IMPUTING NUMERIC AND QUALITATIVE VARIABLES SIMULTANEOUSLY

Michael Bankier, Manchi Luc, Christian Nadeau and Pat Newcombe
Michael Bankier, 15Q R.H. Coats Bldg., Statistics Canada, Ottawa, Ontario K1A 0T6, Canada

KEY WORDS: Minimum Change Hot Deck Imputation, Nonresponse, Inconsistent Responses.

1. Introduction

Many minimum change hot deck imputation systems are based on the imputation methodology proposed by Fellegi and Holt (1976). For example, CANEDIT and GEIS at Statistics Canada and DISCRETE and SPEER at USBC are based on the Fellegi and Holt imputation methodology.

A New Imputation Methodology (NIM) will be used in the 1996 Canadian Census to carry out Edit and Imputation (E&I) for the variables age, sex, marital status, common-law status and relationship. A typical edit and imputation (E&I) problem is displayed in Table 1 for a 6 person failed edit household (only the first three people are displayed).

Table 1:Failed Edit Household

Relationship	Marital Status	Age
Person 1	Married	38
Spouse	Married	35
Mother	Blank	41

In the Table 1, there is a blank response for marital status, and the age of the mother is inconsistent with the age of her son (Person 1). Data borrowed from a household which passed the edits, is used to impute (see Table 2) a marital status of widowed for the mother plus increase her age to 59.

Table 2:Imputed Household

Relationship	Marital Status	Age
Person 1	Married	38
Spouse	Married	35
Mother	Widowed	59

The NIM allows, for the first time, the simultaneous hot deck imputation of qualitative and continuous or discrete numeric variables for large E&I problems.

The Fellegi and Holt algorithm first determines the minimum number of variables to impute and then performs the imputation, possibly by searching for donors. The NIM, in contrast, first searches for donors and then determines the minimum number of variables to impute. Changing the order of these operations allows the NIM to solve larger and more complex E&I problems. The NIM does require donors, however, to be able to carry out imputation.

In this paper, the relatively simple algorithms used to implement the NIM in a computationally efficient way will be illustrated using the above example.

Section 2 gives the objectives and an overview of the NIM. Section 3 provides a simple example illustrating the NIM. Section 4 compares the NIM to the Fellegi/Holt algorithm. Section 5 gives additional details of the NIM. Section 6 shows how to evaluate imputation actions efficiently. Section 7 provides some concluding remarks.

Additional details on the NIM methodology are given in Bankier, Fillion, Luc and Nadeau (1994) and Bankier, Luc, Nadeau and Newcombe (1995). A technical report is available from the authors if the reader would like more information.

2. Objectives and Overview of the NIM

Based on the discussion in the 1994 paper, the objectives for an automated hot deck imputation methodology should be as follows:

(a) The imputed household should closely resemble the failed edit household. This is achieved, given the donors available, by imputing the minimum number of variables in some sense. The underlying assumption (which is not always true in practice) is that a respondent is more likely to make only one or two errors rather than several.

(b) The imputed data for a household should come from a single donor, if possible, rather than two or more donors. In addition, the imputed household should closely resemble that single donor. Achieving these objectives will tend to ensure that the combination of imputed and unimputed responses for the imputed household is plausible.

(c) Equally good imputation actions, based on the available donors, should have a similar chance of being selected to avoid falsely inflating the size of small but important groups in the population (e.g. persons whose age is over 100).

These objectives are achieved under the NIM by first identifying as potential donors those passed edit households which are as similar as possible to the failed edit household. By this it is meant that the two households should match on as many of the qualitative variables as possible while having small differences between the numeric variables.

Households with these characteristics will be called close to each other or nearest neighbours. (A term will be underlined when it is first defined.) Then, for each nearest neighbour, the smallest subsets of the non-matching variables (both numeric and qualitative) which, if imputed, allow the imputed household to pass the edits, are identified. An imputation action which passes the edits will be called feasible. One of these feasible imputation actions which imputes the smallest number of variables possible (which will be called a near minimum change imputation action) is randomly selected. As a result, the imputed household will be as similar as possible to the failed edit household while closely resembling the donor.

These near minimum change imputation actions can be identified efficiently for each nearest neighbour being considered as a donor for the failed edit household as follows:

(a) Only edit rules that one of the possible imputation actions can fail are retained for each failed household/nearest neighbour pair. This results in many fewer edit rules being needed to evaluate the imputation actions.

(b) Variables most likely to need imputation are considered first. Thus, blanks/invalids are imputed first followed by variables which enter the edits that the household failed and finally the other variables.

(c) When generating imputation actions for a failed edit household/nearest neighbour pair, only those which are

- near the optimum (i.e. are near minimum change)
- and are essentially new (i.e. no subset of the variables being imputed would pass the edits)

are evaluated for feasibility. Imputation actions that are not essentially new are discarded because one or more variables is being unnecessarily imputed. This violates the principle of making as little change to the data as possible.

Some of the concepts in this section are defined more precisely in Section 5 and in Bankier et al (1995).

3. An Example Illustrating the NIM Algorithm

The failed edit household displayed in Table 1 will be used to illustrate the NIM algorithm. The Table 1 household matches and hence fails the edit rule in the leftmost column of the Table 3 decision logic table (DLT), i.e. Person 3 is the mother of Person 1 (Relat(3) = Mother) but the age difference between the mother and Person 1 is less than 15 years (Age(3) - Age(1) < 15).

A search among the passed edit households is done to identify the nearest neighbours to the Table 1 household. Preference, if possible, is given to those households which are geographically close. One of these nearest neighbours is listed in Table 4 below. The five responses in Table 4 that do not match the responses of Table 1 are underlined. The distance between failed edit household and the nearest neighbour (which is a measure of the number of non-matching variables) is 3 + 0.1 +0.1 = 3.2. The two 0.1 terms are for the two ages that differ by 2 years (and hence are near matches) while the count of three is for the other three variables that do not match closely. More information on the distance measure is given in Section 5 and Bankier et al (1995).

