} = 0$. On the contrary, to change it means to put $x^i_{jk} = 1$. Since we change, $y^i_{jk} = 1$. Altogether, $y^i_{jk} = x^i_{jk}$. When, instead, $e^i_{jk} = 1$, to keep it unchanged means to put $x^i_{jk} = 1$. Since we do not change, $y^i_{jk} = 0$. On

the contrary, to change it means to put $x^i_{jk} = 0$. Since we change, $y^i_{jk} = 1$. Altogether, $y^i_{jk} = 1 - x^i_{jk}$.

By using the above results, we can rewrite the objective function (3). Therefore, the problem of error localization can be modeled as follows, where the objective function and a consistent number of constraints have a *set covering* structure (see for instance [12]).

$$\min_{x^i_{jk}, \bar{x}^i_{jk} \in \{0,1\}} \sum_{i=1}^{l} \sum_{j=1}^{m} \sum_{k=1}^{n_j} (1 - e^i_{jk}) c^i_{jk} x^i_{jk} + \sum_{i=1}^{l} \sum_{j=1}^{m} \sum_{k=1}^{n_j} e^i_{jk} c^i_{jk} \bar{x}^i_{jk}$$

$$\text{Subject to} \begin{cases} A^\pi x + A^\nu \bar{x} & \geq 1 \\ B^\pi x + B^\nu \bar{x} + Bz & \geq b \\ x + \bar{x} & = 1 \\ z^i_j \in \{0, \ldots, U\}, & i = 1 \ldots l, \ j = 1 \ldots m \\ x^i_{jk}, \bar{x}^i_{jk} \in \{0,1\}, & i = 1 \ldots l, \ j = 1 \ldots m, \ k = 1 \ldots n_j \end{cases} \tag{4}$$

Conversely, in the case of imputation through a donor, relation between x^i_{jk} and y^i_{jk} depends on the values of e^i_{jk} and d^i_{jk}.

$$y^i_{jk} = \begin{cases} x^i_{jk} \ (= 1 - \bar{x}^i_{jk}) & \text{if} \ \ e^i_{jk} = 0 \text{ and } d^i_{jk} = 1 \\ 1 - x^i_{jk} \ (= \bar{x}^i_{jk}) & \text{if} \ \ e^i_{jk} = 1 \text{ and } d^i_{jk} = 0 \\ 0 & \text{if} \ \ e^i_{jk} = d^i_{jk} \end{cases}$$

In fact, when $e^i_{jk} = 0$ and $d^i_{jk} = 1$, not to copy the element means to put $x^i_{jk} = 0$. Since we do not change, $y^i_{jk} = 0$. On the contrary, to copy the element means to put $x^i_{jk} = 1$. Since we change, $y^i_{jk} = 1$. Altogether, $y^i_{jk} = x^i_{jk}$. When, instead, $e^i_{jk} = 1$ and $d^i_{jk} = 0$, not to copy the element means to put $x^i_{jk} = 1$. Since we do not change, $y^i_{jk} = 0$. On the contrary, to copy the element means to put $x^i_{jk} = 0$. Since we change, $y^i_{jk} = 1$. Altogether, $y^i_{jk} = 1 - x^i_{jk}$. Finally, when $e^i_{jk} = d^i_{jk}$, we cannot change e^i_{jk}, hence $y^i_{jk} = 0$.

Note, however, that even when we do not change x^i_{jk} from e^i_{jk} to d^i_{jk}, we still could need to change z^i_j from g^i_j to f^i_j. For instance, this could help in satisfying some mathematical constraints without changing too many values. In order to guide the choice of values for z^i_j, information obtained by the x^i_{jk} variables is used. We take for z^i_j the value of the donor when a) changes on the x^i_{jk} are made, or b) when, even if for all k the x^i_{jk} do not change, we need

to take f_j^i instead of g_j^i.

$$z_j^i = \begin{cases} g_j^i \text{ if } \forall k \in \{1,\dots,n_j\} \;\; y_{jk}^i = 0 \;\text{ and }\; w_j^i = 0 \\ f_j^i \text{ if } \exists k \in \{1,\dots,n_j\} : y_{jk}^i = 1 \text{ or if } w_j^i = 1 \end{cases}$$

