Published in Computers in Human Behavior, Vol 13, N0. 4, 1997, pp.487-503.

MODELS OF COMPLEX HUMAN SCREENING AND CORRECTING OF SOCIAL DATA

Svein Nordbotten
Department of Information Science
University of Bergen
N-5020 Bergen, Norway

Abstract:
National statistical agencies spend annually great budgets in collecting continuously a huge number of data on individual persons, households business activities, etc., to serve the information needs of a number of national and international, government and private users. Substantial parts of their budgets are consumed in checking and improving the quality of the data collected. Because of their complexities, these tasks have depended on the handling of specialists. To save both processing time and resources, to improve the processing for solving these tasks have high priority. The present paper outlines research carried out in Norway on using the neural network paradigm to improve the data quality checking and improvement in large scale data masses.

Keywords:
Neural networks, statistical editing, statistical imputation, data quality control, data quality improvement.

1. Background

Management of national affairs and business require extensive knowledge about the state and development in a country. Many countries have f.ex. held population censuses for centuries to determine the allocation of representatives to their parliaments from different geographical regions, to make fiscal decisions and to plan development of education, social support, health care and a series of other important tasks of modern country management.

In the United States, the US Bureau of the Census experienced in the previous century that the work load of processing the enormous amount of data which had to be collected in

censuses every tenth year became overwhelming, and a search for tools to reduce the load was initiated resulting in the first data processing equipment. In the middle of the 20th century, the amount of required information about each individual had increased and another problem appeared. The consistency and completeness of the data collected had so far been checked by a staff of specialists performing tasks which were named statistical coding, editing and imputation.

The coding task consisted in transforming textual information from the returned questionnaires to a standard numeric form convenient for automatic processing. The textual information might be about profession, education, etc. The responses could frequently be misspelled, in a number of different ways. A human coder was usually a specialist on the respective standard of professions or education, and used his special knowledge to interpret the written textual information and allocate the correct code.

The editing specialists had a more extensive task and were instructed to screen each individual record to confirm that the record gave an acceptable description of an individual. If a suspicious record was identified, it should be put aside for correctional treatment by a third group of specialists. It was usually impossible to list all types of acceptable individual profiles, and except for some general guidelines, much of the screening had to rely on the knowledge, experience and personal judgement of the specialists. The third group of specialists which had the responsibility for processing the suspicious records, had an even more difficult task. Should a new expensive collection be administered, or should the suspicious record be corrected according to the specialists' judgement? If the last approach was chosen, a partially new record was constructed in a process which has been referred to as imputation.

These quality control activities often counted for up to 50 % of multi-million dollar budgets for a population census or survey. For more than 40 years, efforts have been invested to develop methods for simulating the work of the specialists by means of electronic computers. It was recognized that these human processes were complex and difficult, if not impossible to specify. An extensive international cooperation headed by the United Nations and its subsidiaries engaged in the problem (Nordbotten 1963). During the next decades, statistical theory and methodology was developed and adjusted for computer-based editing (Kovas 1995). Still, the problem of specifying the criteria and rules for coding, editing and imputations was not satisfactory solved.

In a field far from statistics, theory and methods for modelling and simulation of activities in the human brain by means of computer by means of the neural network paradigm, were developed. The remarkable feature of these models was that subject to certain conditions, they were not only able to simulate human decision making, but also simulate human learning. In the last part of the 1980, the neural network approach had reached a state at which it could be applied within a wide range of fields (Rumelhart 1986).

Could the specification of statistical coding, editing and imputation processes be approached by applying the learning simulations of the neural network paradigm? In the following section of this paper, a short introduction to basics of the neural network paradigm and simplified models of human training for editing and imputation, are presented. In the third

section, a brief survey of Norwegian research in application of this type of models is given.

2. Neural Network Approach

2.1 Neural network preliminaries.

An artificial neural network can be conceived as a simplified model of a human brain composed by input, processing and output neurons mutually connected. A graphical model structure is illustrated in Figure 1. The components are extremely simple. Input neurons can receive a set of stimuli from f.ex. sensory cells represented as binary values. Stimuli to the input neurons are propagated through the network from left to right by means of connections each of which is characterized by a single, real number called the connection weight The left-side layer of neurons is called the input layer, in the middle is the hidden layer and at the right side is the output layer.

