Automatic editing with soft edits

Sander Scholtus

The views expressed in this paper are those of the author(s) and do not necessarily reflect the policies of Statistics Netherlands

Discussion paper (201130)

Statistics Netherlands

The Hague/Heerlen, 2011

Explanation of symbols

.	= data not available
*	= provisional figure
**	= revised provisional figure (but not definite)
x	= publication prohibited (confidential figure)
–	= nil or less than half of unit concerned
–	= (between two figures) inclusive
0 (0.0)	= less than half of unit concerned
blank	= not applicable
2010–2011	= 2010 to 2011 inclusive
2010/2011	= average of 2010 up to and including 2011
2010/’11	= crop year, financial year, school year etc. beginning in 2010 and ending in 2011
2008/’09– 2010/’11	= crop year, financial year, etc. 2008/’09 to 2010/’11 inclusive

Due to rounding, some totals may not correspond with the sum of the separate figures.

Publisher
Statistics Netherlands
Henri Faasdreef 312
2492 JP The Hague

Prepress
Statistics Netherlands
Grafimedia

Cover
Teldesign, Rotterdam

Information
Telephone +31 88 570 70 70
Telefax +31 70 337 59 94
Via contact form:
www.cbs.nl/information

Where to order
E-mail: verkoop@cbs.nl
Telefax +31 45 570 62 68

Internet
www.cbs.nl

ISSN: 1572-0314

60083201130 X-10

Automatic editing with soft edits

Sander Scholtus

Summary: Current algorithms for automatic editing used by National Statistical Institutes are often based on the Fellegi-Holt paradigm. A considerable limitation of these algorithms is that they treat all edits as hard constraints. That is to say, an edit failure is always attributed to an error in the data. In manual editing, however, subject-matter specialists also make extensive use of soft edits, i.e. constraints that identify (combinations of) values that are suspicious, but not necessarily incorrect. The inability of automatic editing methods to handle soft edits partly explains why many differences between manually edited and automatically edited data are found in practice. The object of this paper is to present a new formulation of the error localisation problem which can distinguish between hard and soft edits. Moreover, it is shown how this problem may be solved by an extended version of the error localisation algorithm of De Waal and Quere (2003).

Keywords: automatic editing, Fellegi-Holt paradigm, branch-and-bound algorithm, hard and soft edits, numerical data, categorical data

1 Introduction

An important part of every statistical process is *data editing*, i.e. detecting and correcting errors and missing values in the collected data. National Statistical Institutes (NSIs) have traditionally relied on manual editing, where the data is checked and, if necessary, adjusted by subject-matter experts. Unfortunately, this approach can be very time-consuming and expensive. In order to increase the efficiency of their statistical processes, NSIs have been developing alternative methods, such as *selective editing* and *automatic editing*. This paper focuses on the latter approach. We refer to De Waal et al. (2011) and the references therein for a discussion of selective editing and other forms of statistical data editing.

The goal of automatic editing is to accurately detect and correct errors and missing values in a data file in a fully automated manner, i.e. without human intervention. Provided that automatic editing leads to data of sufficient quality, it can be used as a partial alternative to manual editing. We refer to De Waal and Coutinho (2005) and De Waal et al. (2011) for an overview of current automatic editing techniques.

In practice, automatic editing implies that the data is made consistent with respect to a set of constraints: the so-called *edits*. Examples of edits include:

$$\textit{Profit} = \textit{Total Turnover} - \textit{Total Costs} \tag{1}$$

and

$$\textit{Profit} \leq 0.6 \times \textit{Total Turnover}. \tag{2}$$

At Statistics Netherlands, the software package SLICE was developed for the automatic editing of the structural business statistics; see De Waal (2005). SLICE edits a record of data by solving two separate problems: first the *error localisation problem*, i.e. determining which variables are erroneous or missing, and second the *consistent imputation problem*, i.e. determining new values for these variables that satisfy all the edits. The present paper mainly focuses on the error localisation problem.

Current algorithms for solving the error localisation problem at NSIs (including the one used by SLICE) are often based on the *Fellegi-Holt paradigm*; see Fellegi and Holt (1976). According to this paradigm, one tries to make a record satisfy all the edits by changing the smallest possible number of original values. It is also possible to distinguish between a priori suspicious and less suspicious values by associating a *confidence weight* to each variable. According to the generalised Fellegi-Holt paradigm, the error localisation problem is then solved by minimising the sum of the confidence weights of the variables that have to be adjusted to satisfy all the edits.

Looking back at the two examples of edits given above, it is interesting to note a conceptual difference between them. Edit (1) is an example of an edit that has to hold by definition, so that every combination of values that fails this edit necessarily contains an error. Edits of this type are commonly known as *hard edits*, *fatal edits*, or *logical edits*. Edit (2), on the other hand, is an example of an edit that identifies combinations of values that are implausible, but not necessarily incorrect. In this example, records for which *Profit* is larger than 60% of *Total Turnover* are considered suspicious. However, it is conceivable that such a combination of values is occasionally correct. Edits of this type, which do not identify errors with certainty, are known as *soft edits* or *query edits*.

An important limitation of existing algorithms for automatic editing is that they necessarily treat all edits as hard edits. That is to say, a failed edit is always attributed to an error in the data. In manual editing, however, subject-matter specialists also make extensive use of soft edits. During automatic editing, these soft edits are either not used at all, or else interpreted as hard edits. Both solutions are unsatisfactory, because in the first case some errors may be missed during automatic editing, and in the second case some correct values may be wrongfully identified as erroneous. In fact, the inability of automatic editing methods to handle soft edits partly explains why many differences between manually edited and automatically edited data are found in practice.

The object of this paper is to present a new formulation of the automatic error localisation problem which can distinguish between hard edits and soft edits. In addition, the paper shows how the error localisation algorithm of SLICE can be adapted to solve this new error localisation problem.

The remainder of this paper is organised as follows. Section 2 provides a brief summary of methods for solving the error localisation problem based on the Fellegi-Holt paradigm. A distinction between hard and soft edits is introduced in the error localisation problem in Section 3. The next two sections extend the theory behind the error localisation algorithm of SLICE to the case that not all edits have to be satisfied. Based on these theoretical results, an algorithm that solves the error localisation problem for hard and soft edits is introduced in Section 6. In Section 7, the new algorithm is illustrated by means of two small examples. Section 8 briefly discusses the consistent imputation problem for the case that not all soft edits have to be satisfied. Section 9 mentions the first experiences with a practical implementation of the new algorithm. A more complex version of the error localisation problem, which can also take the sizes of soft edit failures into account, is considered in Section 10. Finally, some concluding remarks follow in Section 11.

2 Background

2.1 Edits

The problem to be discussed in this paper entails, in its most general form, the detection of erroneous and missing values in a record containing both categorical variables $(v_1, \ldots, v_m)$ and numerical variables $(x_1, \ldots, x_p)$. These variables are supposed to satisfy a set of restrictions (edits), each of which can be written in one of the following forms:

$$\begin{aligned} \psi^k: \quad & \text{IF} && (v_1, \ldots, v_m) \in F_1^k \times \cdots \times F_m^k \\ & \text{THEN} && (x_1, \ldots, x_p) \in \left\{ \vec{x} \,|\, a_{k1}x_1 + \cdots + a_{kp}x_p + b_k \geq 0 \right\} \end{aligned} \tag{3}$$

or

$$\begin{aligned} \psi^k: \quad & \text{IF} && (v_1, \ldots, v_m) \in F_1^k \times \cdots \times F_m^k \\ & \text{THEN} && (x_1, \ldots, x_p) \in \left\{ \vec{x} \,|\, a_{k1}x_1 + \cdots + a_{kp}x_p + b_k = 0 \right\}. \end{aligned} \tag{4}$$

In these expressions, F_j^k is a subset of D_j, the domain of allowed values for the categorical variable v_j, and a_{kj} and b_k are known constants.

A record $(v_1^0, \ldots, v_m^0, x_1^0, \ldots, x_p^0)$ is said to *fail* an edit if the categorical IF-condition is true (i.e. $v_j^0 \in F_j^k$ for all $j = 1, \ldots, m$), but the numerical THEN-condition is false (i.e. either $a_{k1}x_1^0 + \cdots + a_{kp}x_p^0 + b_k < 0$ or $a_{k1}x_1^0 + \cdots + a_{kp}x_p^0 + b_k \neq 0$, depending on the form of the edit). Otherwise, we say that the edit is *satisfied* by that record. It should be noted that an edit is always satisfied by any record for which the categorical IF-condition is false, regardless of the status of the numerical THEN-condition. A record is called *consistent* if it satisfies every edit.

A categorical variable v_j is said to be *involved* in an edit if and only if $F_j^k \neq D_j$, since any edit with $F_j^k = D_j$ is failed or satisfied regardless of the value of v_j. In the same way, a numerical variable x_j is said to be involved in an edit if and only if $a_{kj} \neq 0$. We may assume that $F_j^k \neq \emptyset$; a degenerate edit with $F_j^k = \emptyset$ is never failed, and hence it can be discarded with no loss of information. The same holds for any edit with a numerical THEN-condition that is always true.

Two important special cases of (3) and (4) are edits that involve only categorical or only numerical variables. A purely categorical edit has the following form:

$$\psi^k: \text{ IF } (v_1, \ldots, v_m) \in F_1^k \times \cdots \times F_m^k \text{ THEN } \emptyset. \tag{5}$$

Edit (5) is failed by each record for which the categorical condition is true. A purely numerical edit can be written as

$$\psi^k: \ (x_1, \ldots, x_p) \in \left\{ \vec{x} \,|\, a_{k1}x_1 + \cdots + a_{kp}x_p + b_k \geq 0 \right\} \tag{6}$$

or

$$\psi^k: \ (x_1, \ldots, x_p) \in \left\{ \vec{x} \,|\, a_{k1}x_1 + \cdots + a_{kp}x_p + b_k = 0 \right\}. \tag{7}$$

Edits (6) and (7) are failed by each record for which the numerical conditions are false.

Edits (1) and (2) above are examples of purely numerical edits. A simple example of a purely categorical edit is:

$$\text{IF } (\textit{Age}, \textit{ Marital Status}) \in \{< 16\} \times \{\text{Married}\} \text{ THEN } \emptyset.$$

This edit states that persons aged less than 16 years cannot be married. Finally, an example of a mixed edit is:

$$\text{IF } \textit{Age} \in \{< 12\} \text{ THEN } \textit{Income} = 0.$$

According to this edit, persons aged less than 12 years do not have a positive income.

Most edits that occur in practice can be expressed in one of the forms (3) and (4), although this may require some rewriting and possibly the introduction of auxiliary variables; see De Waal (2005) for more details.

2.2 The Error Localisation Problem

For a given record $(v_1^0, \ldots, v_m^0, x_1^0, \ldots, x_p^0)$ and a collection of edits, it is straightforward to verify which values in the record are missing and whether any of the edits are failed.

However, given that some of the edits are failed, solving the error localisation problem – i.e. determining which values in the record are causing the edit failures – is much less straightforward. On the one hand, most edits involve more than one variable, and on the other hand, most variables are involved in more than one edit.

In order to solve the error localisation problem automatically, we have to adopt a formal strategy for finding erroneous values. According to the well-known (generalised) Fellegi-Holt paradigm, one should search for a subset of the variables which (a) can be imputed such that the adjusted record satisfies all edits, and (b) minimises the following target function:

$$D_{FH} = \sum_{j=1}^{m} w_j^C y_j^C + \sum_{j=1}^{p} w_j^N y_j^N . \tag{8}$$

Here, w_j^C and w_j^N denote the confidence weights of the categorical and numerical variables, respectively. The target variables y_j^C and y_j^N describe the structure of the solution:

$$y_j^C = \begin{cases} 1 & \text{if } v_j \text{ is to be imputed} \\ 0 & \text{otherwise} \end{cases}$$

and

$$y_j^N = \begin{cases} 1 & \text{if } x_j \text{ is to be imputed} \\ 0 & \text{otherwise} \end{cases}$$

Since variables with missing values have to be imputed with certainty, we set y_j^C or y_j^N equal to 1 when v_j^0 or x_j^0 is missing.

When trying to solve the error localisation problem, we have to be careful that by imputing a variable to resolve one edit failure, we do not cause other, previously satisfied edits to become failed. In fact, one might be tempted to argue as follows: it is clear that a subset of the variables can only be a solution to the error localisation problem if every failed edit involves at least one of these variables, i.e. if the failed edits are “covered” by these variables, so, let us choose the minimal subset with this property. Unfortunately, this condition is necessary but not sufficient for a subset of the variables to be a solution to the error localisation problem.

Consider, for instance, the following two numerical edits: $x_1 \geq x_2$ and $x_2 \geq x_3$. The unedited record $(x_1^0, x_2^0, x_3^0) = (4, 5, 6)$ fails both edits. Since x_2 is involved in both edits – that is, the failed edits are “covered” by x_2 –, one might try to obtain consistency with respect to the edits by changing only the value of x_2. This turns out to be impossible, because the imputed value would have to satisfy $4 \geq x_2$ and $x_2 \geq 6$.

The first logically sound approach to solving the error localisation problem was given by Fellegi and Holt (1976). They showed that, in order to determine whether a set of variables can be imputed to satisfy all edits simultaneously, it is necessary to derive so-called *essentially new implied edits* from the original set of edits. By adding these implied edits to the original set of edits, one obtains the so-called *complete set of edits*. For this larger set of edits, it does hold that any subset of the variables which “covers” all failed edits is a feasible solution to the error localisation problem.

In the example above, the complete set of edits consists of the original edits and the essentially new implied edit $x_1 \geq x_3$. Since the latter edit is also failed and x_2 is not involved in this edit, it is clear that imputing only x_2 does not solve the error localisation problem. On the other hand, the three failed edits are “covered” by $\{x_1, x_3\}$, and it is easy to see that imputing new values for x_1 and x_3 is indeed a feasible solution to the error localisation problem. In fact, imputing any combination of values with $x_1 \geq 5$ and $x_3 \leq 5$ leads to a consistent record in this example.

Fellegi and Holt (1976) also proposed a method to derive the complete set of edits from the original set of edits. Having obtained the complete set of edits, the error localisation problem can be solved straightforwardly for any record, by determining which edits are failed and finding the minimal subset of the variables that “covers” these edits. Unfortunately, the number of essentially new implied edits can be extremely large in practice, which means that deriving the complete set of edits is not always computationally feasible.

The next subsection focuses on a different error localisation algorithm, due to De Waal and Quere (2003), which makes use of implied edits without deriving the complete set of edits. This algorithm has been implemented in the software package SLICE and has been found to be computationally feasible in practice at Statistics Netherlands.

2.3 The Branch-and-Bound Algorithm of SLICE

A detailed description of the error localisation algorithm implemented in SLICE can be found in De Waal and Quere (2003), De Waal (2003), and De Waal et al. (2011). Here, we only discuss properties of the algorithm that will be used later in this paper.

SLICE uses a branch-and-bound algorithm to solve the error localisation problem according to the Fellegi-Holt paradigm. For each record, the algorithm sets up a binary tree; see Figure 1 for an example. In the root node of the tree, we start with the original set of edits and we select one of the variables. From the root node, two branches are added to the tree. In the first branch, the original value of the selected variable in the record is assumed to be correct, and in the second branch, this value is assumed to be erroneous. Both assumptions correspond with a transformation of the set of edits, to be described below, after which the selected variable is not involved in the edits anymore. We say that the selected variable has been treated. Next, one of the remaining variables is selected and the operation is repeated.

