UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Budapest, Hungary, 14-16 September 2015)

Topic (v): Emerging methods and data revolution

New results on automatic editing using hard and soft edit rules

Prepared by Sander Scholtus, Statistics Netherlands, Netherlands

I. Introduction

1. To find errors in collected data, statistical institutes typically look for inconsistencies with respect to a set of edit rules. An edit rule is called 'hard' if it identifies erroneous (combinations of) values with certainty and 'soft' if it identifies (combinations of) values that are implausible but not necessarily incorrect. Human editors use both types of constraints. Current methods for automatic editing treat all edit rules as hard constraints and therefore can make only limited use of the information contained in soft edit rules. To extend the possibilities of automatic editing, a new error localisation method was developed at Statistics Netherlands that can distinguish between hard and soft edit rules. Experiences with this new method so far were limited to simulation studies on small synthetic data sets.

2. In this paper, two new results are presented on error localisation with hard and soft edits. Firstly, it is shown that the extended error localisation problem can be re-written in a form that involves only hard constraints. This implies that existing software may be used to solve this problem. Secondly, the new approach is evaluated in a more realistic context, using actual data from the Dutch Structural Business Statistics. These results are presented in Section III and Section IV of the paper, respectively. In preparation of these results, Section II reviews some existing theory of automatic error localisation with hard edits and its extension to soft edits. Some conclusions follow in Section V.

II. Automatic error localisation

A. Data and edit rules

3. Let $\boldsymbol{r} = (\boldsymbol{v}^T, \boldsymbol{x}^T)^T = (v_1, \ldots, v_m, x_1, \ldots, x_n)^T$ denote a generic record of data that consists of m categorical variables and n numerical variables (where either m or n may be zero). It is assumed that each categorical variable v_j takes values in a finite domain D_j and that each numerical variable x_j is restricted in practice to some large interval $(-M, M)$ in $\mathbb{R}$. A particular instance of a record is denoted as $\boldsymbol{r}^0 = (v_1^0, \ldots, v_m^0, x_1^0, \ldots, x_n^0)^T$. Some of the values v_j^0 and x_j^0 may be missing in the original data. It is assumed here that all missing values are erroneously missing and thus require imputation.

4. *Edit rules* (or *edits*) specify constraints that should be satisfied by the data. It is assumed here that all edits can be written in the so-called *normal form* (De Waal, 2003):

$$e_k : \text{IF } \big((v_1, \ldots, v_m) \in F_{k1} \times \cdots \times F_{km}\big) \text{ THEN } (a_{k1}x_1 + \cdots + a_{kn}x_n \odot b_k). \quad (1)$$

Here, $F_{kj} \subseteq D_j$ is a non-empty subset of the domain of v_j; $a_{k1}, \ldots, a_{kn}$ and b_k are numerical constants; and $\odot$ denotes either the $\leq$, $<$, or $=$ operator. The categorical variable v_j (the numerical variable x_j) is said to be *involved* in edit e_k when $F_{kj} \neq D_j$ (when $a_{kj} \neq 0$). A given record $\boldsymbol{r}^0$ is said to *fail* edit e_k

when the IF condition is true and the THEN condition is false; this occurs precisely when $v_j^0 \in F_{kj}$ for all involved categorical variables and the values x_j^0 of the involved numerical variables violate the (in)equality. For instance, the restriction that persons under the age of 18 cannot be married is represented by the following edit rule:

$$\text{IF } (Marital\ Status \in \{\text{"Married"}\}) \text{ THEN } (-Age \leq -18).$$

5. In its general form, (1) denotes a mixed edit. As special cases, there exist purely categorical and purely numerical edits. A purely categorical edit has the form

$$e_k : \text{IF } \big((v_1, \dots, v_m) \in F_{k1} \times \cdots \times F_{km}\big) \text{ THEN } \emptyset, \tag{2}$$

where $\emptyset$ denotes the empty set, with the interpretation that edit (2) is failed whenever its IF condition is true. A purely numerical edit has the form

$$e_k : a_{k1}x_1 + \cdots + a_{kn}x_n \odot b_k. \tag{3}$$

This type of edit is failed whenever the (in)equality is violated. For instance, the categorical edit

$$\text{IF } \big((Gender, Pregnant) \in \{\text{Male}\} \times \{\text{"Yes"}\}\big) \text{ THEN } \emptyset$$

states that men cannot be pregnant, while the numerical balance edit

$$Total\ Costs - Staff\ Costs - Other\ Costs = 0$$

states that the sum of the values of *Staff Costs* and *Other Costs* should equal the value of *Total Costs*.