All hidden and output neurons transform their input to a binary response. The transformation function can be

$$r = \begin{cases} 0 & \text{if } \Sigma < \tau \\ 1 & \text{if } \Sigma >= \tau. \end{cases} \qquad (1)$$

where Σ = sum of all stimuli multiplied by the connection weights from the input neuron to the respective neuron. τ denotes the only parameter specified for the function and called the function's threshold value.

By propagating the input stimuli from a record through the network, either the output neuron A or the output neuron R will respond by emitting a stimulus activating the human motor system to put the questionnaire in the bin of 'accepted' or the bin of 'rejected' questionnaires.

A model of a neural network would not be complete with out including a model simulating learning. The learning mode in the type of neural networks used in the applications with which we are concerned , simulates a human learning from examples with tasks and the corresponding solutions. As an individual starts learning without any knowledge, i.e. the area of the brain we imagine is reserved for the considered task, has an arbitrary content, the model is initiated with small random connection weights. As the human learner, the model is presented with a set of examples of tasks which have been solved by a human teacher. For each case, the model is first required to try to solve the task itself and is then presented with the teachers solution.

The model simulates learning by adjusting its weights to obtain an improved solution to each problem according to

$$w'_{ij}-w_{ij}=(t_j-o_j)*s_i \qquad i=1..M,\ j=1..N. \qquad (2)$$

where w'_{ij} is the adjusted value of the connection weight from input neuron i to output neuron j, w_{ij} its value before adjustment, t_j and r_j are the values of teacher's solution and the value of the student's response, respectively, and finally s_i is the stimulus from input neuron i.

After having been presented to all available examples, the presentations are repeated to see how well the model have learned. To obtain a satisfactory learning, the process frequently have to be repeated a number of times before differences (t_j-r_j) are within predefined limits. The final test must be on a set of examples not included among those in the training set.

More complex models can include several layers of hidden neurons, recursive feedback connections, etc.

2.2 Simulating editing .

A simplified editing task for a junior editor is to sort questionnaires from a survey in two bins: accepted (A) and rejected (R) responses to the questions indicated in Figure 2. The accepted records can proceed to the next processing step, while the rejected records requires repeated observation or correction.

A record representing an individual in age category >=16 and in marital status category M is acceptable. However, if the next record contains age category <16 and marital group M, it should probably be rejected. Records with both age categories and/or both marital status categories either left blank or marked, should be considered invalid and rejected.

For this very simple task, the human editor has only to learn to distinguish 16 different types of records listed in Figure 3. Three of the possible types are acceptable. An easy solution would be to let a senior editor construct the list for his junior colleague to learn. Such a list could of course also be easily programmed for a computer without the need for a complicated neural network model.

After the list has been learned, we assume that only a tiny part of the brain network of the editor's brain will be occupied for this particular task. We assume that this part can be represented by the model in we introduced in Figure 1. The editor's eye cells, produce the stimuli to the 4 input neurons, i_1 .. i_4. The first 2 neurons receive stimuli from the age category fields of the questionnaire while the last 2 lines from the marital status fields. When the editor sees an recorded mark X in a age or a marital status category, the corresponding sensory cell emits a stimulus of strength 1 to the connected input neuron.

The input neurons are connected to 5 hidden processing neurons. The respective connections have all weights determining the influence of the stimulus on the behaviour of the receiving processing neuron. The sum of all stimulus*weight products is the input to the

processing neuron. Each of the hidden neurons a..e has a special task.

Neuron a has the task to keep track of and react if the number of stimuli from input neurons representing the age category fields exceeds 1. By multiplying the two stimuli from input neurons by weights 1 and setting a threshold 2 for neuron a, this neuron will according to the model in Figure 1, respond with 1 if 2 age category fields are marked. On the other hand, if both input neurons are inactive, i.e. emitting 0, because both age fields are blank, neuron b will receive a product sum equal to 0 which will cause the neuron to respond with 1 when its threshold is set to 0. As soon as one or both age fields are marked, the input product sum to neuron b becomes negative and the neuron will be inactive. To summarize, if neuron a responds with 1, both age category fields have been marked, and if neuron b responds with 1, both age categories are left blank.