For each z_j^i, n_j quantities v_{jk}^i are defined. They are 0 or 1 when the corresponding y_{jk}^i are 0 or 1.

$$v_{jk}^i = (x_{jk}^i(1 - e_{jk}^i)\; d_{jk}^i) + ((1 - x_{jk}^i)\; e_{jk}^i(1 - d_{jk}^i))$$

we have that the condition $\exists k \in \{1,\dots,n_j\} : y_{jk}^i = 1$ becomes $\sum_k v_{jk}^i = 2$, and that the condition $\forall k \in \{1,\dots,n_j\}\; y_{jk}^i = 0$ becomes $\sum_k v_{jk}^i = 0$. Therefore, z_j^i is f_j^i when $\sum_k v_{jk}^i = 2$, and we need to choose between f_j^i and g_j^i otherwise.

By using the above, we can rewrite the objective function (3). Therefore, the problem of imputation through a donor can be modeled as follows. Again, the objective function and a consistent number of constraints have a set covering structure.

$$\min_{\substack{x_{jk}^i, \bar{x}_{jk}^i \in \{0,1\}, \\ w_j^i \in \{0,1\}}} \sum_{i=1}^{l}\sum_{j=1}^{m}\sum_{k=1}^{n_j}(1 - e_{jk}^i)d_{jk}^i c_{jk}^i x_{jk}^i + \sum_{i=1}^{l}\sum_{j=1}^{m}\sum_{k=1}^{n_j} e_{jk}^i(1 - d_{jk}^i)c_{jk}^i \bar{x}_{jk}^i + \\ + \sum_{i=1}^{l}\sum_{j=1}^{m} \tilde{c}_j^i w_j^i$$

$$\text{Subject to} \begin{cases} A^\pi x + A^\nu \bar{x} & \geq 1 \\ B^\pi x + B^\nu \bar{x} + Bz & \geq b \\ \quad x + \;\bar{x} & = 1 \\ z_j^i = f_j^i(w_j^i + \frac{\sum_k v_{jk}^i}{2}) + g_j^i(1 - w_j^i - \frac{\sum_k v_{jk}^i}{2}) & i = 1\dots l,\; j = 1\dots m \\ w_j^i \leq \; 1 - \frac{\sum_k v_{jk}^i}{2} & i = 1\dots l,\; j = 1\dots m \\ z_j^i \;\in\; \{0,\dots,U\}, & i = 1\dots l,\; j = 1\dots m \\ x_{jk}^i, \bar{x}_{jk}^i \in \; \{0,1\}, & i = 1\dots l,\; j = 1\dots m,\; k = 1\dots n_j \\ w_j^i \in \{0,1\}, & i = 1\dots l,\; j = 1\dots m \end{cases} \tag{5}$$

The presence of the group of covering constraints and of that of equalities constraints allows the use of an additional separation routine during the branch-and-cut solution of the above described models.

4 Solving the Problems

The practical behavior of the proposed procedure is evaluated both from the computational and from the data quality points of view, as follows. Two large data sets representing correct questionnaires were initially perturbed by introducing errors. After this, detection of erroneous questionnaires was performed, as a trivial task. The proposed procedure is then used for the imputation of such erroneous questionnaires.

Data used for experimentations arise from the Italian Census of Population 1991. They consist in 45,716 four-person households and 20,306 six-person households (from a single region). Data perturbation consists in randomly introducing non responses (blank answers or out-of-range answers) or other valid responses (other values belonging to the feasible domain). In each data set the demographic fields `relation to head of the house`, `sex`, `marital status`, `age`, `years married` were perturbed at the four different increasing error levels (50, 100, 150, 200) described in Table 1.

Level	Perturbation	`relat.`	`sex`	`mar.st.`	`age`	`y.marr.`
050	non resp.	0.26	0.25	0.65	0.20	0.85
	other valid resp.	2.04	1.59	1.00	1.61	0.15
100	non resp.	0.52	0.50	1.30	0.40	1.70
	other valid resp.	4.08	3.17	2.00	3.22	0.30
150	non resp.	0.78	0.75	1.95	0.60	2.55
	other valid resp.	6.12	4.76	3.00	4.83	0.45
200	non resp.	1.04	1.00	2.60	0.80	3.40
	other valid resp.	8.16	6.34	4.00	6.44	0.60

Table 1: Percentages of non responses or other valid responses artificially introduced in the affected fields.

