EFFICIENT METHODOLOGY WITHIN THE CANADIAN CENSUS EDIT AND IMPUTATION SYSTEM (CANCEIS)

Michael Bankier, Paul Poirier and Martin Lachance
Paul Poirier, Statistics Canada, Ottawa, Canada, K1A 0T6 Paul.Poirier@statcan.ca

KEY WORDS: minimum change; editing and imputation; inconsistent responses;

1. Introduction

Many minimum change imputation systems are based on the approach proposed by Fellegi and Holt (1976). For example, CANEDIT and GEIS at Statistics Canada and DISCRETE and SPEER at United States Bureau of the Census (USBC) use (or had as their starting point) the Fellegi/Holt imputation methodology. A somewhat different approach was used in the 1996 Canadian Census to impute for nonresponse and resolve inconsistent responses for the variables age, sex, marital status, common-law status and relationship for all persons in a household simultaneously. The method used is called the Nearest-Neighbour Imputation Methodology (NIM). This implementation of the NIM allowed, for the first time, the simultaneous hot deck imputation of qualitative and quantitative variables for large Edit and Imputation (E&I) problems. Bankier (1999) provides an overview of the NIM algorithm In this paper, the algorithm to be used in the 2001 Canadian Census is described in detail.

The main difference between the NIM and Fellegi/Holt is that the NIM searches for nearest-neighbour donors first and then determines the minimum change imputation action based on these donors. The Fellegi/Holt methodology determines the minimum number of variables to impute and then searches for donors. Reversing the order of these operations confers significant computational advantages to the NIM while still meeting the well accepted Fellegi/Holt objectives of imputing the fewest variables possible and preserving sub-population distributions. The NIM can, however, only be used to carry out imputation using donors while Fellegi/Holt can be used with any imputation methodology.

For the 2001 Canadian Census, a more generic implementation of the NIM has been developed. It is called the CANadian Census Edit and Imputation System (CANCEIS). It is written in the ANSI C programming language and runs off flat ASCII files. As a result, with only minor modifications, it can run on many platforms such as the PC or mainframe and under different operating systems. Besides the demographic variables, it will be used to perform E&I for the labour, mobility, place of work and mode of transport variables. For the 2006 Canadian Census, it is planned to use CANCEIS to process all census variables including the income variables.

The objective of this paper is to describe briefly how CANCEIS determines the minimum number of variables to impute for a failed record/donor pair in a highly efficient fashion when dealing with a mixture of quantitative and qualitative variables. More details regarding the NIM are provided in Bankier (2000).

2. Specification of Edit Rules

For the variables being edited, the user specifies a series of edit rules (or edits for short) which indicate which responses or combinations of responses are either impossible or highly implausible. These response patterns are to be eliminated through imputation. If a record matches one or more of these edit rules, it is said to fail the edits and will be called a failed record. If a record matches no edit rules, it is said to pass the edits and will be called a passed record. The edit rules are specified in a series of decision logic tables (DLTs).

Table 1 gives an example of a DLT. A series of propositions are listed in the first column followed by three columns which each represents an edit rule. A household fails edit rule 2, for example, if the first proposition is false and the fourth and fifth propositions are true. In this DLT, Relashionship(2) represents the relationship of the person listed second on the questionnaire to the person listed first on the questionnaire. Class(Spouse) represents the response class or set of responses {Married_Spouse, Common_Law_Spouse} where Relationship(2) = Class(Spouse) is considered true if it is equal to one or the other of these two responses in the response class. Relationship(2) is a qualitative variable but the responses such as Married_Spouse are actually just labels with the data for Relashionship(2) being stored on the data file as integers, e.g. the code 2 may represent Married_Spouse. The notation p1 in Table 1 represents a variable position person whom, in this example, can take on the values p1 = 2 to 6 for a six person household. CANCEIS makes five replicates of Table 2, for p1 = 2, 3, 4, 5 and 6 to save the user from having to specify these replicates manually.