Neurons d and e are connected to input neurons 3 and 4 in the same way, and the respective neuron will detect if both categories for marital status are marked or left blank.

The last hidden neuron c is set with threshold 2. It is connected to input neuron 1 (age: <16) and neuron 4 (marital status: M) by weights 1 while the connection between input neurons 2 and 3 to hidden neuron c are 0. Neuron c will respond with 1 only for the combination of age:<16 and marital status: M.

When one or more hidden neuron responds with a 1, the brain model signals that the record is unacceptable. We complete the model of the network with 2 output neurons, neuron R and Neuron A. Output neuron R is set with threshold value 1, while neuron A is set with threshold 0. Each output neuron is connected to all the hidden processing neurons, output neuron R with connection weights 1 and output neuron A with connection weights -1.

If one or more processing neurons are responding to input by 1, i.e. the record is unacceptable for some reason, the product sum input to neuron R will be >=1, i.e. equal or greater in value than the threshold, and the neuron will respond with 1 indicating that the record should be rejected. At the same time, the product sum to neuron A will be <0, and this output neuron will respond with 0 indicating that the record is not acceptable.

When all hidden neurons are responding with 0, the input product sum of neuron R will be 0, and the neuron will respond with 0 indicating that the record cannot be classified as unacceptable. At the same time, the input product sum to neuron A will also be =0, but this neuron has a lower threshold and will be activated and respond with 1 indicating that the record can be classified as acceptable. We can also see that if one or more of the hidden neurons are active, i.e. one or more unacceptable fact have been observed in the questionnaire, neuron A will because of its negative connection weight be silent, while neuron R will receive a product sum 1 or larger and become active responding with 1. If the model receives input stimuli, it will always respond with one and only one of the two output neurons.

We can easily convince ourselves that the model gives a correct solutions for each of the possible 16 combinations listed in Figure 3 by computing the input product sums to each hidden neuron, applying the transformation in (2) and computing the input product sums to

the two output neurons and again applying the transformation (2).

2.3 Simulating training

The connection weights values used in the editing model represent knowledge acquired by the editor. Before the model was trained for the particular task, these weights had random values reflecting no knowledge. Assume that the novice editors obtained their knowledge either by studying the how experienced specialists processed questionnaires or by somebody competent corrected their editing. After some time they have acquired knowledge for performing the process themselves as specialists. In the model, this learning process is represented by a gradual adjustment of the initially random weights to values representing the knowledge embedded in a set of exemplars consisting of questionnaires edited by competent editors.

To illustrate how the model is adjusted by learning functions, consider a simplified part of a net with two input neurons receiving stimuli from age recording, and a hidden neuron which does respond to one or two age markings, but not to two blank fields. The net is shown in Figure 4. The attached table illustrates how the initial weights, 0.07 and 0.05, according to formula (2) change for each record presented. Already after one presentation of the records, the weights of this partial net has adjusted to yield correct responses, i.e. it has 'learned' how to edit.

To simulate the learning of the complete network is slightly more complicated, and will usually require more than a single presentation of the set before it performs satisfactory.

2.4 Simulating imputation

The bin of rejected questionnaires can be treated in two ways. The respondents could be revisited and interviewed more carefully to obtain acceptable answer, or their original responses could be corrected. If only a part of the rejected questionnaires could be corrected without a significant quality risk, both resources and time could be saved.

Consider a questionnaire with age category <16 marked, but with both marital status categories blank. Three possibilities exist: (1) The <16 is correct, and the marital status should be U, (2) the marital status should be U, but the <16 is incorrect and should be >=16, or (3) marital status should be M, and the <16 is incorrect and should be >=16. If we from experience know that the conditional probability for an erroneous answer is low if a related field is blank, we may decide that the rejected questionnaires with either both age categories, or both marital status categories blank should be imputed, i.e. one of the blank categories should be filled in by the editor. This type of unacceptable questionnaires is named partial non-response. In the case mentioned above, the editor should obviously mark the blank marital status category U. If the age category marked had been >=16, the marking of a marital status category would have been more difficult. The editor may. however, from

his knowledge of the population know that the probability to do a correct marking would be significantly higher if he marked M and not U.

