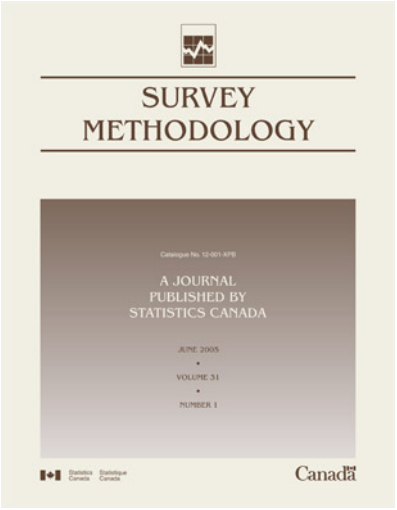
Catalogue no. 12-001-XIE

Survey Methodology

2005



Statistics Canada Statistique Canada

Canada

How to obtain more information

Specific inquiries about this product and related statistics or services should be directed to: Business Survey Methods Division, Statistics Canada, Ottawa, Ontario, K1A 0T6 (telephone: 1 800 263-1136).

For information on the wide range of data available from Statistics Canada, you can contact us by calling one of our toll-free numbers. You can also contact us by e-mail or by visiting our website.

National inquiries line	1 800 263-1136
National telecommunications device for the hearing impaired	1 800 363-7629
Depository Services Program inquiries	1 800 700-1033
Fax line for Depository Services Program	1 800 889-9734
E-mail inquiries	infostats@statcan.ca
Website	www.statcan.ca

Information to access the product

This product, catalogue no. 12-001-XIE, is available for free. To obtain a single issue, visit our website at www.statcan.ca and select Our Products and Services.

Standards of service to the public

Statistics Canada is committed to serving its clients in a prompt, reliable and courteous manner and in the official language of their choice. To this end, the Agency has developed standards of service that its employees observe in serving its clients. To obtain a copy of these service standards, please contact Statistics Canada toll free at 1 800 263-1136. The service standards are also published on www.statcan.ca under About Statistics Canada > Providing services to Canadians.

Statistics Canada
Business Survey Methods Division

Survey Methodology

2005

Published by authority of the Minister responsible for Statistics Canada

© Minister of Industry, 2005

All rights reserved. The content of this publication may be reproduced, in whole or in part, and by any means, without further permission from Statistics Canada, subject to the following conditions: that it is done solely for the purposes of private study, research, criticism, review, newspaper summary, and/or for non-commercial purposes; and that Statistics Canada be fully acknowledged as follows: Source (or "Adapted from", if appropriate): Statistics Canada, name of product, catalogue, volume and issue numbers, reference period and page(s). Otherwise, no part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopy, for any purposes, without the prior written permission of Licensing Services, Marketing Division, Statistics Canada, Ottawa, Ontario, Canada K1A 0T6.

November 2005

Catalogue no. 12-001-XIE
ISSN 1492-0921

Frequency: semi-annual

Ottawa

Cette publication est disponible en français sur demande (nº 12-001-XIF au catalogue).

Note of appreciation

Canada owes the success of its statistical system to a long-standing partnership between Statistics Canada, the citizens of Canada, its businesses, governments and other institutions. Accurate and timely statistical information could not be produced without their continued cooperation and goodwill.

Vol. 29, No. 1, pp. 71-79
Statistics Canada, Catalogue No. 12-001

Solving the Error Localization Problem by Means of Vertex Generation

Ton de Waal [1]

Abstract

To automate the data editing process the so-called error localization problem, *i.e.*, the problem of identifying the erroneous fields in an erroneous record, has to be solved. A paradigm for identifying errors automatically has been proposed by Fellegi and Holt in 1976. Over the years their paradigm has been generalized to: the data of a record should be made to satisfy all edits by changing the values of the variables with the smallest possible sum of reliability weights. A reliability weight of a variable is non-negative number that expresses how reliable one considers the value of this variable to be. Given this paradigm the resulting mathematical problem has to be solved. In the present paper we examine how vertex generation methods can be used to solve this mathematical problem in mized data, *i.e.*, a combination of categorical (discrete) and numerical (continuous) data. The main aim of this paper is not to present new results, but rather to combine the ideas of several other papers in order to give a "complete", self-contained description of the use of vertex generation methods to solve the error localization problem in mixed data. In our exposition we will focus on describing how methods for numerical data can be adapted to mixed data.

Key Words: Chernikova's algorithm; Error localization; Fellegi-Holt paradigm; Fourier-Motzkin elimination; Mathematical programming; Mixed data editing; Statistical data editing; Vertex generation.

1. Introduction

An important problem that has to be solved in order to automate the data editing process is the so-called error localization problem, *i.e.*, the problem of identifying the erroneous fields in an erroneous record. Fellegi and Holt (1976) describe a paradigm for identifying errors in a record automatically. According to this paradigm the data of a record should be made to satisfy all edits by changing the values of the fewest possible number of variables. In due course the original Fellegi-Holt paradigm has been generalized to: the data of a record should be made to satisfy all edits by changing the values of the variables with the smallest possible sum of reliability weights. A reliability weight of a variable is a non-negative number that expresses how reliable one considers the value of this variable to be. A high reliability weight corresponds to a variable of which the values are considered trustworthy, a low reliability weight to a variable of which the values are considered not so trustworthy.

Describing a paradigm for identifying the erroneous fields in an erroneous record is only a first step towards solving the error localization problem, however. The second step consists of actually solving the resulting mathematical problem. This mathematical problem can be solved in several ways, see *e.g.*, Fellegi and Holt (1976); De Waal and Quere (2003), and De Waal (2003). One of these ways is by generating vertices of a certain polyhedron. Unfortunately, the number of vertices of this polyhedron is often too high for this approach to be applicable in practice. Instead, one should therefore generate a suitable subset of the vertices only. There are a number of vertex generation algorithms that efficiently generate such a suitable subset of vertices of a polyhedron. An example of such a vertex generation algorithm is an algorithm proposed by Chernikova (1964, 1965). Probably most computer systems for automatic edit and imputation of numerical data are based on adapted versions of this algorithm. The best-known example of such a system is GEIS (Kovar and Whitridge 1990). Other examples are CherryPi (De Waal 1996), AGGIES (Todaro 1999), and a SAS program developed by the Central Statistical Office of Ireland (see Central Statistical Office 2000). The original algorithm of Chernikova is rather slow for solving the error localization problem. It has been accelerated by various modifications (see Rubin 1975 and 1977; Sande 1978; Schiopu-Kratina and Kovar 1989; Fillion and Schiopu-Kratina 1993).