Once all variables have been treated, the algorithm reaches an end node of the tree. It is seen that, together, the end nodes of the binary tree enumerate all possible choices of erroneous subsets of variables. The transformed set of edits corresponding to an end node does not involve any variables, so it must either be empty or consist of elementary relations such as $1 \geq 0$ and $-1 \geq 0$. The latter example shows that some of these elementary relations may be self-contradicting. As we will discuss below, it is possible to satisfy the original edits by only imputing the variables that have been considered erroneous in the branch leading to an end node, if and only if the transformed set of edits for that end node contains no self-contradicting relations. Hence, all feasible solutions to the error localisation problem may be identified by generating all end nodes of the binary tree.

Since we are only interested in feasible solutions that minimise target function (8), we may in fact reduce the amount of work by pruning a branch of the tree as soon as it

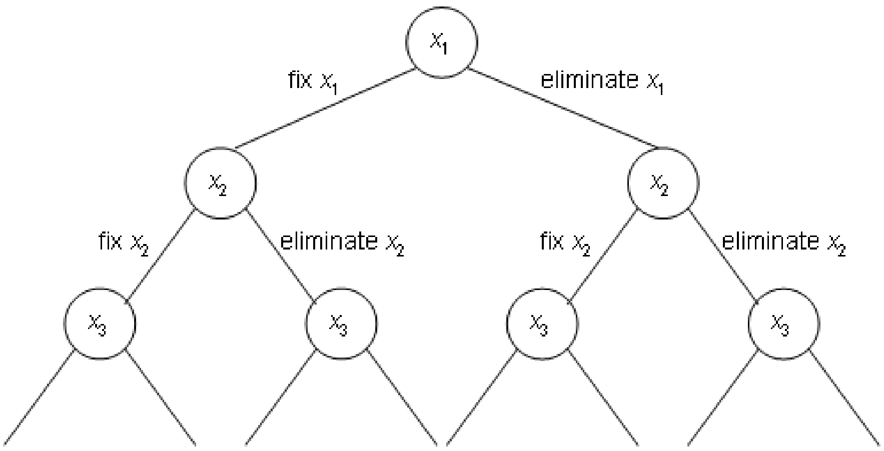

Figure 1: The branch-and-bound algorithm as a binary tree.

becomes clear that it does not lead to end nodes that correspond with feasible solutions, or only to end nodes that correspond with solutions for which the value of (8) exceeds that of the best solution found so far. This is the "bound" part of the branch-and-bound algorithm. In addition, the depth of the tree can be reduced by observing that if the record at hand contains missing values, we may immediately assume that all variables with missing values are erroneous.

We now describe the transformations of the set of edits that occur, depending on whether a variable is assumed to be correct or erroneous. A variable that is assumed to be correct is removed from the edits by simply substituting the original value from the record in the edits. This is called *fixing* a variable to its original value. A variable that is assumed to be erroneous is removed from the edits by a more complex operation, called *eliminating* a variable from the edits. Numerical variables and categorical variables are eliminated by two different, but equivalent methods.

To eliminate a numerical variable, say x_g, from a set of edits having the general forms (3) and (4), we generate implied edits by considering all pairs of edits ψ^s and ψ^t that involve x_g. We first check whether $F_j^s \cap F_j^t \neq \emptyset$ for all $j = 1, \ldots, m$; if any of these intersections yields the empty set, then the pair ψ^s and ψ^t does not generate an implied edit. If the numerical THEN-condition of one of the edits, say ψ^s, is an equality, then this equality may be rewritten as:

$$x_g = -\frac{1}{a_{sg}} \left(a_{s1}x_1 + \cdots + a_{s,g-1}x_{g-1} + a_{s,g+1}x_{g+1} + \cdots + a_{sp}x_p + b_s \right). \tag{9}$$

The numerical THEN-condition of the implied edit is generated by substituting this expression for x_g in the THEN-condition of ψ^t. The categorical IF-condition of the implied edit is found by taking the non-empty intersections $F_j^* = F_j^s \cap F_j^t$ for $j = 1, \ldots, m$.

If the numerical THEN-conditions of ψ^s and ψ^t are both inequalities, the algorithm uses a technique called *Fourier-Motzkin elimination* to generate an implied edit. We first check whether the coefficients of x_g in the two edits have opposite signs, that is $a_{sg}a_{tg} < 0$. Otherwise, this pair of edits does not generate an implied edit. It can be assumed without loss of generality that $a_{sg} < 0$ and $a_{tg} > 0$. This means that the numerical condition of ψ^s can be written as an upper bound on x_g, given the values of the other variables:

$$x_g \leq \frac{1}{-a_{sg}} \left(a_{s1}x_1 + \cdots + a_{s,g-1}x_{g-1} + a_{s,g+1}x_{g+1} + \cdots + a_{sp}x_p + b_s\right). \tag{10}$$

Similarly, the numerical condition of ψ^t can be written as a lower bound on x_g:

$$x_g \geq \frac{1}{-a_{tg}} \left(a_{t1}x_1 + \cdots + a_{t,g-1}x_{g-1} + a_{t,g+1}x_{g+1} + \cdots + a_{tp}x_p + b_t\right). \tag{11}$$

Combining the two bounds and removing x_g, we obtain the implicit condition

$$\begin{aligned}&\frac{1}{-a_{tg}} \left(a_{t1}x_1 + \cdots + a_{t,g-1}x_{g-1} + a_{t,g+1}x_{g+1} + \cdots + a_{tp}x_p + b_t\right)\\ &\qquad \leq \frac{1}{-a_{sg}} \left(a_{s1}x_1 + \cdots + a_{s,g-1}x_{g-1} + a_{s,g+1}x_{g+1} + \cdots + a_{sp}x_p + b_s\right),\end{aligned}$$

which can be written in the general form of a numerical condition as

$$(x_1, \ldots, x_p) \in \left\{\vec{x} \,|\, a_1^* x_1 + \cdots + a_p^* x_p + b^* \geq 0\right\}$$

with $a_j^* = a_{tg}a_{sj} - a_{sg}a_{tj}$ and $b^* = a_{tg}b_s - a_{sg}b_t$. This becomes the numerical THEN-condition of the implied edit. Like before, the categorical IF-condition of the implied edit consists of the non-empty intersections $F_j^* = F_j^s \cap F_j^t$. That is to say, in this case the implied edit generated from ψ^s and ψ^t is

$$\begin{aligned}\psi^* : \quad & \text{IF} \quad && (v_1, \ldots, v_m) \in F_1^* \times \cdots \times F_m^* \\ & \text{THEN} && (x_1, \ldots, x_p) \in \left\{\vec{x} \,|\, a_1^* x_1 + \cdots + a_p^* x_p + b^* \geq 0\right\}\end{aligned} \tag{12}$$

This edit does not involve x_g, since $a_g^* = 0$.

In this manner, implied edits are generated from all pairs of edits that involve x_g. These edits are added to all original edits that do not involve x_g, to find the transformed set of edits obtained by eliminating x_g.

If x_g happens to be involved in a purely numerical equality, i.e. an edit of the form (7), then De Waal and Quere (2003) suggest an alternative technique called the *equality elimination rule*. According to this rule, the purely numerical equality is rewritten as (9) and this expression is substituted in all other edits that involve x_g. The other edits are not combined pairwise. Obviously, the equality elimination rule leads to less implied edits. Unfortunately, we are not able to use it in our new algorithm, as will be seen later.

For the elimination of categorical variables, De Waal and Quere (2003) make the assumption that these variables are only selected when all numerical variables have been either eliminated or fixed. This assumption simplifies the algorithm considerably. It implies that categorical variables are always eliminated from purely categorical edits of the form (5). To eliminate a categorical variable, say v_g, from a set of edits of the form (5), we use a technique that was first described by Fellegi and Holt (1976).

Consider all minimal sets of edits T with the following properties:

$$F_g^*(T) = \bigcup_{k \in T} F_g^k = D_g \tag{13}$$

and

$$F_j^*(T) = \bigcap_{k \in T} F_j^k \neq \emptyset, \qquad j = 1,\ldots,g-1,g+1,\ldots,m. \tag{14}$$

Here, by "minimal" we mean that property (13) does not hold for any set $T' \subset T$. Each of these minimal sets T generates an implied edit

$$\text{IF } (v_1,\ldots,v_m) \in F_1^*(T) \times \cdots \times F_m^*(T) \text{ THEN } \emptyset, \tag{15}$$

which does not involve v_g because of property (13). These implied edits are added to all original edits that do not involve v_g, to find the transformed set of edits obtained by eliminating v_g.

A fundamental property of both elimination techniques, for numerical and categorical variables, is exhibited by the following result.

Theorem 1 *Consider a node in the binary tree with an associated set of edits Ψ_0 and let V_0 be the index set of variables that have not been treated yet. Suppose that a categorical or numerical variable with index g is eliminated or fixed to reach the next node, with an associated set of edits Ψ_1, and define $V_1 := V_0 \backslash \{g\}$. Then there exist values u_j for the variables with $j \in V_1$ that satisfy all edits in Ψ_1, if and only if there also exists a value u_g such that the values u_j for the variables with $j \in V_0$ satisfy all edits in Ψ_0.*

Proof See Theorem 8.1 in De Waal (2003) or Theorem 4.3 in De Waal et al. (2011). □

The above-mentioned correspondence between end nodes without self-contradicting elementary relations and feasible solutions to the error localisation problem follows from a repeated application of this theorem; cf. De Waal (2003) or De Waal et al. (2011).

3 An Error Localisation Problem with Hard and Soft Edits

In the formulation of the error localisation problem given in Section 2.2, which is based on the Fellegi-Holt paradigm, it is tacitly assumed that all edits are hard edits. Consequently, the only subsets of the variables that are considered as feasible solutions to this problem, are those which can be imputed to make the record consistent with respect to all edits. As mentioned in the introduction to this paper, this interpretation of all edits as hard edits during automatic editing can lead to systematic differences between automatic editing and manual editing, because it precludes a meaningful use of soft edits. In this section, we suggest a new formulation of the error localisation problem which distinguishes between hard and soft edits.

Let Ψ denote the set of edits to be used in the error localisation problem. We assume that this set can be partitioned into two disjoint subsets: $\Psi = \Psi_H \cup \Psi_S$. The edits in Ψ_H are hard edits, the edits in Ψ_S are soft edits. From now on, a subset of the variables is considered as a feasible solution to the error localisation problem, if it can be imputed to produce a record that satisfies all edits in Ψ_H. Moreover, we want to use the status of the imputed record with respect to the edits in Ψ_S as auxiliary information in the choice of an optimal solution to the error localisation problem. This may be done by adding a second term to (8).

To make this more precise, the objective of the new error localisation problem is to find a subset of the variables which (a) can be imputed such that the adjusted record satisfies all edits in Ψ_H, and (b) minimises the following target function:

$$D = \lambda D_{FH} + (1-\lambda) D_{soft}, \tag{16}$$

where D_{soft} represents the costs associated with failed edits in Ψ_S. The parameter $\lambda \in [0,1]$ determines the relative contribution of both terms in (16). If the two terms are to be considered equally important, we choose $\lambda = 1/2$. The original Fellegi-Holt paradigm is recovered as a special case by choosing $\lambda = 1$. Thus, the new error localisation problem can be seen as a generalisation of the old one.

In order to use (16) in practice, one has to choose an expression for D_{soft}. Probably the easiest way to assign costs to failed soft edits, is to associate a fixed *failure weight* s_k to each edit in Ψ_S, and to define D_{soft} as the sum of the failure weights of the soft edits that remain failed:

$$D_{soft} = \sum_{k=1}^{K_S} s_k z_k, \tag{17}$$

with K_S the number of edits in Ψ_S and

$$z_k = \begin{cases} 1 & \text{if the } k^{\text{th}} \text{ soft edit is failed} \\ 0 & \text{otherwise} \end{cases}$$

The failure weights might be chosen by subject-matter experts, analogously to the confidence weights, to express the importance that is attached to different soft edits from a subject-matter related point of view. Alternatively, the failure weights might be based on the proportion of records that fail each soft edit in a historical data set which has been edited manually.

A drawback of using fixed failure weights is that they do not take the size of the edit failures into account: every record that fails a particular soft edit receives the same contribution to D_{soft}, namely s_k. This differs from the way soft edits are interpreted by human editors. According to human editors, a failed soft edit points to a combination of values that is suspicious, and the degree of suspicion depends heavily on the size of the edit failure: a small failure is ignored more easily than a large failure. Hence, it seems appropriate to take the size of the edit failures into account in D_{soft}. This point will be taken up in Section 10 and an appendix to this paper, since it leads to some additional difficulties. For now, we assume that expression (17) is used in the error localisation problem.

We should mention that taking soft restrictions into account by adding an appropriate term to a target function is a commonly used technique in mathematical optimisation. To give an example, it has been applied to the so-called benchmarking problem for national accounts, which can be solved by minimising a quadratic target function; see, for instance, Magnus et al. (2000) and Bikker et al. (2010). To our best knowledge, this approach has not been applied previously in the context of the error localisation problem.

4 A Short Theory of Edit Failures, Part One: Numerical Data

Having formulated a new error localisation problem, we will now show how this problem may be solved by an adapted version of the branch-and-bound algorithm of De Waal and Quere (2003). To do this, we first need to derive a similar result to Theorem 1 in the case that some of the edits may be failed. In particular, we want to answer the following question: if certain implied edits are failed, what does this say about the original edits? For convenience, we first examine the case of purely numerical data. The next section examines the case of purely categorical and mixed data.

In the case of purely numerical data, all edits take the form (6) or (7). Moreover, the implied edit (12) reduces to

$$\psi^* : \ (x_1, \ldots, x_p) \in \left\{ \vec{x} \,|\, a_1^* x_1 + \cdots + a_p^* x_p + b^* \geq 0 \right\}. \tag{18}$$

It is a fundamental property of Fourier-Motzkin elimination that a given set of values for $x_1, \ldots, x_{g-1}, x_{g+1}, \ldots, x_p$ satisfies the implied edit (18), if and only if there exists a value for x_g which, together with the other values, satisfies both edits (10) and (11). Actually, this property forms the basis for the proof of Theorem 1. Looking at this equivalence from another point of view, if the values of $x_1, \ldots, x_{g-1}, x_{g+1}, \ldots, x_p$ do *not* satisfy the implied edit (18), then it holds that

$$\begin{aligned} &\frac{1}{-a_{tg}} \left(a_{t1} x_1 + \cdots + a_{t,g-1} x_{g-1} + a_{t,g+1} x_{g+1} + \cdots + a_{tp} x_p + b_t \right) \\ &\qquad > \frac{1}{-a_{sg}} \left(a_{s1} x_1 + \cdots + a_{s,g-1} x_{g-1} + a_{s,g+1} x_{g+1} + \cdots + a_{sp} x_p + b_s \right), \end{aligned}$$

and hence it is impossible to satisfy (10) and (11) simultaneously. However, it is still possible in this case to find a value for x_g that satisfies either edit (10) or edit (11). The same holds when the implied edit is generated by eliminating x_g from a combination of an equality and another edit. While this observation is more or less trivial, it forms the basis for the proof of Theorem 2 below.

Suppose that, at some point during an execution of the branch-and-bound algorithm of De Waal and Quere (2003), q numerical variables have been treated (i.e. either eliminated or fixed), and suppose that the equality elimination rule has not been used. We denote the current set of edits by Ψ_q, and the edits in this set by ψ_q^k. By definition, $\Psi_0 \equiv \Psi$, the original set of edits. It is possible to associate with each current edit ψ_q^k an index set B_q^k, which contains the indices of all the original edits that have been used, directly or indirectly, to derive this edit. In fact, B_q^k can be defined recursively as follows:

- For an original edit ψ_0^k, we define $B_0^k := \{k\}$.
- For an edit ψ_q^k which is derived from one other edit ψ_{q-1}^l, by fixing a variable to its original value or by simply copying the edit, we define $B_q^k := B_{q-1}^l$.
- For an edit ψ_q^k which is derived by eliminating a variable from two other edits ψ_{q-1}^s and ψ_{q-1}^t, we define $B_q^k := B_{q-1}^s \cup B_{q-1}^t$.

