Laboratory of Data Analysis
University of Jyväskylä

EUREDIT - WP6 reports

Evaluation of SOM based Editing and Imputation

(3rd corrected version, May 20. 2003)

Pasi P. Koikkalainen - University of Jyväskylä
with contributions from
Ismo Horppu - University of Jyväskylä
Pasi Piela - Statistics Finland

Contents

1 Introduction

The Self-Organizing Map (SOM) was originally developed by Teuvo Kohonen (1) to imitate the formation of orientation specific neural cells in the brain. Despite the biological plausability of the algorithm remains an open quenstion, the SOM algorithm has been a success in many fields of computational and artificial intelligence.

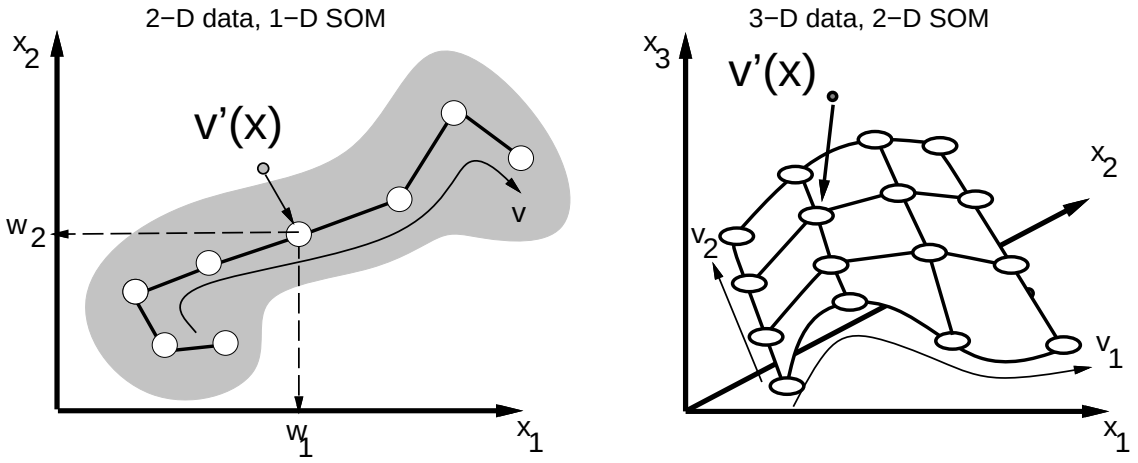
The SOM algorithm and its variants can be characterized and related to other computational and statistical algorithms via three points. If $\mathbf{X}(\omega) = \{X_1(\omega), X_2(\omega), \ldots, X_m(\omega)\} \in \mathbb{R}^m$ is a randomly selected m-dimensional observation $\omega \in \Omega$, then the SOM is

i) a multivatiate algorithm that models the joint distribution $\Pr\{X_1, X_2, \ldots, X_m\}$ of data.

ii) a projection algorithm that constructs a lower dimensional latent space $\mathbf{V} = \{V_1, V_2, \ldots, V_s\}$ (or surface), where $\mathbf{V} \in \mathbb{R}^s$ such that $s \leq m$. Thus the SOM is a mapping from $\mathbb{R}^m$ to $\mathbb{R}^s$, defined by $\mathbf{v}(\mathbf{x}) : \mathbf{X} \rightarrow \mathbf{V}$.
Typically the dimension of SOM is two ($s = 2$), which allows one to describe m dimensional data on a two dimensional surface as shown in figure 1b. This is useful for human assisted data analysis.

iii) a clustering algorithm. The implementation of the SOM uses a discrete set (lattice) of nodes (neurons) to construct the surface $\mathbf{v}$. These nodes can be intepret as data clusters that are smoothed (they borrow strength) along the SOM lattice.

Figure 1: Illustration of 1-D and 2-D SOMs in two and three dimensional data, respectively. An example projection of a data point to the closest SOM node is also shown.



1.1 Brief introduction to the theory of SOM

The characterization *ii)* relates SOM closely to principal curves and surfaces (4), (5), that are defined as smooth regression surfaces

$$\mathbf{x}'(\mathbf{v}) = \mathbb{E}[\mathbf{X} \mid \mathbf{v}'(\mathbf{X}) = \mathbf{v}] + \lambda R_v, \tag{1}$$

where $\mathbb{E}$ denotes expectation, R_v is a smoother (regulator) with some predefined regularization parameter λ, and the mapping $\mathbf{v}'(\mathbf{x})$ projects $\mathbf{x}$ to the closest point $\mathbf{v}$ on the latent space, as defined by

$$\mathbf{v}'(\mathbf{x}) = \arg\min_{\mathbf{v}''} \|\mathbf{x} - \mathbf{x}(\mathbf{v}'')\|. \tag{2}$$

The SOM implements the principal surface $\mathbf{v}$ via a discrete lattice, which can be denoted by replacing $\mathbf{v}$ with $\mathbf{v_i} \in L_s$, where L_s is the set of indices and expressing node locations $\mathbf{x}'(\mathbf{v}_i)$ as weight vectors $\mathbf{w}_i = \mathbf{x}'(\mathbf{v}_i)$. The equation (2) can then be written as a search of the best matching node (bmu)

$$bmu = b(\mathbf{x}) = \arg\min_i \|\mathbf{x} - \mathbf{w}_i\|. \tag{3}$$

A possible embedding of the discrete SOM in data is illustrated in figures 1a and 1b. One should note that the SOM tends to fold inside the data and that all variables $x_1, x_2, \ldots x_m$ are treated in an unsupervised way such that the SOM models the joint distribution of them.

1.1.1 A basic training algorithm

One commonly used training algorithm for the SOM uses a batch iteration where the following steps are repeated until the model converges.

1. A best matching unit (SOM node) is searched for each observation $\mathbf{X}(j)$ of the data set. This divides data between the nodes such that for each node i there is a set of best observations $\Omega_i = \{\mathbf{X}(j) | i = b(\mathbf{X}(j))\}$.
2. The node positions are updated to the (smoothed) centroids of the *bmu* sets:
   $$\mathbf{w}_i = \frac{1}{N_i} \sum_{j \in \Omega_i} \mathbf{X}(j) + \alpha \sum_{k \in Ne(i)} \frac{1}{N_k} \sum_{j \in \Omega_k} \mathbf{X}(j),$$
   where N_i is the cardinality of set Ω_i and $Ne(i)$ defines a lattice neighborhood of SOM node i. Neighborhood implements a kernel smoothing for the SOM that ensures that similar nodes are close to eachother in both $\mathbf{V}$ and $\mathbf{X}$ spaces.

The smoothed updating, step 2., introduces is a difference between the SOM and many other clustering algorithms. The SOM cluster i borrows "strenght" from similar neighbor clusters. Due neighbor smoothing the SOM approximates principal curves (PCs). Differences between the PC and SOM decreases when the number of SOM nodes is increased, as noted by Ritter at al. (6), but this might be difficult formalize with some SOM implementations.

2 SOM Method in EUREDIT-project: The NEAT-DATA algorithm (NEw Algorithm based on Tree-structured self-organizing maps for erroneous and incomplete DATA)

In the EUREDIT-project a tree-structured variant (TS-SOM) of Kohonen's original SOM has been used and modified for the purposes of data editing and imputation. This new algorithm is named as the NEAT-DATA-algorithm.

2.1 Method description

The NEAT-DATA algorithm provides several computational and methodological enchangements to Kohonen's original SOM algorithm. The most important of these features are the following ones.

i) A tree-structured multiresolution learning algorithm (TS-SOM) is a computationally fast method that can be used with large data sets. The TS-SOM is also easier to get well trained than the normal SOM because it does not require experiments to find good parameter adjustmets for different data sets.

ii) Robust estimation of SOM parameters (weights $\mathbf{w}_i$), provides some insensitivity to errors in data. The method uses Huber estimator for continuous variables and a "cut probability" for categorial variables.

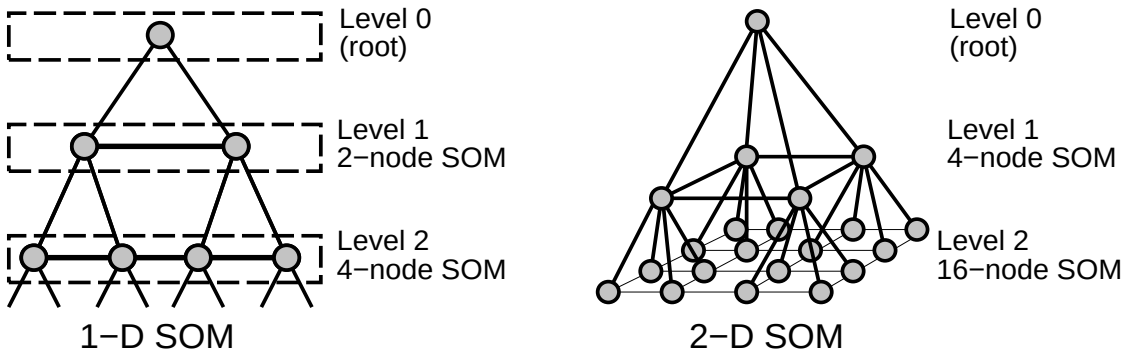
iii) An incomplete data training algorithm that allows one to train the SOM with partially observed samples. This algorithm shares charecteristics with EM-algorithms and multiple path methods.

iv) Several imputation methods that are assisted by the SOM model of joint disrtibution $\Pr(X_1, X_2, \ldots, X_n)$.

2.1.1 The Tree-Sturctured Self-Organizing Map

The TS-SOM algorithm (2) combines a multiresolution representation of the SOM and a tree-search of the best matching unit *bmu*.

Figure 2: The Structure of the Tree-Structures Self-Organizing Map:
a) 1-D TS-SOM, b) 2-D TS-SOM.



When training TS-SOM, several SOMs (layers) with different resolutions are trained, strating from simple ones and increasing the number of nodes by 2^S times when a new layer is introduced, where S is the dimension of the latent SOM surface. Layers are connected such that each node is parent of exactly 2^S child nodes on the next layer, as depicted in figure 2. The number of nodes on layer l is therefore 2^{lS}.

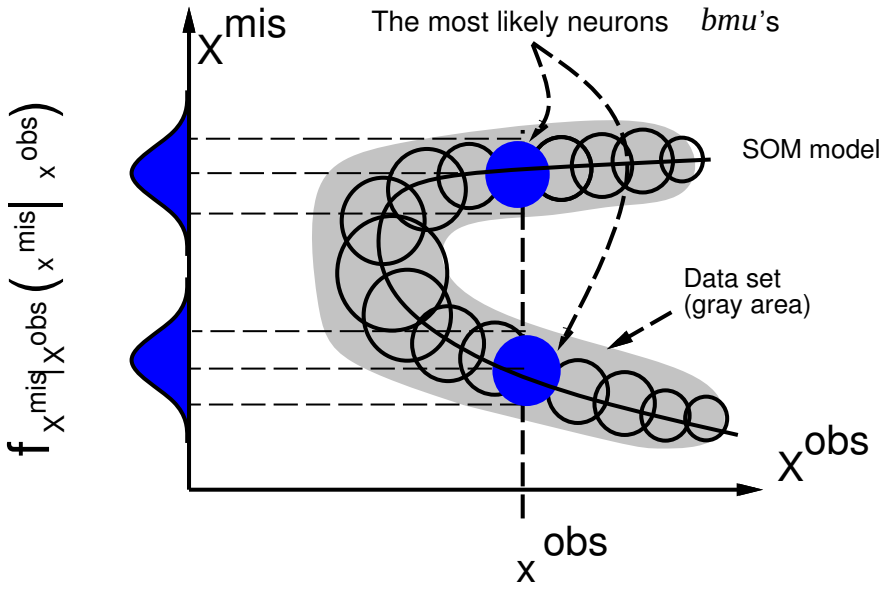
The tree-like structure of the TS-SOM is useful in two ways: i) it can be used as a search tree to speed up the search of the best matching node $b(\mathbf{X}(t))$, and ii) it can be used as a constructive estimator of the data, from highly smoothed solutions (less nodes) to more complex ones.

2.1.2 An Incomplete Data Training Algorithm for SOM

There are currently two versions of the incomplete TS-SOM training algorithm, a multipath version and a random path version. Both algorithms try to incorporate the uncertainty of selecting the best matching node when training with partially observed items.

In the multipath version this is done by replacing the search of one (and only) best matching unit, equation (3), with several, Nb best matching nodes for the given partial observation $\mathbf{x}^{\text{obs}}$.

Figure 3: The role of multiple canditate nodes when unisng SOM with incomplete data.



This idea is illustrated in figure 3, where the best SOM node is not unique for a partial observation $\mathbf{x}^{\text{obs}}$. Therefore several possible canditates are selected. To ensure distribuitonal accurancy, the use of canditates in SOM training and in the imputation of missing values is weighted according their posterior probability $\Pr(i|\mathbf{x}^{\text{obs}})$, which is computed during the search of the best matching units.

2.1.3 Robust training for TS-SOM

Robust training of the TS-SOM is based on an outlier detection. The method examines the distribution of each variable of an input vector with respect to the corresponding distribution in the best matching SOM node (bmu). Those variables, which are considered to be "out of range" from the distribution of the best matching node are marked as outliers. For simplicity the distributions, modelled by SOM nodes, are expected to be symmetric (only diagonal values of the co-variance matrixes are estimated).

For continuous variables the "out of range" criteria is controlled via a training parameter σ_1 (sigma1), There are two implementations of the method. In the first method the outliers are simply omitted. This is done by marking those values as missing. The second robust training option is to use Huber M-estimator, which generally seems to work better than the "mark as missing" alternative.

For categorial variables similar idea was implemented using classification probabilities of the categories.

2.1.4 Preprocessing and variable coding.

The use of the SOM requires that all data are coded into real valued vectors, where the difference between two observations can be measured in Euclidean vector distances. The organization of SOM depends on these distances, which implies that the variabe ranges and the lenghts of the vectors have an affect to the SOM model. Therefore the following preprocessing options are used, if required.