6. The normal form (1) turns out to be sufficiently general to cover most edit rules that are encountered in practice. In particular, the ratio edit $x_1/x_2 \leq c$ can be linearised as $x_1 - cx_2 \leq 0$, which is an edit of the form (3). Conditional edits on numerical variables can also be expressed in the normal form, by introducing auxiliary categorical variables. For instance, the conditional edit rule

$$\text{IF } (Number\ of\ Employees > 0) \text{ THEN } (Staff\ Costs > 0)$$

is logically equivalent to

$$(Number\ of\ Employees \leq 0) \text{ OR } (-Staff\ Costs < 0).$$

By introducing an auxiliary categorical variable v_{aux} with domain {0,1}, one can replace this statement by two edit rules of the form (1):

$$\text{IF } (v_{\text{aux}} \in \{0\}) \text{ THEN } (Number\ of\ Employees \leq 0);$$

$$\text{IF } (v_{\text{aux}} \in \{1\}) \text{ THEN } (-Staff\ Costs < 0).$$

By similar re-writing techniques, a large class of edits is expressible in the normal form; see De Waal (2005) and De Jonge and Van der Loo (2014) for more details.

B. An error localisation problem with hard edits

7. Suppose that, in a given application, a set of edit rules $\mathcal{E} = \{e_1, \dots, e_K\}$ of the form (1) is defined. For the moment, it is assumed that all edit rules are hard constraints. For each record $\boldsymbol{r}^0$ in the data set, one can easily check which edit rules are satisfied and which are failed. (Any edit that involves variables with missing values is considered to be failed here.) A record is called *consistent* with $\mathcal{E}$ if it does not fail any edit rules, and *inconsistent* otherwise. Under the assumption that all edits are hard, any record that is inconsistent must contain at least one erroneous value. The aim of error localisation is to find the erroneous values in a given inconsistent record.

8. To obtain an error localisation problem that can be solved mathematically, an operational criterion is needed. Fellegi and Holt (1976) suggested that, given an inconsistent record $\boldsymbol{r}^0$, the smallest possible subset of variables should be identified as erroneous for which new values can be imputed to obtain a consistent record. This idea can be generalised slightly by assigning *confidence weights* to the variables, to take into account that some variables are more susceptible to errors than others. The objective is then to minimise

$$D_{\mathrm{FH}} = \sum_{j=1}^{m} w_j^C y_j^C + \sum_{j=1}^{n} w_j^N y_j^N, \quad \left(y_j^C, y_j^N \in \{0,1\}\right), \tag{4}$$

under the condition that the record $\boldsymbol{r}^0$ can be made consistent with $\mathcal{E}$ by imputing new values only for those v_j and x_j with $y_j^C = 1$ and $y_j^N = 1$, respectively. Here, w_j^C (w_j^N) denotes the confidence weight of variable v_j (x_j).

9. The above Fellegi-Holt paradigm forms the basis of most error localisation algorithms that have been developed over the past decades, and it is widely used for automatic editing in official statistics. A statistical motivation was provided by Liepins (1980). He showed that minimising (4) is approximately equivalent to maximising the likelihood of the error pattern, under the condition that the errors can be modelled by a stochastic process that satisfies certain – rather strong – assumptions. (In particular, that errors affect one variable at a time, that they are independent across variables, and that the probability for a variable to be observed in error does not depend on its true value.) Importantly, the Fellegi-Holt paradigm is intended for *random errors*; *systematic errors* such as 'thousand errors' have to be handled in a separate (earlier) step by other editing methods (De Waal *et al.*, 2011).

10. Mathematically, the error localisation problem (4) can be cast as a mixed-integer programming problem, as several authors have noted (*e.g.*, Riera-Ledesma and Salazar-González, 2003, De Waal *et al.*, 2011, and De Jonge and Van der Loo, 2014). Several NSIs have developed and implemented error localisation algorithms. Initially, separate tools were developed for error localisation with purely categorical edits/data and purely numerical edits/data, with heuristic procedures to handle occasional combinations of the two. Some more recent tools can handle mixed edits of the form (1) directly, including SLICE (De Waal, 2005) and the R package `editrules` (De Jonge and Van der Loo, 2014).