A set B is called a *representing set* of a collection of sets $B_q^{k_1}, \ldots, B_q^{k_r}$, if it contains at least one element from each of $B_q^{k_1}, \ldots, B_q^{k_r}$; see, for instance, Mirsky (1971, p. 25). It should be noted that, in our case, the elements in a representing set B refer to a subset of Ψ_0, that is, a subset of the original edits.

We can now formulate the following theorem.

Theorem 2 *Suppose that q numerical variables have been treated (without using the equality elimination rule) and that the current set of numerical edits can be partitioned as $\Psi_q = \Psi_q^{(1)} \cup \Psi_q^{(2)}$, where the edits in $\Psi_q^{(1)}$ are satisfied by the original values of the $p-q$ remaining variables, and the edits in $\Psi_q^{(2)}$ are failed. Let B be a representing set of the index sets B_q^k for all $\psi_q^k \in \Psi_q^{(2)}$. Then there exist values for the eliminated variables which, together with the original values of the other variables, satisfy all original edits except those in B.*

Proof The proof of this theorem is given in Appendix A.1. □

The following example demonstrates the use of Theorem 2.

Example

Suppose that there are three variables (x_1, x_2, x_3) that should satisfy the following eight edits:

$$
\begin{array}{lrcr}
\psi_0^1: & x_1 + x_2 + x_3 & = & 20 \\
\psi_0^2: & x_1 - x_2 & \geq & 3 \\
\psi_0^3: & -x_1 + x_2 & \geq & -6 \\
\psi_0^4: & -x_1 + x_3 & \geq & 5 \\
\psi_0^5: & x_1 - x_3 & \geq & -10 \\
\psi_0^6: & x_1 & \geq & 0 \\
\psi_0^7: & x_2 & \geq & 0 \\
\psi_0^8: & x_3 & \geq & 0
\end{array}
$$

The record $(x_1^0, x_2^0, x_3^0) = (10, 1, -3)$ fails the edits ψ_0^1, ψ_0^3, ψ_0^4, and ψ_0^8. Upon eliminating x_1 from the original set of edits, we find the following updated set of edits:

$$
\begin{array}{lrcrl}
\psi_1^1: & -2x_2 - x_3 & \geq & -17 & (B_1^1 = \{1,2\}) \\
\psi_1^2: & 2x_2 + x_3 & \geq & 14 & (B_1^2 = \{1,3\}) \\
\psi_1^3: & x_2 + 2x_3 & \geq & 25 & (B_1^3 = \{1,4\}) \\
\psi_1^4: & -x_2 - 2x_3 & \geq & -30 & (B_1^4 = \{1,5\}) \\
\psi_1^5: & -x_2 - x_3 & \geq & -20 & (B_1^5 = \{1,6\}) \\
\psi_1^6: & x_2 & \geq & 0 & (B_1^6 = \{7\}) \\
\psi_1^7: & x_3 & \geq & 0 & (B_1^7 = \{8\}) \\
\psi_1^8: & 0 & \geq & -3 & (B_1^8 = \{2,3\}) \\
\psi_1^9: & -x_2 + x_3 & \geq & 8 & (B_1^9 = \{2,4\}) \\
\psi_1^{10}: & x_2 - x_3 & \geq & -16 & (B_1^{10} = \{3,5\}) \\
\psi_1^{11}: & 0 & \geq & -5 & (B_1^{11} = \{4,5\}) \\
\psi_1^{12}: & x_2 & \geq & -6 & (B_1^{12} = \{3,6\}) \\
\psi_1^{13}: & x_3 & \geq & 5 & (B_1^{13} = \{4,6\})
\end{array}
$$

The index set B_1^k from Theorem 2 is displayed in brackets next to each edit. Note that the edits $\psi_1^8, \ldots, \psi_1^{13}$ are generated because we do not use the equality elimination rule.

By substituting the original values of x_2 and x_3 in the current set of edits, we see that ψ_1^2, ψ_1^3, ψ_1^7, ψ_1^9, and ψ_1^{13} are failed. The set $B = \{1,4,8\}$ is a representing set for the index sets B_1^2, B_1^3, B_1^7, B_1^9, and B_1^{13}. Hence, according to Theorem 2, there exists a value for x_1 which, together with the original values of x_2 and x_3, satisfies the original edits apart from ψ_0^1, ψ_0^4, and ψ_0^8.

In fact, substituting $x_2^0 = 1$ and $x_3^0 = -3$ into the original edits yields the following restrictions for x_1:

$$\begin{array}{llcr}
\psi_0^1: & x_1 & = & 22 \\
\psi_0^2: & x_1 & \geq & 4 \\
\psi_0^3: & x_1 & \leq & 7 \\
\psi_0^4: & x_1 & \leq & -8 \\
\psi_0^5: & x_1 & \geq & -13 \\
\psi_0^6: & x_1 & \geq & 0 \\
\psi_0^7: & 1 & \geq & 0 \\
\psi_0^8: & -3 & \geq & 0
\end{array}$$

It is easy to see that, if we leave out the first, fourth, and eighth restrictions, then it is possible to find a feasible value for x_1; in fact, any value in the interval $[4,7]$ will do.

Another representing set for B_1^2, B_1^3, B_1^7, B_1^9, and B_1^{13} is given by $B = \{3,4,8\}$. Hence, Theorem 2 guarantees that it is also possible to find a value for x_1 which satisfies the above restrictions, except for the third, fourth and eighth. In this case, imputing $x_1 = 22$ is the only feasible solution. □

The importance of Theorem 2 is that it enables one to evaluate, at each node of the branch-and-bound algorithm, which combinations of the original edits could be satisfied by imputing the variables that have been eliminated so far, and also which edits would remain failed. In particular, if we distinguish between hard and soft original edits, then this result makes it possible to use the branch-and-bound algorithm to find all feasible solutions to the new error localisation problem from Section 3, and also to evaluate, for each feasible solution, which of the soft edits remain failed, and hence evaluate the value of D_{soft}. This idea will be elaborated in Section 6.

Interestingly, the above-defined sets B_q^k may also be used to identify redundant edits, i.e. edits that follow directly from a combination of the other edits. According to a result found independently by Černikov (1963) and Kohler (1967), when q variables have been eliminated by Fourier-Motzkin elimination, all edits with more than $q+1$ elements in B_q^k are redundant; see also Williams (1986) and De Jonge and Van der Loo (2011) for a discussion of this result.

5 A Short Theory of Edit Failures, Part Two: Categorical Data

We now derive a similar result to Theorem 2 for the case of purely categorical data. At the end of this section, we combine the two results so that they may also be applied to mixed data.

In the case of purely categorical data, all edits take the form (5). Let us consider the elimination method for categorical variables described in Section 2.3. A fundamental property of this method is that a given set of values for $v_1, \ldots, v_{g-1}, v_{g+1}, \ldots, v_m$

satisfies the implied edit (15), if and only if there exists a value for v_g which, together with the other values, satisfies all edits ψ^k with $k \in T$. In other words, if the values of $v_1, \ldots, v_{g-1}, v_{g+1}, \ldots, v_m$ do not satisfy (15), then it is not possible to satisfy all edits with $k \in T$ simultaneously. This can be seen by observing that, by property (14), $F_j^*(T) \subseteq F_j^k$ for all $j \neq g$ and all $k \in T$. Hence, if (15) is failed by $v_1, \ldots, v_{g-1}, v_{g+1}, \ldots, v_m$, then plugging these values into an original edit with $k \in T$ produces a non-degenerate univariate edit for v_g. Moreover, every possible value of v_g fails at least one of these univariate edits, because of property (13).

Interestingly, it is still possible in this case to find a value for v_g that satisfies all edits in T but one. This follows from property (13) and the fact that T is a minimal set having this property: for each $k \in T$, F_g^k must contain at least one value from D_g that is not covered by any other F_g^l with $l \in T$, since otherwise $T' = T \setminus \{k\}$ would also satisfy property (13). Thus, for every $k \in T$, there exists a value of v_g that would fail this edit, but none of the other edits in T.

Suppose that, at some point during an execution of the branch-and-bound algorithm of De Waal and Quere (2003), q categorical variables have been treated (i.e. either eliminated or fixed). As before, we denote the current set of edits by Ψ_q, and the edits in this set by ψ_q^k. Again, we associate with each current edit ψ_q^k the index set B_q^k of all original edits that have been used, directly or indirectly, to derive this edit. This time, B_q^k is defined recursively as follows:

- For an original edit ψ_0^k, we define $B_0^k := \{k\}$.
- For an edit ψ_q^k which is derived from one other edit ψ_{q-1}^l, by fixing a variable to its original value or by simply copying the edit, we define $B_q^k := B_{q-1}^l$.
- For an edit ψ_q^k which is derived by eliminating a variable from a set of edits ψ_{q-1}^t $(t \in T)$, we define $B_q^k := \bigcup_{t \in T} B_{q-1}^t$.

We now present the analogue of Theorem 2 for categorical data.

Theorem 3 *Suppose that q categorical variables have been treated and that the current set of categorical edits can be partitioned as $\Psi_q = \Psi_q^{(1)} \cup \Psi_q^{(2)}$, where the edits in $\Psi_q^{(1)}$ are satisfied by the original values of the $m-q$ remaining variables, and the edits in $\Psi_q^{(2)}$ are failed. Let B be a representing set of the index sets B_q^k for all $\psi_q^k \in \Psi_q^{(2)}$. Then there exist values for the eliminated variables which, together with the original values of the other variables, satisfy all original edits except those in B.*

Proof The proof of this theorem is given in Appendix A.2. □

As in the previous section, we illustrate the theorem by means of an example.

Example
There are three categorical variables (v_1, v_2, v_3), with domains $D_1 = \{1,2,3,4\}$ and $D_2 = D_3 = \{1,2,3\}$, which should satisfy the following six edits:

ψ_0^1 : IF $(v_1, v_2, v_3) \in \{1,2\} \times D_2 \times \{2,3\}$ THEN $\emptyset$
ψ_0^2 : IF $(v_1, v_2, v_3) \in \{1,3\} \times \{1,2\} \times D_3$ THEN $\emptyset$
ψ_0^3 : IF $(v_1, v_2, v_3) \in \{2,4\} \times \{1\} \times \{3\}$ THEN $\emptyset$
ψ_0^4 : IF $(v_1, v_2, v_3) \in \{1,2,3\} \times \{2\} \times D_3$ THEN $\emptyset$
ψ_0^5 : IF $(v_1, v_2, v_3) \in \{4\} \times D_2 \times \{2\}$ THEN $\emptyset$
ψ_0^6 : IF $(v_1, v_2, v_3) \in D_1 \times \{3\} \times \{3\}$ THEN $\emptyset$

Consider the unedited record $(v_1^0, v_2^0, v_3^0) = (1,2,2)$. This record fails ψ_0^1, ψ_0^2, and ψ_0^4. Upon eliminating v_1 from the original edits, we obtain the following updated set of edits:

ψ_1^1 : IF $(v_1, v_2, v_3) \in D_1 \times \{1,2\} \times \{2\}$ THEN $\emptyset$ $(B_1^1 = \{1,2,5\})$
ψ_1^2 : IF $(v_1, v_2, v_3) \in D_1 \times \{1\} \times \{3\}$ THEN $\emptyset$ $(B_1^2 = \{2,3\})$
ψ_1^3 : IF $(v_1, v_2, v_3) \in D_1 \times \{2\} \times \{2\}$ THEN $\emptyset$ $(B_1^3 = \{4,5\})$
ψ_1^4 : IF $(v_1, v_2, v_3) \in D_1 \times \{3\} \times \{3\}$ THEN $\emptyset$ $(B_1^4 = \{6\})$

The index sets of generating edits B_1^k are mentioned in brackets. Plugging the original values $v_2^0 = 2$ and $v_3^0 = 2$ into these edits, we see that ψ_1^1 and ψ_1^3 are failed. Since $B = \{5\}$ is a representing set of the sets B_1^1 and B_1^3, Theorem 3 states that it is possible to satisfy all original edits, except for ψ_0^5, by imputing only v_1.

In fact, plugging $v_2^0 = 2$ and $v_3^0 = 2$ into the original edits, we obtain the following univariate restrictions for v_1:

ψ_0^1 : IF $v_1 \in \{1,2\}$ THEN $\emptyset$
ψ_0^2 : IF $v_1 \in \{1,3\}$ THEN $\emptyset$
ψ_0^3 : —
ψ_0^4 : IF $v_1 \in \{1,2,3\}$ THEN $\emptyset$
ψ_0^5 : IF $v_1 \in \{4\}$ THEN $\emptyset$
ψ_0^6 : —

Clearly, we can satisfy all of these restrictions, except for the fifth, by choosing $v_1 = 4$.

In this example, other representing sets of B_1^1 and B_1^3 are given by $B = \{1,4\}$ and $B = \{2,4\}$. It is easy to see that imputing $v_1 = 2$ and $v_1 = 3$, respectively, yields records that fail precisely these original edits. □

Finally, we remark that Theorem 2 and Theorem 3 can be used together when the data is a mix of categorical and numerical variables. This follows from the structure of the branch-and-bound algorithm of De Waal and Quere (2003), where categorical variables are only treated once all numerical variables have been eliminated or fixed. Hence, the two results may be applied consecutively. There is a slight difference in the procedure for eliminating numerical variables, namely that implied edits are only generated from pairs of edits having an overlapping IF-condition; see Section 2.3. However, this does not affect the correctness of Theorem 2.

6 Solving the Error Localisation Problem with Hard and Soft Edits

We now describe an adapted version of the branch-and-bound algorithm of De Waal and Quere (2003) which may be used to solve the error localisation problem defined in Section 3. In the root node of the binary tree, the number of treated variables q is initialised to zero. The initial set of hard edits Ψ_{0H} is the original set of hard edits Ψ_H, and the initial set of soft edits Ψ_{0S} is the original set of soft edits Ψ_S. Moreover, we associate an index set $B^k_{0S} := \{k\}$ to each soft edit $\psi^k_{0S} \in \Psi_{0S}$. We do *not* associate such index sets to the hard edits, because we do not need them.

If the current node of the binary tree is not an end node, then an untreated variable is selected, say x_g or v_g. As in the original algorithm, categorical variables are only selected once all numerical variables have been treated. Two new branches are generated. In the first branch, x_g or v_g is fixed to its original value, and in the second branch, x_g or v_g is eliminated from the current sets of edits Ψ_{qH} and Ψ_{qS}. Both procedures are carried out the same way as in the original algorithm, except that the equality elimination rule may not be used to eliminate numerical variables, as mentioned in Section 4.

In the branch where x_g or v_g is fixed to its original value, a new set of hard edits $\Psi_{q+1,H}$ is obtained from Ψ_{qH}, and a new set of soft edits $\Psi_{q+1,S}$ is obtained from Ψ_{qS}. In addition, we define $B^k_{q+1,S} := B^l_{qS}$ for the soft edit $\psi^k_{q+1,S}$ derived from ψ^l_{qS} by fixing a variable.

In the branch where x_g or v_g is eliminated from the edits, the new set of hard edits $\Psi_{q+1,H}$ consists of all edits from Ψ_{qH} that do not involve x_g or v_g, plus all implied edits that are obtained by eliminating x_g or v_g from a combination of only edits from Ψ_{qH}. The new set of soft edits $\Psi_{q+1,S}$ contains all other edits, i.e.