Table 1: A Decision Logic Table used in the 1996 Canadian Census

Propositions	Rules 1	2	3
Relationship(2) = Class(Spouse)	F	F	F
Relationship(p1) = Grandchild	T		
Age(1) - Age(p1) < 30	T		
Relationship(p1) = Grandparent		T	
Age(p1) - Age(1) < 30		T	
Relationship(p1) = Son/Daughter			T
Age(1) - Age(p1) < 15			T

Table 2 below defines the generic format of DLTs that will be accepted by CANCEIS. It can be seen that a more general form of propositions is allowed to accommodate the more extensive use of quantitative variables. For example,CANCEIS allows a proposition of the form

$$V1 + 2*V2 - 100*V3 + V4 + V5 + V6 \leq 6$$

A DLT can be viewed as a J x (G+1) matrix where the first column is a list of J propositions followed by G columns that each represent an edit rule. The g^{th} edit rule, g = 1 to G, will be represented by $\underset{\sim}{R}_g$, which is a J x 1 vector whose entries are either T, F or b (for blank). The j^{th} proposition, j = 1 to J, takes the form $\Delta_j = \sum_i B_{ji}V_{ai} - c_j \ll 0$ where V_{ai}, i = 1 to I, represent the responses (possibly after imputation) for the I variables being edited, B_{ji} is a coefficient associated with the i^{th} variable and c_j equals a quantitative constant or a set of quantitative constants in the case of a response class associated with a single qualitative variable. The imputed value V_{ai} for the i^{th} variable can be written as

$$V_{ai} = \delta_i V_{pi} + (1-\delta_i) V_{fi} = \delta_i (V_{pi} - V_{fi}) + V_{fi}$$

where V_{fi} represents the value of the i^{th} variable from the failed record while V_{pi} represents the value of the i^{th} variable from the donor being used and δ_i is an indicator variable (where $\delta_i = 1$ if the i^{th} variable is imputed and $\delta_i = 0$ otherwise). When the edit rules are initially applied to determine which records fail and which pass the edits, $\delta_i = 0$ for all i, of course. Finally, the symbol $\ll$ represents one of the signs <, =, >, ≤, ≠ or ≥. It can be seen that the propositions of Table 1 can be easily reformatted to correspond to the $\Delta_j \ll 0$ format.

Table 2: Format of Decision Logic Table Used to Specify CANCEIS Edit Rules

Propositions	Rules 1	...	G
$\Delta_1 = \sum_i B_{1i}V_{ai} - c_1 \ll 0$	T/F/b	...	T/F/b
$\Delta_2 = \sum_i B_{2i}V_{ai} - c_2 \ll 0$	T/F/b	...	T/F/b
...	...	...	...
$\Delta_J = \sum_i B_{Ji}V_{ai} - c_J \ll 0$	T/F/b	...	T/F/b

To evaluate whether a record passes or fails the edits in Table 2, each of the J propositions is evaluated to determine whether it is true (T) or false (F) for that record. The results can be stored in a J x 1 condition result vector $\underset{\sim}{T}$ where the j^{th} entry is set to T or F. The record fails the g^{th} edit rule if the vectors $\underset{\sim}{T}$ and $\underset{\sim}{R}_g$ are equal for those propositions which enter the g^{th} edit rule (i.e. those propositions which have a T or a F entry as opposed to a blank entry in $\underset{\sim}{R}_g$).

CANCEIS takes the edit rules in the DLTs specified by the users and replicates any that include variable position persons as represented by the operators p1, p2 etc. Next, the six possible signs <, =, >, ≤, ≠ and ≥ are reduced to the three signs >, = and < by changing T's to F's and F's to T's in the DLTs for propositions with the signs ≤, ≠ and ≥. Then each proposition is converted into the $\Delta_j \ll 0$ format with

$$\Delta_j = \sum_i B_{ji}V_{ai} - c_j = \sum_i B^*_{ji}\delta_i - c^*_j$$

where $B^*_{ji} = B_{ji}(V_{pi} - V_{fi})$ and $c^*_j = c_j - \sum_i B_{ji}V_{fi}$ because $V_{ai} = \delta_i (V_{pi} - V_{fi}) + V_{fi}$. Expressing Δ_j in terms of the indicator variables δ_i has certain advantages as will be demonstrated in Section 5. The propositions are next stored numerically in terms of their B^*_{ji} and c^*_j values and with the value of the sign $\ll$ being recorded. Then the DLTs are combined by CANCEIS to form a single DLT. If several DLTs contain the same proposition, only one copy of the proposition is retained in the combined DLT. The propositions within the combined DLT are sorted in descending order (from top to bottom) in terms of the