C. An error localisation problem with hard and soft edits

11. During manual editing, subject-matter experts often use soft edits to identify (combinations of) values that are suspicious and may require follow-up. For instance, outlier detection criteria can often be summarised in terms of soft edit rules. In contrast to a hard edit, a soft edit can be failed by a record for which none of the involved variables are erroneous. Hence, the specification of a good set of soft edits is a statistical problem that involves a trade-off between the proportion of false negatives (missed errors) and false positives (correct values identified as suspicious). For instance, in the above example of the ratio edit $x_1/x_2 \leq c$, different proportions of false negatives and false positives are obtained by varying the value of the bounding constant c.

12. An important limitation of automatic error localisation based on the Fellegi-Holt paradigm is that all edits are considered to be hard edits. In the presence of soft edits, the two extreme options that are available for automatic processing are: ignoring them (*i.e.*, use only the hard edits) or taking them into account as if they were hard edits. Neither option is particularly attractive. Di Zio *et al.* (2005) considered a more refined approach in which a subset of the soft edits is taken into account (as hard edits). In fact, they proposed to 'optimise' the set of soft edits for automatic processing, by changing bounds, removing ineffective edits, adding new ones, etc. Here, it should be noted that the 'optimal' balance between false negatives and false positives may be different when soft edits are used for automatic editing compared to manual editing. During manual editing, the main issue with false positives is that they reduce the efficiency of the editing process. During automatic processing, on the other hand, they have a direct adverse influence on the quality of error localisation. Hence, it may be a good idea to re-formulate the set of the soft edits before using them as hard edits for automatic editing.

13. Scholtus (2011, 2013) proposed an extension of the Fellegi-Holt paradigm for automatic editing which avoids the interpretation of soft edits as hard edits. Let $\mathcal{E} = \mathcal{E}^H \cup \mathcal{E}^S$, with $\mathcal{E}^H = \{e_1^H, \ldots, e_{K_H}^H\}$ the subset of hard edits and $\mathcal{E}^S = \{e_1^S, \ldots, e_{K_S}^S\}$ the subset of soft edits. Instead of (4), the objective now becomes to minimise

$$D = \lambda D_{\mathrm{FH}} + (1-\lambda) D_{\mathrm{soft}}, \tag{5}$$

under the condition that a given record can be made consistent with $\mathcal{E}^H$ (rather than $\mathcal{E}$) by imputing new values. Here, D_{FH} is given by (4), D_{soft} is an expression that measures the 'costs' of possible failed soft edits, and $\lambda \in [0,1]$ is a parameter that balances the contributions of the two terms to D.

14. By analogy with D_{FH}, a simple choice for D_{soft} is given by:

$$D_{\text{soft}} = \sum_{k=1}^{K_S} s_k z_k, \quad (z_k \in \{0,1\}), \tag{6}$$

with $z_k = 1$ if soft edit e_k^S is failed by the imputed record and $z_k = 0$ otherwise. Here, s_k is a so-called *failure weight*. With this choice of D_{soft}, the error localisation problem is solved by minimising the following expression:

$$D = \lambda \left(\sum_{j=1}^{m} w_j^C y_j^C + \sum_{j=1}^{n} w_j^N y_j^N \right) + (1-\lambda) \sum_{k=1}^{K_S} s_k z_k. \tag{7}$$

Note that, typically, by increasing the number of variables to impute, one can decrease the number of soft edits that are failed after imputation. Expression (7) weighs the costs of imputing additional variables (measured by the confidence weights) against the costs of failing soft edits (measured by the failure weights). In particular, a soft edit with a higher failure weight is less likely to be failed by the optimal solution to this extended error localisation problem. Various methods for assigning failure weights to soft edits were proposed by Scholtus and Göksen (2012); some of these will be reviewed below in Section IV.

15. Other choices for D_{soft} than (6) were also considered by Scholtus (2011) and Scholtus and Göksen (2012). In particular, a theoretical drawback of expression (6) is that it does not account for the sizes of soft edit failures. As an example, consider the soft ratio edit $x_1/x_2 \leq 5$ and two records with $(x_1^0 = 60,\ x_2^0 = 10)$ and $(x_1^0 = 6000,\ x_2^0 = 10)$. In both cases the edit is failed, so the contribution to (6) would be the same for both records. During manual editing, the second combination of values would probably be considered more suspicious, because the edit failure is 'larger' in some sense. The sizes of soft edit failures can be taken into account in (6) to some extent by working with so-called *quantile edits*; see Scholtus and Göksen (2012). Alternatively, it is possible to choose D_{soft} as a direct function of edit failure sizes, but this leads to a more complicated error localisation problem (Scholtus, 2011).