- all edits from Ψ_{qS} that do not involve x_g or v_g; we define $B^k_{q+1,S} := B^l_{qS}$ for the edit $\psi^k_{q+1,S}$ derived from ψ^l_{qS} this way;
- all implied edits that are obtained by eliminating x_g or v_g from a combination of only edits from Ψ_{qS}; we define $B^k_{q+1,S} := \bigcup_{t \in T} B^t_{qS}$ for the edit $\psi^k_{q+1,S}$ derived from ψ^t_{qS} $(t \in T)$ this way;
- all implied edits that are obtained by eliminating x_g or v_g from a combination of edits from Ψ_{qS} and Ψ_{qH}; we define $B^k_{q+1,S} := \bigcup_{t \in T_1} B^t_{qS}$ for the edit $\psi^k_{q+1,S}$ derived from ψ^t_{qS} $(t \in T_1)$ and ψ^t_{qH} $(t \in T_2)$ this way.

In summary: if an edit is generated only from hard edits, then the new edit is also a hard edit; if any soft edits are involved in its generation, then the new edit is a soft edit. Moreover, the index set B^k_{qS} contains the indices of all the original soft edits that were involved in the generation of ψ^k_{qS} at some point.

Having generated the new sets of edits $\Psi_{q+1,H}$ and $\Psi_{q+1,S}$, we fill in the original values of the variables that have not been treated yet, to check which of these edits are failed. In the old algorithm, there are two possibilities here: either none of the edits are failed and the current branch corresponds with a feasible solution, or at least one of the edits is failed and more variables need to be eliminated. In the new algorithm, three different situations may arise.

First of all, if at least one edit in $\Psi_{q+1,H}$ is failed, then the variables that have been eliminated so far cannot be imputed to satisfy the original hard edits. Hence, more variables need to be eliminated. In this case, we define $q := q + 1$ and continue the generation of branches from the current node.

A second possibility is that none of the edits in $\Psi_{q+1,H}$ or $\Psi_{q+1,S}$ are failed. This means that the variables that have been eliminated so far can be imputed to satisfy all the original edits, both hard and soft. Hence, we have found a feasible solution to the error localisation problem. The value of target function (16) equals $D = \lambda D_{FH}$, i.e. a constant factor times the sum of the confidence weights of the eliminated variables. If this value is smaller than or equal to the value of (16) for the best solution found so far, say $D_{\min}$, then the new solution is stored. Otherwise, it is discarded. Either way, it is not useful to continue the algorithm from the current node, because the value of D can only increase if more variables are eliminated. Hence, we return to the last previous branch that has not been completely searched yet and continue the algorithm from there.

The third and final possibility is that the edits in $\Psi_{q+1,H}$ are satisfied, but at least one edit in $\Psi_{q+1,S}$ is failed. In this case, the variables that have been eliminated so far can be imputed to satisfy the original hard edits, but not all the original soft edits. Hence, a feasible solution to the error localisation problem has been found, but the contribution of D_{soft} to D is non-zero.

According to Theorem 2 or Theorem 3, it is possible to satisfy all original soft edits, except those in a representing set B of the index sets $B^k_{q+1,S}$ for all failed edits in $\Psi_{q+1,S}$. Since this property is shared by all representing sets, we are free to choose B in such a way that D_{soft} is minimised, given the selection of variables to impute. If expression (17) is used for D_{soft}, then the optimal choice of B can be found by solving the following minimisation problem:

$$\begin{array}{l} \min \sum_{k=1}^{K_S} s_k z_k \\ \text{such that:} \\ \sum_{k \in B^l_{q+1,S}} z_k \geq 1, \text{ for all failed } \psi^l_{q+1,S} \in \Psi_{q+1,S} \\ z_k \in \{0,1\}, k = 1,\ldots,K_S \end{array} \tag{19}$$

This is a straightforward binary linear optimisation problem, which can be solved using standard algorithms. The solution consists of a vector $(z_1^*,\ldots,z_{K_S}^*)$ of zeros and ones. The associated optimal representing set[1] is $B^* = \left\{k \,|\, z_k^* = 1\right\}$ and the associated contribution of D_{soft} to D is precisely the minimal value of problem (19), say

$$D^*_{soft} = \sum_{k=1}^{K_S} s_k z_k^* = \sum_{k \in B^*} s_k.$$

As in the previous case, the value $D = \lambda D_{FH} + (1-\lambda) D^*_{soft}$ is compared to $D_{\min}$. If $D \leq D_{\min}$, then the current solution is stored, otherwise it is discarded. Either way, in this case it is meaningful to continue the algorithm from the current node, because eliminating more variables may lead to a lower value of the target function. This can happen, because a solution that imputes more variables typically fails less soft edits, and hence an increase in D_{FH} might be compensated by a decrease in D_{soft}. Therefore, we define $q := q+1$ and continue the generation of branches from the current node.

[1] It should be noted that problem (19) need not have a unique optimal solution. If there is more than one optimal solution, and hence more than one minimal representing set B, then we have to select one according to some additional criterion. A very simple criterion could be to always select the first solution that was found.

The correctness of this algorithm follows from the correctness of the original algorithm of De Waal and Quere (2003) and the theory of Section 4 and Section 5. The index sets B_q^k only have to be computed for the soft edits, because a subset of the variables is never considered as a feasible solution to the error localisation problem when at least one of the hard edits remains failed. This means that, in every application of Theorem 2 or Theorem 3, all implied edits in Ψ_{qH} must be contained in $\Psi_q^{(1)}$. Finally, we note that the new algorithm reduces to the original algorithm of De Waal and Quere (2003) in the case that no soft edits have been specified.

7 Two Examples

To illustrate the algorithm of Section 6, we apply it to two examples. The first example (Section 7.1) contains only numerical variables. The second example (Section 7.2) is somewhat larger and contains a mix of categorical and numerical variables. These examples have appeared previously in De Waal (2003) and Quere and De Waal (2000), respectively, but we have added a distinction between hard and soft edits.

7.1 An Example with Numerical Data

In a fictitious business survey, there are four numerical variables: *total turnover* (T), *profit* (P), *total costs* (C), and *number of employees* (N). The following hard edits and soft edits have been identified:

$$\begin{array}{lrcll}
\psi^1_{0H}: & T-C-P & = & 0 & \\
\psi^2_{0H}: & T & \geq & 0 & \\
\psi^3_{0H}: & C & \geq & 0 & \\
\psi^4_{0H}: & N & \geq & 0 & \\
\psi^5_{0H}: & 550N-T & \geq & 0 & \\
\psi^1_{0S}: & 0.5T-P & \geq & 0 & (B^1_{0S}=\{1\}) \\
\psi^2_{0S}: & P+0.1T & \geq & 0 & (B^2_{0S}=\{2\})
\end{array}$$

We want to edit the following record automatically:

$$(T^0, P^0, C^0, N^0) = (100, 40000, 60000, 5).$$

This record is inconsistent, because it fails the first hard edit. It also fails the first soft edit. The confidence weights of the variables are $(w_T, w_P, w_C, w_N) = (2, 1, 1, 3)$. We choose the failure weights of the two soft edits to be $s_1 = s_2 = 2$. Finally, we choose $\lambda = 1/2$ in (16).

Suppose that the variable P is selected first. In the branch where P is eliminated from the original edits, we obtain the following new set of edits:

$$\begin{array}{lrclll}
\psi^1_{1H}: & T & \geq & 0 & & (\psi^2_{0H}) \\
\psi^2_{1H}: & C & \geq & 0 & & (\psi^3_{0H}) \\
\psi^3_{1H}: & N & \geq & 0 & & (\psi^4_{0H}) \\
\psi^4_{1H}: & 550N-T & \geq & 0 & & (\psi^5_{0H}) \\
\psi^1_{1S}: & -0.5T+C & \geq & 0 & (B^1_{1S}=\{1\}) & (\psi^1_{0H}, \psi^1_{0S}) \\
\psi^2_{1S}: & 1.1T-C & \geq & 0 & (B^2_{1S}=\{2\}) & (\psi^1_{0H}, \psi^2_{0S}) \\
\psi^3_{1S}: & 0.6T & \geq & 0 & (B^3_{1S}=\{1,2\}) & (\psi^1_{0S}, \psi^2_{0S})
\end{array}$$

We have indicated in brackets from which of the previous edits each new edit is derived. For instance, ψ^2_{1S} is obtained by eliminating P from ψ^1_{0H} and ψ^2_{0S}. For the soft edits, the index sets B^k_{1S} are also displayed. The third soft edit ψ^3_{1S} is in fact equivalent to the first hard edit ψ^1_{1H}, which means that it can be discarded.

Upon substituting the original values $(T^0, C^0, N^0) = (100, 60000, 5)$ into the current edits, it is seen that all edits are satisfied except for ψ^2_{1S}. Since all hard edits are satisfied, identifying only the original value of P as erroneous is a feasible solution to the error localisation problem. Moreover, since $B = \{2\}$ is a representing set of B^2_{1S}, it is possible to impute a value for P which satisfies all the original edits except for ψ^2_{0S}. This is in fact the minimal representing set according to problem (19). Hence, the value of target function (16) for this solution equals:

$$D = \frac{1}{2}D_{FH} + \frac{1}{2}D_{soft} = \frac{w_P}{2} + \frac{s_2}{2} = \frac{3}{2}.$$

Possibly, the current solution may be improved by eliminating another variable, say C, from the current set of edits. This yields:

$$\begin{array}{lrcll}
\psi^1_{2H}: & T & \geq & 0 & & (\psi^1_{1H}) \\
\psi^2_{2H}: & N & \geq & 0 & & (\psi^3_{1H}) \\
\psi^3_{2H}: & 550N - T & \geq & 0 & & (\psi^4_{1H}) \\
\psi^1_{2S}: & 1.1T & \geq & 0 & (B^1_{2S} = \{2\}) & (\psi^2_{1H}, \psi^2_{1S}) \\
\psi^2_{2S}: & 0.6T & \geq & 0 & (B^2_{2S} = \{1, 2\}) & (\psi^1_{1S}, \psi^2_{1S})
\end{array}$$

The two new soft edits are both redundant, because they are equivalent to hard edit ψ^1_{2H}. In fact, the remaining original values $(T^0, N^0) = (100, 5)$ satisfy all the current edits. This means that P and C can be imputed to satisfy all the original edits, both hard and soft. The value of target function (16) for this solution equals:

$$D = \frac{1}{2}D_{FH} = \frac{w_P + w_C}{2} = 1.$$

Thus, the new solution improves on the previous solution. Moreover, this solution cannot be improved by eliminating more variables in the current branch of the binary tree, because there are no failed edits remaining.

So far, we have only explored the branch where P is eliminated from the edits. If the algorithm is continued by exploring the rest of the binary tree, it turns out that the best solution found so far (impute P and C) is also the optimal solution. A possible way to impute the record consistently is:

$$(T, P, C, N) = (100, 40, 60, 5).$$

This solution has the nice interpretation that the original values of *profit* and *total costs* were overstated by a factor of $1,000$.

It is of interest to note that, if only the hard edits are used in this example, the first solution found above (impute only P) is the optimal solution to the error localisation problem. In that case, there is only one way to obtain a consistent record:

$$(T, P, C, N) = (100, -59900, 60000, 5).$$

This illustrates that, in this example at least, soft edits are important for finding imputations that are not just consistent with the hard edits, but also plausible.

7.2 An Example with Mixed Data

In the second example, records consist of four categorical variables and three numerical variables $(v_1, v_2, v_3, v_4, x_1, x_2, x_3)$. The domains of the categorical variables are $D_1 = D_3 = \{1,2\}$ and $D_2 = D_4 = \{1,2,3\}$. The following hard and soft edits have been identified:

$$
\begin{array}{lll}
\psi_{0H}^1: & (v_1, v_4) \in \{1\} \times \{1,3\} \Rightarrow \emptyset & \\
\psi_{0H}^2: & (v_2, v_3) \in \{1\} \times \{1\} \Rightarrow \emptyset & \\
\psi_{0H}^3: & (v_1, v_2, v_4) \in \{2\} \times \{1,3\} \times \{1,3\} \Rightarrow \emptyset & \\
\psi_{0H}^4: & v_2 \in \{1,3\} \Rightarrow x_2 = 0 & \\
\psi_{0H}^5: & v_2 \in \{1,3\} \Rightarrow 1250x_1 - x_3 = 0 & \\
\psi_{0H}^6: & (v_2, v_3) \in \{2\} \times \{2\} \Rightarrow 1250x_1 + 12x_2 - x_3 = 0 & \\
\psi_{0H}^7: & (v_2, v_3) \in \{2\} \times \{1\} \Rightarrow 1250x_1 + 12x_2 - x_3 = -1250 & \\
\psi_{0S}^1: & 1250x_1 \geq 15000 & (B_{0S}^1 = \{1\}) \\
\psi_{0S}^2: & v_2 \in \{2\} \Rightarrow 12x_2 \geq 15000 & (B_{0S}^2 = \{2\}) \\
\psi_{0S}^3: & v_2 \in \{2\} \Rightarrow -875x_1 + 12x_2 \geq 0 & (B_{0S}^3 = \{3\}) \\
\psi_{0S}^4: & v_2 \in \{2\} \Rightarrow 1250x_1 - 8.4x_2 \geq 0 & (B_{0S}^4 = \{4\})
\end{array}
$$

For the sake of brevity, we write the edits in a slightly different notation from the rest of the paper. Edit ψ_{0H}^5, for instance, would look as follows in the notation from Section 2.1:

$$
\begin{array}{rll}
\psi_{0H}^5: & \text{IF} & (v_1, v_2, v_3, v_4) \in D_1 \times \{1,3\} \times D_3 \times D_4 \\
& \text{THEN} & (x_1, x_2, x_3) \in \{\vec{x} \mid 1250x_1 - x_3 = 0\}.
\end{array}
$$

Consider the following unedited record:

$$(v_1^0, v_2^0, v_3^0, v_4^0, x_1^0, x_2^0, x_3^0) = (2, 2, 1, 2, 10, 0, 12000).$$

This record fails hard edit ψ_{0H}^7, so it must contain an error. In addition, the record fails the soft edits ψ_{0S}^1, ψ_{0S}^2 and ψ_{0S}^3. In this example, we choose all confidence weights and all failure weights equal to 1, and we choose $\lambda = 1/2$. As in the previous example, we shall only explore one branch of the binary tree.