We can simulate the imputation by the editor by constructing a network model with four input neurons, 8 hidden neurons and 4 output neurons. We don't need to go in details about the interpretation of the hidden neurons and the connection weights, but assume that it is possible to construct the network such that if the input neurons receive stimuli of a partial non-response questionnaire, it will produce responses from the 4 output neurons which represent a valid and acceptable record which is frequently correct.

The specification of the network can again be considered as analogous to the learning of the editor. If the 4 record of Figure 2 with non-response to the age question, i.e. nos. 1, 2, 3, and 4, are used for training the network to make the imputations. In this case with only 2 attributes, neither the editor nor the model will ever learn to consistently impute correct categories. However, the more attributes there are in the questionnaire, the better the basis will be for correct imputations.

It is also possible to model a simultaneous editing and imputation process as we shall see in the next section.

3. Real size applications

3.1 Model dimensions

In real surveys, the number of attributes observed may be several hundreds including observations of categorical and continuous variables, as profession and income. The number of different combinations will be far too large for a formal enumeration and classification, and a systematic description of the rules guiding imputation of non-response would be impossible.

We may imagine a model of the working of an expert's brain receiving hundreds of input stimuli representing a multi-dimensional observation which are distributed to a large number of hidden neurons for parallel processing in several layers, which finally to deliver from a set of output neurons a response indicating an 'accepted', ' rejected' or imputed questionnaire. An outline of some of the problems is given in another paper (Nordbotten 1996b).

3.2 Editing of a population survey

In a study, carried out recently, the purpose was to investigate how well does a neural network model perform as an editor screening the data from a population survey, and making imputation when probable errors were detected (Nordbotten 1995).

To avoid any subjective opinions about what is an error, an experimental strategy to work with synthetic data was adopted. According to this strategy, a stochastic model which could generate synthetic descriptions of individual persons by 9 different attributes was developed. The attributes included were sex, age, marital status, geographical region, children born, education, industry, employment status and income. The model always generated valid records which was named 'true' records. The model had the capability to generate valid individual descriptions. When the individual data were aggregated, sums, averages, proportions and distributions corresponded to what was usually observed in a Scandinavian population. 12.000 synthetic records were randomly generated in two files of size 2.000 and 10.000 records, respectively.

Even the most thorough interview implies a risk for errors. To obtain data which could serve as observed data with possible errors, a second model which generated different types of errors according to specified probability distributions, was developed. This model superimposed different types of errors on the true records and the result was a set of 'raw' records. The researcher could specify different types of errors and error probability distributions. A number of different pairs of 2.000 and 10.000 raw records files were created representing different situations and problems. In this paper we shall only comment on one the situation in which partial non-response errors occur. The error model was specified to generate non-response in about 1/3 of all questionnaires which corresponded to Norwegian experience in similar surveys. Because of multiple non-response in some questionnaires, the total of non-response to questions was 3.318.

When imagining a human editor editing these raw records, a much larger part of the brain than in the previous section, needed to be trained. In the simulation model used, 54 input neurons, 300 hidden neurons and 54 output neurons were specified. This implied about 32.000 connections weights. It was also assumed that the transformation performed by each hidden and output neuron was more complex than indicated in the previous section:

$$r = 1/(1+e^{-\text{input produvtsum}}) \qquad (3)$$

This transformation has the property to return a response between 0 and 1. If the variable was a binary variable, the output was set equal to 0 if the response value was <0.5 and equal to 1 if the response was >=0.5. In case of a continuous variable, the response was scaled.

A pair of 2.000 true and raw records were used to train the network. The training was done by presenting the raw versions as input and the true version as output to the model. During the training session, the connection weights were adjusted to transform each raw record to a true record. To learn how to edit such records, will require a long training time for an editor, and so did the training of the network. Before the network simulated the editing of the 2.000 raw records of the training set satisfactory, it had to cycle through the set about 300 times.

Then the editor simulator was tested on the 10.000 raw records it had not been presented to. Among these, 2.902 records had one or more non-response errors. The records produced by the trained network from the raw records, was called edited records. The model reduced the 3.318 non-response question to 72 which the model did not manage to impute and left blank. Among the imputed non-responses, 288 variables were erroneous.