16. Scholtus (2013) proposed an algorithm for solving the error localisation problem (5) when D_{soft} is a function of $z_1, \ldots, z_{K_S}$ alone. In the next section it is shown that, for the special case with D given by (7), this problem can be re-written so that it involves only hard edits. This implies that existing tools such as SLICE and the R package `editrules` can be used, after some essentially trivial modifications, to perform error localisation with hard and soft edits.

III. An alternative formulation of the problem

17. The idea is to make use of the similarity between (4) and (7) by replacing, in the latter expression, the failure weight of each soft edit by the confidence weight of an auxiliary variable. Basically, this auxiliary variable is just z_k, but it will be denoted by v_{m+k} because of the notational convention used here. (Variables that occur in the data are denoted by v or x, while y and z denote target variables of the error localisation problem.) The details of the argument now follow.

18. Given the original record $\boldsymbol{r} = (v_1, \ldots, v_m, x_1, \ldots, x_n)^T$ to which error localisation problem (7) refers, define $\tilde{\boldsymbol{r}} = \left(v_1, \ldots, v_m, v_{m+1}, \ldots, v_{m+K_S}, x_1, \ldots, x_n\right)^T$ with K_S auxiliary categorical variables. The domain $D_{m+k} = \{0,1\}$ for all $k \in \{1, \ldots, K_S\}$. Suppose that the original soft edit e_k^S is given by

$$e_k^S : \text{IF } \left((v_1, \ldots, v_m) \in F_{k1}^S \times \cdots \times F_{km}^S\right) \text{ THEN } \left(a_{k1}^S x_1 + \cdots + a_{kn}^S x_n \odot b_k^S\right), \tag{8}$$

which is an edit of the form (1). Soft edit (8) can be replaced by the following hard edit $e_{K_H+k}^H$:

$$e^H_{K_H+k} : \text{IF } \left((v_1, \ldots, v_m, v_{m+1}, \ldots, v_{m+K_S}) \right.$$
$$\left. \in F^S_{k1} \times \cdots \times F^S_{km} \times D_{m+1} \times \cdots \times D_{m+k-1} \times \{0\} \times D_{m+k+1} \times \cdots \times D_{m+K_S} \right) \quad (9)$$
$$\text{THEN } \left(a^S_{k1} x_1 + \cdots + a^S_{kn} x_n \odot b^S_k \right).$$

Note the following: if $v_{m+k} = 0$, then the hard edit $e^H_{K_H+k}$ is satisfied by $\tilde{\boldsymbol{r}}$ if, and only if, the original soft edit e^S_k is satisfied by $\boldsymbol{r}$; if $v_{m+k} = 1$, then $e^H_{K_H+k}$ is satisfied regardless of the status of e^S_k. Furthermore, note that (9) is again an edit rule of the general form (1).

19. The following confidence weights are assigned to the variables in the extended record $\tilde{\boldsymbol{r}}$:

$$\begin{aligned} \tilde{w}^C_j &= \lambda w^C_j, && (j \in \{1, \ldots, m\}), \\ \tilde{w}^N_j &= \lambda w^N_j, && (j \in \{1, \ldots, n\}), \\ \tilde{w}^C_{m+k} &= (1-\lambda) s_k, && (k \in \{1, \ldots, K_S\}), \end{aligned}$$

with w^C_j, w^N_j, s_k, and λ as given in the original formulation of the error localisation problem. Finally, the auxiliary variables $v_{m+1}, \ldots, v_{m+K_S}$ are all initialised to the value 0 in the input data.

20. It is not difficult to show that the following two problems are equivalent:
1. the error localisation problem given by (7) applied to the original record $\boldsymbol{r}$ with the set of hard edits $\mathcal{E}^H$ and the set of soft edits $\mathcal{E}^S$;
2. the Fellegi-Holt-based error localisation problem (4) applied to the extended record $\tilde{\boldsymbol{r}}$ with the set of hard edits $\tilde{\mathcal{E}} = \mathcal{E}^H \cup \{e^H_{K_H+1}, \ldots, e^H_{K_H+K_S}\}$.