We begin by treating the numerical variables and decide to first eliminate x_1 from the edits. This yields the following implied edits:

$$
\begin{array}{llll}
\psi_{1H}^1: & (v_1, v_4) \in \{1\} \times \{1,3\} \Rightarrow \emptyset & & (\psi_{0H}^1) \\
\psi_{1H}^2: & (v_2, v_3) \in \{1\} \times \{1\} \Rightarrow \emptyset & & (\psi_{0H}^2) \\
\psi_{1H}^3: & (v_1, v_2, v_4) \in \{2\} \times \{1,3\} \times \{1,3\} \Rightarrow \emptyset & & (\psi_{0H}^3) \\
\psi_{1H}^4: & v_2 \in \{1,3\} \Rightarrow x_2 = 0 & & (\psi_{0H}^4) \\
\psi_{1S}^1: & v_2 \in \{1,3\} \Rightarrow x_3 \geq 15000 & (B_{1S}^1 = \{1\}) & (\psi_{0H}^5, \psi_{0S}^1) \\
\psi_{1S}^2: & (v_2, v_3) \in \{2\} \times \{2\} \Rightarrow -12x_2 + x_3 \geq 15000 & (B_{1S}^2 = \{1\}) & (\psi_{0H}^6, \psi_{0S}^1) \\
\psi_{1S}^3: & (v_2, v_3) \in \{2\} \times \{1\} \Rightarrow -12x_2 + x_3 \geq 16250 & (B_{1S}^3 = \{1\}) & (\psi_{0H}^7, \psi_{0S}^1) \\
\psi_{1S}^4: & v_2 \in \{2\} \Rightarrow 12x_2 \geq 10500 & (B_{1S}^4 = \{1,3\}) & (\psi_{0S}^1, \psi_{0S}^3) \\
\psi_{1S}^5: & (v_2, v_3) \in \{2\} \times \{2\} \Rightarrow 20.4x_2 - 0.7x_3 \geq 0 & (B_{1S}^5 = \{3\}) & (\psi_{0H}^6, \psi_{0S}^3) \\
\psi_{1S}^6: & (v_2, v_3) \in \{2\} \times \{1\} \Rightarrow 20.4x_2 - 0.7x_3 \geq -875 & (B_{1S}^6 = \{3\}) & (\psi_{0H}^7, \psi_{0S}^3)
\end{array}
$$

$$\begin{array}{lllll}
\psi^{7}_{1S}: & v_2 \in \{2\} \Rightarrow x_2 \geq 0 & & (B^{7}_{1S} = \{3,4\}) & (\psi^{3}_{0S}, \psi^{4}_{0S}) \\
\psi^{8}_{1S}: & (v_2,v_3) \in \{2\} \times \{2\} \Rightarrow -20.4x_2 + x_3 \geq 0 & & (B^{8}_{1S} = \{4\}) & (\psi^{6}_{0H}, \psi^{4}_{0S}) \\
\psi^{9}_{1S}: & (v_2,v_3) \in \{2\} \times \{1\} \Rightarrow -20.4x_2 + x_3 \geq 1250 & & (B^{9}_{1S} = \{4\}) & (\psi^{7}_{0H}, \psi^{4}_{0S}) \\
\psi^{10}_{1S}: & v_2 \in \{2\} \Rightarrow 12x_2 \geq 15000 & & (B^{10}_{1S} = \{2\}) & (\psi^{2}_{0S})
\end{array}$$

If we fill in the original values of the other variables in these edits, it turns out that only ψ^{3}_{1S}, ψ^{4}_{1S}, ψ^{6}_{1S}, and ψ^{10}_{1S} are failed. Hence, all hard edits can be satisfied by only imputing x_1, and we have found a feasible solution to the error localisation problem. The minimal representing set of B^{3}_{1S}, B^{4}_{1S}, B^{6}_{1S}, and B^{10}_{1S} is $B = \{1,2,3\}$. This shows that if we impute x_1, three of the original soft edits must be failed. Thus, the value of target function (16) is

$$D = \frac{1}{2}D_{FH} + \frac{1}{2}D_{soft} = \frac{1}{2} + \frac{3}{2} = 2$$

for this solution.

Since there are still failed soft edits, we continue to explore the current branch of the binary tree. Suppose that we decide to fix both x_2 and x_3 to their respective original values, $x^{0}_{2} = 0$ and $x^{0}_{3} = 12000$. We obtain the following set of edits:

$$\begin{array}{lll}
\psi^{1}_{3H}: & (v_1,v_4) \in \{1\} \times \{1,3\} \Rightarrow \emptyset & \\
\psi^{2}_{3H}: & (v_2,v_3) \in \{1\} \times \{1\} \Rightarrow \emptyset & \\
\psi^{3}_{3H}: & (v_1,v_2,v_4) \in \{2\} \times \{1,3\} \times \{1,3\} \Rightarrow \emptyset & \\
\psi^{1}_{3S}: & v_2 \in \{1,3\} \Rightarrow \emptyset & (B^{1}_{3S} = \{1\}) \\
\psi^{2}_{3S}: & (v_2,v_3) \in \{2\} \times \{2\} \Rightarrow \emptyset & (B^{2}_{3S} = \{1\}) \\
\psi^{3}_{3S}: & (v_2,v_3) \in \{2\} \times \{1\} \Rightarrow \emptyset & (B^{3}_{3S} = \{1\}) \\
\psi^{4}_{3S}: & v_2 \in \{2\} \Rightarrow \emptyset & (B^{4}_{3S} = \{1,3\}) \\
\psi^{5}_{3S}: & (v_2,v_3) \in \{2\} \times \{2\} \Rightarrow \emptyset & (B^{5}_{3S} = \{3\}) \\
\psi^{6}_{3S}: & (v_2,v_3) \in \{2\} \times \{1\} \Rightarrow \emptyset & (B^{6}_{3S} = \{3\}) \\
\psi^{7}_{3S}: & v_2 \in \{2\} \Rightarrow \emptyset & (B^{7}_{3S} = \{2\})
\end{array}$$

Since all numerical variables have now been treated, we are left with a set of purely categorical edits. Upon inspecting these edits, it can be seen that the variables v_1 and v_4 are only involved in hard edits. Since it is already possible to satisfy all hard edits by only imputing x_1, it is not useful to consider any solutions where these variables are imputed along with x_1. Therefore we decide immediately to fix these variables to their original values $v^{0}_{1} = 2$ and $v^{0}_{4} = 2$. This leads to the following set of edits:

$$\begin{array}{lll}
\psi^{1}_{5H}: & (v_2,v_3) \in \{1\} \times \{1\} \Rightarrow \emptyset & \\
\psi^{1}_{5S}: & v_2 \in \{1,3\} \Rightarrow \emptyset & (B^{1}_{5S} = \{1\}) \\
\psi^{2}_{5S}: & (v_2,v_3) \in \{2\} \times \{2\} \Rightarrow \emptyset & (B^{2}_{5S} = \{1\}) \\
\psi^{3}_{5S}: & (v_2,v_3) \in \{2\} \times \{1\} \Rightarrow \emptyset & (B^{3}_{5S} = \{1\}) \\
\psi^{4}_{5S}: & v_2 \in \{2\} \Rightarrow \emptyset & (B^{4}_{5S} = \{1,3\}) \\
\psi^{5}_{5S}: & (v_2,v_3) \in \{2\} \times \{2\} \Rightarrow \emptyset & (B^{5}_{5S} = \{3\}) \\
\psi^{6}_{5S}: & (v_2,v_3) \in \{2\} \times \{1\} \Rightarrow \emptyset & (B^{6}_{5S} = \{3\}) \\
\psi^{7}_{5S}: & v_2 \in \{2\} \Rightarrow \emptyset & (B^{7}_{5S} = \{2\})
\end{array}$$

Suppose that we decide to eliminate v_2. We find:

$$\begin{array}{llll}
\psi_{6S}^{1}: & v_3 \in \{2\} \Rightarrow \emptyset & (B_{6S}^{1} = \{1\}) & (\psi_{5S}^{1}, \psi_{5S}^{2}) \\
\psi_{6S}^{2}: & v_3 \in \{1\} \Rightarrow \emptyset & (B_{6S}^{2} = \{1\}) & (\psi_{5S}^{1}, \psi_{5S}^{3}) \\
\psi_{6S}^{3}: & \emptyset & (B_{6S}^{3} = \{1,3\}) & (\psi_{5S}^{1}, \psi_{5S}^{4}) \\
\psi_{6S}^{4}: & v_3 \in \{2\} \Rightarrow \emptyset & (B_{6S}^{4} = \{1,3\}) & (\psi_{5S}^{1}, \psi_{5S}^{5}) \\
\psi_{6S}^{5}: & v_3 \in \{1\} \Rightarrow \emptyset & (B_{6S}^{5} = \{1,3\}) & (\psi_{5S}^{1}, \psi_{5S}^{6}) \\
\psi_{6S}^{6}: & \emptyset & (B_{6S}^{6} = \{1,2\}) & (\psi_{5S}^{1}, \psi_{5S}^{7})
\end{array}$$

It is seen that the original value $v_3^0 = 1$ fails the edits ψ_{6S}^{2}, ψ_{6S}^{3}, ψ_{6S}^{5}, and ψ_{6S}^{6}. Hence, the current solution (imputing only x_1 and v_2) does not lead to a record that satisfies all soft edits. Since $B = \{1\}$ is the minimal representing set of B_{6S}^{2}, B_{6S}^{3}, B_{6S}^{5}, and B_{6S}^{6}, we conclude that it is possible to find values for x_1 and v_2 that satisfy all original edits except for ψ_{0S}^{1}. The value of target function (16) for this solution is:

$$D = \frac{1}{2}D_{FH} + \frac{1}{2}D_{soft} = \frac{2}{2} + \frac{1}{2} = \frac{3}{2},$$

which is an improvement on the previous solution (imputing only x_1). Moreover, it is clear that the current solution cannot be improved by also imputing v_3, because the edits ψ_{6S}^{3} and ψ_{6S}^{6} are failed by definition.

8 (Quasi-)Consistent Imputation

A solution to the error localisation problem for a record simply consists of a list of variables to impute. As we mentioned in the introduction, finding this list of variables to impute is only half of the task of automatic editing. The other half is to find actual values to impute that satisfy all (hard) edits, i.e. solving the consistent imputation problem. Provided that the error localisation problem has been solved correctly, suitable values are guaranteed to exist, but they still need to be found.

In some cases, it is possible to find an appropriate imputation model that produces values that satisfy all edits directly (Tempelman, 2007; De Waal et al., 2011). However, this direct approach can easily become too complex in practical applications. NSIs therefore often apply a two-step approach to obtain consistent imputations. First, the erroneous variables are imputed by a basic imputation method, such as regression or hot deck imputation, which takes certain statistical properties of the data set into account, but not the edits. This yields an imputed record $(v_1, \ldots, v_m, x_1, \ldots, x_p)$ which may be inconsistent. Next, the initial imputations are minimally adjusted to satisfy the edits, according to some distance function. This yields an adjusted record $(\tilde{v}_1, \ldots, \tilde{v}_m, \tilde{x}_1, \ldots, \tilde{x}_p)$ which is consistent. It should be noted that the adjusted record only deviates from the original, unedited record for variables that have been identified as erroneous, because the values of the other variables are considered fixed during both the imputation and the adjustment step.

For the adjustment step, De Waal (2003) suggested minimising the following distance function with mixed categorical and numerical data:

$$\sum_{j=1}^{m} w_j^C \delta_j(v_j, \tilde{v}_j) + \sum_{j=1}^{p} w_j^N |x_j - \tilde{x}_j|, \tag{20}$$

where $\delta_j(v_j, \tilde{v}_j)$ is a metric for the j^{th} categorical variable, and w_j^C and w_j^N denote the confidence weights, as before. For many categorical variables, especially nominal variables, one would typically choose the simple metric given by $\delta_j(v_j, \tilde{v}_j) = 0$ if $v_j = \tilde{v}_j$, and $\delta_j(v_j, \tilde{v}_j) = 1$ otherwise.

De Waal (2003) observed that, for mixed data, solving the above-mentioned minimisation problem to optimality in practice may be a formidable task. He suggested to use a heuristic procedure instead, by applying a simplified version of the branch-and-bound algorithm of De Waal and Quere (2003) for error localisation. This heuristic algorithm reconstructs only one branch of the binary tree, namely the branch that corresponds with the optimal solution to the error localisation problem found previously. In this branch, the variables to be imputed are eliminated from the edits, and the other variables are fixed to their original values.

We denote the set of implied edits after eliminating the q^{th} variable by Ψ_q. Suppose that Q variables have to be imputed. The edits in Ψ_{Q-1} are univariate edits that must be satisfied by the last variable to have been eliminated. If this variable is categorical, then we adjust the originally imputed value v_j to a new value $\tilde{v}_j$ that satisfies the univariate edits *and* minimises $\delta_j(v_j, \tilde{v}_j)$. This adjusted value is plugged into the edits from Ψ_{Q-2}, thus producing a set of univariate edits for the penultimate variable to have been eliminated. We continue in this manner until all categorical variables have been imputed consistently. Assuming that the error localisation problem has been solved correctly, Theorem 1 guarantees that we can find a suitable value to impute at each stage of the algorithm.

Once all categorical variables have been given adjusted values, we are left with the problem of finding adjusted values for the numerical variables that minimise

$$\sum_{j=1}^{p} w_j^N |x_j - \tilde{x}_j|$$

such that the numerical edits are all satisfied. This problem can be formulated as a linear programming problem and solved by the well-known simplex method; see De Waal (2003). Again, a repeated application of Theorem 1 guarantees that the linear programming problem is feasible.

We remark that, for mixed data at least, the algorithm is indeed a heuristic algorithm: the size of the adjustment is minimised for each categorical variable separately, but this does not necessarily lead to a set of adjusted values that minimises (20). Nevertheless, according to De Waal (2003), the heuristic method is likely to give acceptable results in practice.

Under the new error localisation problem with hard and soft edits, it is possible that the optimal solution cannot be imputed consistently with respect to all soft edits. The algorithm of De Waal (2003) can still be used to solve the consistent imputation problem in this case, provided that we remove the soft edits that are failed by the optimal solution. This is more or less trivial, since a list of failed soft edits is automatically provided by the new error localisation algorithm in the form of the minimal representing set B associated with the optimal solution. We illustrate the procedure by revisiting the example from Section 7.2.

Example
In Section 7.2, it was found that imputing x_1 and v_2 is a solution to the error localisation problem, and that these variables can be imputed such that only soft edit ψ^1_{0S} remains failed. We shall now use the algorithm of De Waal (2003) to find suitable imputations. In general, we would start by imputing initial values and adjust these to satisfy the edits. However, the first step can be skipped in this example, because it turns out that the imputable values are uniquely determined by the edits.

We start by plugging the original values of the variables that will not be imputed (i.e. $v^0_1 = 2$, $v^0_3 = 1$, $v^0_4 = 2$, $x^0_2 = 0$, and $x^0_3 = 12000$) into the edits ψ^k_{0H} and ψ^k_{0S}, leaving out edit ψ^1_{0S}. This gives a reduced set of edits for x_1 and v_2, which we write in the notation of Section 2.1:

$$\text{IF } v_2 \in \{1\} \text{ THEN } \emptyset; \tag{21}$$

$$\text{IF } v_2 \in \{1,3\} \text{ THEN } 1250x_1 = 12000; \tag{22}$$

$$\text{IF } v_2 \in \{2\} \text{ THEN } 1250x_1 = 10750; \tag{23}$$

$$\text{IF } v_2 \in \{2\} \text{ THEN } \emptyset; \tag{24}$$

$$\text{IF } v_2 \in \{2\} \text{ THEN } -875x_1 \geq 0; \tag{25}$$

$$\text{IF } v_2 \in \{2\} \text{ THEN } 1250x_1 \geq 0. \tag{26}$$

Eliminating the only remaining numerical variable x_1 yields a set of univariate categorical edits for v_2:

$$\text{IF } v_2 \in \{1\} \text{ THEN } \emptyset; \tag{27}$$

$$\text{IF } v_2 \in \{2\} \text{ THEN } \emptyset. \tag{28}$$

It is seen that the only value from the domain of v_2 to satisfy both (27) and (28) is $\tilde{v}_2 = 3$, and hence we impute this value. Plugging $\tilde{v}_2 = 3$ into edits (21)-(26) yields one univariate numerical edit for x_1:

$$1250x_1 = 12000. \tag{29}$$

In order to satisfy (29), we have to impute the value $\tilde{x}_1 = \frac{48}{5} = 9.6$. Hence, the algorithm for (quasi-)consistent imputation produces the following adjusted record:

$$(\tilde{v}_1, \tilde{v}_2, \tilde{v}_3, \tilde{v}_4, \tilde{x}_1, \tilde{x}_2, \tilde{x}_3) = (2, 3, 1, 2, \frac{48}{5}, 0, 12000).$$

The reader may verify that this record indeed satisfies all original edits from Section 7.2, except for ψ^1_{0S}. □

9 Application[2]

In order to test the new error localisation algorithm in practice, a prototype implementation was written using the R programming language. This prototype draws heavily

[2] The empirical results in this section were collected by Sevinç Göksen as part of her Master's thesis research. A more comprehensive description of this research will be given in a separate report.

on the existing error localisation functionality in R that was made available in the editrules package (De Jonge and Van der Loo, 2011; Van der Loo and De Jonge, 2011). In particular, the editrules package contains an implementation of the original branch-and-bound algorithm of De Waal and Quere (2003).