Table 3: Decision Logic Table of Edit Rules

Relat(3) = Mother	Y	Y	-	-
Age(3) - Age(1) < 15	Y	-	-	-
Age(3) < 30	-	Y	-	-
Relat(3) = Grandmother	-	-	Y	Y
Age(3) - Age(1) < 30	-	-	Y	-
Age(3) < 45	-	-	-	Y

Table 4: Nearest Neighbour to Table 1 Household

Relationship	Marital Status	Age
Person 1	Married	36
Spouse	Married	37
Mother-In-Law	Widowed	59

Most edit rules in Table 3 can be discarded since they will never be failed by any of the imputation actions generated by the Table 1 failed edit household and the Table 4 nearest neighbour. For example, person 3 is age 41 in Table 1 and age 59 in Table 4. Hence the third proposition (Age(3) < 30) will never be true and hence the second edit rule can be discarded. Similarly, neither Table 1 or Table 4 have any grandmothers present and thus the third and fourth edit rules of Table 3 can be discarded. Thus the only Table 3 edit rule remaining is the one failed by the Table 1 household. When this process is repeated with all six person household edits (240 edits in 62 DLTs), only the two edit rules (see Table 5) remain. See Bankier et al (1995) for more information on this simplification process.

Table 5: Edit Rules Remaining After Simplification

Relat(3) = Mother	Y	-
Age(3) - Age(1) < 15	Y	-
Relat(3)=Mother-in-Law	-	Y
Age(3) - Age(2) < 15	-	Y

Any blank/invalid responses will be imputed. Thus any edit rules forbidding blank or invalid responses are not listed in Table 5. The $2^4 - 1 = 15$ imputation actions based on the four variables (Relat(3), Age(1), Age(2) and Age(3)) which enter the two edits of Table 5 will be evaluated.

Imputation actions based on the three variables (Relat(3), Age(1) and Age(3)) which enter the edit rule that the Table 1 household failed will be evaluated first. Imputing Relat(3) or Age(1) alone or as a pair is not sufficient for the Table 1 household to pass the edits but imputing Age(3) is. Imputations actions (there are 7 of them) which involve imputing Age(3) along with one or more of Relat(3), Age(1) and Age(2) are immediately discarded since they are not essentially new.

Then the variable Age(2) is introduced. It is assessed whether imputing it alone or with Relat(3) and/or Age(1) is sufficient for the Table 1 household to pass the edits. None of these imputation actions pass the edits. Thus, of 15 possible imputation actions, only the one which involves imputing Age(3) is retained. This generates the imputation action displayed in Table 2.

This process of identifying imputation actions is repeated with a number of other nearest neighbour households. Let D_{fa} represent the distance from the imputation action to the failed edit household (i.e. a measure of how many variables are imputed). Let D_{ap} represent the distance of the imputation action to the nearest neighbour used (i.e. a measure of plausibility). The five imputation actions with the smallest D_{fpa} are retained where

$$D_{fpa} = \alpha D_{fa} + (1-\alpha) D_{ap}$$

The parameter α (which can fall in the range (0.5, 1]) is often set to 0.75 or 0.9 to place more importance on imputing the minimum number of variables. Then one of these five imputation actions is randomly selected to be the actual imputation action used for the failed edit household.

4. Comparison of NIM and Fellegi/Holt

In previous Censuses, CANEDIT, an implementation of the Fellegi/Holt algorithm, was used to do E&I. The NIM and CANEDIT imputation actions were compared for 12,000 failed edit households. Approximately 98% had the same number of variables imputed. The majority of the remaining variables had one additional variable imputed by the NIM because of the more rigorous NIM edits based on age rather than decade. CANEDIT used decade rather than age in the edits because the computational costs were otherwise too large.

In a few cases, the NIM will impute more than the minimum number of variables if this results in a more plausible imputation action. This is illustrated in Table 6 below. The household fails the edit that there should be at least a 15 year age difference between the parent and the child. The CANEDIT imputation increases the age of Person 1 from 35 to 45 by changing the decade of birth. This results in the CANEDIT edit being satisfied that the parent should be born in an earlier decade than the child. The NIM changes person 3 to the wife of Person 1 plus the marital status of the couple is changed. This creates a more plausible imputation action than CANEDIT.

Table 6: Imputing More Than the Minimum

Failed Edit Household		
Relationship	Marital Status	Age
Person 1	Divorced	35
Son	Single	8
Daughter	Widowed	36
CANEDIT Imputation		
Person 1	Divorced	45
Son	Single	8
Daughter	Widowed	36
NIM Imputation		
Person 1	Married	35
Son	Single	8
P1's Spouse	Married	36

The advantages of the NIM can be summarized as follows. Its costs tend to increase linearly as the number of edit rules and variables increase. With Fellegi/Holt, the costs increase exponentially. With the NIM, simple algorithms are used while sophisticated linear programming techniques are required with Fellegi/Holt. Fellegi/Holt always imputes the minimum number of variables. The NIM will occasionally impute more than the minimum if this results in a more plausible imputation action. The NIM can be extended fairly easily to non-linear numeric edits and to derived variables (e.g. an edit rule "Number of males in a common-law relationship does not equal number of females in a common-law relationship" could be used). The Fellegi/Holt algorithm is not easily extended.

5. Additional Details of the NIM

The households being edited are split into a number of disjoint strata which are further sub-divided into disjoint imputation groups that are processed independently. For example, six person households form one stratum. This stratum is then split into imputation groups of approximately 20,000 geographically close households each (20,000 is represented by a parameter which can be changed). The donor household for a failed edit household comes from the same imputation group.

The edits can be specified either as a group of conflict rules or as a group of validity rules. Conflict rules define invalid responses (often including blanks) for individual variables plus responses that are considered inconsistent for two or more variables. If a household matches the responses given by one or more conflict rules, then it fails the edits. If it does not match any conflict rule, it passes the edits. Validity rules define combinations of responses for several variables that are considered valid and consistent. If a household matches the responses given by one or more validity rules, it passes the edits. If it does not match any validity rule, it fails the edits. For the rest of this paper, it will be assumed that we are dealing with conflict rules. The algorithms described in Section 6, however, can be easily extended to validity rules.