number of edit rules that they enter. The edit rules in the combined DLT are sorted in ascending order (from left to right) in terms of the number of propositions that enter an edit rule. If several rules are found to be identical in terms of their propositions and pattern of T's and F's, only one copy is kept. If the propositions entering one rule are a subset of the propositions entering a second rule and the pattern of T's and F's for this subset of propositions are identical for the two rules, the second rule is dropped because any records which fail the second rule would also fail the first rule. The use of a single sorted combined DLT (which will be called the sorted DLT) improves the computational efficiency of the E&I process as will be seen later.

Because the pattern of T's and F's for this sorted DLT usually forms a sparse matrix (i.e. many blanks are present), the pattern of T's and F's are stored as a list along with information on their location in the matrix. The edit rules used to identify nonresponse (which is defined here to include invalid responses) are not included in the sorted DLT but are stored separately. These are the first edits to be applied since the majority of records generally fail because of nonresponse only. These are also the first responses to be imputed because it is known that these variables must be imputed with certainty.

3. Efficient Editing of Records

In this section, the method used to efficiently determine which records pass or fail the edits will be described. First, if there is nonresponse to any of the variables in a record, the record fails the edits and proceeds immediately to imputation. Otherwise, the edit rules in the sorted DLT are evaluated from left to right (and the propositions from top to bottom) to determine if the record fails at least one of these edit rules because of inconsistent responses. It is first determined if the condition result is T or F for the first proposition which enters the first edit rule. CANCEIS immediately flags as dropped any edit rules that the first proposition enters whose value for that proposition does not equal the condition result since they can never be failed by that record. In addition, the proposition itself is flagged as dropped because it is known that it is satisfied by any edit rules that remain. Next, the leftmost remaining edit rule is identified (this may still be the first edit rule if it was not dropped) and the first proposition not dropped that enters that edit rule has its condition result determined. CANCEIS again flags as dropped any edit rules that this second proposition enters whose value for that proposition does not equal the condition result since they can never be failed by that record. In addition, the second proposition is flagged as dropped. This process continues until all the edit rules have been dropped (in which case the record passes the edits) or an edit rule has not been dropped and all the propositions which enter it have been evaluated (in which case, the record fails this edit). CANCEIS then proceeds in the same manner to apply the full set of edits to the next record.

4. Criteria For Selection of Donors and Imputation Actions

Below are *listed* the criteria used to select donors and determine which IA to retain for the failed record. The criteria used are based on distance measures which are very general and which include the option of imputing the minimum number of variables possible given the available donors. The class of distance measures used can be made even more broad with minor modifications to CANCEIS.

CANCEIS finds at least 40 (this number can vary) *passed records* (called nearest neighbours or donors for short) in the group of records being processed (which is called an imputation group) that are closest to the *failed record* in terms of a distance measure. These donors are used to generate IAs. The distance measure is

$$D_{fp} = D(\underset{\sim}{V}_f, \underset{\sim}{V}_p) = \sum_{i=1}^{I} w_i D_i(V_{fi}, V_{pi})$$

where the distance between the response of the failed record (V_{fi}) and the response of the passed record (V_{pi}) for the i[th] variable is a function which falls in the range $0 \le D_i(V_{fi}, V_{pi}) \le 1$. If $V_{fi} = V_{pi}$ then $D_i(V_{fi}, V_{pi}) = 0$ while if $|V_{fi} - V_{pi}|$ is large then $D_i(V_{fi}, V_{pi}) \simeq 1$. Intermediate values of $|V_{fi} - V_{pi}|$ generate values between 0 and 1. In the case of qualitative variables, if $V_{fi} \neq V_{pi}$ then *generally* $D_i(V_{fi}, V_{pi}) = 1$. The form of the distance measure can be different for *each variable* as long as it respects the above minor restrictions. The weights w_i of the variables (which are non-negative) can be given smaller values for variables where it is considered less important that they match (with, for example, variables considered more likely to be in error). In the 1996 Canadian Census, however, all w_i were set to one. The distance measure can include auxiliary variables which are defined as variables that enter the distance measure but not the edits. A variable will be said to enter an edit rule if it appears in at least one proposition

that enters that edit rule. To ensure the best donors are selected, the failed record occupants can be reordered in various ways to see which results in the smallest distance compared to a particular *passed record*. Smaller distances may result through reordering because, for example, children can be listed in ascending order based on age in one household and descending order in another household.