To test the prototype, an artificial data set was constructed by selecting twelve numerical variables $(x_1, \ldots, x_{12})$ from the structural business statistics questionnaire of 2007 for the wholesale sector. We selected all records pertaining to medium-sized businesses (with 10-100 employees) that had been edited manually during regular production, and divided these into two data sets of 728 records each. Both of the original data sets were considered error-free. We introduced a substantial number of random errors into one of the data sets by applying the following procedure:

- in 4% of the original non-zero values, two digits were interchanged;
- in 4% of the original non-zero values, a random digit was added;
- in 4% of the original non-zero values, a random digit was omitted;
- in 4% of the original non-zero values, a random digit was replaced by another digit;
- 4% of the original non-zero values were multiplied by 25;
- 4% of the original non-zero values were divided by 25 and rounded to the nearest integer;
- 6% of the original non-zero values were replaced by zero;
- 5% of the original zero values were replaced by random integers from $[1, 1000]$;
- 10% of the original values of x_{11} and x_{12} were multiplied by -1.

This procedure was carried out in such a way that at most one change could occur in each value. The second data set was left error-free and used as reference data (see below).

Table 1 shows the hard and soft edits that were applied to the test data. The hard edits were copied from the regular production system. The soft edits were identified by examining a number of univariate and bivariate distributions in the error-free reference data. We did not use a particularly rigourous approach to choose these soft edits, because this application was only intended as a first exploratory test of the new algorithm.

The error localisation algorithm was applied to the data set with artificial errors using several different set-ups. Throughtout, all confidence weights w_j were chosen equal to one, and the parameter λ in (16) was chosen equal to $1/2$. We considered the following approaches:

A. The first test used only the hard edits from Table 1.

B. The second test used all edits from Table 1, with the interpretation of all edits as hard edits.

Table 1: The edits that were used in the test application.

hard edits:	$x_1 + x_2 = x_3$
	$x_2 = x_4$
	$x_5 + x_6 + x_7 = x_8$
	$x_3 + x_8 = x_9$
	$x_9 - x_{10} = x_{11}$
	$x_i \geq 0$ $(i = 1, \ldots, 10$ and $i = 12)$
soft edits:	$x_2 \geq 0.5x_3$
	$x_3 \geq 0.9x_9$
	$x_5 + x_6 \geq x_7$
	$x_9 \geq 50x_{12}$
	$x_9 \leq 5000x_{12}$
	$x_{11} \leq 0.4x_9$
	$x_{11} \geq -0.1x_9$
	$x_{12} \geq 1$
	$x_{12} \geq 5$
	$x_{12} \leq 100$

C. The third test used all edits from Table 1, with a distinction between hard and soft edits. Each soft edit received the same fixed failure weight $s_k = 1$.

D. The fourth test was similar to the third test, but with fixed failure weights that differed between soft edits. We calculated the fraction of records in the reference data set that satisfied each soft edit, and used these fractions for s_k. Thus, a soft edit received a lower failure weight if it was failed more often in the reference data set, and vice versa. The rationale behind this is that all soft edits failures occurring in the reference data were caused by unusual, but correct combinations of values, since the reference data were considered error-free. By associating low weights to soft edits that are often failed in the reference data, we ensure that these edits may also be failed more easily in the edited version of the test data.

Since the distribution of errors in our test data set was completely known, we could directly evaluate the performance of each automatic error localisation approach. We used several quality indicators for this. Consider the following 2×2 contingency table:

		detected:	
		error	no error
true:	error	TP	FN
	no error	FP	TN

In this table, TP, FN, FP, and TN denote, respectively:

- the number of values that were correctly identified as errors (true positives);
- the number of values that were incorrectly identified as non-errors (false negatives);

Table 2: Results of automatic error localisation for the artificial data.

	quality indicators			
approach	α	β	γ	δ
A	0.364	0.047	0.115	40%
B	0.232	0.131	0.153	37%
C	0.227	0.060	0.096	47%
D	0.253	0.037	0.083	52%

- the number of values that were incorrectly identified as errors (false positives);
- the number of values that were correctly identified as non-errors (true negatives).

Such a contingency table was constructed for the outcome of each of the editing approaches A, B, C, and D.

The first quality indicator measures the proportion of true errors that were missed by the algorithm:

$$\alpha = \frac{FN}{TP+FN}.$$

The second quality indicator measures the proportion of correct values that were mistaken for errors by the algorithm:

$$\beta = \frac{FP}{FP+TN}.$$

The third quality indicator measures the overall proportion of wrong decisions made by the algorithm:

$$\gamma = \frac{FN+FP}{TP+FN+FP+TN}.$$

These three indicators evaluate the performance of the algorithm with respect to identifying individual values as correct or erroneous. They have been used in previous evaluation studies; see, for instance, Pannekoek and De Waal (2005). To evaluate the performance of the algorithm from a slightly different angle, we also calculated the percentage of records for which the algorithm detected exactly the right combination of erroneous values, and this indicator is denoted by δ. A good editing approach should have low scores on the α, β, and γ measures, but a high score on the δ measure.

Table 2 shows the values of the quality indicators for the above-mentioned editing approaches A, B, C, and D. It can be seen that approach B is outperformed by the other approaches on all measures, except for the proportion of missed errors. Thus, using the soft edits as if they were hard edits does not work well for this data set; in fact, better results are achieved by approach A, which does not use the soft edits at all. It can also be seen that approaches C and D, which use the new algorithm to take the soft edits into account, yield better results than approaches A and B, which use the old algorithm. Overall, approach D appears to achieve the best results in this experiment. Compared with approach A, approach D in fact correctly identifies more errors *and* more correct values.

It should be noted that, under the old definition of the error localisation problem, approaches A and B represent the two extreme options for using soft edits that are available: either not using them, or using them as hard edits. As a compromise between these options, one could also decide to use only a subset of the soft edits as hard edits and discard the others. We did not test this approach during the experiment. One might expect that it leads to scores on the α, β, γ, and δ measures in between those of approaches A and B.

10 Sizes of Soft Edit Failures

We mentioned in Section 3 that it is intuitively appealing to take the sizes of soft edit failures into account in the error localisation problem. An obvious way to achieve this is to choose an expression for D_{soft} that depends on the amount by which each soft edit is failed, where larger soft edit failures yield higher values of D_{soft}. Unfortunately, the error localisation problem then becomes more difficult to solve, because the sizes of the soft edit failures depend on the choice of variables to impute. This is easily seen in the example from Section 7.1. In this example, the first soft edit was failed by the original record with a left-hand-side of -39950, while the second soft edit was satisfied. Imputing the value -59900 for P led to a record that satisfied the first soft edit, but failed the second soft edit with a left-hand-side of -59890.

As this example illustrates, if D_{soft} depends on the sizes of the soft edit failures, then it actually depends on the imputed values for the variables that are selected for imputation. However, the imputed values are unknown during an execution of the error localisation algorithm from Section 6. We could, of course, first determine all feasible solutions to the error localisation problem, then find the corresponding imputed values and use these to compute D_{soft} for all solutions, and finally select the best solution. This would require much more work than the original algorithm (for one thing, we would lose the ‘bound’ part of the branch-and-bound algorithm), and it seems doubtful whether this approach would be computationally feasible in practice.

We have worked out an alternative approach, which does not explicitly impute all feasible solutions, but rather infers lower bounds on the sizes of the soft edit failures from the implicit edits. The interested reader is referred to Appendix B for details. Appendix B also mentions possible ways to transform the edit failure sizes into an expression for D_{soft}.

Sevinç Göksen pointed out that, in theory, it is also possible to associate higher failure weights with larger soft edit failures – at least to some extent – in our original formulation of the error localisation problem. To illustrate this, we consider the following soft edits from Table 1:

$$x_{12} \geq 1;$$

$$x_{12} \geq 5.$$

We suppose that expression (17) is used for D_{soft} and we choose all fixed failure weights equal to 1 for convenience. It is easy to see that any record that fails the first edit (i.e. any record with $x_{12} < 1$) also fails the second edit. Hence, a record with $x_{12} < 1$ receives a contribution of 2 to D_{soft}, whereas a record with $1 \leq x_{12} < 5$, which only fails the second edit, receives a contribution of 1. This is in line with the intuitive interpretation of this pair of edits, namely that a value $1 \leq x_{12} < 5$ is considered suspicious, while a value $x_{12} < 1$ is considered *highly* suspicious.

More generally, if we consider a numerical soft edit of the form

$$a_{k1}x_1 + \cdots + a_{kp}x_p + b_k \geq 0,$$

then we may replace this edit by a system of analogous soft edits:

$$\begin{array}{rcl} a_{k1}^{(1)}x_1 + \cdots + a_{kp}^{(1)}x_p + b_k^{(1)} & \geq & 0, \\ & \vdots & \\ a_{k1}^{(R)}x_1 + \cdots + a_{kp}^{(R)}x_p + b_k^{(R)} & \geq & 0, \end{array}$$

with $a_{kj}^{(r)} = a_{kj}$ for all $j = 1, \ldots, p$ and $r = 1, \ldots, R$, and where the constant terms are chosen such that $b_k = b_k^{(1)} < b_k^{(2)} < \cdots < b_k^{(R)}$. Similarly to the above example, any record that fails the r^{th} edit from this system automatically also fails the first $r-1$ edits, for $r = 2, \ldots, R$. Thus, a larger edit failure implicitly receives a higher failure weight, simply because it fails more soft edits. In practice, however, this approach should be used sparingly, because it can easily make the number of implied edits prohibitively large.

11 Conclusion

In this paper, we proposed a new formulation of the error localisation problem which can take the distinction between hard and soft edits into account. In addition, we showed that a modified version of the branch-and-bound algorithm of De Waal and Quere (2003) can be used to solve this new error localisation problem. Since subject-matter experts also use the conceptual difference between hard and soft edits during manual editing, it seems probable that the new error localisation algorithm can be used to increase the quality of automatic editing. This is also indicated by the first empirical results reported in Section 9, although it should be stressed that these results were obtained with data containing synthetic errors. Applications are currently being investigated of the new error localisation algorithm to realistic data.

It remains an open problem how the costs of soft edit failures may best be modelled, i.e. how the term D_{soft} in (16) should be defined. The different results with approaches C and D in Section 9 demonstrate that the quality of automatic error localisation may be improved by a suitable choice of failure weights. It will be interesting to see to what extent the quality of automatic editing may be improved further by experimenting with:

- different choices of failure weights s_k;
- different choices of confidence weights w_j;
- different choices of the balancing parameter λ in (16);
- other forms of D_{soft} than expression (17), including forms that depend on the sizes of the soft edit failures.

With respect to the final point, it should be noted that the algorithm from Section 6 may be used to solve the error localisation problem for all choices of D_{soft} that can be expressed as functions of only $z_1, \ldots, z_{K_S}$. One simply uses the appropriate expression for D_{soft} as the target function in problem (19). In practice, problem (19) is easier to solve if D_{soft} is a linear function of $z_1, \ldots, z_{K_S}$, but non-linear functions are also allowed. On the other hand, if D_{soft} is a function of the sizes of the soft edit failures, then we have to resort to a more complex approach, as outlined in Appendix B. It remains to be seen whether this more complex approach also leads to better results in practice.

Acknowledgements

The author would like to thank Jeroen Pannekoek and Ton de Waal for many helpful suggestions, Edwin de Jonge and Mark van der Loo for programming the branch-and-bound algorithm in R, and Sevinç Göksen for collecting experimental results.

References

Bikker, R., Daalmans, J. and Mushkudiani, N. (2010), 'A Multivariate Denton Method for Benchmarking Large Data Sets'. Discussion Paper 10002, Statistics Netherlands, The Hague.

Černikov, S. N. (1963), 'The Solution of Linear Programming Problem by Elimination of Unknowns'. In *Soviet Mathematics DOKLADY 2*.

De Jonge, E. and Van der Loo, M. (2011), 'Manipulation of Linear Edits and Error Localization with the Editrules Package'. Discussion Paper 201120, Statistics Netherlands, The Hague.

De Waal, T. (2003), 'Processing of Erroneous and Unsafe Data'. PhD Thesis, Erasmus University, Rotterdam.

De Waal, T. (2005), 'SLICE 1.5: a Software Framework for Automatic Edit and Imputation'. Working Paper, UN/ECE Work Session on Statistical Data Editing, Ottawa.

De Waal, T. and Coutinho, W. (2005), 'Automatic Editing for Business Surveys: an Assessment for Selected Algorithms', *International Statistical Review* **73**, pp. 73–102.

De Waal, T., Pannekoek, J. and Scholtus, S. (2011), *Handbook of Statistical Data Editing and Imputation*, John Wiley & Sons, Hoboken, New Jersey.

De Waal, T. and Quere, R. (2003), 'A Fast and Simple Algorithm for Automatic Editing of Mixed Data', *Journal of Official Statistics* **19**, pp. 383–402.

Fellegi, I. P. and Holt, D. (1976), 'A Systematic Approach to Automatic Edit and Imputation', *Journal of the American Statistical Association* **71**, pp. 17–35.

Hedlin, D. (2003), 'Score Functions to Reduce Business Survey Editing at the U.K. Office for National Statistics', *Journal of Official Statistics* **19**, pp. 177–199.

Kohler, D. A. (1967), ‘Projections of Convex Polyhedral Sets’. Operational Research Center Report ORC 67-29, University of California, Berkeley.

Little, R. J. A. and Smith, P. J. (1987), ‘Editing and Imputation of Quantitative Survey Data’, *Journal of the American Statistical Association* **82**, pp. 58–68.

Magnus, J. R., Van Tongeren, J. W. and De Vos, A. F. (2000), ‘National Accounts Estimation Using Indicator Ratios’, *Review of Income and Wealth* **46**, pp. 329–350.

Mirsky, L. (1971), *Transversal Theory*, Academic Press, Inc., New York.

Pannekoek, J. and De Waal, T. (2005), ‘Automatic Edit and Imputation for Business Surveys: the Dutch Contribution to the EUREDIT Project’, *Journal of Official Statistics* **21**, pp. 257–286.

Quere, R. and De Waal, T. (2000), ‘Error Localization in Mixed Data Sets’. Internal Report (BPA-nr. 2285-00-RSM), Statistics Netherlands, Voorburg.

Tempelman, D. C. G. (2007), ‘Imputation of Restricted Data’. PhD Thesis, University of Groningen.

Van der Loo, M. and De Jonge, E. (2011), ‘Manipulation of Categorical Edits and Error Localization with the Editrules Package’. Discussion Paper (forthcoming), Statistics Netherlands, The Hague.

Williams, H. P. (1986), ‘Fourier’s Method of Linear Programming and Its Dual’, *The American Mathematical Monthly* **93**, pp. 681–695.