It should be noted that the experiment did not include a test of the performance of human editors on the material. As a matter of fact, the experiments told us how well the model learned from and performed compared with an ideal super editor who never made any errors. A test including trained editors would have given us information about how well the editors performed compared with the true material, and how well the model competed with the editors.

3.3 Improving the results from a population census

In the Nordic countries, official identification number have been in use for several decades, and administrative registers using these numbers as keys have also become an important data source for social researcher and statisticians. Already in 1980, the Danish Population Census obtained the data for the required population tables from administrative sources. The time and content of Population Censuses are determined by international agreements, and the next round of national censuses will be around year 2000. It is expected that the required content will exceed what can be obtained from administrative registers even in countries with well suited administrative registers, and it is anticipated that supplementary sample surveys may be needed.

In cooperation with the Norwegian Bureau of Statistics, a study of using the neural network approach was carried out in 1995 based on data from the previous census (Nordbotten 1996c). The main objective for the study was training a network on merged survey and administrative data from a sample of individuals to impute the values of 49 'survey' variables based on 96 values of administrative variables for each individual not in the sample. If these predictions were satisfactory, the predicted data would be a better basis for computing statistics for small groups than traditional sample survey estimators.

The study made use of population data for a municipality with 17.326 inhabitants which had paid extra to get all inhabitants surveyed. An experiment was designed in which 1.845 individuals were drawn to simulate a survey sample. 10 neural network models were specified and trained to predict survey variable values based on the values of the administrative variables for individuals. All models had 96 input neurons and 25 hidden neurons while the number of output neurons varied from 2 to 9.

The training of the nets required up to 2000 repeated presentations of the training sample records. After the training, the 15 records not used in the training were used as test basis. For each individual, all survey variables were predicted. In this way, two complete sets of

records with survey variable values became available for all individuals, the set of all observed values and the mixed set of 1.845 observed values and about 15.481 predicted values.

Statistics for the complete population and for different sub-populations were computed in three different ways, compared and evaluated. First, based on the sample of the 1.845 individuals, ordinary usual sample estimates for the municipality and different sub-areas, were computed. Second, based on the set of mixed records, aggregate statistics were computed for the same areas, and finally, based on the set of observed values, aggregates were developed as benchmarks for the first two sets of statistics.

The results from the study indicated that the statistics based on the mixed set of observed and imputed record values rendered statistics with acceptable accuracy for a breakdown into small areas with only a few hundred inhabitants, and significantly better results than the ordinary sample estimators for these areas. Work has also been done to develop methods for predicting the accuracy of statistics from the sets observed and imputed data (Nordbotten 1996d).

3.4 Coding information

Frequently, in statistical surveys the respondents are asked to fill out their profession, education, place of birth, etc. in a field of the questionnaire. Effective statistical processing requires, however, precise codes for each distinguished category within the attribute in question. The feasible categories is assumed to be defined in a list of unique names with attached numerical code. Leaving this coding to the respondents would require that huge code-books had to be distributed along with the questionnaires. Experience indicates also that the quality of the responses would be low. Traditionally, the agencies were left with the solution to have a staff of trained coders to transform the answers to the correct symbol code. During the last few decades, optical reading equipment has made it possible to transcribe the visual images of information written on the questionnaires to a machine-readable form.

Different types of written information represent different reading challenges. The easiest task is to read symbols with standard fonts from a printer. On the easy side are the well known mark reading, and reading of handwritten Arabic numerals. Both representing few reading problems for a coder. On the opposite side is the reading of handwritten information This problem is particularly difficult in statistical surveys because of the great variation in handwriting among the thousand of respondents.

In a statistical questionnaire, the expected response will frequently be a single word of limited length. It is therefore realistic to request the respondents to write their response by means of capital handprinted letters separated in a sequence of small frames. In the study to be described, we imagined a model of a human coder who inspected the questionnaires, read

the letters, and checked if the interpreted sequence of letters was within the list of feasible names. If not, the coder determined the most likely correct sequence based on the letters identified and knowledge about spelling errors.

The first part of the task was assumed carried out by comparing the image in each individual letter frame with the patterns of letter variations the coder knew and selecting the most probable letter. The second part was assumed to be done by checking the sequence of identified letters against the list of valid sequences. If there was a match, the code was attached to the questionnaire.
In the case of no match, the coder was assumed to consider for each name in the list if the current sequence could be a misspelling. The coder would consider the possibility of letters added to or deleted from the correct spelling, etc. The less changes needed in the current sequence to match a name in the list, the higher probability was assigned to the name as a correct candidate for the sequence. After the current sequence had been analysed against each name in the list, the name assigned with the highest probability was selected as the correct name, and its code assigned to the record.