Only the last three of these papers focus on the error localization problem. Sande (1978) discusses the error localization problems for numerical data, categorical data and mixed data. The discussion of the error localization problem in mixed data is very brief, however. Schiopu-Kratina and Kovar (1989) and Fillion and Schiopu-Kratina (1993) propose a number of improvements on Sande's method for solving the error localization problem for numerical data. They do not consider the error localization problems for numerical data. They do not consider the error localization problems for categorical or mixed data.

In the present paper we examine how vertex generation methods can be used to solve the error localization problem in mixed data, *i.e.*, a combination of categorical (discrete) and numerical (continuous) data. The main aim of this paper is not to present new results, but rather to combine the ideas

1. Ton de Waal, Statistics Netherlands, PO Box 40000, 2270 JM Voorburg, The Netherlands. E-mail: twal@cbs.nl.

of the above-mentioned papers in order to give a "complete", self-contained description of the use of vertex generation methods to solve the error localization problem in mixed data. We will especially describe how modifications to accelerate Chernikova's algorithm for numerical data can also be used for mixed data.

The remainder of the present paper is organized as follows. Section 2 gives a formal definition of the edits that we consider as well as a number of examples. Section 3 formulates the error localization problem as a mixed integer programming problem. Section 4 describes how the error localization problem can be solved by generating vertices of an appropriate polyhedron. We describe how Chernikova's algorithm can be used to generate these vertices in sections 5 and 6. In these sections we also describe modifications to the algorithm in order to improve its performance. Section 7 concludes the paper with a brief discussion. In the Appendix we give Rubin's description of Chernikova's algorithm. In this paper proofs are omitted for most results. The interested reader is referred to the literature for those proofs.

2. The Edits

2.1 Formal Definition of the Edits

We denote the categorical variables by v_i $(i = 1, ..., m)$ and the numerical variables by x_i $(i = 1, ..., n)$. For categorical data we denote the domain, *i.e.*, the set of possible values, of variable i by D_i. We assume that every edit E^j $(j = 1, ..., J)$ is written in the following form: edit E^j is satisfied by a record $(v_1, ..., v_m, x_1, ..., x_n)$ if and only if the following statement holds true:

$$\text{IF } v_i \in F_i^j \quad \text{for} \quad i = 1, ..., m$$
$$\text{THEN } (x_1, ..., x_n) \in \left\{ \mathbf{x} \mid \begin{array}{r} a_{1j} x_1 + ... + a_{nj} x_n \\ + b_j \geq 0 \end{array} \right\}, \quad (2.1)$$

where $F_i^j \subset D_i$ $(j = 1, ..., J)$. Numerical variables may attain negative values. For non-negative variables an edit of type (2.1) needs to be introduced in order to ensure non-negativity. A numerical equality can be expressed as two inequalities.

All edits have to be satisfied simultaneously. A record that satisfies all edits is called a consistent record. The condition after the IF – statement, *i.e.*, "$v_i \in F_i^j$ for all $i = 1, ..., m$", is called the IF – condition of edit j $(j = 1, ..., J)$. The condition after the THEN – statement is called the THEN – condition. If the IF – condition does not hold true, the edit is always satisfied, irrespective of the values of the numerical variables. If the set in the THEN – condition of (2.1) is the entire n – dimensional real vector space, then the edit is always satisfied and may be discarded. If the set in the THEN – condition of (2.1) is empty, then the edit is failed by any record for which the IF – condition holds.

In many practical cases, certain kinds of missing values are acceptable, *e.g.*, when the corresponding questions are not applicable to a particular respondent. We assume that for categorical variables such acceptable missing values are coded by special values in their domains. Non-acceptable missing values of categorical variables are not coded. The optimization problem of section 3 will identify these missing values as being erroneous. We also assume that numerical THEN – conditions are only be triggered if none of the values of the variables involved may be missing. Hence, If – in a certain record – a THEN – condition involving a numerical variable of which the value is missing is triggered by the categorical values, then either the missing numerical value is erroneous or at least one of the categorical values.

2.2 Examples of Edits

Below we illustrate what kind of edits can be expressed in the form (2.1) by means of a number of examples.

1. $Turnover - Profit \geq 0.$ (2.2)

   This is an example of a numerical edit. For every combination of categorical values the edit should be satisfied. The edit can be formulated in our standard form as:

   IF $v_i \in D_i$ for all $i = 1, ..., m$
   THEN (*Profit*, *Turnover*) $\in$
   $\{(Profit, Turnover) \mid Turnover - Profit \geq 0\}.$ (2.3)

   In the remaining examples we will be slightly less formal with our notation. In particular, we will omit the terms "$v_i \in D_i$" from the edits.

2. IF (*Gender* = "*Male*")
   THEN (*Pregnant* = "*No*"). (2.4)

   This is an example of a categorical edit. It can be formulated in our standard form as:

   IF (*Gender* = "*Male*") AND (*Pregnant* = "*Yes*")
   THEN Ø. (2.5)

3. IF (*Occupation* = "*Statistician*")
   THEN (*Income* $\geq$ 1,000 Euro). (2.6)

   This is a typical example of a mixed edit. Given certain values for the categorical variables, a certain numerical constraint has to be satisfied.

4. IF (*Occupation* = "*Statistician*")
   OR (*Education* = "*University*")
   THEN (Income $\geq$ 1,000 Euro). (2.7)

   This edit can be split into two edits given by (2.6) and

   IF (*Education* = "*University*")
   THEN (*Income* $\geq$ 1,000 Euro). (2.8)

5. IF (*Tax on Wages* > 0)
   THEN (*Number of Employees* ≥ 1). (2.9)

Edit (2.9) is not in standard form (2.1), because the IF−condition involves a numerical variable. To handle such an edit, one can carry out a pre-processing step to introduce an additional categorical variable *TaxCond* with domain {"*False*", "*True*"}. Initially, *TaxCond* is given the value "*True*" if *Tax on Wages* > 0 in the unedited record, and the value "*False*" otherwise. The reliability weight *TaxCond* is set to zero. We can now replace (2.9) by the following three edits to type (2.1):

IF (*TaxCond* = "*False*")
THEN (*Tax on Wages* ≤ 0), (2.10)

IF (*TaxCond* = "*True*")
THEN (*Tax on Wages* ≥ ε), (2.11)

IF (*TaxCond* = "*True*")
THEN (*Number of Employees* ≥ 1), (2.12)

where ε is a sufficiently small positive number.