The following eight different data sets are therefore obtained:

$$(4\_050, 4\_100, 4\_150, 4\_200, 6\_050, 6\_100, 6\_150, 6\_200)$$

The set of rules used for experimentations are real rules, developed by experts of the Italian Statistic Office. Note that the possibility of using a large set of rules is required for improving the accuracy of an imputation procedure. The considered set is in fact quite large compared to other census cases, and consist in:

32 logic individual rules (to be repeated for each individual $i \in I$);
35 logic interpersonal with 2 individuals rules (to be repeated for each couple of individuals $(i, i') \in I$);
2 logic interpersonal with 3 individuals rules (to be repeated for each triple of individuals $(i, i', i'') \in I$);
1 logic-mathematic individual rule (to be repeated for each individual $i \in I$);
55 logic-mathematical interp. with 2 ind. rules (to be repeated for each couple of individuals $(i, i') \in I$);
2 logic-mathematical interp. with 3 ind. rules (to be repeated for each triple of individuals $(i, i', i'') \in I$);
1 logic-mathematical interp. with 4 ind. rule (to be repeated for each quadruple of individuals $(i, i', i'', i''') \in I$).

For each erroneous questionnaire q^e, the error localization problem (4) is solved at first, obtaining a value $z^\star_{\text{loc}}$ of the cost function. After this, a number $\sigma(q^e)$ of donor questionnaires is used. Such donors $\{q^d_1, \ldots, q^d_\sigma\}$ are selected among the correct records of the data set, by choosing the nearest ones to q^e, according to our distance function δ. Consequently, for each erroneous questionnaire q^e, $\sigma(q^e)$ problems of imputation through a donor (5) are solved, obtaining $\sigma(q^e)$ values $\{z^\star_{\text{imp }1}, \ldots, z^\star_{\text{imp }\sigma}\}$ for the cost function. By construction, such values are all greater than or equal to $z^\star_{\text{loc}}$. The corrected questionnaire q^c is finally obtained by choosing the best result among such imputations through a donor, as the one having the smallest value for the described cost function. Moreover, the number $\sigma(q^e)$ is increased when the quality of the above imputations through a donor is not satisfactory. The quality is not satisfactory when the values $\{z^\star_{\text{imp }1}, \ldots, z^\star_{\text{imp }\sigma}\}$ are all higher than $z^\star_{\text{loc}}$ multiplied by a fixed parameter $s > 1$. This means that the donors selected so far are not good, and therefore other donors should be selected for q^e. In this experimentation, σ is initially set to 5 and possibly increased until a maximum of 15, s is set to 1.4. Altogether, for each erroneous questionnaire q^e, $\sigma(q^e) + 1$ optimization problems are solved.

Problems are solved by using a commercial implementation of a state-of-the-art branch-and-bound routine for integer programming (ILOG Cplex 7.1). However, such solver allows the user to define specific separation subroutines to be used within its framework, obtaining therefore a branch-and-cut procedure. Since most of the constraints have a structure similar to those of set covering problems, a separation routine for the set covering polytope was used in order to generate valid cuts. Such separation routine, described in [13], is based on projection operations, which were possible thanks to the presence of the equality constraints above called *coupling constraints*. Since such cut generation is a relatively costing operation, it is preferable to perform it only at the very first levels of the branching tree, where its effect is greater. Each single imputation problem that is solved corresponds to an integer linear programming problem in which only some variables are generated: all variables

corresponding to fields involved in failed rules, together with all other variables connected by the rules to the former ones. Therefore, such problems do not have all the same number of variables and, consequently, of constraints. The average number of variables per problem is 3000, while the average number of constraints is 3500. Computational times in minutes for solving each data set (on a Pentium III 800MHz PC) are reported in Table 2. As observable, each single imputation problem is solved in extremely short times. Therefore, large data sets are imputed in very reasonable times. Also, this would allow the use of a more numerous set of rules. Consequently, accuracy improvements of a general process of data imputation are made possible.