Edit rules are specified in more than one DLT because up to 7000 edit rules are required to evaluate all possible pairs and permutations of persons in an eight person household. The conflict rules in a DLT are assumed to be connected by "or"s and the DLTs themselves are assumed to be connected by "or"s. This means that a household fails the edits if it matches one or more conflict rule in one or more DLT. For the rest of this paper, it will be assumed that we are dealing with S DLTs that are connected by "or"s. The algorithms described below, however, can be easily extended to DLTs connected by "and"s.

Within an imputation group, it will be assumed that F households fail the edits while P households pass the edits. The responses for the households that fail and pass the edits will be labelled by $\underset{\sim}{V}_f=[V_{fi}]$, f= 1 to F and $\underset{\sim}{V}_p=[V_{pi}]$, p = 1 to P respectively. These are I x 1 vectors containing the responses for the I variables that enter the edit rules.

It is too costly to evaluate, for each failed edit household, the imputation actions of all passed edit households. Often a sufficient number of nearest neighbours are discovered by examining just the 1000 passed edit households geographically closest to the failed edit household. Also, usually only the imputation actions for the closest nearest neighbours (in terms of the distance measure) have to be assessed because only they will generate near minimum change imputation actions.

The distance between a failed edit household and a passed edit household will be defined as

$$D_{fp}=\sum_{i=1}^{I} w_i D_i(V_{fi}, V_{pi})$$

where the weights w_i (which are non-negative) can be given smaller values for variables where it is considered less important that they match. All these weights were set to 1, however, when the NIM was tested.

In the above distance measure, the distance function $D_i(V_{fi}, V_{pi})$ can be different for each variable i. In the 1996 Census, however, one distance function will be used for qualitative variables while a second distance function will be used for the numeric variables. For the qualitative variables, let $D_i(V_{fi}, V_{pi})=1$ if $V_{fi}\neq V_{pi}$ (the ith qualitative variable does not match for the two households) and let $D_i(V_{fi}, V_{pi})=0$ otherwise. For the numeric age variables, $0\leq D_i(V_{fi}, V_{pi})\leq 1$ where $D_i(V_{fi}, V_{pi})$ will be close to or equal to 0 if the difference between V_{fi} and V_{pi} is small while $D_i(V_{fi}, V_{pi})$ will be close to or equal to 1 if the difference between V_{fi} and V_{pi} is large. See Bankier et al (1995) for more details.

Let

$$\underset{\sim}{V}_a=diag(\underset{\sim}{\delta})\,\underset{\sim}{V}_p+diag(\underset{\sim}{1}-\underset{\sim}{\delta})\,\underset{\sim}{V}_f$$

represent a potential imputation action $\underset{\sim}{V}_a$ where $\underset{\sim}{V}_a=[V_{ai}]$. Also, $diag(\underset{\sim}{\delta})$ represents an I x I matrix with $\underset{\sim}{\delta}$ running down the diagonal and zeros elsewhere and $\underset{\sim}{1}$ represents an I x 1 vector of 1's. $\underset{\sim}{1}$ in this paper will always be a vector of 1's. Its number of rows, however, will vary depending on the context in which it is used. In addition, $\underset{\sim}{\delta}=[\delta_i]$ is an I x 1 vector of indicator variables showing which variables will be imputed where $\delta_i = 1$ if the ith variable is imputed while $\delta_i = 0$ otherwise.

Next, it is desired to write D_{fpa} in terms of the imputation action vector $\underset{\sim}{\delta}$. Because $\underset{\sim}{\delta}$ is a binary vector (i.e. the cells of the vector can only take on the values of 0 or 1), expressing the distance measure in terms of it makes the optimum imputation actions easier to determine computationally. It should first be noted that

$$D_{fa}=\sum_{i=1}^{I} w_i D_i(V_{fi}, V_{ai})=\sum_{i=1}^{I} w_i\delta_i D_i(V_{fi}, V_{pi})$$
$$=\sum_{i=1}^{I} w_{fpi}\delta_i = \underset{\sim}{w}'_{fp}\underset{\sim}{\delta}$$

where $\underset{\sim}{w}_{fp}=[w_{fpi}]$ is a I x 1 vector with $w_{fpi}=w_i D_i(V_{fi}, V_{pi})\geq 0$.

It can be easily shown that $D_{fa}+D_{ap}=D_{fp}$. Thus

$$D_{fpa}=(2\alpha-1)\,D_{fa}+(1-\alpha)\,D_{fp}$$
$$=(2\alpha-1)\,\underset{\sim}{w}'_{fp}\underset{\sim}{\delta}+(1-\alpha)\,D_{fp} \qquad (1)$$

Let min D_{fpa} represent the minimum value of D_{fpa} when all nearest neighbour, passed edit households $\underset{\sim}{V}_p$ and all feasible imputation

actions $\underset{\sim}{V}_a$ (based on these nearest neighbours) are considered for that failed edit household $\underset{\sim}{V}_f$. Any feasible, essentially new imputation actions with D_{fpa} = min D_{fpa} will be called minimum change imputation actions.

Any feasible, essentially new imputation actions $\underset{\sim}{V}_a$ which satisfy

$$D_{fpa} \leq \gamma \min D_{fpa} \qquad (2)$$

where $\gamma \geq 1$ will be called near minimum change imputation actions. In tests done to date, γ was set equal to 1.1. Values of γ greater than 1 are allowed because the near minimum change imputation actions, for practical purposes (particularly with numeric variables), are nearly as good as the minimum change imputation actions. Imputation actions which are not near minimum change imputation actions are discarded because the principle of making as little change to the data as possible when carrying out imputation would be violated.

In practice, however, it is desired to keep only the five near minimum change imputation actions with the smallest D_{fpa}. To achieve this, the value of γ is adjusted downwards towards 1 as the processing proceeds.