3. The Error Localization Problem as a Mixed Integer Programming Problem

We assume that the values of the numerical variables are bounded. That is, we assume that for the i^{th} numerical variable $(i = 1, ..., n)$ constants α_i and β_i exist such that

$$\alpha_i \leq x_i \leq \beta_i \tag{3.1}$$

for all consistent records. In practice, such values α_i and β_i always exist although they may be very large, because numerical variables that occur in data of statistical offices are bounded. The values of α_i and β_i may be negative. If the value of the i^{th} numerical variable is missing, we code this by assigning a value less than α_i or larger than β_i to x_i. Numerical variables for which the value should be missing, *e.g.*, because the corresponding question was non-applicable, will nonetheless receive a value after the termination of the algorithm that is described in subsequent sections, but this value may subsequently be ignored.

For the i^{th} categorical variable, let $D_i = \{c_{ik},\ k = 1, ..., g_i\}$ $(i = 1, ..., m)$ be its domain. We introduce the binary variable γ_{ik}

$$\gamma_{ik} = \begin{cases} 1 & \text{if the value of categorical variable } i \text{ equals } c_{ik} \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

To the i^{th} categorical variable there corresponds a vector $(\gamma_{i1}, ..., \gamma_{ig_i})$ such that $\gamma_{ik} = 1$ if and only if the value of this categorical variable equals c_{ik}, otherwise $\gamma_{ik} = 0$. For each categorical variable i of a consistent record the relation

$$\sum_k \gamma_{ik} = 1 \tag{3.3}$$

has to hold, *i.e.*, exactly one categorical value should be filled in. The vector $(\gamma_{i1}, ..., \gamma_{ig_i})$ will also be denoted by $\boldsymbol{\gamma}_{\mathbf{i}}$. If the value of the i^{th} categorical variable $(i = 1, ..., m)$ is missing, we set all γ_{ik} equal to zero $(k = 1, ..., g_i)$. In terms of the binary variables γ_{ik} an edit j given by (2.1) can be written as

$$a_{1j}x_1 + \ldots + a_{nj}x_n + b_j \geq M\left(\sum_{i=1}^{m}\left(\sum_{c_{ik} \in F_i^j} \gamma_{ik} - 1\right)\right), \tag{3.4}$$

where a positive M is chosen so that $-M$ is less than the lowest possible value of $a_{1j}x_1 + \ldots + a_{nj}x_n + b_j$. If the IF−condition of (2.1) and condition (3.3) hold true, the right-hand side of (3.4) equals zero. Consequently, the THEN−condition of (2.1) has to hold true for the numerical variables. If the IF−condition of (2.1) does not hold true, by (3.2) the right-hand side of (3.4) equals a large negative value. Consequently, (3.4) holds true irrespective of the values of numerical variables.

If (2.1) is not satisfied by a record $(v_1^0, ..., v_m^0, x_1^0, ..., x_n^0)$, or equivalently if (3.4) is not satisfied by $(\boldsymbol{\gamma}_{\mathbf{1}}^{\mathbf{0}}, ..., \boldsymbol{\gamma}_{\mathbf{m}}^{\mathbf{0}}, x_1^0, ..., x_n^0)$, then we seek values e_{ik}^P $(k = 1, ..., g_i;\ i = 1, ..., m)$, e_{ik}^N $(k = 1, ..., g_i;\ i = 1, ..., m)$, z_i^P $(i = 1, ..., n)$ and z_i^N $(i = 1, ..., n)$ that have to satisfy certain conditions mentioned below. The e_{ik}^P and the e_{ik}^N correspond to positive and negative changes, respectively, in the value of γ_{ik}^0. Likewise, the z_i^P and the z_i^N correspond to positive and negative changes, respectively, in the value of x_i^0. The vector $(e_{i1}^P, ..., e_{ig_i}^P)$ will also be denoted as $\mathbf{e}_{\mathbf{i}}^{\mathbf{P}}$ and the vector $(e_{i1}^N, ..., e_{ig_i}^N)$ as $\mathbf{e}_{\mathbf{i}}^{\mathbf{N}}$.

The objective function we consider in this paper is given by

$$\sum_{i=1}^{m} w_i^c \left(\sum_k e_{ik}^P\right) + \sum_{i=1}^{n} w_i^r \left(\delta(z_i^P) + \delta(z_i^N)\right), \tag{3.5}$$

where w_i^c is the reliability weight of the i^{th} categorical variable $(i = 1, ..., m)$, w_i^r the reliability weight of the i^{th} real-valued variable $(i = 1, ..., n)$, $\delta(x) = 1$ if $x \neq 0$ and $\delta(x) = 0$ otherwise. The objective function (3.5) is the sum of the reliability weights of the variables for which a new value must be imputed. Note that minimizing (3.5) is equivalent to minimizing

$$\sum_{i=1}^{m} w_i^c \left(\sum_k e_{ik}^N\right) + \sum_{i=1}^{n} w_i^r \left(\delta(z_i^P) + \delta(z_i^N)\right). \tag{3.6}$$

The objective function (3.6) is the sum of the reliability weights of the variables of which the original values must be modified. The value of the objective function (3.5) is equal to the value of the objective function (3.6) plus the sum of reliability weights of the categorical variables for which the original value was missing.