In a recent study, the coding of place of birth was used as a the task to be done. The universe of birth sites were limited to a list of 368 US cities. In each questionnaire, one of the city names was represented as a sequence of letter patterns. To simulate the coder's learning of different versions of the 26 capital letters in the English alphabet, a set of standard fonts was designed as patterns formed by a 8*8 grid as shown in Figure 5. Each pattern had 64 rectangles which were either white (value 0) or black (value (1). Twenty different versions of each letter were generated with small random variations of the standard pattern. The 520 patterns each assigned to a specific letter, was used to train a neural network which received each pattern as a binary vector with 64 elements.

Several test sets of randomly drawn city names were obtained from the list. By means of a model simulating misspelling, each city name had the risk of being scrambled in different ways. The risk of scrambling was varied for each set of city names. Then each resulting letter sequence was represented by the corresponding sequence of standard letter patterns shown in Figure 5. Then each pattern was randomly distorted simulating the respondents' individual handprinting variation by a model similar to the one used for creating different versions of each font pattern. Different distorting probability models were used on different test sets. A handprinted city name would thus be represented by a sequence of patterns.

A hybrid model consisting of a module for recognising distorted letter patterns, and a module for determining the most probable resolution to a misspelled name, was implemented. The different test sets were processed by the model simulating the work of human coders. The experiments gave a number of interesting results. The recognition of handprinted names without spelling errors seemed to be almost perfect. A test set including city names which had either extra letters added or deleted from the city names, and the letter patterns of the scrambled names distorted seemed also to do quite well.

The coding activity described in this secton can be considered as a special case of an editing and imputing problem.

4. Final remarks

The results of studies on using computerized models of complex tasks usually performed by human specialists in the domain of large scale data processing in official statistical information systems, indicate that model paradigms from the field of Artificial Intelligence are promising.

The studies referred to have in part been carried out on synthetic data. The advantage of a research strategy using synthetic data is that influence of unwanted factors can be eliminated, and conclusions be strictly limited to the factors studied. The serious disadvantage is, however, that important real life aspects may be overlooked. For this reason, research on real data is now being carried out tyo confirm the conclusions drawn from the synthetic data.

5. Acknowledgement

The present paper and the studies on which it is based, was prepared as part of the author's research duties at the University of Bergen. Parts of the work referred to was carried out under cooperation contracts between the Statistical Information System (SIS) project at the Department of Information Science, and the Central Bureau of Statistics in Norway, and Statistics Sweden. The project has also benefitted from information exchange with a number of other national statistical agencies.

6. References

Kovar, J.G. and Whitridge, P.J. (1995): Imputation of Establishment Survey Data, Business Survey Methods.

Nordbotten, S. (1965): Automatic Editing of Individual Statistical Observations, Statistical Standards and Studies, Handbook No.2, United Nations, N.Y.

Nordbotten, S. (1995): Editing Statistical Records by Neural Networks, Journal of Official Statistics, Vol. .. No.4, pp 391-411.

Nordbotten, S. (1996a): A Model of Automatic Coding from Boxes of Hanprinted

Characters. Proceedings of the 1996 Annual Research Conference, US Bureau of the Census. US Department of Commerce. Washinton , DC. pp. 701-809.

Nordbotten, S. (1996b): Editing and Imputation by Means of Neural Networks. Statistical Journal of UN/NC. IOS Prees. Vol. 12, No. 4. pp. 119-129.

Nordbotten, S. (1996c): Neural Network Imputation Applied on Norwegian 1990 Population Census Data Utilizing Administrative Registers. To be published in Journal of Official Statistics.

Nordbotten, S. (1996d): Predicting the Accuracy of Imputed Proportions. Department of Information Science. University of Berrgen.

Rumelhart, D.E. and McClelland, J.L. (1986): Parallel Distributed Processing - Explorations in Microstructure of Cognition, Vol. 1: Foundation. MIT Press, Cambridge, Mass.