Appendix A Proofs

A.1 Proof of Theorem 2

In order to prove Theorem 2, it is convenient to prove first an auxiliary lemma. Suppose that Ψ_q is obtained from Ψ_{q-1} by eliminating x_g. We define, for each edit ψ_q^k, the index set A_q^k of the edit(s) in Ψ_{q-1} from which it has been derived. That is to say, we define $A_q^k := \{l\}$ if ψ_q^k is obtained by copying the edit ψ_{q-1}^l, and we define $A_q^k := \{s,t\}$ if ψ_q^k is obtained by eliminating a variable from the pair of edits $\psi_{q-1}^s, \psi_{q-1}^t$.

Lemma 1 *Consider the situation of Theorem 2 for $q \geq 1$, and suppose that x_g has been eliminated to obtain Ψ_q from Ψ_{q-1}. Let A be a representing set of the index sets A_q^k belonging to all $\psi_q^k \in \Psi_q^{(2)}$. Then there exists a value for x_g which, together with the original values of the variables that are involved in Ψ_q, satisfies all edits in Ψ_{q-1} except those in A.*

Proof (of Lemma 1) By construction, A contains all indices of failed edits from Ψ_{q-1} which do not involve x_g. Hence, the only way for the lemma to be false would be, if there existed two edits that involve x_g, say ψ_{q-1}^s and ψ_{q-1}^t, with $s \notin A$ and $t \notin A$, such that it is not possible to find a value for x_g that satisfies both edits simultaneously. In this case, an implied edit in Ψ_q is generated by eliminating x_g from ψ_{q-1}^s and ψ_{q-1}^t. (At this point, we need the assumption that the equality elimination rule has not been used.) Moreover, by the fundamental property of Fourier-Motzkin elimination, this implied edit must be failed by the original values of the other variables. In other words, the implied edit must be an element of $\Psi_q^{(2)}$. But this would contradict the assumption that A is a representing set of A_q^k for all $\psi_q^k \in \Psi_q^{(2)}$. Hence, it is impossible to find such a pair of edits, and the lemma follows. □

The proof of Theorem 2 now proceeds by induction on the number of treated variables q. For $q = 0$, the statement is trivial. For $q = 1$, the theorem follows as a special case of Lemma 1; note that $B_1^k \equiv A_1^k$. We suppose therefore that the statement has been proved for all $q \in \{0, 1, \ldots, Q-1\}$, and we consider the case $q = Q$, with $Q \geq 2$.

If Ψ_Q is obtained from Ψ_{Q-1} by fixing a variable to its original value, and B is a representing set of the sets B_Q^k for the failed edits from Ψ_Q, then by construction B is also a representing set of the sets B_{Q-1}^k for the failed edits from Ψ_{Q-1}. Thus, in this case, the statement for $q = Q$ follows immediately from the induction hypothesis.

Thus, we are left with the case that Ψ_Q is obtained from Ψ_{Q-1} by eliminating a variable, say x_g. We define, for each $\psi_Q^k \in \Psi_Q^{(2)}$, the index set A_Q^k of the edit(s) from Ψ_{Q-1} from which ψ_Q^k is derived, just as above. Next, we use B to construct a set A, by applying the following procedure to each $\psi_Q^k \in \Psi_Q^{(2)}$:

- If ψ_Q^k is obtained by copying ψ_{Q-1}^l (so that $A_Q^k = \{l\}$ and $B_Q^k = B_{Q-1}^l$), then we add l to A.
- If ψ_Q^k is obtained by eliminating x_g from ψ_{Q-1}^s and ψ_{Q-1}^t (so that $A_Q^k = \{s,t\}$ and $B_Q^k = B_{Q-1}^s \cup B_{Q-1}^t$), then we add s to A if B contains an element of B_{Q-1}^s, and we add t to A otherwise.

It is easy to see that this procedure produces a representing set A of the index sets A_Q^k for all $\psi_Q^k \in \Psi_Q^{(2)}$.

According to Lemma 1, there exists a value for x_g which, together with the original values of the $p-q$ variables that have not been treated, satisfies the edits in Ψ_{Q-1} except those in A. That is to say, Ψ_{Q-1} can be partitioned similarly to Ψ_Q as $\Psi_{Q-1} = \Psi_{Q-1}^{(1)} \cup \Psi_{Q-1}^{(2)}$, where $\Psi_{Q-1}^{(2)}$ contains the edits with indices in A. Moreover, it is not difficult to see that the above procedure implies that B is a representing set of the index sets B_{Q-1}^k for all $\psi_{Q-1}^k \in \Psi_{Q-1}^{(2)}$. Hence, the induction hypothesis establishes that, given the original values of the variables that have not been eliminated *and* given the chosen value for x_g, there exist values for the other eliminated variables that satisfy all the original edits except those in B. This shows that the statement holds for $q = Q$ and completes the proof of Theorem 2.

A.2 Proof of Theorem 3

To prove Theorem 3, we start again with an auxiliary lemma. Analogously to the numerical case, when Ψ_q is obtained from Ψ_{q-1} by eliminating v_g, we define the index set A_q^k of edits in Ψ_{q-1} from which the edit $\psi_q^k \in \Psi_q$ is derived. To be precise, we define $A_q^k := \{l\}$ if ψ_q^k is obtained by copying the edit ψ_{q-1}^l, and we define $A_q^k := T$ if ψ_q^k is obtained by eliminating a variable from the set of edits ψ_{q-1}^t $(t \in T)$. In contrast with the numerical case, here A_q^k may contain more than two elements.

Lemma 2 *Consider the situation of Theorem 3 for $q \geq 1$, and suppose that v_g has been eliminated to obtain Ψ_q from Ψ_{q-1}. Let A be a representing set of the index sets A_q^k belonging to all $\psi_q^k \in \Psi_q^{(2)}$. Then there exists a value for v_g which, together with the original values of the variables that are involved in Ψ_q, satisfies all edits in Ψ_{q-1} except those in A.*

Proof (of Lemma 2) By construction, A contains all indices of failed edits from Ψ_{q-1} which do not involve v_g. Hence, the only way for the lemma to be false would be, if there existed edits that involve v_g, say $\psi_{q-1}^{t_1}, \ldots, \psi_{q-1}^{t_r}$, with $A \cap \{t_1, \ldots, t_r\} = \emptyset$, such that it is not possible to find a value for v_g that satisfies these edits simultaneously, given the values of the other variables. Clearly, this could only happen if $F_g^{t_1} \cup \cdots \cup F_g^{t_r} = D_g$, since otherwise any value for v_g outside $F_g^{t_1} \cup \cdots \cup F_g^{t_r}$ would work. We may assume without loss of generality that $T' = \{t_1, \ldots, t_r\}$ is a minimal set having this property. Furthermore, it must hold in this case that for all variables involved in Ψ_q, the original value of v_j is contained in all sets $F_j^{t_1}, \ldots, F_j^{t_r}$. In other words, T' must have properties (13) and (14). This means that T' would generate an implied edit in Ψ_q which, as discussed at the beginning of Section 5, must be failed by the original values of the remaining variables. But this would contradict the assumption that A is a representing set of A_q^k for all $\psi_q^k \in \Psi_q^{(2)}$. □

The proof of Theorem 3 is now completely analogous to that of Theorem 2, with Lemma 2 taking the role of Lemma 1. The only slight difference occurs in the procedure to transform B to A, where the second bullet is replaced by:

- If ψ_Q^k is obtained by eliminating v_g from ψ_{Q-1}^t $(t \in T)$ (so that $A_Q^k = T$ and $B_Q^k = \bigcup_{t\in T} B_{Q-1}^t$), then we add one $t \in T$ to A such that B contains an element of B_{Q-1}^t.

Appendix B Using Edit Failure Sizes in D_{soft}

In this appendix, we discuss an extension of the theory in the main text to the case that D_{soft} is a function of the sizes of the soft edit failures. As mentioned in Section 3, taking the edit failure sizes into account is intuitively appealing. However, this approach leads to two technical difficulties. First, one has to choose a measure for edit failure sizes; some suggestions are given in Appendix B.1. Second, the algorithm from Section 6 has to be adapted to solve a more complicated error localisation problem; this adaptation is treated in Appendix B.2.

B.1 Measuring the Size of an Edit Failure

For purely numerical edits, there exists a natural measure of edit failure size. A record $(x_1^0,\dots,x_p^0)$ can be said to fail an edit of the form (6) by the amount

$$e^k = \max\left\{0, -(a_{k1}x_1^0 + \cdots + a_{kp}x_p^0 + b_k)\right\}, \tag{30}$$

and an edit of the form (7) by the amount

$$e^k = \left|a_{k1}x_1^0 + \cdots + a_{kp}x_p^0 + b_k\right|. \tag{31}$$

In both cases, e^k represents the absolute amount by which the left-hand-side of the edit would have to be shifted in order for the edit to become satisfied. That is to say, $e^k = 0$ if a record satisfies an edit, and $e^k > 0$ otherwise.

Thus, for purely numerical edits, it is natural to express D_{soft} as a function of the soft edit failures e^k $(k = 1,\dots,K_S)$. It seems advantageous to standardise the edit failure sizes first, since the magnitude of e^k may be very different for different edits. In a slightly different context, Hedlin (2003) suggested to use the *Mahalanobis distance* for this. The Mahalanobis distance of two vectors $\vec{a}$ and $\vec{b}$, which (supposedly) originate from a distribution with covariance matrix $\mathbf{S}$, is defined as

$$D_M(\vec{a},\vec{b}) = \sqrt{(\vec{a}-\vec{b})'\mathbf{S}^{-1}(\vec{a}-\vec{b})}.$$

As a special case, the Mahalanobis distance between $\vec{a}$ and the mean vector $\vec{\mu}$ of the distribution is often used as a measure of outlyingness for $\vec{a}$:

$$d_M(\vec{a}) = D_M(\vec{a},\vec{\mu}) = \sqrt{(\vec{a}-\vec{\mu})'\mathbf{S}^{-1}(\vec{a}-\vec{\mu})}.$$

See also Little and Smith (1987) for a different application of $d_M(\vec{a})$ in the context of error localisation.

Let $\vec{e} = (e^1,\dots,e^{K_S})'$ denote the vector of soft edit failures for a particular record. (To keep the notation as simple as possible, we omit a separate index for records.) An interesting expression for D_{soft} as a function of soft edit failures for purely numerical edits may be:

$$D_{soft} = D_M(\vec{e},\vec{0}) = \sqrt{\vec{e}'\mathbf{S}_{\vec{e},ref}^{-1}\vec{e}}, \tag{32}$$

where $\vec{0}$ denotes the K_S-dimensional vector of zeros and $\mathbf{S}_{\vec{e},ref}$ is the covariance matrix of $\vec{e}$ in a previously edited reference data set. Since the point $\vec{e} = \vec{0}$ corresponds with the absence of soft edit failures, expression (32) represents the overall magnitude of all soft edit failures in a record. A useful feature of the Mahalanobis distance is that it also takes possible correlations between e^k and e^l ($k \neq l$) into account.

For categorical and mixed edits of the forms (3) and (4), e^k only measures the numerical part of the edit failure. For a mixed record $(v_1^0, \ldots, v_m^0, x_1^0, \ldots, x_p^0)$, the numerical failure size e^k equals expression (30) or (31) if $v_j^0 \in F_j^k$ for all $j = 1, \ldots, m$, and $e^k = 0$ otherwise. In order to measure the overall edit failure size of a categorical or mixed edit, including the categorical part, we suggest a different expression, which can be seen as a rough estimate of the expected change that is needed in a record to satisfy the edit.

First, we may compute, for each variable that is involved in an edit of the form (3) or (4), a prior probability that the variable is erroneous, based on the reciprocal values of the confidence weights:

$$\pi_{kj}^C = \frac{(1/w_j^C) I(F_j^k \neq D_j)}{\sum_{j'=1}^m (1/w_{j'}^C) I(F_{j'}^k \neq D_{j'}) + \sum_{j'=1}^p (1/w_{j'}^N) I(a_{kj'} \neq 0)}$$

for categorical variables v_j ($j = 1, \ldots, m$), and

$$\pi_{kj}^N = \frac{(1/w_j^N) I(a_{kj} \neq 0)}{\sum_{j'=1}^m (1/w_{j'}^C) I(F_{j'}^k \neq D_{j'}) + \sum_{j'=1}^p (1/w_{j'}^N) I(a_{kj'} \neq 0)}$$

for numerical variables x_j ($j = 1, \ldots, p$). Here, $I(A) = 1$ if condition A is true and $I(A) = 0$ otherwise.

Next, for a given record $(v_1^0, \ldots, v_m^0, x_1^0, \ldots, x_p^0)$ and for each variable that is involved in an edit, we compute the minimal amount by which that variable would have to be changed in order to satisfy the edit, assuming that none of the other variables are changed. For a numerical variable x_j with $a_{kj} \neq 0$, it is not difficult to see that the absolute value of the required minimal amount equals

$$d_{kj}^N = \begin{cases} \frac{e^k}{|a_{kj}|} & \text{if the } k^{\text{th}} \text{ soft edit is failed and } a_{kj} \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

where e^k is defined above.

For each categorical variable v_j with $F_j^k \neq D_j$, it is possible to satisfy the edit by changing the value v_j^0 (if necessary) to any value from $D_j \backslash F_j^k$. Assuming that – as in Section 8 – a metric $\delta_j : D_j \times D_j \to \mathbb{R}_{\geq 0}$ has been chosen for each categorical variable, the required minimal amount may thus be defined as

$$d_{kj}^C = \begin{cases} \min_{a \in D_j \backslash F_j^k} \delta_j(v_j^0, a) & \text{if } F_j^k \neq D_j \\ 0 & \text{otherwise} \end{cases}$$

Using these ingredients, an alternative measure of edit failure size may now be defined as follows:

$$\varepsilon^k = \sum_{j=1}^m \pi_{kj}^C d_{kj}^C + \sum_{j=1}^p \pi_{kj}^N d_{kj}^N. \tag{34}$$

This measure estimates the expected value of the absolute amount by which the values in the record have to be changed in order to satisfy the edit. It should be noted that the estimate is rather rough, since it does not take the possibility into account that more than one variable might be changed simultaneously, nor that the minimal change required to satisfy one edit might cause another edit to become failed.

Analogously to e^k, it holds that $\varepsilon^k = 0$ if the k^{th} soft edit is satisfied, and $\varepsilon^k > 0$ otherwise, where larger values of ε^k are to be interpreted as larger soft edit failures. Hence, we can define an analogous expression to (32) for D_{soft} in the case of categorical and mixed edits. Letting $\vec{\varepsilon} = (\varepsilon^1, \ldots, \varepsilon^{K_S})'$ denote the vector of soft edit failures for a particular record, we define

$$D_{soft} = D_M(\vec{\varepsilon}, \vec{0}) = \sqrt{\vec{\varepsilon}' \mathbf{S}^{-1}_{\vec{\varepsilon},ref} \vec{\varepsilon}}, \tag{35}$$

where, similarly to before, $\mathbf{S}_{\vec{\varepsilon},ref}$ is the covariance matrix of $\vec{\varepsilon}$ in a previously edited reference data set.

Example

In the example from Section 7.1, the original record

$$(T^0, P^0, C^0, N^0) = (100, 40000, 60000, 5)$$

does not satisfy the soft edit

$$0.5T - P \geq 0.$$

Using expression (30), it is seen that $e = 39950$, as was already mentioned in Section 10.