Equalization of variable ranges.
In the min-max equalization the equalization is: min-max-eq: $[\min X_r, \max X_r] \rightarrow [0,1]$.
In the robust equalization fractiles are used: rob-1%-eq: $[\text{frac}_{1\%} X_r, \text{frac}_{99\%} X_r] \rightarrow [0,1]$.
In the variance equalization the equalization is: var-eq: $[\overline{X}_r - \text{std}_{X_r}, \overline{X}_r + \text{std}_{X_r}] \rightarrow [0,1]$.

Normalization The normalization scales the lenght of the data vectors such that they are on the surface of a unit hypesphere, $\|\mathbf{X}\| = 1$.

Log-transform makes the following transformation: $X' = \log X + 1$.

The coding of categorial variables is done via dummy coding, where a new indicator variable is created for each category. Thus each categorial scalar variable is replaced with a vector of zero-one indicators:

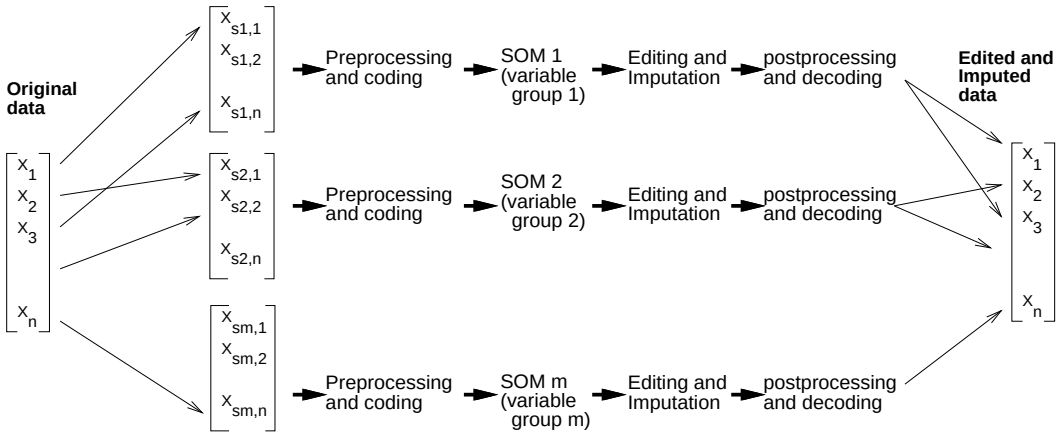
$$X^{\text{cat}} \rightarrow \mathbf{X} = \begin{bmatrix} X_{\text{cat1}} \\ X_{\text{cat2}} \\ \dots \\ X_{\text{catN}} \end{bmatrix}, \text{ where } X_r = \begin{cases} 1 \text{ if } r = \text{cat}_r \\ 0 \text{ if } r \neq \text{cat}_r \end{cases}$$

Special values of variables like, *not applicable*, *not asked*,*can't say*, etc. can be coded as separate categories or as missing values during the training of the SOM model. The appropriate choice depends on the model design. After training the original values are (usually) returned. Sometimes zero values of continuous variables are also categorized as a specific ZERO category to ensure the preservation of values like zero income, for example.

2.1.5 Variable selection for SOM based editing and imputation

Because the SOM models a joint distribution $\Pr(X_1, X_2, \dots, X_n)$ of the given input data set, the performance of the method in editing and imputation depends quite strongly about a proper selection of variables. **Note a difference to supervised methods: the editing and imputation is NOT conditionalized with a set of background variables. All model variables are "equal".**

Figure 4: The role of variable selections with SOM models in editing and imputation.



The problem with the SOM model is that it is sensitive to all types of scatter in data, not only to those effects that are behind the errors and/or missingness. On the other side, the method does require strong assumptions about the causes of bad data. With incomplete and robust training options, **the method does not require clean training data**.

In EUREDIT experiments the variables of the data sets were groupped into several SOM models, as depicted in figure 4. Then the same SOM model was used to edit and impute all variables of the model. There were typically 4-8 variables per SOM model. In the case of categorial variables the input dimension of the SOM training vectors can be large (over 30 dimensions) due dummy coding.

2.1.6 Parameters for editing and imputation procedures

There are a couple of parameters to control the training and the use of the TS-SOM model. In the experiments the parameters were selected in order to optimize the performance of editing and imputation. These parameters are summarized as follows.

SOM layer determines the complexity (smoothness) of the SOM model. This is defined by the number of SOM nodes, which is itself defined by the SOM layer. The bigger the layer is the more there are nodes (data clusters) and the more complex the model is.

sigma1 (for continuous variables only) defines the robustness of the SOM training algorithm. Observations that are over the distance sigma1×STD from SOM nodes are considered to be outliers, where STD is the standard deviation. Small sigma1 makes the SOM more robust.

sigma2 (for continuous variables only) controls the SOM editing procedure, which is used after the training. It scales the variance of the of the estimated distribution around the SOM surface. The probability of error of an observation increases when sigma2 decreases.

Train cut (categorial variables only) is the "cut probability" that marks the observation as an outlier during the training if the posterior probability of category in SOM cluster is smaller than **Train cut**.

Edit cut is the "cut probability" (Pr_cut) that marks the observation as an error. For categorial variables this vehaves like **Train cut**, and for continuous variables as described below.

The reader should note that the NEAT-data algorithm was under development during the EUREDIT project. Because of this, the role of the parameters can be a little confusing[1]. In the case of continuous variables the role of the parameters is illustrated in figure 5. The parameter **sigma1** controls the robustness of the algorithm during the training by scaling the estimated standard deviation of the local data clusters. The problematic part of the algorithm is related to parameters **sigma2**, and **Edit cut**, which are adjusted for outlier detection and imputation after the training. Unfortunately the effect of these parameters to outlier detection and imputation is strongly dependent of eachother. As a default one would expect that **sigma2** is close to one and **Edit cut** to be something like 0.96. In some cases, however, one needs large **sigma2** and **Edit cut** that is closer to zero than one. This type of behavior can be seen in some of the experiments of this document.

2.1.7 Editing procedures for SOM

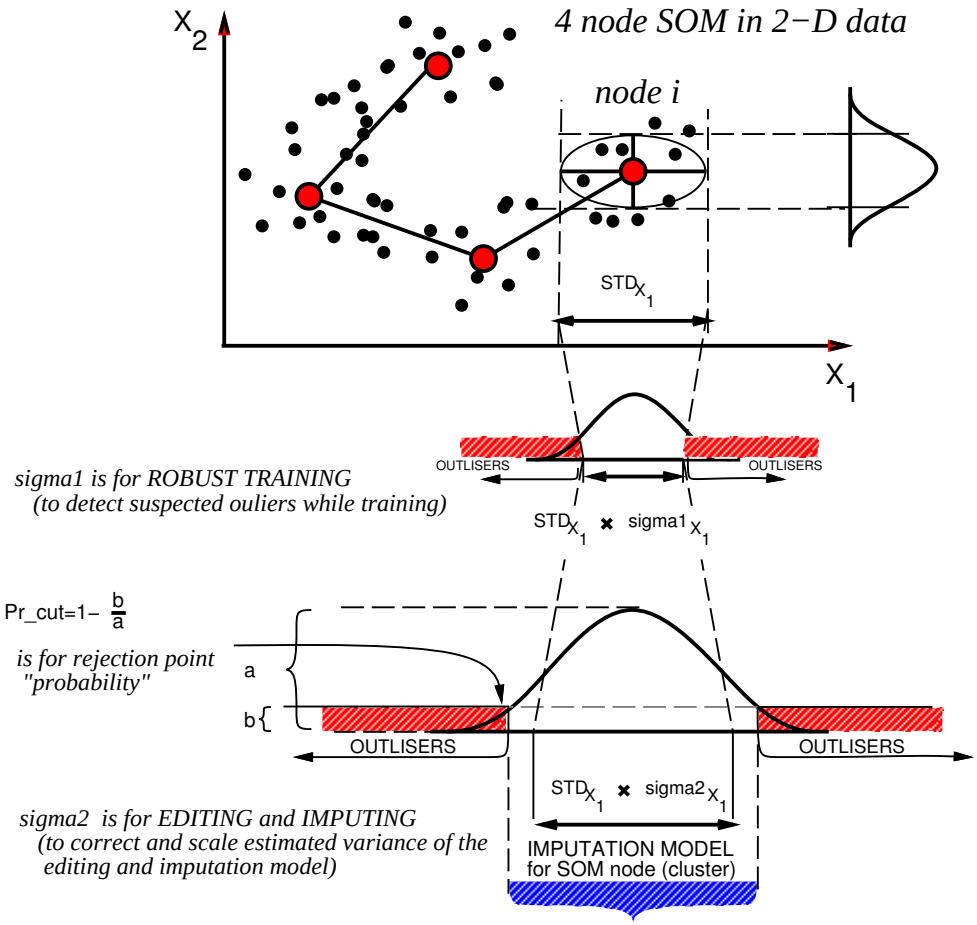
After training the TS-SOM with robust options the SOM can be used for outlier detection. First for each observation the best matching unit is searched. Then for each variable we examine its distance from the mean and compare this to some predefined distance, which should be defined as a function of error probability.

It is more difficult to decide the probability of an error since there is no objective criteria for this. The method assumes that true data is normal distributed with mean $w_{i,r}$ and variance $\mathrm{var}_{i,r} = (\mathrm{std}_{\mathrm{bmu},r} \times \mathbf{sigma2})^2$, but it lacks exact definition for erroneous samples.

The final probablility of an error is selected to be

[1] We have improved the technology after the experiments, described this document, were made.

Figure 5: An illustration about the role of training, imputation and editing parameters.



$$\Pr(\texttt{error}|X_r) = \begin{cases} P_e & \text{if } P_r^{\text{cut}} < P_e \leq 1 \\ 0 & \text{if } P_e < P_r^{\text{cut}}, \end{cases} ,$$

where $P_e = \frac{N(X_r, \text{var}_{i,r}) - N(0, \text{var}_{i,r})}{N(X_r, \text{var}_{i,r})}$, and P_r^{cut} is the **Edit cut** parameter for variable r. The parameter $\sigma_{\mathbf{2}}$ reshapes the normal assumption of correct data, if nessessary. One should note that there is a relationship between the P_e and a P-value. We know now (but it is too late) that we should have used P-values rather than P_e, because P-values are easier to intepret.

In case of **categorial variables** $X_r \in \{0, 1\}$ even simpler outlier detection mechanism was used. Now

$$P_e = |w_{i,r} - X_r| \quad \text{which can be intepret as } |\overline{X}_{r|i}^{smoothed} - X_r|$$

and

$$\Pr(\texttt{error}|X_r) = \begin{cases} P_e & \text{if } P_e > P_r^{\text{cut}} \\ 0 & \text{if } P_e < P_r^{\text{cut}}, \end{cases}$$

.

2.1.8 Imputation procedures for SOM

We have implemented six different imputation procedures. All imputation routines use the TS-SOM as a model of "correct" distribution. First step of the imputation is to find a set of `Nb` best candiatate

neurons (clusters). Imputation is done using one neuron i only, which can be selected according the posterior probability $\Pr(i|\mathbf{x}^{\text{obs}})$.

For all SOM nodes a local density approximation of data is made such that $\mathbf{w}_i$ approximates the local mean of data and $\sigma_{i,r} = \text{std}_{i,r} \times$ **sigma2** is a local measure of spread from the local mean.

Actual values are chosen according to local neuron statistics, where the six possibilities for missing X_r^{mis} values are:

0 use mean values
$X_r^{\text{imp}} = w_{i,r}$

1 pick a random sample from truncated Normal pdf.
$X_r^{\text{imp}} \sim N(w_{i,r}, \sigma_{i,r}^2)_{|w_{i,r}-2\sigma_{i,r}<X_r^{\text{imp}}<w_{i,r}+2\sigma_{i,r}}$

2 pick a random sample according to uniform pdf.
$X_r^{\text{imp}} \sim U(w_{i,r} - \sigma_{i,r}, w_{i,r} + \sigma_{i,r})$

3 Use random donor from cluster i
$X_r^{\text{imp}} = X_j$, where $j = Rand(0..N_i), j \in \Omega_i$

4 Use nearest neighbor donor from cluster i
$X_r^{\text{imp}} = X_r(k)^{\text{obs}}$, where $k = \arg\min_{k'} \|\mathbf{X}(k')^{\text{obs}} - \mathbf{X}(j)^{\text{obs}}\|$

5 Use node specific MLP regression model for imputation
$X_r^{\text{imp}} = f_r^{\text{MLP}_i}(\mathbf{X}^{\text{obs}}|i)$

Methods 0,1 and 2 are **model based**, where everything is determined by the SOM model. The imputed values are taken from the data model of the "best" SOM node i for the given observation $\mathbf{X}_r$.

The random donor method, method 3, is a **model assisted Hot deck** imputation system, where the selection of the donor subset is determined by the "best" SOM node. As the number of SOM nodes increases, model complexity grows, and the donor subset becomes smaller until there is only one observation per SOM node. Then the system behaves like the nearest neighbor donor imputation system.

Methods 4 and 5 are SOM assisted hyprid methods, where a nearest neighbor, 4, or a MLP neural network, 5, is used for data that is captured by a SOM node. A notable difference between these and the rest of the methods is that different background varibles can (and sometimes must) be used for the SOM and the imputation method for the local subsets. The disadvantage of our implementation of the MLP method is that it can take only fully observed values as inputs.

2.1.9 Software issues

A software for SOM based Data Editing and Imputation has been developed in the university of Jyväskylä (JyU) by the research group on Software Engineering and Computational Intelligence (SE&CI). Software has been build on the top of NDA (Neural Data Analysis) software platform. In EurEdit project some new methodology for data editing and imputation has been implemented in the NDA kernel, as well as a new user interface, specially made for editing and imputing, has been build.

The software for data editing and imputation is our attempt to cover all aspects of typical *data production process* (DPP), as we think it is done in official statistics and industrial data management.

We consider the following tasks as fundamental requirements of what our software should be able to do.

a) Data manipulation, reorganization and visualization are common tasks in data analysis. These are already implemented in the NDA, but some effort has been done, and still need to be done, to make these operations user friendly.

b) Use of external knowledge, such as edit rules, must be supported. We have done this with a simple rule converter that translates rules of Euredit data sets to out NDA type of expressions.

c) Variable selections and case spesific edit/imputation operations should allowed. The user should be able do them with minimal effort and see easily all the choices and selections he has done.

d) Experimenting and playing with data should be easy.

e) There should be support for the visualization of results, summaries etc.