Only nonmatching variables (those with $V_{fi} \neq V_{pi}$) are, of course, considered for imputation. Various subsets of these nonmatching variables are imputed to determine which are the optimum imputations for a *failed record*/donor pair. Each of these subsets, when imputed, will be called an imputation action (IA). An IA can be defined more formally as

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

where $\delta = [\delta_i]$ is an I x 1 vector of the indicator variables δ_i while $diag(\delta)$ represents an I x I matrix with δ running down the main diagonal. Those IAs which fail the edits are discarded. For those that remain (which are called feasible IAs),

$$D_{fpa} = \alpha D_{fa} + (1-\alpha) D_{ap} = (2\alpha - 1) D_{fa} + (1-\alpha) D_{fp}$$

is calculated where $D_{fa} = D(V_f, V_a)$, $D_{ap} = D(V_a, V_p)$ and it can be shown that $D_{fp} = D_{fa} + D_{ap}$. α is a parameter which was set to $\alpha = 0.9$ in the 1996 Canadian Census. D_{fpa} is a weighted average of the distance D_{fa} of the IA to the failed record and the distance D_{ap} of the IA to the donor. Placing an emphasis on minimizing D_{fa} (by having $\alpha = 0.9$), means that CANCEIS will tend to modify the data of V_f as little as possible through imputation. Placing some weight on D_{ap}, however, means that some importance is given to having a plausible IA, i.e. one that resembles a record that passed the edits without imputation. Only values of α in the range (.5, 1] are considered since with $\alpha < 0.5$, D_{fpa} becomes smaller as D_{fa} becomes larger (i.e. maximum change imputation!) while with $\alpha > 1$, D_{fpa} becomes smaller as D_{fp} becomes larger (i.e. donors that resemble the failed record less well are preferred!).

For the feasible IAs , the minimum value of D_{fpa} is determined and is labeled $\min D_{fpa}$. Any feasible IAs with $D_{fpa} = \min D_{fpa}$ will be called minimum change IAs. Those feasible IAs with a D_{fpa} that satisfies the equation $D_{fpa} \leq \gamma \min D_{fpa}$ where $\gamma \geq 1$ ($\gamma = 1.1$ in the 1996 Canadian Census), are called near minimum change imputation actions (NMCIAs) and are retained on a List of NMCIAs. Values of γ greater than 1 are allowed because the NMCIAs, for practical purposes (particularly with quantitative variables), are nearly as good as the minimum change IAs. IAs, which are not NMCIAs, are discarded because otherwise the principle of making as little change to the data as possible when carrying out imputation is being violated.

Only NMCIAs which are essentially new (i.e. no subset of the variables being imputed based on that donor would pass the edits) are retained. IAs that are not essentially new are discarded because one or more variables are being unnecessarily imputed. Doing this again satisfies the principle of making as little change to the data as possible.

A size measure $M_{fpa} = (\min D_{fpa}/D_{fpa})^t$ is defined for each of the NMCIAs. CANCEIS selects a single NMCIA for the failed record V_f with probability proportional to M_{fpa}. If t = 0, all NMCIAs will have equal probability of selection. If t = ∞, then all minimum change IAs will have equal probability of being selected and all other IAs will have zero probability of being selected. A value of t somewhere between these two extremes will usually be chosen so that minimum change IAs will be selected with somewhat higher probability than IAs with D_{fpa} close but not equal to $\min D_{fpa}$.

5. Imputation of essential variables and simplifying the edit rules

The initial IA is generated for a failed record/donor pair by first imputing all nonresponse variables. It is then determined if this initial IA fails the edits. Simultaneously, it is also assessed whether additional variables are always to be imputed for the feasible IAs generated by that failed record/donor pair and whether some edit rules can be dropped because they will never be failed.