In fact, the first problem minimises expression (7) under the condition that the imputed record satisfies all edits in $\mathcal{E}^H \cup \mathcal{E}^S$ *except* the soft edits e^S_k with $z_k = 1$. The second problem minimises

$$\sum_{j=1}^{m} \lambda w^C_j y^C_j + \sum_{j=1}^{n} \lambda w^N_j y^N_j + \sum_{k=1}^{K_S} (1-\lambda) s_k y^C_{m+k} \quad (10)$$

under the condition that the imputed extended record satisfies all edits in $\tilde{\mathcal{E}}$. It was seen above that the soft edit e^S_k is allowed to be failed if, and only if, $v_{m+k} = 1$. This in turn requires that $y^C_{m+k} = 1$, since one has to impute v_{m+k} to obtain $v_{m+k} = 1$. Hence, the role of y^C_{m+k} in (10) is identical to that of z_k in (7), and it follows that the two minimisation problems are equivalent.

21. As mentioned above, the importance of this result is that existing implementations of error localisation algorithms based on the Fellegi-Holt paradigm with only hard edits can be used directly to solve the new problem with hard and soft edits. All that is required is that the input data, edit rules, and confidence weights be modified slightly before the algorithm is run. Potentially, this could greatly increase the applicability of the new error localisation problem with hard and soft edits. There are two limitations to this result. Firstly, the existing error localisation tool should be able to handle mixed edits of the form (1); this is true for instance of SLICE and `editrules`, but not of all existing tools. Secondly, the result is limited to the error localisation problem with D_{soft} given by (6). More general forms would require a different algorithm, for instance that of Scholtus (2011, 2013).

IV. Application

A. Data and edit rules

22. Previous evaluation studies of the error localisation problem (5) were based on relatively small data sets (with respect to the number of variables and edits) and restricted to purely numerical edits of the form (3). In this section, results will be discussed of a simulation study with mixed edit rules and real data from the Dutch Structural Business Statistics (SBS) of 2007 on wholesale. All results were obtained in

the R environment for statistical computing (R Development Core Team, 2015), using the above-mentioned `editrules` package and the alternative problem formulation of Section III.

23. All responding units had completed one of two SBS questionnaires on wholesale: units in size classes 1, 2, and 3 (corresponding to businesses with less than 10 employees) received a shorter questionnaire, while units in size classes 4 to 9 (businesses with 10 employees or more) received the long version. For this simulation study, only records were retained that had been edited manually during regular production. These manually edited data were assumed to be error-free. During regular production, the data of the previous year would be available as reference data, for instance to estimate failure weights (see subsection IV.C below). For the purpose of this study, a random subset of about half of the records was used as reference data. The other records were used as test data. To this end, various automatic editing methods were applied to the original test data prior to manual editing and the results were compared to the manually edited test data. Since the focus here is on methods for automatic localisation of random errors, as a starting point a version of the test data was used from which several types of systematic error, including 'thousand errors', had already been removed.

Table 1. Data used in the study.

	size classes 1–3	size classes 4–9
number of records (test data)	140	846
number of records (test data; ≤ 10 errors)	126	800
number of records (reference data)	143	853
number of variables	69	89
number of hard edit rules	93	111
number of soft edit rules	24	37

24. Table 1 lists some properties of the data. All variables are numerical, with one exception: size class. However, it is not allowed to change the value of size class during editing, as a coordinated version of this variable is maintained in the general business register. Most variables contained a substantial fraction of missing data. Of all the SBS variables, a subset of 22 'core variables' is published regularly by Statistics Netherlands. The evaluation results below focus mostly on these core variables. Moreover, we restrict attention to records in the test data that contain at most 10 errors. (According to the manually edited test data, the largest number of errors in a single record is 42.)

25. The edit rules (hard and soft) were obtained directly from the production database. These consisted mostly of balance edits, non-negativity edits, ratio edits, and conditional edits. The structure of the edit rules was very similar between the short and long versions of the questionnaire.

B. Evaluation criteria

26. Since the manually edited version of the test data was considered error-free, the quality of error localisation could be evaluated directly. To this end, the following contingency table was computed for all individual *non-missing* values in the test data:

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

From this table, the proportions of false negatives, false positives, and overall wrong decisions were obtained (De Waal *et al.*, 2011, pp. 410–411):

$$\alpha = \frac{FN}{TP + FN}; \quad \beta = \frac{FP}{FP + TN}; \quad \delta = \frac{FN + FP}{TP + FN + FP + TN}.$$

It should be noted that all error localisation methods identify the missing values correctly. To enable a clear comparison of the quality of error localisation between different methods, missing values were therefore suppressed in the computation of α, β, and δ. As an additional measure the value $\rho^c = 1 - \rho$ was computed, with ρ the fraction of records in the test data for which exactly the right solution to the error localisation problem was obtained. Lower values of α, β, δ, and ρ^c indicate a better quality of error localisation. It may be helpful to note that $1 - \alpha$ and $1 - \beta$ are also known as *sensitivity* and *specificity*.