The objective function (3.5) is to be minimized subject to the following constraints:

$$e_{ik}^P,\ e_{ik}^N \in \{0,\ 1\}, \qquad (i = 1,\ ...,\ m) \qquad (3.7)$$

$$z_i^P,\ z_i^N \geq 0, \qquad (i = 1,\ ...,\ n) \qquad (3.8)$$

$$e_{ik}^P + e_{ik}^N \leq 1 \qquad (i = 1,\ ...,\ m) \qquad (3.9)$$

$$\sum_k e_{ik}^P \leq 1, \qquad (i = 1,\ ...,\ m) \qquad (3.10)$$

$$e_{ik}^N = 0 \quad \text{if } \gamma_{ik}^0 = 0 \qquad (i = 1,\ ...,\ m) \qquad (3.11)$$

$$\sum_k (\gamma_{ik}^0 + e_{ik}^P - e_{ik}^N) = 1, \qquad (i = 1,\ ...,\ m) \qquad (3.12)$$

$$\alpha_i \leq x_i^0 + z_i^P - z_i^N \leq \beta_i \qquad (i = 1,\ ...,\ n) \qquad (3.13)$$

and

$$\sum_{i=1}^{n} a_{ij}\,(x_i^0 + z_i^P - z_i^N) + b_j \geq M\left(\sum_{i=1}^{m}\left(\sum_{c_{ik} \in F_i^j} (\gamma_{ik}^0 + e_{ik}^P - e_{ik}^N) - 1\right)\right) \qquad (3.14)$$

for all edits $j = 1,\ ...,\ K$.

Relation (3.9) expresses that a negative correction and a positive one may not be applied to the same reported value of a categorical variable. Relation (3.10) expresses that at most one value may be imputed, *i.e.*, estimated and subsequently filled in, for a categorical variable, and relation (3.11) that a negative correction may not be applied to a categorical value that was not filled in. Relation (3.12) ensures that a value for each categorical variable is filled in, even if the original value was missing. Relation (3.13) states that the value of a numerical variable must be bounded by the appropriate constants. In particular, relation (3.13) also states that the value of a numerical variable may not be missing. Finally, relation (3.14) expresses that the modified record should satisfy all edits given by (2.1).

After solving this optimization problem the resulting, modified record is given by

$$(\boldsymbol{\gamma_1^0} + \mathbf{e_1^P} - \mathbf{e_1^N}, ..., \boldsymbol{\gamma_m^0} + \mathbf{e_m^P} - \mathbf{e_m^N}, x_1^0 + z_1^P - z_1^N,\ ...,\ x_n^0 + z_n^0 - z_n^N).$$

This modified record is consistent, *i.e.*, satisfies all edits. A solution to the above mathematical problem corresponds to a solution to the error localization problem, which simply consists of a list of variables of which the values have to be changed without specifying their new values. There may be several optimal solutions to the error localization problem. Our aim is to find all these optimal solutions. Note that the above optimization problem is a translation of the generalized Fellegi-Holt paradigm in mathematical terms.

We end this section with two remarks. First, note that in practice only one e_{ik}^N–variable for each variable i is needed, namely for the index k for which $\gamma_{ik}^0 = 1$. The other e_{ik}^N equal zero. In the present paper we use g_i binary e_{ik}^N–variables for each variable i to cover all possible cases. Second, note that in an optimal solution to the above optimization problem either $z_i^P = 0$ or $z_i^N = 0$, and that, similarly, in any feasible solution either $e_{ik}^P = 0$ or $e_{ik}^N = 0$ (or both).

4. Vertex Generation Methods and Error Localization for Mixed Data

In this section we explain how vertex generation methods can be used to solve the error localization problem in mixed data. To this end we show that a minimum of (3.5) subject to (3.7) to (3.14) is attained in a vertex of a certain polyhedron P described by linear, non-integer constraints. Suppose a minimum of (3.5) subject to (3.7) to (3.14) is attained in a point given by:

1. $e_{ik}^N = 0$ for $(i,\ k) \in I_e^N$, $e_{ik}^N = 1$ otherwise,
2. $e_{ik}^P = 0$ for $(i,\ k) \in I_e^P$, $e_{ik}^P = 1$ otherwise,
3. $z_i^P = 0$ for $i \in I_z^P$, $z_i^P \neq 0$ otherwise,
4. $z_i^N = 0$ for $i \in I_z^N$, and $z_i^N \neq 0$ otherwise,

for certain index sets I_e^N, I_e^P, I_z^N and I_z^P. We now consider the problem of minimizing the linear function given by

$$\sum_{(i,k) \in I_e^N} e_{ik}^N + \sum_{(i,k) \notin I_e^N} (1 - e_{ik}^N) + \sum_{(i,k) \in I_e^P} e_{ik}^P + \sum_{(i,k) \notin I_e^P} (1 - e_{ik}^P) + \sum_{i \in I_z^P} z_i^P + \sum_{i \in I_z^N} z_i^N \qquad (4.1)$$

subject to (3.8) to (3.14) and

$$0 \leq e_{ik}^N,\ e_{ik}^P \leq 1. \qquad (4.2)$$

Subject to (3.8) to (3.14) and (4.2), which together form our polyhedron P, the function (4.1) is non-negative. Moreover, its value equals zero only for the point given by 1 to 4 above. In other words, our selected minimum of (3.5) subject to (3.7) to (3.14) is also the minimum of (4.1) subject to (3.8) to (3.14) and (4.2).

It is well known that a linear function subject to a set of linear constraints attains its minimum, if such a minimum exists, in a vertex of the feasible polyhedron described by the set of linear constraints (see *e.g.*, Chvátal 1983). So, the minimum of (4.1) subject to (3.8) to (3.14) and (4.2), zero, is attained in a vertex of the feasible polyhedron P described by (3.8) to (3.14) and (4.2). We conclude that the point given by 1 to 4 above, *i.e.*, an arbitrary optimum of (3.5) subject to (3.7) to (3.14), is a vertex of the polyhedron defined by (3.8) to (3.14) and (4.2).

The above observation implies that the minimum of (3.5) subject to (3.7) to (3.14) can be found by generating all vertices of the polyhedron given by (3.8) to (3.14) and (4.2). From these vertices we select the vertices that satisfy (3.7). From those latter vertices we subsequently select the vertices for which the value of the objective function (3.5) is minimal. These vertices correspond to the optimal solutions to the error localization problem.

5. Chernikova's Algorithm and the Error Localization Problem

Chernikova's algorithm (Chernikova 1964 and 1965) was designed to generate the edges of a system of linear inequalities given by

$$\mathbf{Cx} \geq \mathbf{0} \tag{5.1}$$

and

$$\mathbf{x} \geq \mathbf{0}, \tag{5.2}$$

where $\mathbf{C}$ is a constant $n_r \times n_c -$matrix and $\mathbf{x}$ an $n_c -$dimensional vector of unknowns. The algorithm is described in the Appendix. It can be used to find the vertices of a system of linear inequalities because of the following lemma (see Rubin 1975 and 1977).