To do this, CANCEIS starts by evaluating the first proposition (which will *be called* the j^{th} proposition) for the first edit rule in the sorted DLT to determine if the proposition has a constant condition result for all possible IAs. Let us assume, for simplicity, that « represents < for the j^{th} proposition. In addition, it will be assumed that the condition result

of $\Delta_j^0 < 0$ is T where Δ_j^0 represents the value of Δ_j for the initial IA. At least one IA can be generated where the condition result of $\Delta_j < 0$ is F (i.e. $\Delta_j \geq 0$) if $\max \Delta_j = \Delta_j^0 + \sum_{i+} B_{ji}^* \geq 0$ is true, where $\sum_{i+} B_{ji}^*$ represents the summation of those values of B_{ji}^* which are positive but only for variables not already imputed (i.e. $\delta_i = 0$). Otherwise, the condition result is constant.

If the condition result is constant over all possible IAs for the j th proposition, any edit rules that this proposition enters that do not match the constant condition result can be dropped since no IAs can fail these dropped edits. In addition, the proposition itself can be dropped since it is known that its condition result matches the remaining edit rules. This process is known as simplifying the edit rules.

If the condition result is not constant over all possible IAs for the j th proposition, it is determined, for each unimputed variable, if not imputing that variable will cause the condition result to be constant over all remaining IAs. Any such variable with this characteristic is called an essential to impute variable for that proposition. To reiterate, assuming again that both $\Delta_j^0 < 0$ and $\max \Delta_j = \Delta_j^0 + \sum_{i+} B_{ji}^* \geq 0$ are true, this means that the condition result is not constant. Then any unimputed variable with a positive B_{ji}^* and $\max \Delta_j - B_{ji}^* < 0$ is essential to impute for that proposition because any IAs which do not impute that variable will not be able to change the proposition's condition result. Section 7.2 documents similar methods used to simplify the edit rules and determine essential to impute variables when the condition result of $\Delta_j^0 < 0$ is F and/or when « equals > or =. The concepts of essential not to impute and inutile variables are also introduced in that section.

If the first edit rule is dropped or if the condition result for the proposition just analysed does not match the first edit rule (and hence the edit rule is not failed), CANCEIS takes the next leftmost available edit and identifies the first proposition not already dropped by the above method. Otherwise, the next undropped proposition entering the first edit rule is identified. This next proposition has the above process applied to determine if its condition result is constant (if it is, some edit rules may be dropped) and whether it contains any essential to impute variables. This process of evaluating rules and propositions continues until either no more edit rules remain or some edit rules remain but none are failed by the initial IA (in either case the initial IA passes the edits and CANCEIS stops because no other IAs would be essentially new for that donor) or the leftmost edit rule remaining has had all its propositions evaluated.

If all the propositions have been dropped for this leftmost edit rule, this means that it is impossible to generate an IA which passes this edit rule for the failed record/donor pair. This is because all the dropped propositions have a constant condition result for all the IAs and the condition results match those of this leftmost remaining edit rule. In this case, the process would start again with another donor. This situation can only occur for the initial IA if some variables that enter the edits are not allowed to be imputed (these are called unimputable variables). If, however, all nonmatching variables can be imputed, the resulting IA can become identical to the donor by imputing all these variables and hence at least one IA exists which passes the edits.

If all the propositions have not been dropped for this leftmost edit rule, it is determined if the initial IA passes this edit. If it passes, the processing described in next paragraph is carried out. If it fails this edit, the intersection of the essential to impute variables for the propositions remaining is determined and this intersection represents the essential to impute variables for this failing rule or essential variables for short. These are essential to impute because if they are not imputed it will not be possible to change the condition result for any of the propositions which enter this failing rule. These essential variables are imputed and the value of Δ_j^0 is updated to reflect this (this will be called the updated initial IA). It should be noted that even if the essential variables are imputed, the resulting IA may still fail this leftmost edit rule.

Then the next edit rule remaining to the right of the leftmost edit rule just processed is identified and the first undropped proposition in that edit rule (if any) is identified. This proposition has the process above applied to determine if its condition result is constant (if it is, the edit rules are simplified, if possible) and whether it contains any essential to impute variables. As the propositions are processed, edit rules are progressively evaluated, simplified and have essential variables imputed until the rightmost edit rule remaining has been processed or until the process terminates for a donor because all possible IAs fail an edit rule or it terminates because the initial updated IA passes the edits. If the processing terminates for that donor, it then recommences with a new donor. If the process has not terminated and if one or more essential to impute variables have been imputed, the edit rules are applied again starting with the leftmost edit rule. This iterative process continues until it terminates or until a pass from left to right through the edit rules does not result in any additional essential to impute variables

being identified.