Then, substituting equation (1) into equation (2) and rearranging the equation, the only feasible imputation actions $\underset{\sim}{\delta}$ that would be retained are those where

$$\underset{\sim}{w}'_{fp}\underset{\sim}{\delta} \leq \frac{\gamma \min D_{fpa} - (1-\alpha) D_{fp}}{2\alpha - 1} \qquad (3)$$

As mentioned above, the cells of $\underset{\sim}{w}_{fp}$ are non-negative. And since $\underset{\sim}{\delta}$ is a binary vector, $\underset{\sim}{w}'_{fp}\underset{\sim}{\delta} \geq 0$. Thus if the right hand side of the inequality of equation (3) is negative, it is known that there are no imputation actions $\underset{\sim}{\delta}$ which satisfy this equation (and hence equation (2)) for that $\underset{\sim}{V}_f$ and $\underset{\sim}{V}_p$.

The I variables can be classified into four groups for the $\underset{\sim}{V}_f$ and $\underset{\sim}{V}_p$ being considered:

(i) Those variables with blank/invalid responses for the failed edit household will be called Type 1 variables. It is known that these variables will always be imputed. Thus the cells of $\underset{\sim}{\delta}$ which correspond to Type 1 variables will equal 1.

(ii) Those variables which match for $\underset{\sim}{V}_f$ and $\underset{\sim}{V}_p$ will be called Type 2 variables. It is known that these variables will never be imputed. Thus the cells of $\underset{\sim}{\delta}$ which correspond to Type 2 variables will equal 0.

(iii) Those variables which are not Type 1 or Type 2 and which do not enter the simplified edits for $\underset{\sim}{V}_f$ and $\underset{\sim}{V}_p$ will be called Type 3 variables. Simplified edits are those remaining after dropping edit rules that no imputation action of $\underset{\sim}{V}_f$ and $\underset{\sim}{V}_p$ matches. It is known that these variables will never be imputed because they do not enter the simplified edits. Any imputation action involving Type 3 variables will not be essentially new. Thus the cells of $\underset{\sim}{\delta}$ which correspond to Type 3 variables will equal 0.

(iv) Those variables which are not Type 1, 2 or 3 variables will be called Type 4 variables. It is not known initially whether they will be imputed or not. Let I_4 represent the number of Type 4 variables.

Rewriting equation (3), the only imputation actions $\underset{\sim}{\delta}$ that will be retained are those where

$$\underset{\sim}{w}'_{fp4}\underset{\sim}{\delta}_4 \leq \frac{\gamma \min D_{fpa} - (1-\alpha) D_{fp}}{2\alpha - 1} - \underset{\sim}{w}'_{fp1}\underset{\sim}{1} \qquad (4)$$

where $\underset{\sim}{w}_{fp4}$ contains the $\underset{\sim}{w}_{fp}$ weights for Type 4 variables while $\underset{\sim}{w}'_{fp1}$ contains the $\underset{\sim}{w}_{fp}$ weights for Type 1 variables. In addition, $\underset{\sim}{\delta}_4 = [\delta_{4i}]$ is an I_4 x 1 vector which contains the cells of $\underset{\sim}{\delta}$ which correspond to the I_4 Type 4 variables.

6. Evaluating Imputation Actions Efficiently

6.1 Defining Groups of Imputation Actions

(a) Let the 2^{I_4} possible imputation actions based on the I_4 Type 4 variables be represented by the columns of the I_4 x 2^{I_4} matrix $\underset{\sim}{\delta}^*$.

(b) Reorder the variables in $\underset{\sim}{w}_{fp4}$ in descending order (from top to bottom) based on what generation the variables are. Within a generation of variables, reorder the variables in $\underset{\sim}{w}_{fp4}$ in descending order based on the size of the weights stored in the $\underset{\sim}{w}_{fp4}$ vector's cells. What generation each of the variables is will be derived iteratively.

(c) Edit rules that $\underset{\sim}{V}_f$ fails will be called Generation 0 edit rules. Variables which enter the Generation 0 edit rules will be called Generation 0 variables. All possible combinations of Generation 0 variables will be imputed (these are the Generation 0 imputation actions) and evaluated first in Section 6.2.3. Generation 0 imputation actions which pass the Generation 0 edit rules and satisfy equation (4) but fail some of the other edit rules (the latter will be called Generation 1 edit rules) will be retained. Generation 0 imputation actions which fail the Generation 0 edit rules will be discarded. This is done because these discarded imputation actions will still fail the Generation 0 edits regardless of which additional non-Generation 0 variables are imputed. Generation 0 imputation actions which pass all the edit rules (including the Generation 0 edit rules) will

not have additional non-Generation 0 variables imputed because such imputation actions would not be essentially new. Thus only Generation 0 imputation actions which pass the Generation 0 edits but fail some of the other edits should have additional non-Generation 0 variables imputed. Generation 0 imputation actions which do not satisfy equation (4), will still not satisfy equation (4) if additional variables are imputed. Thus, these Generation 0 imputation actions should be discarded as well.

(d) Variables which enter the Generation 1 edit rules but which are not Generation 0 variables will be called Generation 1 variables. All possible combinations of the Generation 1 variables will be imputed for the retained Generation 0 imputation actions (these are the Generation 1 imputation actions) and evaluated next in Section 6.2.3. Generation 1 imputation actions which pass the Generation 1 edit rules and satisfy equation (4) but fail some of the other edit rules (the latter will be called Generation 2 edit rules) will be retained for use in point (e).

(e) Variables which enter the Generation 2 edit rules but which are not Generation 0 or 1 variables will be called Generation 2 variables. All possible combinations of the Generation 2 variables will be imputed for the retained Generation 1 imputation actions (these are the Generation 2 imputation actions) and evaluated next in Section 6.2.3.

(f) This process will continue until Generation g when no Generation g+1 variables are found. This will occur

- if all Type 4 variables have been assigned to one of the g generations
- some Type 4 variables have not been assigned to a generation but there are no Generation g imputation actions which pass the Generation g edit rules and satisfy equation (4) but fail some other edit rules.

If some Type 4 variables have not been assigned a generation, they are said to be not related to the Generation 0 variables.