Lemma 5.1. *The vector* $\mathbf{x^0}$ *is a vertex of the system of linear inequalities*

$$\mathbf{Ax} \leq \mathbf{b} \tag{5.3}$$

and

$$\mathbf{x} \geq \mathbf{0} \tag{5.4}$$

if and only if $\{(\lambda \mathbf{x^0} \mid \lambda)^T,\ \lambda \geq 0\}$ *is an edge of the cone described by*

$$(-\mathbf{A} \mid \mathbf{b}) \begin{pmatrix} \mathbf{x} \\ \xi \end{pmatrix} \geq \mathbf{0} \tag{5.5}$$

and

$$\begin{pmatrix} \mathbf{x} \\ \xi \end{pmatrix} \geq \mathbf{0}. \tag{5.6}$$

Here $\mathbf{A}$ *is an* $n_r \times n_c -$*matrix,* $\boldsymbol{b}$ *an* $n_r -$*vector,* $\mathbf{x}$ *an* $n_v -$*vector, and* ξ *and* λ *scalar variables.*

For notational convenience we write

$$n_c = n_v + 1 \tag{5.7}$$

throughout this paper. The matrix in (5.5) is then an $n_r \times n_c -$matrix just like in (5.1), so we can use the same notation as in Rubin's formulation of Chernikova's algorithm.

If Chernikova's algorithm is used to determine the edges of (5.5) and (5.6), then after the termination of the algorithm the vertices of (5.3) and (5.4) correspond to those columns j of $\mathbf{L^{n_r}}$ (see Appendix) for which $l^{n_r}_{n_c,\,j} \neq 0$. The entries of such a vertex $\mathbf{x}'$ are given by

$$x'_i = l^{n_r}_{ij} / l^{n_r}_{n_c,\,j} \quad \text{for} \quad i = 1, \ldots, n_v. \tag{5.8}$$

Now, we explain how Chernikova's algorithm can be used to solve the error localization problem in mixed data. The set of constraints (3.8) to (3.14) and (4.2) can be written in the form (5.3) and (5.4). We can find the vertices of the polyhedron corresponding to this set of constraints by applying Chernikova's algorithm to (5.5) and (5.6). Vertices of the polyhedron defined by (3.8) to (3.14) and (4.2) are given by columns $y^{n_r}_{*s}$ for which $u^{n_r}_{is} \geq 0$ for all i and $l^{n_r}_{n_c,\,s} > 0$, where n_c is the number of rows of the final matrix $\mathbf{L^{n_r}}$ (see Appendix). In our case, n_c equals the total number of variables z_i^P, z_i^N, e_{ik}^P and e_{ik}^N plus one (corresponding to ξ in (5.5) and (5.6)), *i.e.*, $n_c = 2n + 2G + 1$, where $G = \sum_i g_i$. The values of the variables z_i^P, z_i^N, e_{ik}^P and e_{ik}^N in such a vertex are given by the corresponding values $l^{n_r}_{js} / l^{n_r}_{n_c,\,s}$.

Two technical problems must be overcome when Chernikova's algorithm is applied to solve the error localization problem for mixed data. First, the algorithm must be sufficiently fast. Second, the solution found must be feasible for the error localization problem for mixed data, *i.e.*, the values of the variables e_{ik}^P and e_{ik}^N must be either 0 or 1. Both problems can be overcome by removing certain "undesirable" columns from the current matrix $\mathbf{Y^k}$, *i.e.*, by deleting columns that cannot yield an optimal solution to the error localization problem. That such undesirable columns may indeed be removed from the current matrix $\mathbf{Y^k}$ is essentially demonstrated by Rubin (1975 and 1977). We state this result as Theorem 5.1.

Theorem 5.1. *Columns that cannot yield an optimal solution to the error localization problem because they contain too many non-zero entries may be removed from an intermediate matrix.*

To accelerate Chernikova's algorithm, we aim to limit the number of vertices that are generated as much as possible. Once we have found a (possibly suboptimal) solution to the error localization problem for which the objective value (3.5) equals η, say, we from then on look only for vertices corresponding to solutions with an objective value at most equal to η. A minor technical problem is that we cannot use the objective function (3.5) directly when applying Chernikova's algorithm, because the values of e_{ik}^P, e_{ik}^N, z_i^N and z_i^P are not known during the execution of this algorithm. Therefore, we introduce a new objective function that associates a value to each column of the matrix $\mathbf{Y^k}$ (see Appendix). Assume that the first G entries of a column l^k_{*s} of $\mathbf{L^k}$ correspond to the $e_{ik}^P -$variables, the next G entries to the $e_{ik}^N -$variables, the next n entries to the $z_i^P -$variables, and the subsequent n entries to the $z_i^N -$variables. We define the following objective function

$$\sum_{i=1}^{m} w_i^c \left(\sum_{k=1}^{g_i} \delta(l_{t,s}^k) \right) + \sum_{i=1}^{n} w_i^r \times (\delta(l_{2G+i,s}^k) + \delta(l_{2G+n+i,s}^k)), \quad (5.9)$$

where $t = \sum_{l=1}^{i-1} g_l + r$ for each pair $\{i, r\}$ ($i = 1, ..., m$; $r = 1, ..., g_i$). Differences between (3.5) and (5.9) are that for each e_{ik}^P or e_{ik}^N in (3.5) several variables $l_{t,s}^k$ occur in (5.9), and that the e_{ik}^P and e_{ik}^N attain values in $\{0,1\}$ whereas the $l_{t,s}^k$ can attain any value between zero and one. If column y_{*s}^k of $\mathbf{Y^k}$ corresponds to a solution to the error localization problem, then the value of the objective function (5.9) for y_{*s}^k equals the value of the objective function (3.5) for this solution. This implies that we can use the objective function (5.9) to update the value of η.

The computing time of Chernikova's algorithm can be further reduced by noting that in an optimal solution to the error localization problem either $z_i^P = 0$ or $z_i^N = 0$ (or both). This implies that in Step 7 of Chernikova's algorithm (see Appendix) columns y_{*s}^k and y_{*t}^k need not be combined if one of these columns corresponds to $z_i^P \neq 0$ and the other to $z_i^N \neq 0$. Theorem 5.1 implies that not combining such columns is allowed.