For evaluation purposes a DLL-library version of the software has also been made. This has been made by the request of NAG and it will be included in the NAG evaluation software of the EUREDIT project.

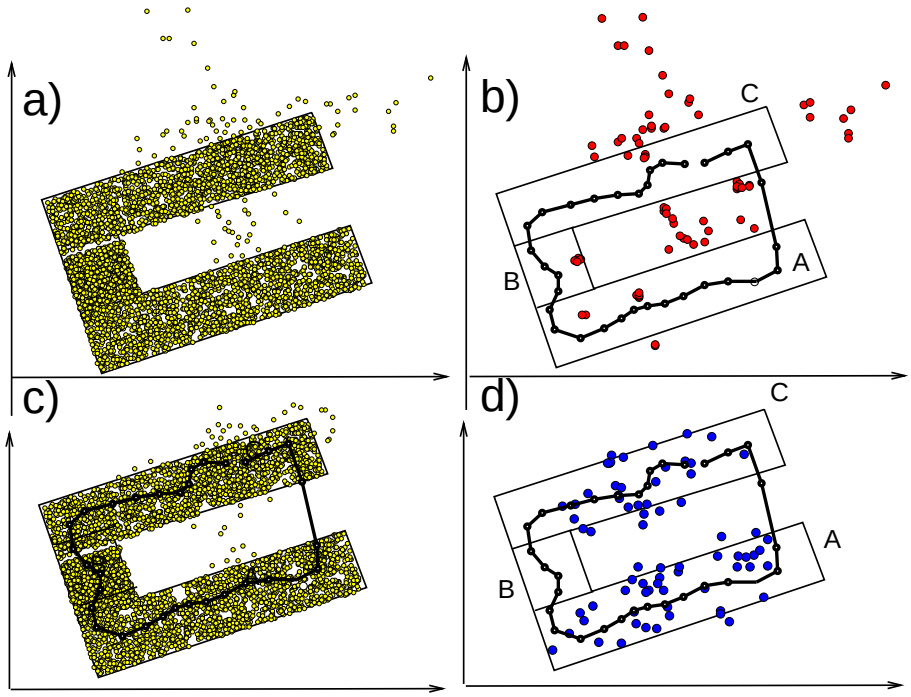
2.1.10 An illustrative example

In the following toy example some potential errors (200 points) have been added to uniformly distributed data sets in regions A, B and C, where there are 1600, 800 and 1600 clean data samples, respectively. This data is shown in figure 6 a).

For model building a rejection boundary $\sigma_1 = 3 \times$ std was used. After training, all samples out of $\sigma_2 = 6 \times$ std were marked as errors, as shown in figure 6 b). The cleaned data is then shown in figure 6 c).

After error detection the erroneous variables of the observations have been marked as missing. Their new positions have been imputed using SOM with Normal pdf rule. The result of outlier imputation is shown in figure 6 d).

Figure 6: An illustration of error detction and imputation with the TS-SOM.



2.2 Evaluation of data sets with SOM

This evaluation consists of a brief description of SOM based data analysis for LFS and ABI data, a technical description of all experiments for LFS, ABI, SARS, EPE, and GOESP data sets, and a brief discussion of the main results. Because the cross comparison of methods is done separately for each data set, this presentation omits most of the results from other EurEdit partners. Only ONS experiment with nearest neighbor imputation is used as a baseline with all SOM evaluation results.

The experiments named with initial J have been done by the Univeristy of Jyväskylä (JyU), while the experiments with initial F were made by the Statistics Finland (StatFI). The different viewpoint of JyU and StatFI is that JyU people are computer scientists while StatFI are representatives of national statistical institutions (NSIs). During 2001 and 2002 Statistics Finland has been testing and evaluating different versions of NDA Editing and Imputation system (a software for SOM), that was developed in the University of Jyväskylä.

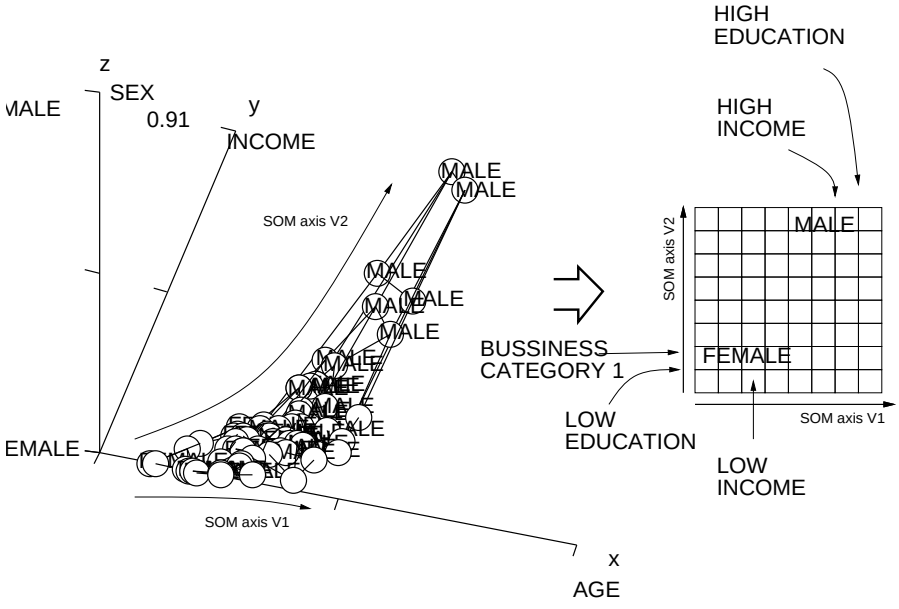
Statistics Finland has always used SAS System for Windows very intensively in testing any method for comparisons, data-analysis, data preparations, checkings, upgrades etc. This Integrated modelling approach to imputation and error localisation (IMAI) is presented in The Standard Methods.

However, it should be noted that NDA/DPP, done by JyU, does not actually require any other program. Anything related to data analysis or preprocessing can be made by NDA/DPP itself, as it has been done in the experiments by the University of Jyävskylä.

2.2.1 Dataset: The Danish Labour Force Survey Y2 (DLFS)

In the DFLS experiments all the SOM models use the INCOME variable and several background variables, which typically include AGE, SEX and BUSSINESS. The INCOME varaible is continuous, the AGE can be used either as a continuous or as a categorial variable, and the rest of the variables are categorized via dummy coding. One SOM model is build for an experiment because there is missingness in the INCOME variable only.

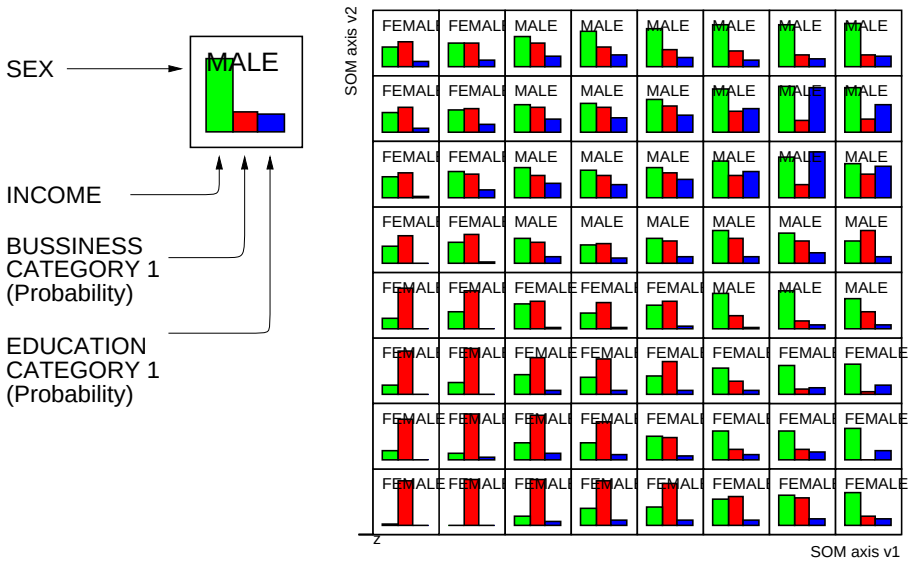
Figure 7: A SOM model that is organized in LFS data and an illustration of the idea of SOM based data analysis.



A typical organization of the SOM using DLFS data is illustrated in figures 7 and 8. The first figure

7 shows how the SOM is positioned in the data set. The same SOM is also shown in the second figure 8, which depicts some representative LFS variables after projecting the data set onto the SOM lattice, as it is often done in SOM based data analysis. The figure reveals clearly that SOM is able to capture relationships between the INCOME, SEX, EDUCATION and BUSSINESS variables.

Figure 8: An illustration of some LFS variable dependensies using SOM.



Originally six SOM experiments were made by the University of Jyväskylä (JyU, J) and one by the Statistics Finland (StatFI, F). To be able to compare the results with standard methods StatFI made also a control experiment (FL20001) using nearest neighbor method.

Table 1: Summary of SOM experiments with the DLFS data set.

Experiment		Description	SOM nodes	runtime
JL20001		SOM + Mean imputation	4096	21 Sec
JL20002		SOM + MLP	64	127 Sec
JL20003	*	SOM + Random donor	4096	56 Sec
JL20004		SOM + Normal pdf (from data)	4096	201 Sec
JL20005	*	SOM + Nearest neighbor	256	17 Sec
JL20006		SOM + Normal pdf (from SOM)	4096	100 Sec
FL20002	*	old SOM + Nearest neighbor	64	45 Sec
FL20001	*	nearest neightbor (no SOM)	-	7min 43 sec

An overall summary of all experiments is given in the table 1. The four experiments that are marked with a star (*) are selected as the most representative ones and their details are explaned in the following text. In the experiments the complexity of SOM, defined by the number of nodes, was optimized for maximal performance. The reader may note that experiments that use a combination of SOM and another method (MLP or nearest neighbor) require less nodes than those where all modelling was done with SOM. Another observation is that the use of SOM for nearest neighbor provides significant computational speedups.

Technical Summary of DLFS experiments

The following tables provide full technical details of selected four DLFS experiments. The first experiment JL20003 demonstrates SOM assisted hot deck imputation with a relatively complex SOM model. Due the large number of SOM nodes we may expect that the results are closer to nearest neighbor imputation than complete random hot deck method.

JL20003 technical details (SOM + random donor)	
Software	Windows+NDA
Hardware	Intel Celeron/700MHz + 256MB RAM
Set up time (by human)	2 minutes
Imputation run time	36 seconds
Other processing run time	20 seconds (data preprocess/TS-SOM model build)
Complete run time	56 seconds
Parameters	sigma2=1, layer 6 (4096 nodes)
Preprocess (continuous)	min-max equalization
Preprocess (categorial)	dummy coding
SOM MODEL	
Scope	imputation of INCOME variable
Variables (continuous)	INCOME, AGE
Variables (categorial)	SEX, EDUCATIO, BUSINESS, UNEMPLOY, MARRIAGE, PHONE

The second selected experiment JL20005 is SOM assisted nearest neighbor imputation. One should remember that the SOM node selection procedure introduces randomness to imputation. This is a notable difference to the normal deterministic nearest neighbor method.

JL20005 technical details (SOM + nearest neighbor)	
Software	Windows+NDA
Hardware	Intel Celeron/700MHz + 256MB RAM
Set up time (by human)	2 minutes
Imputation run time	6 seconds
Other processing run time	11 seconds (data preprocess/TS-SOM model build)
Complete run time	17 seconds
Parameters	sigma2=1, layer 4 (256 nodes)
Preprocess (continuous)	min-max equalization
Preprocess (categorial)	dummy coding
SOM MODEL	
Scope	imputation of INCOME variable
Variables (continuous)	INCOME, AGE
Variables (categorial)	SEX, EDUCATIO, BUSINESS, UNEMPLOY, MARRIAGE, PHONE
nearest neighbor	SEX, AGE, EDUCATIO, BUSINESS
variables	UNEMPLOY, MARRIAGE, PHONE

The pure nearest neighbor experiment FL20001 serves as a reference to SOM based methods. Of particular note is the high computational complexity of the method.

FL20001 technical details (nearest neighbor)	
Software	Windows + NDA
Hardware	IBM Pentium III 500Mhz + 256MB RAM
Set up time (by human)	5 minutes
Imputation run time	7min 42 seconds
Other processing run time	1 seconds
Complete run time	7min 43 seconds
Scope	Imputation of INCOME variable
nearest neighbor	AGE, AREA, BUSINESS, CHILDREN, COHABIT
variables	EDUCATION, MARRIAGE, PHONE, SEX and UNEMPLOY

The SOM assisted nearest neighbor experiment FL20002 by StatFI uses all LFS parameters (execpt RESPONCE and REF) for model building. When compared to similar donor method without SOM the compuatatonal speedup is significant.

FL20002 technical details (nearest neighbor)	
Software	Windows + NDA
Hardware	IBM Pentium III 500Mhz + 256MB RAM
Set up time (by human)	5 minutes
Imputation run time	29 seconds
Other processing run time	15 seconds
Complete run time	45 seconds
Parameters	sigma2=1, layer 3 (64 nodes)
Preprocess (continuous)	min-max equalization
SOM MODEL	
Scope	Imputation of INCOME variable
variables	all the 13 available variables
nearest neighbor	AGE, AREA, BUSINESS, CHILDREN, COHABIT
variables	EDUCATION, MARRIAGE, PHONE, SEX and UNEMPLOY

Results for DLFS/Y2

In all SOM experiments the main motivation was to preserve distributional accurancy rather than predict single observations. To evaluate how well this objective is achieved we have studied the confusion matrix between true and imputed values when developing the method. A typical example of such a joint distribution is shown in figure 9 for LFS develpment data.

The actual evaluation experiments were then done using the experiances, learned from development data. Some results of these experiments are summarized in the table 2 together with reference statistics. The best SOM results seem to be obtained using SOM assisted random donor method.