(g) Generation 0 variables should have their imputation actions assessed first because it is known that at least one Generation 0 variable has to be imputed for the imputation actions to pass the edits that $\underset{\sim}{V}_f$ failed. Generation 1 variables should be assessed next because it is known that at least one Generation 1 variable has to be imputed for imputation actions, based on the remaining Generation 0 imputation actions, to pass the Generation 1 edits. This process will be repeated with Generation 2, 3 etc. variables. It is known that some Generation 0 variables will always be imputed. Generation 1, 2 , 3 etc. variables will generally be imputed with progressively less frequency until we reach the variables (if any) that are not related to the Generation 0 variables. It is known that these unrelated variables will never enter any near minimum change imputation actions and hence will never be imputed. They can be converted to Type 3 variables and the count I_4 of the number of Type 4 variables can be reduced accordingly. A further discussion of the implications of processing variables in terms of what generation they belong to is given in Section 6.4.

(h) Reorder the rows in $\underset{\sim}{\delta}^*$ such that the variables take the same order as in $\underset{\sim}{w}_{fp4}$.

(i) Finally, reorder the columns of $\underset{\sim}{\delta}^*$ such that the matrix equals

$$\underset{\sim}{\delta}^* = [\underset{\sim}{\delta}_0^* \ \underset{\sim}{\delta}_1^* \ \underset{\sim}{\delta}_2^* \cdots \underset{\sim}{\delta}_{I_4}^*] \qquad (5)$$

where sub-matrix $\underset{\sim}{\delta}_0 = \underset{\sim}{0}$ (a single column of zeros) while sub-matrix $\underset{\sim}{\delta}_i^*$ ($i > 0$) contains all imputation actions where the i^{th} variable (counting from the bottom of $\underset{\sim}{w}_{fp4}$) from the vector $\underset{\sim}{w}_{fp4}$ is imputed (along with all possible combinations of variables that occur below the i^{th} variable in $\underset{\sim}{w}_{fp4}$) but no variables that occur above the i^{th} variable in $\underset{\sim}{w}_{fp4}$ are imputed. The rows of the transpose of $\underset{\sim}{\delta}^*$, when viewed as binary numbers, will be arranged in ascending order. The imputation actions in sub-matrix $\underset{\sim}{\delta}_i^*$ will be called Imputation Action Group i or Group i for short in the sections which follow.

(j) The $\underset{\sim}{\delta}^*$ matrix ordered in this fashion will be illustrated by a simple example. Assume that $I_4 = 4$. Then $\underset{\sim}{\delta}^*$, after reordering, will take the following form:

$\underset{\sim}{\delta}_0^*$	$\underset{\sim}{\delta}_1^*$	$\underset{\sim}{\delta}_2^*$	$\underset{\sim}{\delta}_3^*$	$\underset{\sim}{\delta}_4^*$
0	0	00	0000	11111111
0	0	00	1111	00001111
0	0	11	0011	00110011
0	1	01	0101	01010101

In this example, the rows of the transpose of $\underset{\sim}{\delta}^*$, when viewed as binary numbers, are arranged in ascending order from 0 to 15 with one row for each of the $2^4 = 16$ possible imputation actions.

(k) The i'th sub-matrix of the above equation (5), i' = 1 to I_4, can be constructed from the other sub-matrices using the following equation:

$$\underset{\sim}{\delta}_{i'}^* = [\underset{\sim}{\delta}_0^* \ \underset{\sim}{\delta}_1^* \cdots \underset{\sim}{\delta}_{(i'-1)}^*] + \underset{\sim}{1}_{i'} \underset{\sim}{1}' \qquad (6)$$

where $\underset{\sim}{1}_{i'}$ is a I_4 x 1 vector with a 1 in the cell corresponding to the i'th variable (counting from the bottom of $\underset{\sim}{w}_{fp4}$) and with zeros elsewhere. In Section 6.2, various checks are done which result in imputation actions possibly being dropped for the 0^{th} to $(i'-1)^{th}$ submatrices. These imputation actions should be dropped before the i'th sub-matrix is generated since this will reduce the number of imputation actions that it and subsequent sub-matrices

contain.

6.2 Assessing Imputation Actions Plus Simplifying DLTs Further

Sections 6.2.1 to 6.2.3 show an effective method to determine which imputation actions pass the edits and which fail the edits while at the same time simplifying the S DLTs further. Then Section 6.3 shows how to implement the algorithm of Section 6.2 efficiently.

6.2.1 Initial Comments

(a) Before any imputation actions of the first passed edit household are assessed, min D_{fpa} will be initialized to the maximum possible value of D_{fpa} for that failed edit household and the first passed edit household, i.e.

$$\min D_{fpa} = (2\alpha - 1)\,(w'_{fp1}\underset{\sim}{1} + w'_{fp4}\underset{\sim}{1}) + (1-\alpha)\,D_{fp}$$

using equation (1). This maximum value would occur if all the Type 4 variables were imputed along with the Type 1 variables. The min D_{fpa} will be updated during the processing of the first passed edit household to reflect the smallest value of D_{fpa} encountered. The value of min D_{fpa} at the end of processing of one passed edit household will be used at the start of processing of the next passed edit household for that failed edit household.

(b) The procedures below will be carried out separately for each passed edit household having imputation actions assessed for the failed edit household.

(c) The imputation actions for a passed edit household will be processed sequentially based on the i = 0 to I_4 imputation action groups $\underset{\sim}{\delta}^*_i$ defined in Section 6.1.

6.2.2 Processing Imputation Action Group i = 0

(a) The sole imputation action in Group i = 0 has no Type 4 variables imputed. This imputation action will satisfy equation (4) because the right hand side of equation (4) is non-negative. If the right hand side of equation (4) was negative, the passed edit household would have been discarded without having its imputation actions assessed.

(b) Next, this imputation action should be assessed against the S DLTs to determine if it passes the edits. If it does, then no further imputation actions have to be considered since this is the only essentially new imputation action.