Data set	Number of households	# of problems solved	Total time
4_050	45,716	320,656	53.0
4_100	45,716	346,223	96.4
4_150	45,716	385,680	130.5
4_200	45,716	416,074	157.9
6_050	20,306	145,322	85.8
6_100	20,306	160,371	139.8
6_150	20,306	186,434	174.5
6_200	20,306	198,121	202.6

Table 2: Imputation times in minutes for 4 persons household and 6 persons households.

The statistical performances of the proposed methodology, implemented in a software system called DIESIS (Data Imputation Editing System - Italian Software) has also been strictly evaluated and compared with the performance of the Canadian Nearest-neighbour Imputation Methodology (CANCEIS) [1] by a simulation study based on real data from the 1991 Italian Population Census. We report here the summarized results, while for details we refer to [11]. CANCEIS has been selected for the comparative statistical evaluation because at the time of writing it is deemed to be the best specific methodology to automatically handle hierarchical demographic data. The quality of imputed data was evaluated by comparing the original questionnaires (here known) with the corrected ones. We report in Table 3 the value of some particularly meaningful statistical indicator: the percentage of not modified values erroneously imputed by the procedure (E_{true}); the percentage of modified values not imputed (E_{mod}); the percentage of imputed values for which imputation is a failure (I_{imp}). Therefore, lower values correspond to a better data quality. Reported value is computed as average on the demographic fields `relation to head of the house`, `sex`, `marital status`, `age`, `years married`. Results

of such comparison are very encouraging: the quality of the imputation performed by the proposed procedure is generally comparable, and sometimes better, than CANCEIS.

	DIESIS			CANCEIS		
Data set	E_{true}	E_{mod}	I_{imp}	E_{true}	E_{mod}	I_{imp}
4_050	0.04	24.61	15.02	0.09	25.62	16.07
4_100	0.09	26.02	15.48	0.17	26.01	16.69
4_150	0.13	26.32	16.20	0.26	27.16	18.10
4_200	0.19	27.25	17.10	0.40	28.40	19.10
6_050	0.08	31.20	20.47	0.15	32.13	20.94
6_100	0.16	31.44	20.29	0.32	32.67	21.64
6_150	0.25	32.83	21.45	0.48	33.88	23.41
6_200	0.35	33.01	21.88	0.66	35.11	24.26

Table 3: Percentage of not modified values erroneously imputed (E_{true}), percentage of modified values not imputed (E_{mod}), percentage of imputed values for which imputation is a failure (I_{imp}).

The proposed procedure introduces surprisingly few changes in fields that were not perturbed, is able to discover more than two times out of three the values which were modified, and imputes values which are generally correct. Note that, when randomly modifying values, the record can still appear correct, in the sense that it still satisfies the rules, so detection of perturbed values inherently has no possibility of being always exact. Note, moreover, that for fields having many values, such as the case of `age`, the correct imputation is extremely difficult. Detailed results on the Italian Census 2001 correction should be made available, as far as concerning information that can be made publicly accessible, at the Italian Statistic Office web site [3] .

5 Conclusions

Imputation problems are of great relevance in every process of data collecting. They also arise when cleaning databases which can contain errors. Imputation problems have been tackled in several different manners, but satisfactory data quality and computational efficiency appear to be at odds. A discrete

[3] `www.istat.it`

mathematics model of the whole imputation process allows the implementation of an automatic procedure for data imputation. Such procedure repairs the data using donors, ensuring so that the marginal and joint distribution within the data are, as far as it is possible, preserved. The sequence of arisen integer programming problems can be solved to optimality by using state-of-the-art implementation of branch-and-cut procedures. Related computational problems for considered data sets are completely overcome. Each single imputation problem is solved to optimality in extremely short times (always less than 1 second). Therefore, computational limits of a generic imputation procedure can be pushed further by using the proposed approach. Also, this would allow the use of a more numerous set of rules. Consequently, considerable accuracy improvements of a generic process of data imputation are made possible.