C. Results

27. Table 2 displays the results for several variants of the Fellegi-Holt-based error localisation problem (4), with all specified edits treated as hard constraints. The second and third rows refer to the case that ignores the soft edit rules, while the fourth and fifth rows refer to the case with all soft edits treated as hard ones. Both variants were run once with all confidence weights equal to 1 ("no weights") and once with the confidence weights from the production database ("weights"). For completeness, evaluation results are also shown for the case of no error localisation (first row). Table 2a contains the above-mentioned evaluation measures, computed for the 22 core variables.

Table 2a. Evaluation results for Fellegi-Holt-based error localisation (22 core variables).

editing approach	size classes 1–3				size classes 4–9			
	α	β	δ	ρ^c	α	β	δ	ρ^c
no error localisation	1.000	0.000	0.096	0.571	1.000	0.000	0.063	0.549
only hard edits (no weights)	0.779	0.012	0.085	0.500	0.813	0.007	0.058	0.493
only hard edits (weights)	0.784	0.008	0.082	0.500	0.833	0.007	0.059	0.499
all edits as hard edits (no weights)	0.618	0.093	0.143	0.849	0.650	0.060	0.097	0.694
all edits as hard edits (weights)	0.638	0.091	0.143	0.857	0.649	0.060	0.097	0.693

Table 2b. Evaluation results for Fellegi-Holt-based error localisation (continued): number of identified errors; mean (t_{mean}) and median (t_{med}) computation time per record (in seconds); number of records with a possibly suboptimal solution (N_{sub}).

editing approach	size classes 1–3				size classes 4–9			
	errors	t_{mean}	t_{med}	N_{sub}	errors	t_{mean}	t_{med}	N_{sub}
no error localisation	0	–	–	–	0	–	–	–
only hard edits (no weights)	125	0.45	0.21	0	497	0.24	0.18	0
only hard edits (weights)	127	0.36	0.17	0	507	0.30	0.18	0
all edits as hard edits (no weights)	448	0.65	0.25	0	2168	0.91	0.20	0
all edits as hard edits (weights)	462	2.15	0.30	0	2187	1.15	0.22	0

28. It is seen that, with all these approaches, the quality of error localisation was rather poor. The main issue is that a substantial number of true errors were not found (large values of α). Taking all soft edits into account as if they were hard constraints improved the value of α, but at the cost of a much larger fraction of false positives β. On balance, ignoring the soft edits appears to be preferable to using them as hard constraints. It may be interesting to note that the variant "only hard edits (weights)" is currently used for automatic editing during regular SBS production at Statistics Netherlands. An unexpected result is that the inclusion or exclusion of confidence weights hardly affected the quality of error localisation. For brevity, attention is therefore restricted to variants with all confidence weights equal to 1 in what follows.

29. Table 2b shows some additional properties. The first column contains the total number of errors (with missing values excluded) found by the algorithm; this number refers to the full set of variables, not just the core ones. The actual number of errors in the test data was 455 for size classes 1–3 and 2404 for size classes 4–9. Thus, it is seen that most approaches listed in Table 2 identified far too few errors, which explains the large fraction of false negatives. A possible explanation for this phenomenon is that the data contain a large number of missing values. Since the error localisation algorithm aims to minimise the number of imputations per record, and missing variables have to be imputed with certainty, observed values will be identified as erroneous only if the edit rules cannot be satisfied by imputing the missing variables alone. On the other hand, subject-matter experts may identify additional errors in the observed values during manual editing, because they are not bound by an explicit optimisation criterion. The presence of a large number of missing values might also explain the above-mentioned lack of effect of setting different confidence weights.

30. Table 2b also shows the mean (t_{mean}) and median (t_{med}) computation time per record required by the algorithm. In all variants, the maximal computation time per record was set to 10 minutes. If the algorithm could not be completed within 10 minutes, the best – possibly suboptimal – solution found was used. The column N_{sub} lists the number of times this occurred. For the variants in Table 2, the optimal

solution was always found within the allowed time. (In theory, it is possible for the algorithm to find no feasible solution at all in the first 10 minutes, but this did not happen during this study.)