6.2.3 Processing Imputation Action Groups i > 0

(a) Groups i = 1 to I_4 will be assessed sequentially starting with i = 1. Let i = i' represent the group currently being assessed. It will be assumed in the discussion below that i' is a Generation g' variable.

(b) Generate Group i' from Groups 0 to i' - 1 using equation (6) of Section 6.1.

(c) Each imputation action in Group i' will be assessed against equation (4). If equation (4) is not satisfied, the imputation action in Group i' is dropped. This is done since imputation actions in groups with i > i' generated from the dropped Group i' imputation action would not satisfy equation (4) either. Also, the imputation action in the previous group (i < i') which generated the dropped imputation action from Group i' should not be used to generate imputation actions for other groups with i > i' that belong to Generation g'. This is because, within Generation g', the w_{fp4} weights are sorted in descending order. Thus none of the generated imputation actions within Generation g', based on this previous group imputation action, will satisfy equation (4). The previous group imputation action, however, can be used to generate imputation actions for other groups with i > i' that belong to Generation g' + 1. It is possible, however, that this previous group imputation action may be dropped by the checks of point (f) below before Generation g' + 1 variables are processed.

(d) Each imputation action in Group i' remaining after point (c) is assessed against the S DLTs to determine if it passes the edits.

(d1) If it passes the edits, the imputation action should be placed in the list of near minimum change imputation actions. It should be removed from Group i' since any other imputation action generated later based on the removed imputation action will not be essentially new. It should also be assessed whether the left hand side of equation (4) for this removed imputation action is less than the right hand side of equation (4) if $\gamma = 1$. If this is so, determine D_{fpa} for this removed imputation action and set min D_{fpa} equal to this value. In addition, other imputation actions in Group i' which are not essentially new in terms of this removed imputation action should also be removed. Only imputation actions which are larger than the removed imputation action (when viewed as binary numbers) should be evaluated to see if they are not essentially new. This is because, for an imputation action to not be essentially new, it must have ones in the same positions as the removed imputation action plus at least one additional one.

(d2) If it fails edits, it should be retained in Group i'.

(e) If the value of min D_{fpa} is changed in step (d1), determine if equation (4) is still satisfied for all previously retained imputation actions in groups i = 0 to i'. The retained imputation actions include those identified as near minimum change imputation actions and those which failed the edits but which were retained to generate other imputation actions in later groups. Any found that no longer satisfy equation (4) should be dropped. Assumed a dropped imputation

action comes from Group i''. If the weights in $\underset{\sim}{w}_{fp4}$ for variables with i > i'' equal or exceed the weight for i = i'' then the imputation action in the previous group which generated the dropped imputation action from Group i'' is also dropped.

(f) After steps (c), (d) and (e) have been applied to each imputation action in Group i', it will be assessed
- if some imputation actions remaining in Group i' can be dropped (in step (f1)) because no imputation actions in groups with i > i' generated from these dropped imputation actions will pass the edits or
- if any edit rules can be dropped (in step (f2)) because no imputation actions in Groups with i > i' will match them. Propositions where there is no variable with i > i' entering will be called <u>i ≤ i' propositions</u>. Steps (f1) and (f2) will only be carried out if there is one or more i ≤ i' proposition. Edit rules where only i ≤ i' propositions enter will be called <u>i ≤ i' edit rules</u>. Step (f1) should be applied before step (f2) to allow edit rules to be discarded more quickly.

(f1) Drop any Group i = 0 to i' imputation action which matches one or more i ≤ i' edit rules in at least one of the S DLTs.

(f2) Drop any edit rules which, for each Group i = 0 to i' imputation action remaining, have one or more non-matches for the i ≤ i' propositions. Then drop any propositions which do not enter any of the remaining edit rules. Then identify any of the i > i' variables which no longer enter any of the remaining propositions of the DLTs as a result of dropping these edit rules and propositions. Label these variables as Type 3 (i.e. they no longer enter the edits) and reduce the count I_4 of Type 4 variables correspondingly. Additional groups of imputation actions will only be generated for those variables i > i' which have not been converted to Type 3 variables.

(g) At the end of the analysis of each Group i' imputation actions, it will be determined if any imputation actions remain in Groups i = 0 to i'. If the answer is no, all near minimum change imputation actions have been determined for that passed edit household for the failed edit household. If the answer is yes, go to step (h).

(h) At the end of the analysis of the Group i' imputation actions, processing will continue with Group i' + 1 if all S DLTs still contain some edits. If, however, one or more of the S DLTs has had all its edit rules deleted, steps (h1) and (h2) will be carried out. Edits were deleted only if none of the remaining imputation actions in Groups i = 0 to I_4 matched the deleted edits.

(h1) <u>If some DLTs still contain some edits</u> then any DLTs with no edit rules remaining can be ignored and processing will continue with the Group i' + 1.

(h2) <u>If no DLTs contain any edits</u> then there are no more Type 4 variables and all near minimum change imputation actions have been determined for that passed edit household for the failed edit household.

<u>6.3 Assessing Imputation Actions Efficiently</u>

(a) The I_4 + 1 imputation action groups were generated and assessed sequentially in Section 6.2. This was computationally efficient because deleted imputation actions were not used to generate imputation actions in a later group. Also, edit rules that were deleted and Type 4 variables that were converted to Type 3 variables were not assessed by later imputation action groups.

(b) Splitting the imputation actions into groups and then evaluating them sequentially has significant computational advantages as well when evaluating equation (4) and the DLTs as will be shown in this section.

(c) First, some notation will be defined. Assume that the s^{th} simplified DLT (s = 1 to S) has M_{s4} propositions and that it lists J_{s4} edit rules. Let the total number of propositions and edit rules in the simplified S DLTs be represented by M_4 and J_4 respectively. Information on the propositions in the DLTs will be provided in the following three matrices:

- $\underset{\sim}{B}$ - a M_4 x 1 matrix containing, in machine readable form, the part of the M_4 propositions to the left of the sign (e.g. Relat(3)).
- $\underset{\sim}{C}$ - a M_4 x 1 matrix providing the constant to the right of the sign in the proposition. This constant will either be a response class or an individual response (e.g. Mother) in the case of single qualitative variable proposition or it will be a numeric constant in the case of other types of propositions.
- $\underset{\sim}{X}$ - a M_4 x 1 matrix providing the signs separating the variables from the constants in the propositions. The following numbers represent the various signs:
  1 - ≤
  2 - =
  3 - <
  4 - >
  5 - ≠
  6 - ≥

It will be assumed that the propositions in the above three matrices are arranged by DLT such that the propositions in the first DLT come first, the propositions in the second DLT come second etc.