If the updated initial IA passes the edits, CANCEIS does not generate any more IAs for that failed record/donor pair because they would not be essentially new. If the updated initial IA still fails the edits, CANCEIS applies the algorithm described in Section 6 to impute additional variables such that the optimal feasible IAs are generated. The simplified edits derived above will be used to determine if the IAs generated in Section 6 pass or fail the edits. Bankier (1999) provides some simple examples to illustrate the simplification of the edit rules and the identification of essential variables.

6. Imputation of Other Variables

The updated initial IA is the first IA to be placed on the Generating List. The first proposition in the leftmost edit of the simplified sorted DLT failed by the updated initial IA is identified. Then the leftmost (i.e. the first one listed by the user *in the DLT*) nonmatching unimputed variable in this proposition is imputed for the updated initial IA to create a new IA. This new IA is immediately discarded if its D_{fpa} is too large for it to be added to the List of NMCIAs generated by other donors. Otherwise, the algorithm specified in Section 5 is used to identify the essential to impute variables (if any) and determine if the new IA, after the imputation of these essential variables, passes or fails the edits. If it passes and its D_{fpa} is not too large, it is added to the List of NMCIAs. If it fails, it is added to the Generating List unless all IAs which can be generated from it fail the edits. Let us assume that the second IA is added to the Generating List. The next nonmatching unimputed variable in this leftmost edit failed by the updated initial IA is identified (looking at the first proposition entering this failing edit rule, then the second proposition entering etc.). Two new IAs are created by imputing this variable for the two IAs on the Generating List. If the second variable is already imputed in the second IA on the Generating List (because it was an essential variable), however, the second new IA is not created. These two new IAs are then assessed in a similar fashion to determine if they should be dropped, or should be added to either the List of NMCIAs or the Generating List.

Once all nonmatching unimputed variables in the leftmost failing edit rule have been used to generate IAs, the updated initial IA is dropped since any additional IAs generated from it will continue to fail the leftmost failing edit rule regardless of additional variables imputed. The IA remaining on the Generating List with the smallest D_{fpa} is then found. The leftmost failing edit rule of this second IA is identified and the process described above is repeated to generate more IAs. Once this second IA is dropped, CANCEIS selects the IA remaining on the Generating List with the smallest D_{fpa} and repeats the process.

Besides checking to see if new IAs should be dropped before adding them to the Generating List, it is also checked if IAs already on the Generating List can be dropped because any additional IAs that could generated from them would always fail the edits or because the D_{fpa} for these generated IAs would be too large to be added to the List of NMCIAs. Finally, IAs are dropped from the List of NMCIAs or the Generating List because they are not essentially new in terms of other IAs on the List of NMCIAs which were generated by the same donor.

The above process continues until there are no more IAs on the Generating List. If, at some point, there is only one IA on the Generating List, the current simplified edits, before any more IAs are generated, are replaced by the simplified edits for that single IA. In Section 7, it is shown that this process will generate all the NMCIAs for a failed record/donor pair. Bankier (1999) provides a simple example of the generation of IAs using this approach.

7. Concluding Remarks

CANCEIS, with its highly efficient editing and imputation algorithms, shows great promise for solving very general imputation problems involving a large number of edit rules and a large number of qualitative and quantitative variables when minimum change donor imputation is appropriate. The Fellegi/Holt minimum change E&I algorithm, however, should still be the method of choice for smaller imputation problems if there may not be sufficient donors available or if it is more appropriate to use another method to perform imputation.

References

Bankier, M. (1999), "Experience with the New Imputation Methodology Used in the 1996 Canadian Census with Extensions for Future Censuses", Proceedings of the Workshop on Data Editing, UN-ECE, Italy (Rome).

Bankier, M. (2000), "Imputing Numeric and Qualitative Variables Simultaneously", Social Survey Methods Division Report, Statistics Canada, Dated February 21, 2000.

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.