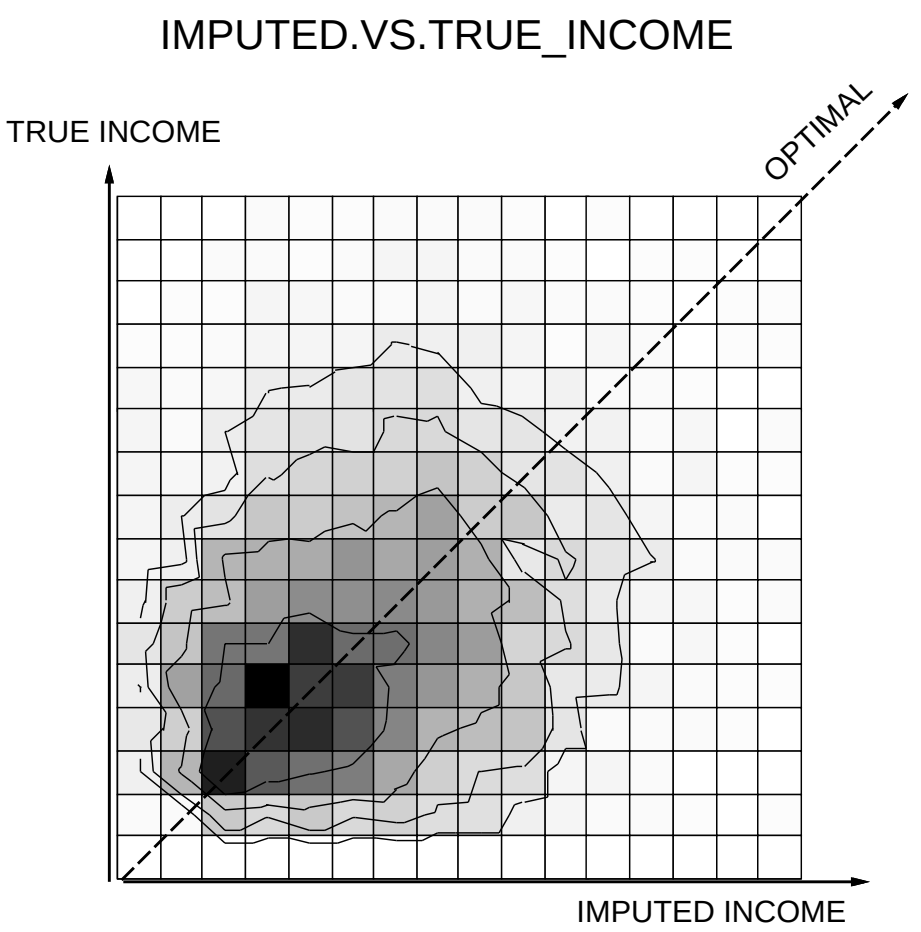
Table 2: Selected results of the SOM imputation for DLFS data set.

Experiment	dL1	K-S	m_1	m_2	method
JL20003	64992	0.03593	401.94205	3.61E+9	SOM+ random donor
JL20005	60266	0.04335	947.32263	2.53E+9	SOM+ nearest neighbor
FL20002	72160	0.068	6572.14	8.48E+8	SOM+ nearest neighbor
FL20001	66379	0.050	1322.10	1.81E+9	ref: nearest neighbor

The experiments that were made by JyU seem to be perform better than the experiments by StatFI. This is mainly explained by the better use of the SOM model. Since the SOM constructs a model of the joint distribution of variables, it is important to find a good subset of variables for model building.

Statistics Finland has stressed that the usual unit level metrics such as DL1 has its values somewhere between 60,000 and 85,000, and they are all poor. This is clearly because of a lack of background

Figure 9: A typical confusion table (pdf) between true and imputed values of INCOME.



information. Much more important in these data is thus the preservation of the aggregate/domain level values that are here well described by mean, variance and quartiles. Moreover, regression based methods naturally give smaller DL-values but decreases the preservation of the mean and the variance at the same time.

The results correspond somewhat to expectations from the tests based on the development data. That is, in StatFI experiments the TS-SOM did not manage to enhance the results but it made those very much faster because of its nature of the very low computational complexity. These large differences are highly considerable when using large datasets such as the LFS development data of 200,000 observations.

2.2.2 Dataset: European Community Household Survey Y2 (GSOEP)

The SOM GSOEP experiments by JyU should be regarded are "preliminary tests". The time that was spent with these was considerably lower than with other EUREDIT data sets. Originally four experiments made by JyU but only two, JG20003 and JG20004, are described here as representative examples of SOM based imputation. The difference between the two is that the variance for local models is estimated differently and that JG20003 uses more SOM nodes than JG20004.

Technical Summary of JG20003 and JG20004: SOM + Normal pdf	
Software	Windows+NDA
Hardware	Intel Celeron/700MHz + 256MB RAM
Set up time (by human)	10 minutes
JG20003	(smoothed estimate of node variance)
Imputation run time	44 seconds
Other processing run time	62 seconds (data preprocess/TS-SOM models build)
Complete run time	106 seconds
Parameters	sigma2=1, layer 4 (256 nodes)
JG20004	(variance from data subclusters, not smoothed)
Imputation run time	35 seconds
Other processing run time	24 seconds (data preprocess/TS-SOM models build)
Complete run time	59 seconds
Parameters	sigma2=1, layer 3 (64 nodes)
Remarks	Run times are totals for building 6 TS-SOM models and imputing for 6 × 2-variables (INCOME9x, HHINCO9x)
Parameters	sigma2=1, layer 4 (256 nodes)
Preprocess (continuous)	min-max equalization
Preprocess (categorial)	dummy coding
SOM models for years yy,	$yy \in \{91, 92, 93, 94, 95, 96\}$
Imputation of	INCOMEyy, HHINCOyy
Variables (continuous)	INCOMEyy, HHINCOyy, Y.O.B
Variables (categorial)	SEX, BETRyy, ERWTYPyy

A summary of selected evaluation statistics is shown in the table 3 together with one ONS experiment OG20001 (donor imputation). All we can say about these preliminary experiments is that they seem to be more competetive for later years. This is likely due the structure and missingness pattern of the data set. Although the GSOEP data set has a panel structure, the proposed imputation models were done for each year independently. One would expect that another type of design, based on the panel, would improve the results.

Table 3: Summary of some GSOEP imputation results for INCOME variable

INCOME variable							
stat	EXPERIMENT/year	91	92	93	94	95	96
dL1	JG20003	18473	18093	21113	19995	21235	22137
dL1	JG20004	20627	21593	24126	24673	26193	28541
dL1	OG20001	13171	13164	20716	23404	23824	22640
K-S	JG20003	0.09139	0.15841	0.10006	0.13049	0.17612	0.16218
K-S	JG20004	0.11098	0.12385	0.08039	0.12207	0.13373	0.10656
K-S	OG20001	0.02077	0.02246	0.07230	0.08178	0.07045	0.07611
m1	JG20003	1399	1037	2261	2214	1105	668
m1	JG20004	2231	1540	1528	5256	2463	2520
m1	OG20001	170	178	5763	7140	4089	3064

2.2.3 Dataset: UK Annual Business Inquiry Y2 (ABI)

A total of six experiments were made for ABI/Y2 data set. These esperiments are summarized in the table 4. The extra "experiment" XA20000 is random donor imputation without any assisting model. One may expect that any SOM and other model based imputation methods should be better than XA20000.

Table 4: Summary of SOM experiments with the ABI Y2 data set.

Experiment		Description	models	nodes	edit rules	runtime
JA20001		SOM + Mean&Nearest neighbor	4	16-256	NO	29 Sec
JA20002		SOM + Mean&Nearest neighbor	4	16-256	YES	36 Sec
JA20003		SOM + MLP&Nearest neighbor	5	16	NO	170 Sec
JA20004	*	SOM + MLP&Nearest neighbor	5	16	YES	171 Sec
JA20006		SOM + Normal pdf	11	256-1024	NO	14 Sec
JA20006	*	SOM + Normal pdf	11	256-1024	YES	16 Sec
XA20000		full random donor	-	-	-	-

Categorial variables were dummy coded

The only categorial variables were: EMPREG and FORMTYPE

Continuous variables (all the rest)

All continuous variables were LOG transformed and MIN-MAX equalized.

Special values (short form):

Short form questionnaire's non applicable (-9) values (= not asked) were set to missing data values in all experiments for variables EMPWAG, EMPNI, EMPENS, EMPRED, PUREN, PURCOTH, PURHIRE, PURINS, PURTRANS, PURTELE, PURCOMP, PURADV, PUROTHSE, TAXRATES, and TAXOTHE.

Special values (long form):

Long form questionnaire's special values 0 and -9 (= true non applicable value) were coded as own categories in experiments 5 and 6 for variables PURHIRE, PURTRANS, PURCOMP, PUROTHAL, TAXOTHE, ASSACQ, ASSDISP, and CAPWORK.

Pre-edit rules (experiments 2, 4, and 6):

Some experiments were preprocessed with simple linear logical edit rules, that included **zero rule** and **summing rule** as describe below:

- **Zero rule**
  If total (= sum) is zero then all components must be zero (if they are missing). NOTE: a) multivariate missing situtations (ie. two or more components have missing data and total is zero) were processed too, b) only missing data values were edited.
- **Summing rules**
  If only one component is missing and the total (= sum) and its components are observed, then the missing value is trivial to solve arithmetically. The used rules are

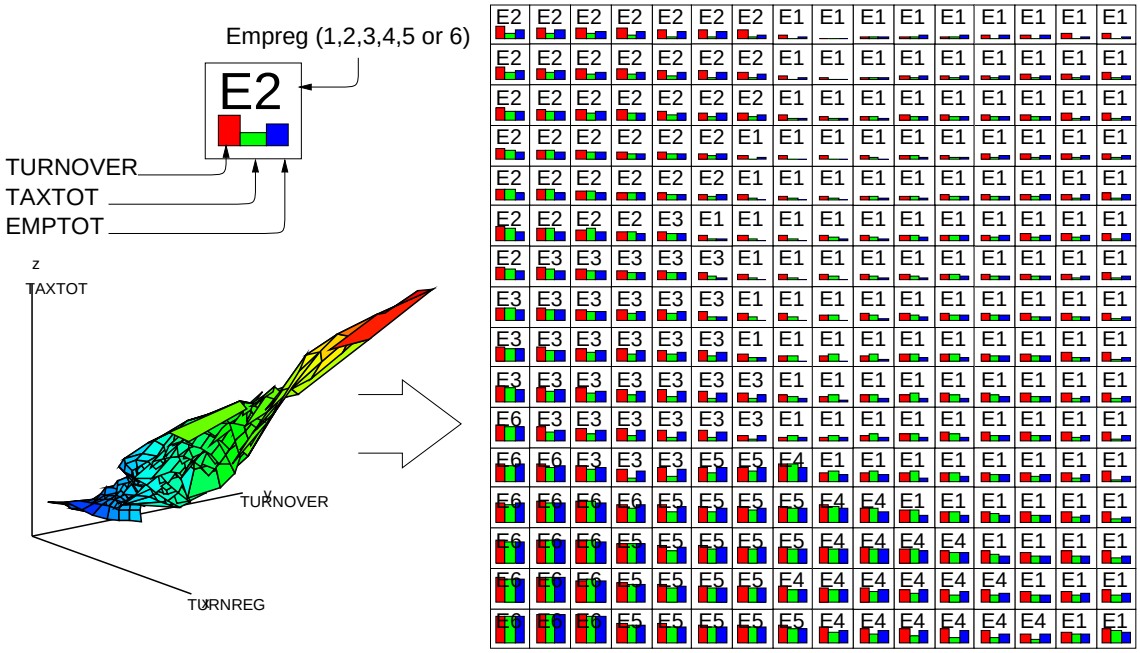
7	emptotc=	empwag+empni+empens+empred
9	purtot =	puren+purcoth+puresale+purhire+purins+purtrans+ purtele+purcomp+puradv+purothse
11	purtot =	puresale+purothal
16	taxtot =	taxrates+taxothe,

  where the number refers to the metafile rule of the ABI Y2 data set.

There are a total of 26 variables with missingess in the ABI Y2 data set, which makes the imputation process quite laborous, regardless of the methodology. To assist the process we used some simple variable selection techniques. The most important variables were also examined with SOM based data analysis, as illustrated in figure 10, to see if (or if not) the model is able to represent the imputed variables truthfully. For example, in the figure 10, the TURNOVER is clearly dependent on the TURNREG, EMPREG and TAXTOT variables.

In the left part of the figure a 2-D SOM is illustrated in the data space of TAXTOT, TURNREG and TURNOVER variables. The same SOM is then shown in the right part of the figure such that the local averages of variables in the SOM nodes (data clusters) are illustrated as bars and with a truth label for EMPREG category.

Figure 10: An illustration of SOM based data analysis for ABI Y2 data sets.



The motivation for six different SOM experiments was to examine the strenghts and weaknesses of SOM based imputation strategies. As expected, pre-edit rule can improve the imputation statistics, but not as much as one would expect. Therefore it is more interesting to compare a hybrid SOM approach with a "pure" SOM, where hybrid referes to SOM + (MLP or nearest neighbor) method and the pure SOM uses node statistics of SOM clusters directly for imputation. These two approaches are tested in experiments JA20004 and JA20006, respectively.

In JA20004 experiment five SOM models were build, with only 16 nodes per model. Then for each subset of data a MLP or nearest neighbor imputation method was used to do the actual imputation. In the JA20006 experiment a more basic SOM architecture with normal pdf imputation in the SOM nodes was used with 256-1024 neurons per model. These models are summarized in the following tables.