We now consider the problem of constructing a feasible solution to the error localization problem for mixed data. This problem can, of course, be solved by first generating vertices without taking into account that values of e_{ik}^P and e_{ik}^N must be either 0 or 1 and then selecting the best vertices that possess this property, but this rather inefficient so we suggest a different approach. It suffices to ensure that for each variable i ($i = 1, ..., m$) at most one e_{ik}^P differs from zero, and that the e_{ik}^N and e_{ik}^P equal either zero or one after the termination of the algorithm. We can ensure that for each i at most one e_{ik}^P differs from zero in the following way. If in Step 7 of Chernikova's algorithm the entry of y_{*s}^k corresponding to $e_{ik_1}^P$ differs from zero and the entry of y_{*t}^k corresponding to $e_{ik_2}^P$ ($k_2 \neq k_1$) differs from zero as well, then columns y_{*s}^k and y_{*t}^k are not combined to generate a new column. We can also ensure that the e_{ik}^N equal either zero or one after the termination of the algorithm. For each i this is a problem only for the unique $e_{ik_0}^N$ for which $\gamma_{ik_0}^0 = 1$. We introduce variables $\tilde{e}_i$ that can attain values between zero and one. These variables have to satisfy

$$e_{ik_0}^N + \tilde{e}_i = 1. \quad (5.10)$$

Relation (5.10) is treated as a constraint for the values of the variables $e_{ik_0}^N$ and $\tilde{e}_i$. Because the value of $e_{ik_0}^N$ has to be either zero or one, we demand that either $e_{ik_0}^N = 0$ or $\tilde{e}_i = 0$. This can be ensured in the same manner as for the z_i^p and the z_i^N. Finally, we have to ensure that the e_{ik}^P equal either zero or one after the termination of the algorithm. This is automatically the case if for each i at most one e_{ik}^P differs from zero, at most one e_{ik}^N equals one and the remaining e_{ik}^N equal zero, because relation (3.12) has to hold true. We have already ensured that these conditions are satisfied, so all e_{ik}^P equal zero or one after the termination of the algorithm. With the adaptations described above Chernikova's algorithm can be applied to solve the error localization problem in mixed data. Theorem 5.1 again implies that these modifications are allowed.

6. Adapting Chernikova's Algorithm to the Error Localization Problem

6.1 Advanced Adaptations

In this section we consider more advanced adaptations of Chernikova's algorithm in order to make the algorithm better suited for solving the error localization problem. Sande (1978) notes that when two columns in the initial matrix $\mathbf{Y^0}$ have exactly the same entries in the upper matrix $\mathbf{U^0}$, they will be treated exactly the same in the algorithm. The two columns are always combined with the same other columns, and never with each other. Keeping both columns in the matrix only makes the problem unnecessarily bigger. One of the columns may therefore be temporarily deleted. After the termination of the algorithm, the solutions to the error localization problem involving the temporarily deleted column can easily be generated.

A correction patter associated with column y_{*s}^k in an intermediate matrix $\mathbf{Y^k}$, where $\mathbf{Y^k}$ can be split into an upper matrix $\mathbf{U^k}$ and lower matrix $\mathbf{L^k}$ with n_r and n_c rows respectively (see Appendix), is defined as the n_c – dimensional vector with entries $\delta(y_{js}^k)$ for $n_r < j \leq n_r + n_c$. For each z_i^P, z_i^N, e_{ik}^P, and e_{ik}^N a correction pattern contains an entry with value in $\{0, 1\}$. Sande (1978) notes that Theorem 5.1 implies that once a vertex has been found, all columns with correction patterns with ones on the same places as in the correction patter of this vertex can be removed.

The concept of correction patterns has been improved upon by Fillion and Schiopu-Kratina (1993), who note that it is not important how the value of a variable is changed, but only whether the value of a variable is changed or not. A *generalized correction pattern* associated with column y_{*s}^k in an intermediate matrix $\mathbf{Y^k}$ is defined as the $(m+n)$ – dimensional vector of which the j^{th} entry equals 1 if and only if an entry corresponding to the j^{th} variable in column y_{*s}^k is different from 0, and 0 otherwise. Here m denotes the number of categorical variables and n the number of numerical variables. For each variable involved in the error localization problem, a generalized correction pattern contains an entry with value in $\{0, 1\}$. Again Theorem 5.1 implies that once a vertex has been found, all columns with generalized correction patterns with ones on the same places as in the generalized correction pattern of this vertex can be deleted.

Fillion and Schiopu-Kratina (1993) define a *failed row* as a row that contains at least one negative entry placed on a column of which the last entry is non-zero. They note that in order to solve the error localization problem we can already terminate Chernikova's algorithm as soon as all failed rows have been processed. This result is stated as Theorem 6.1.

Theorem 6.1. *If an intermediate matrix contains no failed rows, then all* (*generalized*) *patterns corresponding to vertices for which* (5.9) *is minimal have been found.*

The final adaption of Fillion and Schiopu-Kratina (1993) to Chernikova's algorithm is a method to speed-up the algorithm in case of missing values. Suppose the error localization problem has to be solved for a record with missing values. For each numerical variable of which the value is missing we first fill in an arbitrary value, say zero. Next, only the entries corresponding to variables with non-missing values are taken into account when calculating the value of function (5.9) for a column. An optimal solution to the error localization problem is given by the variables corresponding to a determined optimal generalized correction pattern plus the variables with missing values. In this way, unnecessary generalized correction patterns according to which many variables with non-missing values should be changed are discarded earlier than in the standard algorithm.

6.2 Duffin's Rules

Chernikova's algorithm does not generate any redundant columns, *i.e.*, columns whose information is already contained in another column. Its problem is, however, that in order to achieve this the algorithm requires a considerable amount of computing time. This is for a substantial part caused by its Step 7 where a time-consuming check has to be performed to prevent the generation of redundant columns. Duffin (1974) demonstrates that this step can be split into two parts. In Duffin's version of the algorithm Step 7 consists of two parts:

- For each pair (s, t) for which $y^k_{rs} \times y^k_{rt} < 0$ we choose $\mu_1, \mu_2 > 0$ such that $\mu_1 y^k_{rs} + \mu_2 y^k_{rt} = 0$ and adjoin the column $\mu_1 y^k_{*s} + \mu_2 y^k_{*t}$ to $\mathbf{Y^{k+1}}$.
- Delete (some of) the redundant columns of $\mathbf{Y^{k+1}}$.