31. For the error localisation problem with hard and soft edits (5), only variants with D_{soft} given by (6) were considered in this study. Given this basic format, a variant is completely specified by the choice of failure weights $s_1, \ldots, s_{K_S}$ and weighting parameter λ. Variants with $\lambda \in \{0.3, 0.5, 0.7\}$ were considered in this study. With $\lambda = 0.5$, both terms in expression (7) are weighted equally. Lower (higher) values of λ place more (less) emphasis on D_{soft}, which means that soft edits are less (more) likely to be failed by the optimal solution.

32. For the failure weights, the following choices were considered:

$$s_k^A = 1;$$
$$s_k^B = P\left(z_k^{\text{clean}} = 0\right);$$
$$s_k^C = P\left(z_k^{\text{clean}} = 0 \middle| z_k^{\text{raw}} = 1\right).$$

Here, z_k^{clean} denotes the indicator z_k evaluated on an error-free record and z_k^{raw} denotes z_k evaluated on the corresponding unedited record. That is, s_k^B denotes the probability that an arbitrary error-free record does not fail the k^{th} soft edit, and s_k^C denotes the associated conditional probability given that the unedited version of the record *does* fail the k^{th} soft edit. These probabilities were estimated from the empirical distribution of edit failures in the reference data. The above failure weights were proposed, with motivation, by Scholtus and Göksen (2012).

33. Table 3 displays evaluation results for several editing approaches with hard and soft edits for units that completed the long questionnaire. Results with $\lambda = 0.7$ were consistently worse and these are omitted here. In comparison with the approaches in Table 2, most variants with hard and soft edits yielded a slight improvement in the quality of error localisation, in particular regarding the fraction of false negatives α and the fraction of records with a perfect solution ρ. The best overall performance was obtained with s_k^C and $\lambda = 0.3$. Nonetheless, the differences between approaches were small and the quality of error localisation remained rather poor in all cases. Moreover, a drawback of the introduction of soft edit rules into the error localisation problem is that it led to significantly longer computation times. The vast majority of records could still be solved to optimality within 10 minutes.

Table 3. Evaluation results for error localisation with hard and soft edits, for size classes 4–9. The first set of columns refers to the 22 core variables.

editing approach	α	β	δ	ρ^c	errors	t_{mean}	t_{med}	N_{sub}
approaches with $\lambda = 0.5$								
failure weights A	0.750	0.012	0.058	0.479	786	19.59	0.41	13
failure weights B	0.778	0.010	0.058	0.479	638	23.69	0.38	17
failure weights C	0.778	0.011	0.059	0.483	660	31.31	0.43	24
approaches with $\lambda = 0.3$								
failure weights A	0.724	0.018	0.062	0.516	1085	3.94	0.36	0
failure weights B	0.724	0.019	0.063	0.529	1085	2.82	0.33	0
failure weights C	0.731	0.011	0.056	0.459	889	5.52	0.33	2
'ideal' subset as hard edits	0.703	0.011	0.054	0.440	865	0.57	0.20	0

34. Scholtus and Göksen (2012) also suggested several other methods for choosing failure weights, and some of these were also tested in this study (not shown). This did not lead to any significant improvements. By choosing the failure weights and other parameters of the error localisation problem (7), one implicitly decides, for each record, which soft edit rules should be failed after imputation and which should be satisfied. An 'ideal' choice of parameters would yield an imputed data set in which each record satisfies the 'ideal' subset of soft edit rules, *i.e.*, the subset that is satisfied by the error-free version of that record. Since the error-free data were available in this study, the 'ideal' subset was known for each record. Hence, it was possible to run a variant of error localisation problem (4) with, for each record, the 'ideal' subset of soft edit rules taken into account as hard constraints, and the remaining soft edits

omitted. The results of this variant are shown in the last row of Table 3. It is seen that, even under these 'ideal' conditions, only a modest improvement was obtained compared to the approach that did not use any soft edits. A similar result was found for units that completed the short questionnaire.

D. Discussion

35. Overall, the results of automatic editing with hard and/or soft edits in this study were rather disappointing. Taking the current set of soft edits into account did not lead to a substantial improvement in the quality of error localisation, while in fact there was a lot of room for improvement (*e.g.*, only about one in every five errors was identified correctly by the Fellegi-Holt-based method without soft edits). It should be noted that these results were obtained on data that were selected for manual editing during regular production of the Dutch SBS. Given the selective editing strategy that is used for the Dutch SBS (Hoogland, 2006; Hoogland and Smit, 2008), these records represent the relatively 'difficult' cases, in particular among smaller units. It is possible – perhaps even likely – that better results would have been obtained for units that were edited automatically during regular production.