We can also evaluate expression (34). The confidence weights of T and P are 2 and 1, respectively, and we find the following prior probabilities:

$$\pi_T = \frac{1/2}{1/2+1} = 1/3,$$

and

$$\pi_P = \frac{1}{1/2+1} = 2/3.$$

Plugging $e = 39950$ into expression (33), we can compute the minimal amounts by which T^0 or P^0 would have to be changed in order to satisfy the edit:

$$d_T = \frac{39950}{0.5} = 79900,$$

and

$$d_P = \frac{39950}{1} = 39950.$$

It is easily verified that the edit indeed becomes satisfied if we either increase the value of T to $100 + 79900 = 80000$ or decrease the value of P to $40000 - 39950 = 50$.

Expression (34) now yields

$$\varepsilon = \frac{1}{3} \times 79900 + \frac{2}{3} \times 39950 = 53266\frac{2}{3}.$$

By comparison, if the same procedure is applied to the record

$$(T^0, P^0, C^0, N^0) = (100, 60, 40, 5),$$

which also fails the above edit, we find $e = 10$ and $\varepsilon = 13\frac{1}{3}$. These different values quantify the intuitive notion that the first record fails the edit "more" than does the second record. □

As the above example shows, for purely numerical edits, the values of e^k and ε^k are generally different. The main theoretical advantage of ε^k is that this measure is suitable for numerical, categorical, and mixed edits. It remains to be seen whether this measure also leads to good results in practice.

B.2 Dynamic Edit Failures

In this section, we describe an extension to the algorithm from Section 6 which can be used to solve the error localisation problem when D_{soft} is a function of the sizes of the soft edit failures. In fact, the extended algorithm is mostly the same as before. The only difference occurs in nodes where it is possible to satisfy all hard edits, but not all soft edits. Previously, we used the binary linear programming problem (19) to find the best representing set of failed soft edits. Instead, we now have to determine *all* minimal representing sets B of $B^k_{q+1,S}$ for the failed soft edits. Here, by "minimal", we mean that there is no strict subset $B_1 \subset B$ that is also a representing set of $B^k_{q+1,S}$. For each of these representing sets B, we determine the amounts by which the original failed soft edits are minimally failed, and use these amounts to compute the associated minimal value of D_{soft}. Finally, we choose the representing set B^* with the lowest minimal value of D_{soft}.

The difficult part in this approach lies in the evaluation of the minimal amounts by which the soft edits are failed. For this, we first need to extend the theory from Sections 4 and 5. The material in this section is rather technical and makes use of some of the notation from Appendix A. Throughout this section, we assume that all edits have the general form (3). It should be noted that an edit of the form (4) may always be replaced by two equivalent edits of the form (3).

We write the numerical condition of the original edit ψ^k_0 as $f^k_0 \geq 0$, with

$$f^k_0(x_1,\ldots,x_p) = \sum_{j=1}^{p} a_{kj}x_j + b_k, \quad k = 1,\ldots,K_0.$$

Here, K_0 denotes the number of original edits. Next, we consider the implied edits ψ^k_1 that are created by eliminating a numerical variable, say x_g. The numerical condition of such an implied edit may be written as $f^k_1 \geq 0$, with f^k_1 a linear combination of f^t_0 $(t = 1,\ldots,K_0)$:

$$f^k_1(x_1,\ldots,x_p) = \sum_{t=1}^{K_0} \lambda^k_{1,t} f^t_0(x_1,\ldots,x_p) = \sum_{t\in A^k_1} \lambda^k_{1,t} f^t_0(x_1,\ldots,x_p),$$

with A^k_1 as defined in Appendix A.1. It should be noted that $\lambda^k_{1,t} > 0$ if $t \in A^k_1$ and $\lambda^k_{1,t} = 0$ otherwise. If A^k_1 refers to a pair of original edits that involve x_g, then the coefficients $\lambda^k_{1,t}$ are chosen such that

$$\sum_{t\in A^k_1} \lambda^k_{1,t} a_{tg} = 0;$$

see (12) in Section 2.3. The same expression also holds in the case that f_1^k is obtained by copying an original edit that does not involve x_g; in that case, we have $\lambda_{1,t}^k = 1$ for the unique $t \in A_1^k$.

More generally, when q numerical variables have been eliminated, the numerical condition of an implied edit ψ_q^k is $f_q^k \geq 0$, where f_q^k may be written as a linear combination of f_0^t $(t = 1, \ldots, K_0)$:

$$f_q^k(x_1, \ldots, x_p) = \sum_{t=1}^{K_0} \Lambda_{q,t}^k f_0^t(x_1, \ldots, x_p) = \sum_{t \in B_q^k} \Lambda_{q,t}^k f_0^t(x_1, \ldots, x_p), \tag{36}$$

with B_q^k as defined in Section 4. The coefficients $\Lambda_{q,t}^k$ are defined recursively by

$$\Lambda_{q,t}^k = \sum_{s \in A_q^k} \lambda_{q,s}^k \Lambda_{q-1,t}^s, \tag{37}$$

with A_q^k as defined in Appendix A.1 and $\lambda_{q,s}^k$ analogous to $\lambda_{1,t}^k$. It should be noted that $\Lambda_{q,t}^k = 0$ if $t \notin B_q^k$. Moreover, it holds that

$$\sum_{t \in B_q^k} \Lambda_{q,t}^k a_{tg} = 0$$

for all variables x_g that have been eliminated so far. For convenience, we define $\Lambda_{0,t}^t = 1$ and $\Lambda_{0,t}^k = 0$ for all $t \neq k$, to ensure that $\Lambda_{1,t}^k = \lambda_{1,t}^k$.

Lemma 3 *Consider again the situation of Theorem 2. Suppose that ψ_q^k is failed by the original values of the remaining variables and let e_q^k denote the size of the failure of the numerical condition as measured by expression* (30). *If $B_q^k \cap B = \{b\}$, then the variables that have been eliminated can imputed such that all edits ψ_0^t with $t \in B_q^k \setminus \{b\}$ are satisfied, while the size of the failure of the numerical condition of edit ψ_0^b as measured by* (30) *equals at least $e_q^k / \Lambda_{q,b}^k$.*

Proof Theorem 2 implies that all edits ψ_0^t with $t \in B_q^k \setminus \{b\}$ can be satisfied. Let $(\tilde{x}_1, \ldots, \tilde{x}_p)$ be the numerical part of an adapted record in which only the eliminated variables have been imputed and which satisfies all edits in $B_q^k \setminus \{b\}$. Using expression (36), the failure size e_q^k of edit ψ_q^k can be written as

$$\sum_{t \in B_q^k} \Lambda_{q,t}^k f_0^t(\tilde{x}_1, \ldots, \tilde{x}_p) = -e_q^k.$$

Since, by assumption, $f_0^t(\tilde{x}_1, \ldots, \tilde{x}_p) \geq 0$ for all $t \in B_q^k \setminus \{b\}$, it must hold that

$$\Lambda_{q,b}^k f_0^b(\tilde{x}_1, \ldots, \tilde{x}_p) = -e_q^k - \sum_{t \in B_q^k \setminus \{b\}} \Lambda_{q,t}^k f_0^t(\tilde{x}_1, \ldots, \tilde{x}_p) \leq -e_q^k,$$

where we have used that $\Lambda_{q,t}^k > 0$ for all $t \in B_q^k$. The lemma now follows. □

Now suppose that, more generally, there are several failed implied edits ψ_q^k, with edit failure sizes $e_q^k > 0$, and that B is a representing set for the associated index sets B_q^k. We write $B_q^{k*} = B_q^k \cap B$. Moreover, let $M(e_0^1, \ldots, e_0^{K_0})$ be a real-valued function of the individual edit failure sizes of the original edits ψ_0^k, and let $M_0(a_t; t \in B)$ be a related function obtained by substituting in M the values $e_0^t = a_t$ for $t \in B$ and $e_0^t = 0$ for $t \notin B$. We consider the following optimisation problem:

$$\begin{aligned} &\text{minimise } M_0(-f_0^t;\, t \in B) \\ &\text{such that:} \\ &\textstyle\sum_{t \in B_q^{k*}} \Lambda_{q,t}^k f_0^t \leq -e_q^k, \text{ for all } k \text{ with } e_q^k > 0, \end{aligned} \tag{38}$$

where the target variables are f_0^t $(t \in B)$.

Theorem 4 *Consider again the situation of Theorem 2. Let $\hat{f}_0^t$ $(t \in B)$ be the optimal solution of problem* (38) *for a given representing set B of the index sets B_q^k associated with all failed edits ψ_q^k. If the eliminated variables are imputed such that all edits ψ_0^t with $t \notin B$ are satisfied, then the total edit failure of the edits ψ_0^t with $t \in B$, as measured by the function M, is at least equal to $M_0(-\hat{f}_0^t;\, t \in B)$.*

Proof The case that $B_q^{k*} = \{b\}$ is treated in Lemma 3. More generally, it can be shown analogously to the proof of Lemma 3 that the numerical failures of the original edits in B_q^{k*} have to satisfy

$$\sum_{t \in B_q^{k*}} \Lambda_{q,t}^k f_0^t(\tilde{x}_1, \ldots, \tilde{x}_p) \leq -e_q^k.$$

Problem (38) considers these inequality conditions simultaneously for all failed edits ψ_q^k. By construction, the optimal solution of (38) consists of the smallest possible combination of values for f_0^t $(t \in B)$ – i.e. the smallest possible as measured by the function M – that satisfies all inequality conditions. □

In the context of the error localisation problem from Section 3, the function M in problem (38) should be replaced by D_{soft}. Depending on the form of D_{soft}, it may be easy or difficult to solve this problem. In Appendix A.1, we suggested a definition for D_{soft} based on the Mahalanobis distance. This would lead to a quadratic programming problem in (38).

Example

To illustrate the use of Theorem 4, we revisit the example from Section 4. The equality edit ψ_0^1 is tacitly replaced here by two equivalent inequality edits:

$$\begin{array}{lrcr} \psi_0^{1a}: & x_1 + x_2 + x_3 & \geq & 20 \\ \psi_0^{1b}: & -x_1 - x_2 - x_3 & \geq & -20 \end{array}$$

It was seen in Section 4 that, after elimination of x_1 from the original edits, five implied edits are failed: ψ_1^2, ψ_1^3, ψ_1^7, ψ_1^9, and ψ_1^{13}. It was also seen that $B = \{1,4,8\}$ is a representing set for the associated index sets B_1^2, B_1^3, B_1^7, B_1^9, and B_1^{13}. A more careful analysis reveals that, when ψ_0^1 is replaced by ψ_0^{1a} and ψ_0^{1b}, this representing set becomes $B = \{1a,4,8\}$.

For the failed implied edits, we list the edit failure sizes e_1^k and the coefficients $\Lambda_{1,t}^k$:

$$\begin{array}{lll} \psi_1^2: & e_1^2 = 15, & \Lambda_{1,1a}^2 = 1, \Lambda_{1,3}^2 = 1 \\ \psi_1^3: & e_1^3 = 30, & \Lambda_{1,1a}^3 = 1, \Lambda_{1,4}^3 = 1 \\ \psi_1^7: & e_1^7 = 3, & \Lambda_{1,8}^7 = 1 \\ \psi_1^9: & e_1^9 = 12, & \Lambda_{1,2}^9 = 1, \Lambda_{1,4}^9 = 1 \\ \psi_1^{13}: & e_1^{13} = 8, & \Lambda_{1,4}^{13} = 1, \Lambda_{1,6}^{13} = 1 \end{array}$$

For instance, ψ_1^2 is obtained by taking a linear combination of ψ_0^{1a} and ψ_0^3, with coefficients $\Lambda_{1,1a}^2 = \Lambda_{1,3}^2 = 1$. Moreover, plugging the original values $x_2^0 = 1$ and $x_3^0 = -3$ into ψ_1^2 yields the expression $-15 \geq 0$, which shows that $e_1^2 = 15$.

We are now ready to formulate problem (38) for this example:

$$\begin{aligned}
&\text{minimise } M_0(-f_0^{1a}, -f_0^4, -f_0^8) \\
&\text{such that:} \\
&f_0^{1a} \leq -15, \\
&f_0^{1a} + f_0^4 \leq -30, \\
&f_0^8 \leq -3, \\
&f_0^4 \leq -12, \\
&f_0^4 \leq -8.
\end{aligned}$$

The optimal solution of this problem is given by: $\hat{f}_0^{1a} = -15 - u$, $\hat{f}_0^4 = -15 + u$, and $\hat{f}_0^8 = -3$, with $0 \leq u \leq 3$. Here, the optimal choice of u depends on the form of M_0. According to Theorem 4, the interpretation of this solution is as follows: when x_1 has to be imputed such that the edits ψ_0^{1b}, ψ_0^2, ψ_0^3, ψ_0^5, ψ_0^6, and ψ_0^7 are all satisfied, then the lowest possible value of $M(e_0^{1a}, e_0^{1b}, e_0^2, e_0^3, e_0^4, e_0^5, e_0^6, e_0^7, e_0^8)$ is obtained by choosing a value for x_1 such that edit ψ_0^{1a} is failed by $e_0^{1a} = 15 + u$, edit ψ_0^4 is failed by $e_0^4 = 15 - u$, and edit ψ_0^8 is failed by $e_0^8 = 3$.

In fact, we already noted in Section 4 that the edits ψ_0^{1b}, ψ_0^2, ψ_0^3, ψ_0^5, ψ_0^6, and ψ_0^7 could be satisfied in this example by imputing any value $x_1 = 7 - u$ with $0 \leq u \leq 3$. It is easy to see that this choice of x_1 yields precisely the above-mentioned edit failures e_0^{1a}, e_0^4, and e_0^8. □

Now, as indicated at the beginning of this section, the theoretical result that we just derived can be incorporated in the error localisation algorithm from Section 6, for the case that D_{soft} depends on the sizes of the soft edit failures. In a node where it is possible to satisfy all hard edits, but not all soft edits, we first find all minimal representing sets B of $B_{q+1,S}^k$ for the failed soft edits. For each of these sets, we set up and solve problem (38) to find the associated minimal value of D_{soft}. Finally, we choose the representing set B^* with the lowest minimal value of D_{soft}.

So far in this section, we have only considered the error localisation problem for numerical data and edits. For mixed data, we assume, as before, that the algorithm does not treat any categorical variable until all numerical variables have been treated. In the first p iterations of the algorithm, the above-mentioned procedure for numerical variables is used. For the implied (purely categorical) edits ψ_p^k, we define $C_p^k := \{k\}$. Next, the algorithm derives implied edits ψ_{p+q}^k by treating the categorical variables. For these implied edits, we define the index sets C_{p+q}^k analogously to B_{p+q}^k:

- For an edit ψ_{p+q}^k which is derived from one other edit ψ_{p+q-1}^l, by fixing a variable to its original value or by simply copying the edit, we define $C_{p+q}^k := C_{p+q-1}^l$.
- For an edit ψ_{p+q}^k which is derived by eliminating a variable from a set of edits ψ_{p+q-1}^t $(t \in T)$, we define $C_{p+q}^k := \bigcup_{t \in T} C_{p+q-1}^t$.

Analogously to Theorem 3, if C is a representing set of C_{p+q}^k for all edits ψ_{p+q}^k that are failed by the remaining categorical variables, then it is possible to impute the eliminated categorical variables such that all edits ψ_p^c with $c \notin C$ are satisfied.

For every such representing set C, we can find another representing set B of the index sets B_p^c for $c \in C$. According to Theorem 2, it is possible to impute the eliminated numerical variables such that all original edits ψ_0^b with $b \notin B$ are satisfied. It should be noted that, by construction, B is also a representing set of the index sets B_{p+q}^k for all failed edits ψ_{p+q}^k. Now, in order to find the minimal value of D_{soft} for the edit failures of ψ_0^b with $b \in B$, it is sufficient to solve problem (38), where the inequality restrictions are given by the failed edits ψ_p^c with $c \in C$. In this manner, we may proceed as before.