JA20004 Experiment techical details	
Software	Windows+NDA
Hardware	Intel Celeron/700MHz + 256MB RAM
Set up time (by human)	10 minutes
Imputation run time	140 seconds
Other processing run time	31 seconds (data preprocess/TS-SOM models build)
Complete run time	171 seconds
Model type	Hybrid: SOM+MLP+Nearest neighbor
Number of SOMs	5
MLP configuration	one hidden layer with 5 neurons, sigmoid activation functions Rprob-algorithm with 250 epochs
SOM 1:	SOM + 16× MLP
Parameters	layer 2 (16 nodes), sigma2=1
To impute	TAXTOT
Train with	TURNREG, EMPTOTC, PURTOT, PURESALE, TURNOVER EMPLOY, ASSACQ
MLP variables	TURNREG, EMPREG, FORMTYPE
SOM 2:	SOM + 16× MLP
Parameters	layer 2 (16 nodes), sigma2=1
To impute	EMPTOTC
Train with	FORMTYPE, TURNREG, EMPREG
MLP variables	TURNREG, EMPREG, FORMTYPE
SOM 3:	SOM + 16× MLP
Parameters	layer 2 (16 nodes), sigma2=1
To impute	To impute EMPLOY, TURNOVER, STOCKBERG
Train with	TURNREG, EMPREG
MLP variables	TURNREG, EMPREG
SOM 4:	SOM + 16× MLP
Parameters	layer 2 (16 nodes), sigma2=1
To impute	PURTOT
Train with	EMPREG, FORMTYPE, PUREN, PURCOTH
MLP variables	EMPREG, FORMTYPE
SOM 5:	SOM + 16× Nearest neighbor imputations
Parameters	layer 2 (16 nodes), sigma2=1
To impute	ASSACQ, ASSDISP and CAPWORK TAXRATES, TAXOTHE, PUREN, PURCOTH, PURESALE, PURHIRE, PURINS, PURTRANS, PURTELE, PUROTHAL, PURADV, PUROTHSE,PURCOMP, EMPWAG, EMPNI, EMPENS, EMPRED, STOCKEND
Train with	TURNREG, EMPTOTC, TAXTOT, PURTOT, CAPWORK
Nearest neighbor variables	TURNREG, EMPTOTC, ASSACQ, ASSDISP, TAXTOT, PURTOT

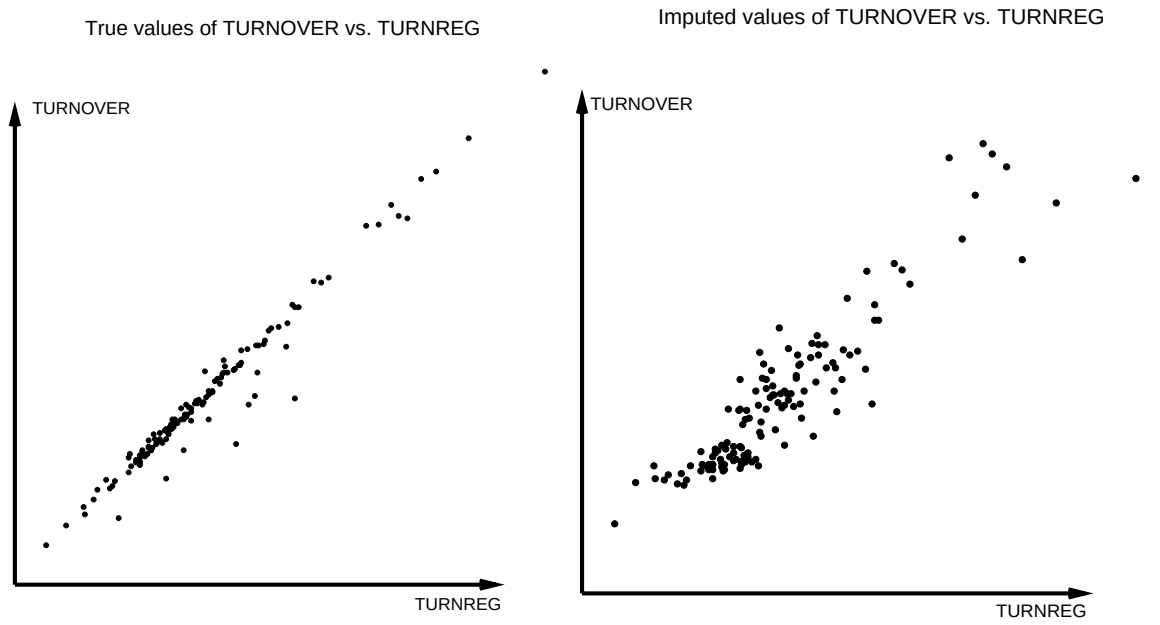
JA20006 Experiment techical details	
Software	Windows+NDA
Hardware	Intel Celeron/700MHz + 256MB RAM
Set up time (by human)	10 minutes
Imputation run time	15 seconds
Other processing run time	1 seconds (data preprocess/TS-SOM models build)
Complete run time	16 seconds
Model type	SOM + Normal pdf
Number of SOMs	11
SOM 1:	SOM + Normal pdf
Parameters	layer 4 (256 nodes), sigma2=1
To impute	TURNOVER, EMPTOTC, PURTOT, PURESALE
Train with	TURNOVER, EMPTOTC, PURTOT,PURESALE, EMPLOY, TURNREG, ASSACQ
SOM 2:	SOM + Normal pdf
Parameters	layer 4 (256 nodes), sigma2=1
To impute	TAXRATES, TAXOTHE, TAXTOT
Train with	TAXRATES, TAXOTHE, TAXTOT,TURNOVER, EMPLOY, EMPTOTC, PURTOT
SOM 3:	SOM + Normal pdf
Parameters	layer 4 (256 nodes), sigma2=1
To impute	STOCKBEG, STOCKEND
Train with	STOCKBEG, STOCKEND, TURNOVER, EMPLOY, PURTOT, EMPTOTC
SOM 4:	SOM + Normal pdf
Parameters	layer 4 (256 nodes), sigma2=1
To impute	EMPWAG, EMPNI, EMPENS, EMPRED
Train with	EMPWAG, EMPNI, EMPENS, EMPRED, EMPTOTC, TURNOVER, PURTOT, TAXTOT, EMPLOY
SOM 5:	SOM + Normal pdf
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	PURTELE, PURADV, PUROTHSE, PURINS, PUREN, PURCOTH
Train with	PURINS, PUREN, PURCOTH, EMPTOTC, TURNOVER, PURTOT, TAXTOT, EMPLOY
SOM 6:	SOM + Normal pdf
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	EMPLOY
Train with	EMPLOY, EMPREG, EMPTOTC, TURNOVER, PURTOT
SOM 7:	SOM + Normal pdf
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	PURTRANS, PURHIRE
Train with	PURTRANS, PURHIRE, EMPLOY, TURNREG, ASSACQ
SOM 8:	SOM + Normal pdf
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	PURCOMP, PUROTHAL
Train with	PURCOMP, PUROTHAL, TURNOVER, PURTOT, EMPLOY, TAXTOT

JA20006 Experiment techical details (cont.)	
SOM 9:	SOM + Normal pdf
Parameters	layer 4 (256 nodes), sigma2=1
To impute	ASSACQ
Train with	ASSACQ, TURNOVER, EMPLOY,TAXTOT, PURTOT
SOM 10:	SOM + Normal pdf
Parameters	layer 4 (256 nodes), sigma2=1
To impute	ASSDISP
Train with	ASSDISP, TURNOVER, EMPTOTC,PURTOT, TAXTOT, EMPLOY
SOM 11:	SOM + Normal pdf
Parameters	layer 4 (256 nodes), sigma2=1
To impute	CAPWORK
Train with	CAPWORK, TURNOVER, EMPTOTC, PURTOT, TAXTOT, EMPLOY

SOM results for ABI/Y2

The imputation performance of the SOM for ABI Y2 data set seems to be quite similar with all SOM based methods, at least from the point of view that there are no clear "outliers" among the imputation results. The SOM seems to be somewhat tolerant to different selections of training variables but as the number of variables increases, the behavior of the method seems to move closer to random donor. Therefore we do recommend that the user should not use more than ten different variables for training, unless there is a good reason for this.

Figure 11: Left: true values of ABI TURNOVER with respect to TURNREG
Right: imputed values of ABI TURNOVER with respect to TURNREG.

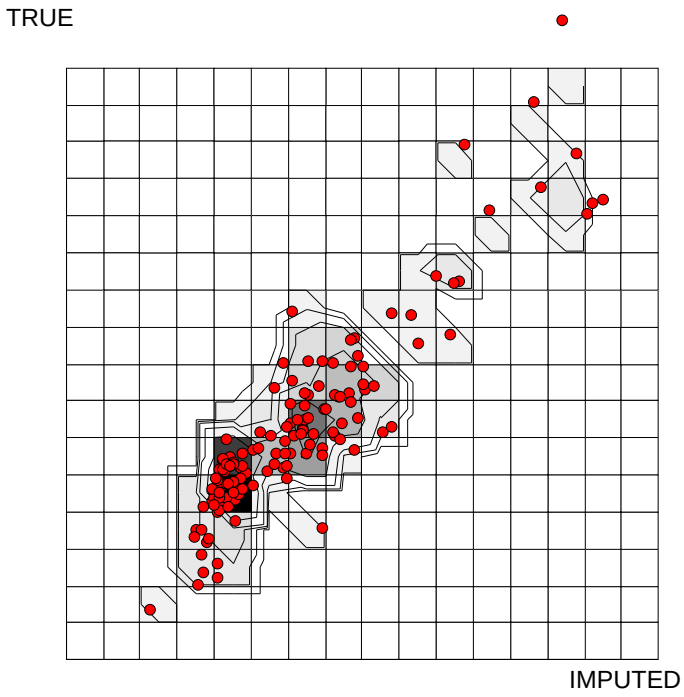


The plots 11 and 12 were made using JA20006 type of model and they illustrate a typical imputation result of the SOM for TURNOVER variable. The reader should also note that the axis does not start from zero, because in this example 1% and 99% fractiles were used for scaling of the data set. The imputation performance can be visualized with respect to some background variables, as it is done

in figure 11. This allows us to see that the current SOM model is able to impute missing values of TURNOVER quite well but it has problems with small values, which is due the border effects that are common in all SOM algorithms. To avoid the problem, we have used zero values as a special category with problematic continuous ABI variables.

The other plot, a confusion matrix (or pdf), as shown in figure 12 allows us to see that the SOM is able to preserve the distrubution of the TURNOVER quite well because the plot is diagonally symmetric. The prediction of individual values seem also quite good since the values are not very far from the diagonal line.

Figure 12: Distribution of true vs. imputed values of missing ABI/Y2 TURNOVER.



Since it is quite impossible to list all the imputation results of the SOM in this context, we investigate only four variables, TURNOVER, TAXTOT, EMPLOY and ASSDISP. Similar to DLFS data set, some selected evaluation statistics of the experiments are summarized in tables 5, 6, 7 and 8.

Table 5: Selected results of the SOM imputation of TURNOVER for the ABI Y2 data set.

EXP.	dL1	dL2	dLinf	K-S	K-S_1	m1	m2	method=SOM+	edits
JA20001	812	30832	72347	0.10216	0.00011	703	10 E+8	mean+neib.	NO
JA20002	894	31295	72714	0.11524	0.00009	592	9 E+8	mean+neib.	YES
JA20003	699	31275	74516	0.10828	0.00007	612	9 E+8	MLP+neib	NO
JA20004	595	15596	34588	0.12900	0.00019	371	7 E+8	MLP+neib	YES
JA20005	1036	30605	68837	0.09580	0.00012	469	9 E+8	Norm.pdf	NO
JA20006	556	13425	28758	0.11619	0.00024	7	8 E+8	Norm.pdf	YES
OA20001	1113	47506	113388	0.14137	0.00013	860	52 E+8	DIS	-
XA20000	11864	92175	468798	0.32970	0.00177	10244	74 E+8	donor	-

The ONS experiment OA20001 is used as a reference of nearest neighbor imputation and XA20000 is a full random donor. One can conclude that for ABI Y2 data the overall performance of SOM based

methods seems to be quite good in comparison to standard nearest neighbor and donor imputations. The role of imputation rules does not seem to be very important.

Table 6: Selected results of the SOM imputation of TAXTOT for the ABI Y2 data set.

EXP.	Slope	dL1	dL2	dLinf	K-S	K-S_1	K-S_2	m1	m2
JA20001	0.012	19	87	806	0.23488	0.00237	0.00009	13.8	11405
JA20002	0.008	44	134	810	0.34598	0.00867	0.00075	38.2	17338
JA20003	1.143	4	29	54	0.12851	0.00063	0.00002	1.9	2669
JA20004	1.000	3	12	29	0.10748	0.00037	0.00001	1.2	352
JA20005	0.463	8	79	172	0.30445	0.00072	0.00003	1.1	6339
JA20006	0.439	8	82	187	0.28364	0.00085	0.00002	0.7	6194
OA20001	0.50	8	97	190	0.09653	0.00041	0.00001	3.5	23915
XA20000	0.0002	408	2451	29189	0.21610	0.01120	0.00033	398	6 E+6

.

Table 7: Selected results of the SOM imputation of EMPLOY for the ABI Y2 data set.

EXP.	Slope	dL1	dL2	dLinf	K-S	K-S_1	K-S_2	m1	m2
JA20001	0.21	8.6	137.3	251.3	0.085	0.00041	0.00001	1.6	11190
JA20002	0.16	9.1	161.2	318.8	0.083	0.00046	0.00001	1.0	12453
JA20003	1.00	3.9	79.0	180.7	0.271	0.00027	0.00002	0.8	15953
JA20004	0.73	5.3	127.5	293.9	0.271	0.00023	0.00002	1.1	19810
JA20005	0.93	4.3	69.5	157.4	0.268	0.00028	0.00001	0.9	14919
JA20006	0.82	4.6	109.2	250.0	0.256	0.00013	0.00001	1.6	18806
OA20001	0.86	5.2	28.8	52.5	0.076	0.00044	0.00000	1.8	6971
XA20000	0.00	105.2	852.3	3726.2	0.302	0.00333	0.00018	95.0	685864

.

Table 8: Selected results of the SOM imputation of ASSDISP for the ABI Y2 data set.