(d) The edit rules associated with the propositions in

the s^{th} simplified DLT will be represented by the following matrix:

$\underset{\sim}{R}_s$ - a M_{s4} x J_{s4} matrix recording the pattern of Y's, N's and blanks for the J_{s4} edit rules for the M_{s4} propositions. The Y's, N's and blanks will be represented by the following numbers

1 - Y
-1 - N
0 - blank or - which means Y or N

(e) A M_4 x 2^{I_4} matrix

$$\underset{\sim}{T}^* = [\underset{\sim}{T}_0^* \ \underset{\sim}{T}_1^* \ \underset{\sim}{T}_2^* \cdots \underset{\sim}{T}_{I_4}^*]$$

will be defined which contains the condition result vectors (defined below in this point) for M_4 propositions of the S DLTs for the 2^{I_4} imputation actions contained in the matrix $\underset{\sim}{\delta}^*$. If the m^{th} proposition is true for an imputation action then the cell in the m^{th} row and in the column of $\underset{\sim}{T}^*$ corresponding to that imputation action will be set to 1. Otherwise it will be set to -1.

(f) A J_4 x 2^{I_4} matrix

$$\underset{\sim}{N}^{**} = [\underset{\sim}{N}_0^{**} \ \underset{\sim}{N}_1^{**} \ \underset{\sim}{N}_2^{**} \cdots \underset{\sim}{N}_{I_4}^{**}]$$

will be defined which contains the number of non-matching propositions for each of the J_4 edit rules of the S DLTs and for each of the 2^{I_4} imputation actions contained in the matrix $\underset{\sim}{\delta}^*$.

(g) The single imputation action contained in $\underset{\sim}{\delta}_0^*$ will be evaluated against the propositions contained in $\underset{\sim}{B}$, $\underset{\sim}{C}$ and $\underset{\sim}{X}$ to determine the condition result vector $\underset{\sim}{T}_0^*$. Then, $\underset{\sim}{T}_0^*$ will be compared to $\underset{\sim}{R}_s$, s = 1 to S so that the number of non-matching propositions can be recorded for each of the J_4 edit rules and stored in $\underset{\sim}{N}_0^{**}$. It will be considered a non-match for a specific proposition

- if the condition result is 1 in $\underset{\sim}{T}_0^*$ and the rule being analysed in $\underset{\sim}{R}_s$ has -1 or
- if the condition result is -1 in $\underset{\sim}{T}_0^*$ and the rule being analysed in $\underset{\sim}{R}_s$ has 1.

It should be noted, however, that the values in the condition result vector $\underset{\sim}{T}_0^*$ and the non-matches vector $\underset{\sim}{N}_0^{**}$ could have already been generated when the household was edited.

(h) The method to evaluate the imputation actions for Groups i = 1 to I_4 will now be described. Let i = i' be the group currently being assessed. Generate the Group i' imputation actions $\underset{\sim}{\delta}_{i'}$ using equation (6). It is easy to see that

$$\underset{\sim}{w}'_{fp4}\underset{\sim}{\delta}_{i'}^* = \underset{\sim}{w}'_{fp4}([\underset{\sim}{\delta}_0^* \ \underset{\sim}{\delta}_1^* \cdots \underset{\sim}{\delta}_{i'-1}^*] + \underset{\sim}{1}_{i'}\underset{\sim}{1}')$$

Earlier steps which assessed $\underset{\sim}{\delta}_0^*$, $\underset{\sim}{\delta}_1^*$ etc. determined the values in the vector $\underset{\sim}{w}'_{fp4}[\underset{\sim}{\delta}_0^* \ \underset{\sim}{\delta}_1^* \cdots \underset{\sim}{\delta}_{i-1'}^*]$. And the quantity $\underset{\sim}{w}'_{fp4}\underset{\sim}{1}_{i'}\underset{\sim}{1}'$ is just a vector containing the weight of the i'^{th} variable in each cell. This vector is added to the already known vector to determine $\underset{\sim}{w}'_{fp4}\underset{\sim}{\delta}_{i'}^*$. The values in $\underset{\sim}{w}'_{fp4}\underset{\sim}{\delta}_{i'}^*$ can be used to determine which imputation actions in $\underset{\sim}{\delta}_{i'}^*$ do not satisfy equation (4) and hence can be dropped.

(i) Initialize

$$\underset{\sim}{T}_{i'}^* = [\underset{\sim}{T}_0^* \ \underset{\sim}{T}_1^* \ \underset{\sim}{T}_2^* \cdots \underset{\sim}{T}_{i'-1}^*] \qquad (7)$$

and

$$\underset{\sim}{N}_{i'}^{**} = [\underset{\sim}{N}_0^{**} \ \underset{\sim}{N}_1^{**} \ \underset{\sim}{N}_2^{**} \cdots \underset{\sim}{N}_{i'-1}^{**}]$$

The number of columns in $\underset{\sim}{T}_{i'}^*$ and $\underset{\sim}{N}_{i'}^{**}$ will always be identical to the number of columns in $\underset{\sim}{\delta}_{i'}^*$.

(j) It is only necessary to update the entries of $\underset{\sim}{T}_{i'}^*$ for those propositions where the i^{th} variable enters. This is to be done for each of the imputation actions appearing in $\underset{\sim}{\delta}_{i'}^*$. The processing to be carried out for a specific proposition and a specific imputation action is described in point (k) below.