The statistical performances of the proposed procedure has been strictly evaluated on real-world problems, and compared with the performance of the Canadian Nearest-neighbour Imputation Methodology, which is deemed to be, at the time of writing, the best methodology to automatically handle hierarchical demographic data. Results are very encouraging both form the computational and from the data quality point of view.

Acknowledgments. The author is grateful to Dr. M. Casaroli, Dr. A. Manzari, Dr. A. Reale and Dr. R. Torelli for their essential support in the experimental part.

References

[1] M. Bankier. Canadian Census Minimum change Donor imputation methodology. In *Proceedings of the Workshop on Data Editing*, UN/ECE, Cardiff, UK, 2000.

[2] E. Boros, P.L. Hammer, T. Ibaraki and A. Kogan. Logical analysis of numerical data. *Mathematical Programming*, 79:163–190, 1997.

[3] R. Bruni, A. Reale, R. Torelli. Optimization Techniques for Edit Validation and Data Imputation. In *Proceedings of Statistics Canada Symposium: Achieving Data Quality in a Statistical Agency*, Ottawa, Canada, 2001.

[4] R. Bruni and A. Sassano. Error Detection and Correction in Large Scale Data Collecting. In *Advances in Intelligent Data Analysis*, Lecture Notes in Computer Science 2189, 84-94, Springer, 2001.

[5] T. De Waal. WAID 4.1: a Computer Program for Imputation of Missing Values. *Research in Official Statistics* 4, 53-70, 2001.

[6] T. De Waal. *Processing of Erroneous and Unsafe Data*. Ph.D. Thesis, ERIM PhD series in Research Management, 2003.

[7] P. Fellegi and D. Holt. A Systematic Approach to Automatic edit and Imputation. *Journal of the American Statistical Association*, 71(353), 17-35, 1976.

[8] E. Franconi, A. Laureti Palma, N. Leone, S. Perri, F. Scarcello. Census Data Repair: a challenging application of Disjunctive Logic Programming. In *Proceedings of 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, (LPAR-2001) Lecture Notes in Artificial Intelligence 2250, Springer, 2001.

[9] G. Greco, S. Greco, E. Zumpano. A Logic Programming Approach to the Integration, Repairing and Querying of Inconsistent Databases. In *Proceedings of the International Conference on Logic Programming*, 2001.

[10] J.J. Hox. A Review of Current Software for Handling Missing Data. *Kwantitatieve Methoden* 62, 123-138, 1999.

[11] A. Manzari and A. Reale. Towards a new system for edit and imputation of the 2001 Italian Population Census data: a comparison with the Canadian Nearest-neighbour Imputation Methodology. in *Proc. of the 53rd Session of the International Statistical Institute*, Seoul, South Korea, 2001.

[12] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. J. Wiley, New York, 1988.

[13] P. Nobili and A. Sassano. A Separation Routine for the Set Covering Polytope. IASI-CNR *report 333*, Rome, Italy 1992.

[14] M. Pierzchala. Editing Systems and Software. In Cox, Binder, Chinnappa, Christianson, Knott (eds.), *Business Survey Methods*, J. Wiley & Sons, New York, 425-441, 1995.

[15] C. Poirier. A Functional Evaluation of Edit and Imputation Tools. *UN/ECE Work Session on Statistical Data Editing*, Working Paper n.12, Rome, Italy, 1999.

[16] C.T. Ragsdale and P.G. McKeown. On Solving the Continuous Data Editing Problem. *Computers and Operations Research* 23, 263-273, 1996.

[17] R. Ramakrishnan and J. Gehrke. *Database Management System*. McGraw Hill, 2000.

[18] United Nations Economic Commission for Europe and the Statistical Office of the European Communities. Recommendations for the 2000 censuses of population and housing in the ECE region. *Technical Report Statistical Standards And Studies* No. 49, UN/ECE Statistical Division, 1998.

[19] H.P. Williams. *Model Building in Mathematical Programming*. J. Wiley, Chichester, 1993.

[20] W.E. Winkler. State of Statistical Data Editing and current Research Problems. *UN/ECE Work Session on Statistical Data Editing*, Working Paper n.29, Rome, Italy, 1999.