36. It should also be noted that this study used the same soft edits that were used during manual editing of the SBS 2007 on wholesale. As mentioned in Section II.C, these edits may not be ideal for direct use in an automatic method. One way to improve the results of automatic error localisation might therefore be to derive a different set of soft edit rules that is more suited to this purpose. The introduction of "quantile edits" to take the sizes of soft edit failures into account might also lead to an improvement (*cf.* Scholtus and Göksen, 2012). Another option is to look at the role of the missing values. As noted above, missing values can 'mask' the presence of errors among the observed values, because error localisation algorithms tend to minimise the number of variables to impute. Most of the missing values in the data sets used in this study were in fact equal to zero. It might be possible to obtain better error localisation results among the observed values if (some of) the missing values are imputed beforehand.

V. Conclusion

37. In this paper, the problem of automatic error localisation given a set of edit rules was discussed. Traditional methods based on the Fellegi-Holt paradigm treat all edit rules as hard constraints. An alternative error localisation method was discussed that can distinguish between hard edit rules (which always have to be satisfied) and soft edit rules (which may be failed at some additional 'cost'). It was shown that a special case of this problem can be re-written in the format of the Fellegi-Holt-based error localisation problem. Thus, existing tools for automatic error localisation can be used to solve this problem. In particular, the alternative error localisation problem fits within the functionality of the R package `editrules` which is open source and available for download at http://cran.r-project.org/.

38. The paper also presented some results of an evaluation study on automatic error localisation using data from the Dutch SBS. The outcome of automatic editing was compared to the outcome of manual editing by subject-matter experts. It was found that there were large differences between the two approaches. In particular, the automatic methods found much smaller numbers of erroneously observed values than the subject-matter experts did. Thus, in this application at least, there is a substantial gap between manual editing and automatic editing based on the Fellegi-Holt paradigm, which one could attempt to close by developing more flexible methods for automatic editing. A slight improvement was in fact obtained by taking soft edits into account, but large differences remained also with the alternative method. Some ideas for further improvements were discussed at the end of Section IV. In particular, the set of soft edits used in the study seems to be suboptimal for the purpose of automatic editing.

VI. References

E. de Jonge and M. van der Loo (2014), *Error Localization as a Mixed Integer Problem with the* `editrules` *Package*. Discussion Paper 2014-07, Statistics Netherlands, The Hague.

T. de Waal (2003), *Processing of Erroneous and Unsafe Data*. PhD Thesis, Erasmus University, Rotterdam.

T. de Waal (2005), *SLICE 1.5: A Software Framework for Automatic Edit and Imputation*. Working Paper, UN/ECE Work Session on Statistical Data Editing, Ottawa.

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

M. Di Zio, U. Guarnera, and O. Luzi (2005), *Improving the Effectiveness of a Probabilistic Editing Strategy for Business Data*. Report, ISTAT, Rome.

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

J. Hoogland (2006), Selective Editing using Plausibility Indicators and SLICE. In: *Statistical Data Editing, Volume 3: Impact on Data Quality*. United Nations, New York and Geneva, pp. 106–130.

J. Hoogland and R. Smit (2008), *Selective Automatic Editing of Mixed Mode Questionnaires for Structural Business Statistics*. Working Paper, UN/ECE Work Session on Statistical Data Editing, Vienna.

G.E. Liepins (1980), *A Rigorous, Systematic Approach to Automatic Data Editing and its Statistical Basis*. Report ORNL/TM-7126, Oak Ridge National Laboratory.

J. Riera-Ledesma and J.J. Salazar-González (2003), *New Algorithms for the Editing and Imputation Problem*. Working Paper, UN/ECE Work Session on Statistical Data Editing, Madrid.

R Development Core Team (2015), *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

S. Scholtus (2011), *Automatic Editing with Soft Edits*. Discussion Paper 201130, Statistics Netherlands, The Hague.

S. Scholtus (2013), Automatic Editing with Hard and Soft Edits. *Survey Methodology* **39**, 59–89.

S. Scholtus and S. Göksen (2012), *Automatic Editing with Hard and Soft Edits – Some First Experiences*. Discussion Paper 201225, Statistics Netherlands, The Hague.