EXP.	dL1	dL2	dLinf	K-S	K-S_1	K-S_2	m1	m2
JA20001	7.7	155.	379.	0.37	0.00040	0.00011	0.30	24686.
JA20002	7.7	131.	315.	0.39	0.00041	0.00012	1.56	22418.
JA20003	7.8	158.	384.	0.29	0.00039	0.00007	0.87	23162.
JA20004	7.8	157.	384.	0.37	0.00041	0.00010	0.54	24676.
JA20005	6.0	169.	314.	0.18	0.00029	0.00001	1.61	11329.
JA20006	9.6	255.	499.	0.17	0.00031	0.00001	0.91	16920.
OA20001	5.4	127.	273.	0.06	0.00024	0.00000	2.76	61616.
XA20000	27.02	300.	1675.	0.06	0.00240	0.00003	23.61	91037.

.

.

2.2.4 Dataset: UK Annual Business Inquiry Y3 (ABI)

ABI Y3 data set contains both errors and missing values. We note that it is more difficult to use the SOM for erroneous data than for incomplete data without any errors. This is because the SOM model is basically a nonparametric method where we have no predifined models for errors. Due the flexibility of the SOM training algoritm, it easily models all errors as well as the rest of data. This might be useful for human assisted data analysis, but is not desired for automated error detection. To solve the problem we have developed robust training procedures for SOM. Our aim was to build a SOM model of clean data from erroneous and incomplete observations.

The robust SOM algorithm tries to detect outliers during the training by measuring the distance of observations from the SOM surface under an assumption that data are Normal distributed around the SOM model. This procedure is quite risky because the indicator of an outlier is dependent on several SOM training parameters including the SOM complexity (TS-SOM layer) and "sigma" parameters, as defined in section 1. Also the method that was used in the EUREDIT project is still experimental, which makes us to believe that an improved and better justified version of robust SOM would give better results than what we were able to achieve in this project.

We made originally 4 experiments but only two of them JA30001 and JA30004 are fully comparable with the results of other EUREDIT partners. Experiments JA30002 and JA30003 are for editing evaluation only because most of the outliers were not imputed but marked as missing, which omits them from the compuation of imputation statistics. Thus the base set of imputation statistics for JA30002 and JA30003 is different to other experiments, and therefore not comparable (it is too optimistic).

After the real experiments were completed, true data was given to us and three extra experiments XA30001, XA30050 and XA30100 were made for comparison. In the "extra" experiments a portion of 1, 50 and 100 per centage of true errors was marked as outliers (with help of true data), and then both missing and marked outliers were imputed using the full random donor method. The reader should note that such a perfect outlier detection is not realistic since no non outliers were incorrectly marked as outliers. The extra experiments are simply baselines that allows us to compare the performance of the imputation methodology. All these experiments are summarised in table 9.

Table 9: Summary of SOM experiments with the ABI Y3 data set.

Experiment		Description	models	nodes	edit rules	robust	runtime
JA30001	*	SOM + Normal pdf	11	16	YES	omit outliers	200 Sec
JA30004	*	SOM + Normal pdf	5	16	YES	Huber	91 Sec
JA30002		SOM (edit only)	11	16	YES	omit outliers	200 Sec
JA30003		SOM (edit only)	5	16	YES	Huber	91 Sec
XA30001		full random donor	1% errors detected and marked as outliers				
XA30050		full random donor	50% errors detected and marked as outliers				
XA30100		full random donor	100% errors detected and marked as outliers				

The coding and preprocessing of ABI Y3 data set is in most parts similar to that of ABI Y2, as summarized in the following.

Categorial variables were dummy coded
The only categorial variables were: EMPREG and FORMTYPE

Continuous variables (all the rest)
All continuous variables were LOG transformed and MIN-MAX equalized.

Special values (short form):
Short form questionnaire's non applicable (-9) values (= not asked) were set to missing data values in all experiments for variables EMPWAG, EMPNI, EMPENS, EMPRED,

PUREN, PURCOTH, PURHIRE, PURINS, PURTRANS, PURTELE, PURCOMP, PURADV, PUROTHSE, TAXRATES, and TAXOTHE.

Special values (long form):
Long form questionnaire's special values 0 and -9 (= true non applicable value) were coded as own categories in experiments 5 and 6 for variables PURHIRE, PURTRANS, PURCOMP, PUROTHAL, TAXOTHE, ASSACQ, ASSDISP, and CAPWORK.

Pre-edit rules :**NO EDIT RULES WERE USED FOR ABI Y3 DATA !**

Specific to Y3 data set is that continuous training variables were robustly min-max equalized using 5% and 95% fractiles. We also note that non perturbed register variables TURNREG and EMPREG were not edited and there no robustness was used for them in the training of TS-SOM models. Other technical details are given in the following tables.

JA30001 technical details	
Software	Windows+NDA
Hardware	IBM 600X laptop/Intel P3 processor/500MHz + 374MB RAM
Set up time (by human)	10 minutes
Edit run time	10 seconds
Imputation run time	10 seconds
Other processing run time	180 seconds (data preprocess/TS-SOM models build)
Complete run time	200 seconds
Remarks	Run times are totals for doing 11 TS-SOM models and editing/imputing 11 variable groups
Model type	Basic: SOM+Normal pdf in nodes
Number of SOMs	11
Outlier handling (training)	Suspected outliers are marked as missing values
SOM 1: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.25 , sigma2= 1.85 , Edit cut= 0.45
To edit/impute	TURNOVER, EMPTOTC, PURTOT
Train with	same as above + TURNREG and EMPREG
SOM 2: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 2.5 , sigma2= 1.25 , Edit cut= 0.25
To edit/impute	TAXRATES, TAXTOT
Train with	same as above + TURNREG and EMPREG
SOM 3: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.0 , sigma2= 0.5 , Edit cut= 0.5
To edit/impute	EMPWAG, EMPNI, EMPENS, EMPRED
Train with	same as above + TURNREG and EMPREG

JA20001 technical details (continue)	
SOM 4: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.0 , sigma2= 1.0 , Edit cut= 0.4
To edit/impute	STOCKBEG, STOCKEND
Train with	same as above + TURNREG and EMPREG
SOM 5: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.0 , sigma2= 0.4 , Edit cut= 0.5
To edit/impute	PURESALE, PURINS, PURTELE, PURADV PUROTHSE, PUREN, PURCOTH
Train with	same as above + TURNREG and EMPREG
SOM 6: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.25 , sigma2= 1.5 , Edit cut= 0.5
To edit/impute	EMPLOY
Train with	same as above + TURNREG and EMPREG
SOM 7: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.25 , sigma2= 0.9 , Edit cut= 0.5
To edit/impute	PURHIRE, PURTRANS, PURCOMP
Train with	same as above + TURNREG and EMPREG
SOM 8: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.25 , sigma2= 1.0 , Edit cut= 0.5
To edit/impute	TAXOTHE
Train with	same as above + TURNREG and EMPREG
SOM 9: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.5 , sigma2= 0.8 , Edit cut= 0.5
To edit/impute	ASSACQ,ASSDISP
Train with	same as above + TURNREG and EMPREG
SOM 10: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.25 , sigma2= 0.75 , Edit cut= 0.5
To edit/impute	PUROTHAL
Train with	same as above + TURNREG and EMPREG
SOM 11: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 2.75 , sigma2= 0.5 , Edit cut= 0.5
To edit/impute	CAPWORK
Train with	same as above + TURNREG and EMPREG

The main differences between JA30001 and JA30004 experiments are the handling of suspected outliers during training and the number of SOM models (11 models in JA30001 and 5 models in JA30004). In JA30001 outliers were marked as missing values during the training. In JA30004 a robust Huber estimator was used to make SOM training robust for errors.

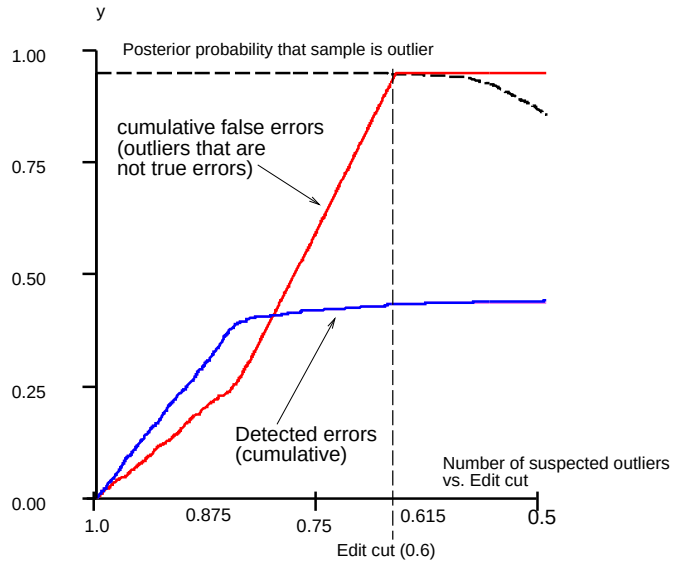
JA30004 technical details	
Software	Windows+NDA
Hardware	Intel Celeron/700MHz + 256MB RAM
Set up time (by human)	10 minutes
Edit run time	10 seconds
Imputation run time	10 seconds
Other processing run time	71 seconds (data preprocess/TS-SOM models build)
Complete run time	91 seconds
Remarks	Run times are totals for doing 5 TS-SOM models and editing/imputing 5 variable groups
Model type	Basic: SOM+Normal pdf in nodes
Number of SOMs	5
Outlier handling (training)	Huber estimator was used
SOM 1: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.0 , sigma2= 0.5, Edit cut= 0.25
To edit/impute	CAPWORK,TAXOTHE
Train with	same as above + TURNREG and EMPREG
SOM 2: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.0 , sigma2= 1.0, Edit cut= 0.25
To edit/impute	PURESALE, PURINS,PURTELE, PURADV, PUROTHSE, PURHIRE, PURTRANS, PURCOMP, PUROTHAL, PUREN, PURCOTH
Train with	same as above + TURNREG and EMPREG
SOM 3: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 2.0 , sigma2= 1.25, Edit cut= 0.25
To edit/impute	TURNOVER, EMPTOTC, PURTOT, TAXRATES, TAXTOT, EMPLOY, EMPWAG, EMPNI, EMPENS, EMPRED
Train with	same as above + TURNREG and EMPREG
SOM 4: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.0 , sigma2= 0.5, Edit cut= 0.25
To edit/impute	ASSACQ,ASSDISP
Train with	same as above + TURNREG and EMPREG
SOM 5: Parameters (same for all edited/imputed variables)	SOM + 16 nodes with Normal pdf layer 2 (16 nodes), sigma1= 3.0 , sigma2= 0.5, Edit cut= 0.25
To edit/impute	STOCKBEG,STOCKEND
Train with	same as above + TURNREG and EMPREG

SOM results for ABI Y3

Unfortunately the performance of SOM in editing and imputation is strongly dependent on parameter selections, the role of which is still under development. This makes the use of the current SOM version quite tricky since there is no clear objective that can be used for the search of optimal editing parameters. Because of this the obtained results are more like indicators of the potential of SOM rather than optimal results that we can achieve.

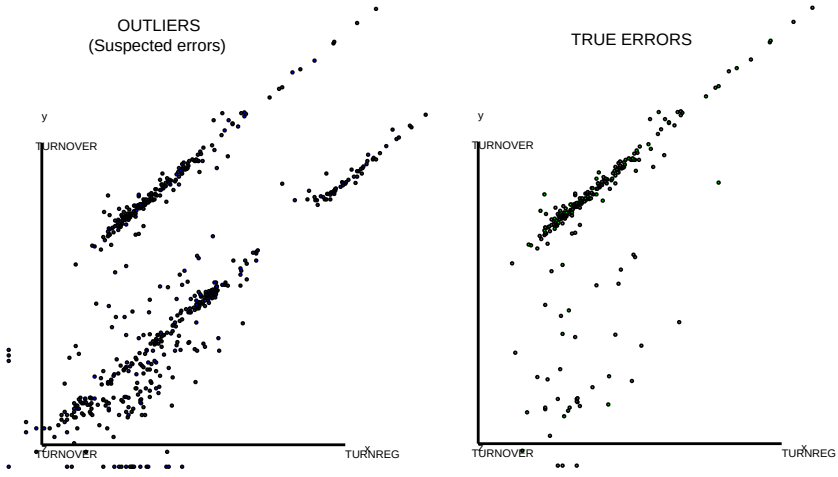
One difficulty of editing is the tradeoff between found errors and false alarms, which is depicted in figure 13. Optimally the number of true errors grows always faster than the number of false ones. In practice however, after some point most of the outliers are false alarms although many of the true errors are still not found. In the case of sec98 ABI Y3 this might be natural since many of the errors seem to be inliers that are impossible to identify probabilistically.

Figure 13: Cumulative number of suspected true and false errors



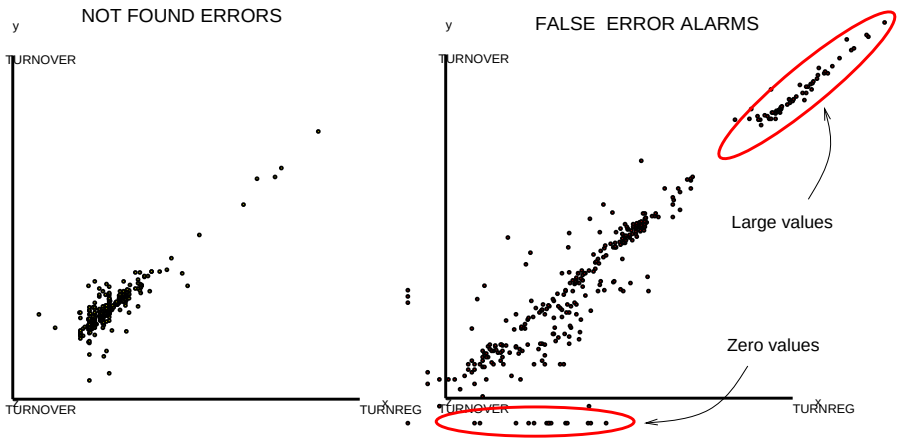
Error detection performance can be visualized also as shown in figures 14 and 15 for TURNOVER variable as a function of TURNREG. Although true outliers are relatively well detected, there are also quite many false alarms, especially among large values of ABI Y3 variables. This is due border effects of SOM training, which states that SOM is not able to model extreme values of data. The problem is most severe in the case of zero values since SOM training does not automatically create a node class for them. In this case the problem can be solved by using an additional category for zero values, but for large values variables such an option does not exist.