(k) The condition result generated for that proposition and that imputation action should be compared to the initial condition result for that proposition and that imputation action given in equation (7). If the condition result is unchanged from the initial result, no more processing of that proposition and imputation action is required. If the condition result is changed (it is converted from a 1 to a -1 or it is converted from a -1 to a 1), the s^{th} DLT that this proposition belongs to should be identified and the edit rules that the proposition enters (i.e. there is a 1 or a -1 rather than a 0 in $\underset{\sim}{R}_s$ for an edit rule for that proposition) should be identified. For each edit rule that the proposition enters, it should be determined if the updated condition result matches the edit rule for that proposition. If it does, the number of non-matches in $\underset{\sim}{N}_{i'}^{**}$ should be decreased by 1 for that edit rule. If it does not match, the number of non-matches in $\underset{\sim}{N}_{i'}^{**}$ should be increased by 1 for that edit rule.

(l) Then the next imputation action appearing in $\underset{\sim}{\delta}_{i'}^*$ will be evaluated for that proposition. After all imputation actions appearing in $\underset{\sim}{\delta}_{i'}^*$ have been evaluated for a proposition, the next proposition that the variable i' enters will have all imputation actions appearing in $\underset{\sim}{\delta}_{i'}^*$ evaluated for it.

(m) After all propositions that have the variable i' enter have been evaluated for all of the $\underset{\sim}{\delta}_{i'}^*$ imputation actions and $\underset{\sim}{T}_{i'}^*$ and $\underset{\sim}{N}_{i'}^{**}$ have been updated, $\underset{\sim}{N}_{i'}^{**}$ can be assessed to determine which imputation actions pass or fail the edits. Let $\underset{\sim}{N}_{si'}^{**}$ be the sub-matrix of $\underset{\sim}{N}_{i'}^{**}$ containing the J_{s4} edit rules of the s^{th} DLT. These sub-matrices will be evaluated in the following fashion for each imputation action. If there is one or more edit rules in any of

the S DLTs with 0 non-matches, the imputation action fails the edits. Otherwise it passes the edits.

(n) When evaluating a proposition where u Type 4 variables enter, it is known that there are 2^u possible imputation actions for that proposition. The 2^u imputation action condition results will gradually be generated as Groups i, i = 0 to I_4 are evaluated. When the condition result for one of the 2^u possible imputation actions is derived, it should be retained so that the proposition will not have to be evaluated a second time for that imputation action.

(o) It will be necessary to have a second matrix similar to $N_{\sim i'}^{**}$ which will keep track of the number of non-matches for the $i \leq i'$ propositions. These counts are needed for steps (f1) and (f2) in Section 6.2.3.

(p) It should also be noted that it is not absolutely necessary to reorder the variables at the start of this algorithm. It is sufficient to assess the imputation actions starting with the one just involving the Generation 0 variable with the smallest weight and then examine variables with progressively larger generation numbers and weights. Also, when dropping propositions and edit rules, it is not necessary to actually delete them from the DLTs. It may be simpler and computationally more efficient to just have indicator vectors which keep track of which propositions and edit rules have been deleted.

6.4 Further Discussion of Generations

After all Generation 0 imputation actions have been assessed, the only ones which will remain (as discussed in Section 6.1) are those which pass the Generation 0 edit rules and satisfy equation (4) but fail some of the other edit rules (these will be called Generation 1 edit rules). In addition, all Generation 0 edit rules will have been discarded as a result of point (f2) of Section 6.2.3.

After all Generation 1 imputation actions have been assessed, the only ones which will remain (for reasons similar to those with the Generation 0 imputation actions) are those which pass the Generation 1 edit rules and satisfy equation (4) but fail some of the other edit rules (these will be called Generation 2 edit rules). The remaining Generation 1 imputation actions were generated from Generation 0 imputation actions which passed the Generation 0 edit rules but failed the Generation 1 edit rules.

Because the remaining Generation 1 imputation actions now pass the Generation 1 (and Generation 0) edit rules, this shows that one or more Generation 1 variables has been imputed for each of the remaining Generation 1 imputation actions. Thus all Generation 0 imputation actions have been discarded by the end of the assessment of the Generation 1 imputation actions. In addition, all Generation 1 edit rules will have been discarded as a result of point (f2) of Section 6.2.3.

After all Generation 2 imputation actions have been assessed, the only ones which will remain (for reasons similar to those with the Generation 0 imputation actions) are those which pass the Generation 2 edit rules and satisfy equation (4) but fail some of the other edit rules (these will be called Generation 3 edit rules). The remaining Generation 2 imputation actions were generated from Generation 1 imputation actions which passed the Generation 1 edit rules but failed the Generation 2 edit rules. Because the remaining Generation 2 imputation actions now pass the Generation 2 (and Generation 0 and 1) edit rules, this shows that one or more Generation 2 variables has been imputed for each of the remaining Generation 2 imputation actions. Thus all Generation 1 imputation actions have been discarded by the end of the assessment of the Generation 2 imputation actions. In addition, all Generation 2 edit rules will have been discarded as a result of point (f2) of Section 6.2.3.

This process will continue in a similar fashion for later generations of variables.

7. Concluding Remarks

The NIM performs minimum change donor imputation for numeric and qualitative variables simultaneously in a computationally feasible fashion. It is applicable to a wide range of surveys. It is in the final stages of testing prior to processing the 1996 Canadian Census demographic variables. It will be generalized to do donor imputation for more variables for the 2001 Canadian Census.

References

Bankier, M., Fillion, J.-M., Luc, M. and Nadeau, C. (1994), "Imputing Numeric and Qualitative Variables Simultaneously", Proceedings of the Section on Survey Research Methods, American Statistical Association, 242-247.

Bankier, M., Luc, M., Nadeau, C. and Newcombe, P. (1995), "Additional Details on Imputing Numeric and Qualitative Variables Simultaneously", Proceedings of the Section on Survey Research Methods, American Statistical Association, 287-292.

Fellegi, I.P. and Holt, D. (1976), "A Systematic Approach to Automatic Edit and Imputation", Journal of the American Statistical Association", March 1976, Volume 71, No. 353, 17-35.