Duffin (1974) gives the following two rules to delete redundant columns of $\mathbf{Y^{k+1}}$.

Refined elimination rule: When t rows have been processed, delete any columns that have been generated by combining $t+2$ or more original columns.

This first rule allows the generation of redundant columns, but is much faster to apply than Step 7 of Chernikova's algorithm. The second rule, the dominance rule, makes sure that no redundant columns are generated. A column y^k_{*u} is called dominated by another column y^k_{*v} if $y^k_{iv} = 0$ implies $y^k_{iu} = 0$.

Dominance rule: Delete any column y^k_{*u} in $\mathbf{Y^k}$ that is dominated by some other column y^k_{*v}.

One could consider using the refined elimination rule during most iterations of Chernikova's algorithm and only resort to the dominance rule when the number of columns becomes too high to be handled efficiently. After all failed rows have been processed the dominance rule has to be applied to remove redundant columns from the final matrix $\mathbf{Y^k}$. One may hope that this leads to an algorithm that is faster than Chernikova's algorithm, but this remains to be tested.

7. Discussion

At Statistics Netherlands a prototype computer program based on the adapted version of Chernikova's algorithm described in section 5 and 6.1 of the present paper has been developed. The possibly more efficient rules described in section 6.2 have not been implemented in this prototype program. For purely numerical data a production version of this program has been used for several years in the day-to-day routine at Statistics Netherlands in order to produce clean data for most of our structural business statistics.

For Statistics Netherlands improving the efficiency of the data editing process for economic, and hence mainly numerical, data is much more important than for social, and hence mainly categorical, data. In particular, edits of type 1 (see *e.g.*, (2.2)) mentioned in section 2.2 are the most important ones for us, followed by edits of type 5 (see *e.g.*, (2.9)). Because improving the efficiency of data editing for numerical data is much more important to us than for social data, the developed prototype program has only been evaluated for purely numerical test data. For these numerical test data, the program has been compared to several other prototype programs, namely a program based on a standard mixed integer programming problem formulation (see *e.g.*, De Waal 2003), a program based on cutting planes (see Garfinkel, Kunnathur and Liepins 1988; Ragsdale and McKeown 1996, and De Waal 2003), and a program based on a branch-and-bound algorithm (see *e.g.*, De Waal and Quere 2003). Our evaluation results show that the computing speed of our program based on the adapted version of Chernikova's algorithm is acceptable in comparison to other algorithms (for details on our evaluation experiments we refer to De Waal 2003). They also show, however , that this program is out-performed by the program based on the branch-and-bound algorithm. Besides being faster than the adapted version of Chernikova's algorithm, the branch-and-bound algorithm is less complex, and hence easier to maintain.

Further improvements to the adapted version of Chernikova's algorithm may reduce its computing time. Examples of such potential improvements are: better selection criteria for the row to be processed, and better

ways to handle missing values. However, these improvement would at the same time increase the complexity of the algorithm, thereby making it virtually impossible for software-engineers at Statistics Netherlands to maintain the program. For the above reasons, computing time for numerical data and complexity of the algorithm, we recently decided to switch to the branch-and-bound algorithm instead of the adapted version of Chernikova's algorithm for our production software. In our latest version of our production software, a version of the branch-and-bound algorithm suitable for a mix of categorical, continuous, and integer data has been implemented. We sincerely hope, however, that the present paper will inspire some readers to find further improvements to Chernikova's algorithm.

Acknowledgements

The views expressed in this paper are those of the author and do not necessarily reflect the policies of Statistics Netherlands. The author would like to thank two anonymous referees for their useful comments.

Appendix: Chernikova's Algorithm

Rubin's formulation (Rubin 1975 and 1977) of Chernikova's algorithm is as follows:

1. Construct the $(n_r + n_c) \times n_c -$matrix

$$\mathbf{Y^0} = \begin{pmatrix} \mathbf{U^0} \\ \mathbf{L^0} \end{pmatrix},$$

where $\mathbf{U^0} = \mathbf{C}$ and $\mathbf{L^0} = \mathbf{I_{n_c}}$: the $n_c \times n_c -$identity matrix. The j^{th} column of $\mathbf{Y^0}$, y^0_{*j}, will also be denoted as

$$y^0_{*j} = \begin{pmatrix} u^0_{*j} \\ l^0_{*j} \end{pmatrix},$$

where u^0_{*j} and l^0_{*j} are the j^{th} columns of $\mathbf{U^0}$ and $\mathbf{L^0}$, respectively.