Figure 14: Detected outliers and true errors of ABI Y3 TURNOVER variable.



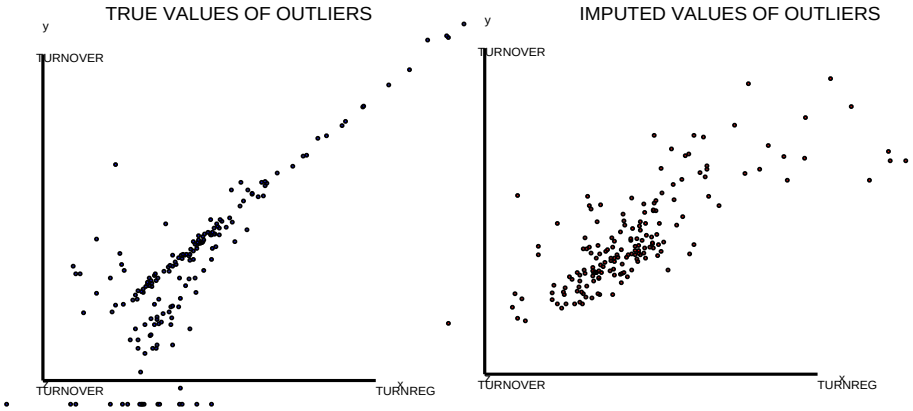
The problem of false errors with TURNOVER variable in SOM outlier detection is clearly demonstrated in figure 15. While the "not found errors" are obviously inliers with respect to TURNREG variable, there are many false alarms in the same area as well, which are due a error detection based on some other covariates than TURNREG. The most alarming is, however, that the SOM picks zeros and larger values of TURNOVER as outliers.

Figure 15: "Not detected errors" and "false alarms" of ABI Y3 TURNOVER.



We note that the imputation performance of SOM for ABI Y3 is not as good as it is for Y2 data set because it is more difficult to build good SOM model from bad data.

Figure 16: True and imputed values of ABI Y3 TURNOVER outliers.

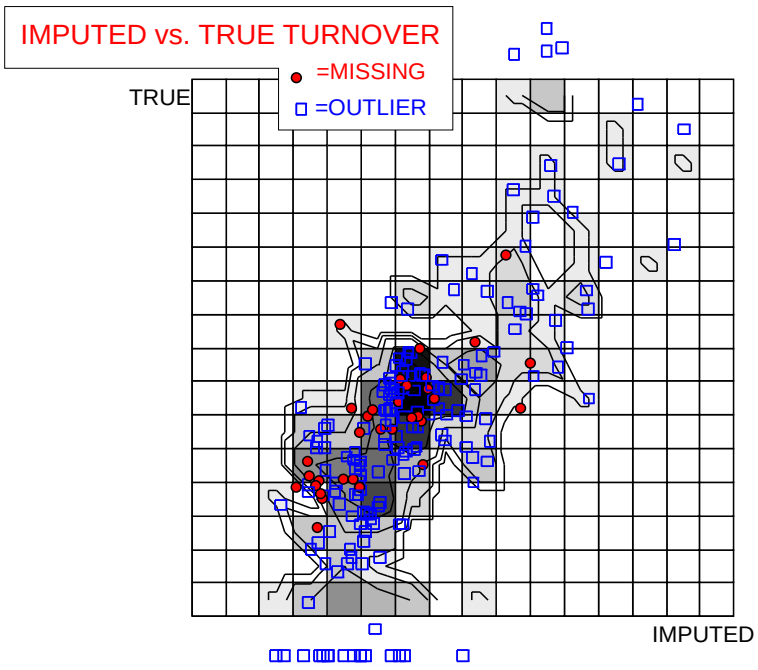


In most cases we have been careful by using simple rather than complex SOM models, which protects us from very bad parameter choices but increases the risk of border effects. Imputation of TURNOVER variable is illustrated in figures 16, 16 and 18.

The role of border effects in imputation is most clearly visible in figure 18, which shows the confusion distribution between true and imputed values of TURNOVER. Distribution is clearly biased (off diagonal) in small values of TURNOVER and it has high variance in large ones.

Figure 17: True and imputed values of missingness for ABI Y3 TURNOVER.

Figure 18: Distribution of true vs. imputed values of ABI Y3 TURNOVER.



The actual editing performance of all SOM experiement for TURNOVER is shown in table 10. Experiment OA30001 is nearest neighbor donor and it is used as a baseline together with random donor experiments XA30001, XA30050 and XA30100. The baselines have no value in editing since there is no objective way to build any kind of standard baseline for error detection. We remaind the reader that in XA30001, XA30050 and XA30100 true data was used to detect 1, 50 and 100 per centage of errors but no false alarms were made (thus beta=0). We note also that JA30003 was intentionally overedited to better understand the behavior of evaluation statistics. We remaind also that JA30002 and JA30003 are valid only for editing statistics, not for imputatation because most of the outliers were marked as missing values.

Table 10: Selected results of the SOM editing of TURNOVER for the ABI Y3 data set.

EXPERIMENT	alpha	beta	delta	RRASE	RER	tj	AREm2
JA30001	0.646	0.0008	0.056	0.0402	3446.	2.7	0.70621
JA30004	0.704	0.0225	0.081	0.0204	169.	6.7	0.96251
JA30002	0.614	0.0229	0.074	-	-	-	-
JA30003	0.044	0.9652	0.885	-	-	-	-
OA30001	1.000	0.0000	0.086	11.2859	2005507.	2.5	10350.29688
XA30001	0.981	0.0000	0.084	0.0196	100230.	8.0	0.00914
XA30050	0.504	0.0000	0.043	0.0071	27118.	6.8	0.06310
XA30100	0.000	0.0000	0.000	0.0000	-10000.	-	0.12929

The corresponding imputation statistics for ABI Y3 TURNOVER are shown in table 11. The problem of border effects is now seen in evaluation statistics as well. Because many large but correct values are marked as outliers, unit level measures , such as DL1 and DL2 indicate large differences between the true and imputed values. On the distributional statistics (Kolmogorov-Smirnov) SOM seems to work quite well despite of border effects, which is due the large variances estimated Normal pdf's in nodes that represent large values of data. Overall SOM seems to behave a little worse than nearest neighbor without any error detection but it is clearly better than random donor without any model assistance.

Table 11: Results of the SOM imputation of TURNOVER for the ABI Y3 data set.

EXP.	Slope	dL1	dL2	dLinf	K-S	K-S_1	K-S_2
JA30001	0.98769	8206.	254083.	542666.	0.41568	0.00044	0.00000
JA30004	0.00112	26408.	286816.	1.5E+6	0.21127	0.00077	0.00001
OA30001	0.62238	622.	3689.	9061.	0.21171	0.00222	0.00013
XA30001	0.00100	4748449.	73E+6	99E+6	0.35359	0.00141	0.00001
XA30050	0.00100	272883.	19E+6	60E+6	0.38725	0.00004	0.00000
XA30100	0.00053	10693.	69566.	964738.	0.38180	0.00518	0.00020

EXP.	m1	m2	MSE
JA30001	8027.	645 E+8.	18992
JA30004	8138.	743 E+8.	518701
OA30001	493.	0.2E +8	14 E+8
XA30001	4747417.	5485 E+12	16 E+8
XA30050	272308.	375 E+12	8 E+8
XA30100	10046.	43 E+8	2.8 E+8

Variable EMPLOY was edited very carefully because it was difficult to tune reliable editing parameters for it. The benefit of light editing is that there are less false alarams and less changes in the variables. Although the model accepts many errors, the overall evaluation result after imputation is often better than what it would be if large number of data records are marked as outliers and imputed, and light editing is still able to detect and correct the most severe errors. The editing performance of EMPLOY variable is summarized in table 12,

Table 12: Selected results of the SOM editing of EMPLOY for the ABI Y3 data set.

EXPERIMENT	alpha	beta	delta	RRASE	RER	tj	AREm2
JA30001	0.97	0.005	0.051	0.00130	225.0	3.90682	0.9610
JA30004	0.99	0.011	0.058	0.00126	63.9	3.05660	0.9922
JA30002	0.98	0.003	0.050	-	-	-	-
JA30003	0.11	0.927	0.889	-	-	-	-
OA30001	1.00	0.000	0.047	0.00176	225.0	3.18052	0.0078
XA30001	1.00	0.000	0.047	0.00172	225.0	3.18052	0.0078
XA30050	0.54	0.000	0.026	0.00108	32.6	2.04919	0.0402
XA30100	0.11	0.000	0.005	0.00004	0.7	-2.71804	0.0726

The imputation performance, which is shown in table 13, shows moderate success when compared to baselines OA30001 and XA30xxx, but one can argure that nearest nearest neighbor imputation without any editing, OA30001, is better in overall.

Table 13: Results of the SOM imputation of EMPLOY for the ABI Y3 data set.

EXP.	Slope	dL1	dL2	dLinf	K-S	K-S_1	K-S_2
JA30001	1.18271	531.	5518.	7189.	0.27652	0.00332	0.00004
JA30004	2.67470	846.	6142.	12238.	0.31876	0.00542	0.00014
OA30001	0.01084	12.	49.	332.	0.23851	0.01027	0.00087
XA30001	0.05822	11.	24.	112.	0.45628	0.02976	0.00692
XA30050	0.00012	239.	2057.	27702.	0.42383	0.00374	0.00026
XA30100	0.00013	189.	1666.	27517.	0.40971	0.00413	0.00022

EXP.	m1	m2	MSE
JA30001	529.8	30473318.	23.6
JA30004	816.4	37950896.	36.7
OA30001	11.7	2808.	2.3
XA30001	9.2	315.	2.3
XA30050	237.3	4233850.	131.6
XA30100	188.1	2776715.	213.0

Variable PURTOT is an example of rather successful use of SOM for ABI Y3 data. Although only about 30 per cent of errors were correctly identified for PURTOT variable, the imputation results are quite satisfactory. A part of this relative success is that almost no false alarms were made in outlier detection, but we may also expect that the not found errors were not severe ones. The editing statistics of PURTOT is summarized in table 14.

Table 14: Selected results of the SOM editing of PURTOT for the ABI Y3 data set.

EXPERIMENT	alpha	beta	delta	RRASE	RER	tj	AREm2
JA30001	0.751	0.00133	0.112	0.051	4778.75	3.22	0.65225
JA30004	0.770	0.01725	0.129	0.024	296.08	7.16	0.94967
JA30002	0.726	0.01554	0.121	-	-	-	-
JA30003	0.032	0.96362	0.825	-	-	-	-
OA30001	1.000	0.00000	0.148	9.841	1686835.12	0.00	7394.80664
XA30001	0.984	0.00000	0.146	0.012	48326.53	8.77	0.01065
XA30050	0.467	0.00000	0.069	0.010	48326.53	6.11	0.10902
XA30100	0.000	0.00000	0.000	0.000	-	-	0.21318

The imputation statistics for PURTOT is shown in table 15. As we see experiment JA30001 is the most successful according to almost all evaluation viewpoints. One reason for success might be that PURTOT was in the same variable group (SOM model) with two other important variables (TURNOVER and EMPTOTC) and more time was spent to optimize the editing parameters for this SOM model. To repeat such success automatically, more development work must be done with the robust SOM training algorithm. This work in now in progress.

Table 15: Results of the SOM imputation of PURTOT for the ABI Y3 data set.

EXP.	Slope	dL1	dL2	dLinf	K-S	K-S_1	K-S_2
JA30001	1.14533	5120	172479	389386	0.320	0.00038	0.00000
JA30004	0.00384	12231	189952	509789	0.130	0.00075	0.00000
OA30001	0.00127	32442	151686	571876	0.159	0.00544	0.00030
XA30001	0.00100	2867659	39058652	43678328	0.288	0.00195	0.00001
XA30050	0.00100	131708	9850125	36266144	0.310	0.00006	0.00000
XA30100	0.00020	9341	85399	1665859	0.313	0.00151	0.00004

EXP.	m1	m2	MSE
JA30001	5032	297E+8	6910
JA30004	7994	364E+8	315236
OA30001	32403	230E+8	1119E+6
XA30001	2867472	1528E+12	1226E+6
XA30050	131336	97E+12	515E+6
XA30100	8887	71E+8	118E+6

2.2.5 Dataset: UK Household Y2 (SARS)

UK Sars data set is special in two ways. It has more records, a total of 492472 observations in Y2 data set, and most of its variables are categorial. Also, because categorial variables for SOM based analysis are dummy coded with one zero-one per category, the number of SOM training variables can be very high.

For a tree structured SOM (TS-SOM) the large number of data records is not a problem. The method is quite fast to train, assuming that there is enough of memory in the computer to keep all data in the RAM. The number of categories, however, in categorial variables can be problematic because SOM models the joint distribution of its training data. When the dimension is large, there is no clear focus for the training and all types of scatters between the categories have an effect to the training result. This is typical example of the curse of dimension, which we decided to tackle with careful division of data into smaller variable sets. Also the large number of records helps because it allows us to use more neurons (nodes, data clusters) that hopefully have less confusion between the categories than what large groups have.

There were originally 3 experiments from JyU and 2 partial experiments by StatFI. These are summarized in the table 16. Experiment XS20001 is a baseline for comparison purposes. It was made using random donor (hot deck) without any model assistance. As before we expect that our method should do better than XS20001.

Table 16: Summary of SOM experiments with the SARS Y2 data set.

Experiment	Description	models	nodes/model	edit rules	runtime
JS20001	SOM + Normal pdf (cont.) + Posterior probability (cat.)	9	1024	some	5400 Sec
JS20002	SOM + random donor	9	1024	some	6000 Sec
JS20003	SOM + random donor	9	16384	some	16200 Sec
FS20001	SOM + nearest neighbour + radom donor + rules	4	63-1024	some	many hours
FS20002	SOM + SAS (hyprid of many)	4	63-1024	some	many hours
XS20001	full random donor				

JyU (JS2000x) Experiment details

In JyU experiments JS20001, JS20002 and JS20003 the following setup for experiments was used.

Categorial variables were dummy coded.

Continuous variables were min-max equalized.