2. $k := 0$
3. If any row of $\mathbf{U^k}$ has all components negative, $\mathbf{x} = \mathbf{0}$ is the only point satisfying (5.1) and (5.2), and the algorithm terminates.
4. If all the elements of $\mathbf{U^k}$ are non-negative, the columns of $\mathbf{L^k}$ are the edges of the cone described by (5.1) and (5.2), and the algorithm terminates.
5. If neither 3 nor 4 holds: choose a row of $\mathbf{U^k}$, say row r, with at least one negative entry.
6. Let $R = \{ j \mid y^k_{rj} \geq 0 \}$. Let v be the number of elements in R. Then the first v columns of the new matrix $\mathbf{Y^{k+1}}$ are all the columns y^k_{*j} of $\mathbf{Y^k}$ for $j \in R$.
7. Examine the matrix $\mathbf{Y^k}$.
   a. If $\mathbf{Y^k}$ has only two columns and $y^k_{r1} \times y^k_{r2} < 0$, then choose $\mu_1, \mu_2 > 0$ such that $\mu_1 y^k_{r1} + \mu_2 y^k_{r2} = 0$. Adjoin the column $\mu_1 y^k_{*1} + \mu_2 y^k_{*2}$ to $\mathbf{Y^{k+1}}$. Go to Step 9.
   b. If $\mathbf{Y^k}$ has more than two columns then let $S = \{(s, t) \mid y^k_{rs} \times y^k_{rt} < 0 \text{ and } t > s\}$, *i.e.*, let S be the set of all pairs of columns of $\mathbf{Y^k}$ whose elements in row r have opposite signs. Let I_0 be the index set of all non-negative rows of $\mathbf{Y^k}$, *i.e.*, all rows of $\mathbf{Y^k}$ with only non-negative entries. For each $(s, t) \in S$, find all $i \in I_0$ such that $y^k_{is} = y^k_{it} = 0$. Call this set $I_1(s, t)$.
      – If $I_1(s, t) = \emptyset$, then y^k_{*s} and y^k_{*t} do not contribute another column to the new matrix.
      – If $I_1(s, t) = \emptyset$, check to see if there is a v not equal to s or t such that $y^k_{iv} = 0$ for all $i \in I_1(s, t)$. If such a v exists, then y^k_{*s} and y^k_{*t} do not contribute a column to the new matrix. If no such v exists, then choose $\mu_1, \mu_2 > 0$ such that $\mu_1 y^k_{rs} + \mu_2 y^k_{rt} = 0$. Adjoin the column $\mu_1 y^k_{*s} + \alpha_2 y^k_{*t}$ to $\mathbf{Y^{k+1}}$.
8. When all pairs in S have been examined, and the additional columns (if any) have been added, we say that row r has been processed. We then define matrices $\mathbf{U^{k+1}}$ and $\mathbf{L^{k+1}}$ by

$$\mathbf{Y^{k+1}} = \begin{pmatrix} \mathbf{U^{k+1}} \\ \mathbf{L^{k+1}} \end{pmatrix},$$

where $\mathbf{U^{k+1}}$ is a matrix with n_r rows and $\mathbf{L^{k+1}}$ a matrix with n_c rows. The j^{th} column of $\mathbf{Y^{k+1}}$, y^{k+1}_{*j}, will also be denoted as

$$y^{k+1}_{*j} = \begin{pmatrix} u^{k+1}_{*j} \\ l^{k+1}_{*j} \end{pmatrix},$$

where u^{k+1}_{*j} and l^{k+1}_{*j} are the j^{th} columns of $\mathbf{U^{k+1}}$ and $\mathbf{L^{k+1}}$, respectively.

9. $k := k + 1$, and go to Step 3.

Chernikova's algorithm can be modified in order to handle equalities more efficiently than treating them as two inequalities. Steps 3, 5 and 6 should be replaced by

3. If any row of $\mathbf{U^k}$ corresponding to an inequality or equality has all components negative or if any row of $\mathbf{U^k}$ corresponding to an equality has all components positive, $\mathbf{x} = 0$ is the only point satisfying (5.1) and (5.2), and the algorithm terminates.
5. If neither 3 nor 4 holds: choose a row of $\mathbf{U^k}$, say row r, with at least one negative entry if the row corresponds to an inequality, and with at least one non-zero entry if the row corresponds to an equality.
6. If row r corresponds to an inequality, then apply Step 6 of the standard algorithm. If row r corresponds to an equality then let $R = \{ j \mid y^k_{rj} = 0 \}$. Let v be the number of elements in R. Then the first v columns of the new matrix $\mathbf{Y^{k+1}}$ are all the columns y^k_{*j} of $\mathbf{Y^k}$ for $j \in R$.

In Step 5 of Chernikova's algorithm a failed row has to be chosen. Rubin (1975) proposes the following simple rule. Suppose a failed row has z entries equal to zero, p positive entries, and q negative ones. We then calculate for each failed row the value $N_{\max} = z + p + pq$ if the row corresponds to an inequality and the value $N_{\max} = z + pq$ if the row corresponds to an equality, and choose a failed row with the lowest value of $N_{\max}$.

References

Central Statistical Office (2000). Editing and calibration in survey processing. Report SMD-37, Ireland.

Chernikova, N.V. (1964). Algorithm for finding a general formula for the non-negative solutions of a system of linear equations. *USSR Computational Mathematics and Mathematical Physics*, 4, 151-158.

Chernikova, N.V. (1965). Algorithm for finding a general formula for the non-negative solutions of a system of linear inequalities. *USSR Computational Mathematics and Mathematical Physics*, 5, 228-233.

Chvátal, V. (1983). *Linear Programming.* W.H. Freeman and Company.

De Waal, T. (1996). CherryPi: A computer program for automatic edit and imputation. Paper presented at the UN/ECE work session on statistical data editing, Voorburg.

De Waal, T. (2003). Processing of erroneous and unsafe data. Ph.D. Thesis, Erasmus University Rotterdam.

De Waal, T., and Quere, R. (2003). A fast and simple algorithm for automatic editing of mixed data. Paper submitted to *Journal of Official Statistics.*

Duffin, R.J. (1974). On Fourier's analysis of linear inequality systems. *Mathematical Programming Study*. North-Holland Publishing Company. 1, 71-97.

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

Fillion, J.M., and Schiopu-Kratina, I. (1993). On the Use of Chernikova's Algorithm for Error Localization. Report, Statistics Canada.

Garfinkel, R.S., Kunnathur, A.S. and Liepins, G.E. (1988). Error localization for erroneous data: Continuous data, linear constraints. *SIAM Journal on Scientific and Statistical Computing*, 9, 922-931.

Kovar, J., and Whitridge, P. (1990). Generalized edit and imputation system. Overview and applications. *Revista Brasileira de Estadistica*, 51, 85-100.

Ragsdale, C.T., and Mckeown, P.G. (1996). On solving the continuous data editing problem. *Computers & Operations Research*, 23, 263-273.

Rubin, D.S. (1975). Vertex generation and cardinality constrained linear programs. *Operations Research*, 23, 555-565.

Rubin, D.S. (1977). Vertex generation methods for problems with logical constraints. *Annals of Discrete Mathematics*, 1, 457-466.

SANDE, G. (1978). An algorithm for the fields to impute problems of numerical and coded data. Report, Statistics Canada.

Schiopu-Kratina, I., and Kovar, J.G. (1989). Use of Chernikova's algorithm in the generalized edit and imputation system. Report, Statistics Canada.

Todaro, T.A. (1999). Overview and evaluation of the AGGIES automated edit and imputation system. Paper presented at the UN/ECE work session on statistical data editing, Rome.