Special values for continuous variables hours (values: -9, 71, and 81) and age (values: 91, 93, 95) were coded as own categories (new dummy variables).

Post processing 1 of imputed **categorial variables**, (when random donor was not used), were done by selecting the final category randomly according to posterior probabilities of the categories. These posterior probabilities were computed by the SOM training algorithm.

Post processing 2. Posterior probabilities for class -9 (non applicable value) were set to 0 (zero) for variables WORKPLCE, DISTWORK, ISCO1, URVISIT, ECONPRIM, ISCO2, QUALSUB, QUALEVEL, MIGORGN, and TERMTIM. Thus no new -9 values were created by imputation.

JS20001 Experiment technical details	
Software	Windows+NDA
Hardware	Intel Celeron/700MHz + 256MB RAM
Set up time	20 minutes
Imputation run time	300 seconds
Other processing run time	5100 seconds (data preprocess/TS-SOM models build)
Complete run time	5400 seconds
Remarks	Run times are for building 9 TS-SOM models and imputing 9 variable groups
Model type	SOM + Posterior probability (categories) + Normal Pdf (continuous varaibles)
Remark	Posterior probabilities are node mean values for categories that are trained with 0,1 dummy values.
Number of SOMs	9
SOM 1:	SOM + node mean (Posterior prob.)+Normal pdf (age)
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	SEX, AGE, MSTATUS, LTILL
Train with	same as above + HOURS, ISCO1, RELAT
SOM 2:	SOM + node mean (Posterior prob.)
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	RELAT
Train with	RELAT, SEX, ECONPRIM, AGE
SOM 3:	SOM + node mean (Posterior prob.)
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	HHSPTYPE, ROOMSNUM, BATH CENHEAT, INSIDEWC, TENURE
Train with	same as above + PERSINHH
SOM 4:	SOM + node mean (Posterior prob.)+Normal pdf (hours)
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	ISCO1, DISTWORK, HOURS, WORKPLCE
Train with	SEX, RELAT
SOM 5:	SOM + node mean (Posterior prob.)
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	COBIRTH
Train with	COBIRTH, SEX, QUALNUM, QUALEVEL
SOM 6:	SOM + node mean (Posterior prob.)
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	URVISIT, ECONPRIM, ISCO2
Train with	same as above + SEX
SOM 7:	SOM + node mean (Posterior prob.)
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	QUALNUM, QUALEVEL, QUALSUB
Train with	same as above + SEX, ISCO1
SOM 8:	SOM + node mean (Posterior prob.)
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	RESIDSTA, MIGORGN, TERMTIM
Train with	same as above + SEX, ISCO1, RELAT, HOURS
SOM 9:	SOM + node mean (Posterior prob.)
Parameters	layer 5 (1024 nodes), sigma2=1
To impute	CARS
Train with	CARS, PERSINHH, HHSPTYPE,ROOMSNUM

JS20002 Experiment technical details	
Software	Windows+NDA
Hardware	Intel Celeron/700MHz + 256MB RAM
Set up time	15 minutes
Imputation run time	400 seconds
Other processing run time	5600 seconds (data preprocess/TS-SOM models build)
Complete run time	6000 seconds
Remarks	Run times are for building 9 TS-SOM models and imputing 9 variable groups
Model type	SOM + random donor
Remark	Very simple SOM models are used ! Imputed variables are not use in SOM training .
Number of SOMs	9
SOM 1:	SOM + random donor
Parameters	layer 5 (1024 nodes)
To impute	SEX, AGE, MSTATUS, LTILL
Train with	HOURS, ISCO1, RELAT
SOM 2:	SOM + random donor
Parameters	layer 5 (1024 nodes)
To impute	RELAT
Train with	SEX, ECONPRIM, AGE
SOM 3:	SOM + random donor
Parameters	layer 5 (1024 nodes)
To impute	HHSPTYPE, ROOMSNUM, BATH CENHEAT, INSIDEWC, TENURE
Train with	PERSINHH
SOM 4:	SOM + random donor
Parameters	layer 5 (1024 nodes)
To impute	ISCO1, DISTWORK, HOURS, WORKPLCE
Train with	SEX, RELAT
SOM 5:	SOM + random donor
Parameters	layer 2 (64 nodes)
To impute	COBIRTH
Train with	SEX, QUALNUM, QUALEVEL
SOM 6:	SOM + random donor
Parameters	layer 5 (1024 nodes)
To impute	URVISIT, ECONPRIM, ISCO2
Train with	SEX
SOM 7:	SOM + random donor
Parameters	layer 5 (1024 nodes)
To impute	QUALNUM, QUALEVEL, QUALSUB
Train with	SEX, ISCO1
SOM 8:	SOM + random donor
Parameters	layer 5 (1024 nodes)
To impute	RESIDSTA, MIGORGN, TERMTIM
Train with	SEX, ISCO1, RELAT, HOURS
SOM 9:	SOM + random donor
Parameters	layer 5 (1024 nodes)
To impute	PERSINHH, HHSPTYPE
Train with	ROOMSNUM

JS20003 Experiment technical details	
Software	Linux+batch NDA (optimized version)
Hardware	AMD Athlon XP 1900 + 2GB RAM
Set up time	20 minutes
Imputation run time	not available
Other processing run time	not available
Complete run time	16200 seconds
Remarks	Complete run time is for a) imputation of all incomplete variables b) long computational time is due to "non intelligent" use of TS-SOM algorithm, ie. models were build for each variable separately. With intelligent use of TS-SOM algorithm the complete run time could easily be reduced to 3600 seconds (or to less).
The experiment is same as JS20001, except *that TS-SOM layer 7 (16384 nodes) was used for imputation.*	

StatFi (FS2000x) Experiment details

Experiments by Statistics Finland (StstFi) differ from the experiments by Univeristy of Jyväskylä (JyU) in two ways. Older version of NDA software was used, which was slower and more laborous than newer versions. The second difference is that Statistics Finland did use their expertise about data content when doing imputations while in JyU experiments a simple "black box" thinking (trial and error) was used.

FS20001 (FS20002) Experiment technical details	
Software	Windows + NDA (old version)
Hardware	IBM Pentium III 500Mhz + 256MB RAM
Set up time	-
Imputation run time	many hours
Other processing run time	hours
Complete run time	many hours
Remarks	Imputation of all the variables except ISCO2 and QUALSUB. FS20002 is similar to FS20001
Model type	SOM + nearest neighbor + random donor
Number of SOMs	4

In StatFI experiments FS20001 (FS20002), because of the size and the number of imputation variables data were partitioned into 11 subgroups by area code, AREA (1, 3-12), before any training. This was mainly due to memory over flow problems in the computer used. This caused the complete run time to be very many hours. After subgrouping, variables have been divided into four groups because of their nature, and different methods that were used for them are presented hereafter. Many of the imputation variables have been tested by various methods number of times. The following imputation variable groups have been created:

Household level	BATH, CENHEAT, HHSPTYPE, INSIDEWC, ROOMSNUM, TENURE
Unit level I	AGE, RELAT, SEX
Unit level II	HOURS
Unit level III	other 13 variables (excluding ISCO2 and QUALSUB)

HOUSEHOLD LEVEL IMPUTATION has been carried out selecting only breadwinners of each household as their representatives. Naturally, after imputation all the missing values in one variable of the same household get same imputed value than the related breadwinner. Possible missingness of RELAT that defines household heads is not an actual problem here. Due to a number of tests for the

household level variables the nearest neighbour imputation at the 3rd level of TS-SOM was chosen here.

- TS-SOM training variables: BATH, CENHEAT, HHSPTYPE, INSIDEWC, ROOMSNUM, TENURE, ECONPRIM, MSTATUS, PERSINHH, SEX, AGE and ISCO1.
- Coding: all the categorical variables were binarized.
- Preprocessing: continuous variable AGE, both PERSINHH and ROOMSNUM were min-max equalized.
- Imputation method: nearest neighbour using Euclidean distances at 3rd level of TS-SOM (64 clusters).
- The NN explanatory variables were ECONPRIM, HHSPTYPE, MSTATUS, PERSINHH and TENURE.

UNIT LEVEL I has been imputed randomly at 4th level of TS-SOM, that is within 256 clusters. Several methods have been tested for AGE. Specifically, the number of imputed 0s seems to be one satisfactory indicator of the goodness of imputation in the development dataset. Nearest neighbour imputation failed when compared to other methods; there were too many imputed 0s. Thus, the zero observations seem to be quite special in these data. Hence, random donor was considered the best method in this case.

TS-SOM imputation for AGE was also upgraded by a classical method (FS20002) where each household were treated as an imputation class (by variable HNUM). Within households, two new subgroups were created by using RELAT: likely older people (RELAT 0,1,2,9,10 or 15) and likely younger people in relation to a household in question. This method left 36,3% unimputed for the TS-SOM random imputation as described below.

- TS-SOM training variables: AGE, CARS, ECONPRIM, HHSPTYPE, ISCO1, LTILL, MSTATUS, PERSINHH, QUALEVEL, RELAT, ROOMSNUM and SEX.
- Coding: all the categorical variables were binarized.
- Preprocessing: continuous variable AGE, both PERSINHH and ROOMSNUM were min-max equalized.
- Imputation method: random donor at 4th level of TS-SOM (256 clusters).

UNIT LEVEL II, namely HOURS, differs from the others in that non-applicable (-9) observations were deleted. According to the development data it was expected that there are not any missing values that should be imputed as not-applicable. Prediction from the normal distribution that is estimated for all the 5th level SOM clusters by calculation of the variance and the mean was chosen to the best imputation method here.

- TS-SOM training variables: AGE, CARS, DISTWORK, ECONPRIM, HHSPTYPE, HOURS, ISCO1, LTILL, MSTATUS, PERSINHH, QUALEVEL, RELAT and SEX.
- Coding: all the categorical variables were binarized.
- Preprocessing: continuous variables AGE and HOURS, and PERSINHH were min-max equalized.
- Imputation method: prediction from the normal distribution at 5th level of TS-SOM (1024 clusters).

UNIT LEVEL III consisting of 13 variables was imputed at 5th level of TS-SOM using mean imputation.

- TS-SOM training variables: AGE, CARS, ECONPRIM, HHSPTYPE, ISCO1, LTILL, MSTATUS, PERSINHH, QUALEVEL, RELAT, ROOMSNUM and SEX.
- Coding: all the categorical variables were binarized.

- Preprocessing: continuous variable AGE and both ROOMSNUM and PERSINHH were min-max equalized.

- Imputation method: mean imputation at 5th level of TS-SOM (1024 clusters).

SOM results for ABI/Y2

Selected results of JyU and StatFi experiments for Sars Y2 are shown in tables 17 and 18. For comparison purposes also ONS nearest neighbor OS20001 and random donor without any model, XS20001, are shown.

Table 17: Selected results of the SOM categorial imputation for Sars Y2 dat.

	CENHEAT			**INSIDEWC**		
EXPERIMENT	W	D	Eps	W	D	Eps
JS20001	2032	0.35	0.34	68.01	0.00380	0.00000
JS20002	53	0.44	0.43	197.88	0.01360	0.00187
JS20003	2718	0.36	0.35	23.30	0.00094	0.00000
FS20001	31	0.40	0.39	18.91	0.00112	0.0000
OS20001	34	0.44	0.43	82.10	0.00746	0.00000
XS20001	5	0.46	0.45	143.88	0.01182	0.00008
	HHSPTYPE			**TENURE**		
EXPERIMENT	W	D	Eps	W	D	Eps
JS20001	1490	0.669	0.663	1967	0.52	0.515
JS20002	100	0.717	0.711	410	0.56	0.557
JS20003	908	0.639	0.632	5965	0.47	0.466
FS20001	315	0.612	0.604	494	0.51	0.506
OS20001	966	0.708	0.701	1758	0.62	0.616
XS20001	21	0.748	0.742	29	0.67	0.671
	DISTWORK			**LTILL**		
EXPERIMENT	W	D	Eps	W	D	Eps
JS20001	0.484	0.66	0.65	369.0	0.147	0.137
JS20002	0.491	0.82	0.82	407.1	0.176	0.166
JS20003	572	0.65	0.64	715.0	0.138	0.128
FS20001	3643	0.85	0.85	570.7	0.224	0.215
OS20001	3202	0.82	0.82	288.2	0.176	0.166
XS20001	7182	0.93	0.92	2.9	0.212	0.203
	RELAT			**SEX**		
EXPERIMENT	W	D	Eps	W	D	Eps
JS20001	1544	0.29	0.283	1613	0.28	0.27
JS20002	12	0.24	0.234	367	0.34	0.33
JS20003	2297	0.30	0.290	1595	0.28	0.27
FS20001	120	0.25	0.240	15	0.29	0.28
FS20002	2258	0.119	0.108	278	0.28	0.27
OS20001	1286	0.35	0.345	654	0.33	0.32
XS20001	7	0.70	0.694	0.0069	0.50	0.49

	ECONPRIM			ISCO2		
EXPERIMENT	W	D	Eps	W	D	Eps
JS20001	2545.1	0.68	0.67	14653	0.95	0.94
JS20002	0.8	0.65	0.64	43	0.92	0.92
JS20003	1597.1	0.69	0.68	13892	0.91	0.91
FS20001	1614	0.63	0.62	na	na	na
OS20001	1948.7	0.70	0.69	5842	0.91	0.91
XS20001	2683.9	0.82	0.81	11562	0.96	0.96

In the case of categorial variables it is a matter of some concern for us that full random donor experiments XS20001 as well as OS20001 are competetive. Therefore the use of SOM for the imputation of complex sets of categorial variables may not be a realistic option, at least for the current version of the algorithm. The benefits are not big enough. The reason for this "not better" performance over "naive" imputation is likely due several reasons. The current SOM training algorithm is not optimal for categorial variables with large numbers of categories. Secondly, the imputation models for categorial variables needs more development. Yet, the overall performance of SOM assisted random donor (JS20002 and partially FS20001) is quite good. In many cases it gives the best results and the worst cases are not too far from the baseline.

For continuous variables (AGE and HOURS) SOM imputation, especially SOM assisted random donor seems to be rater good alternative when copmpared to baseline experiments OS20001 and XS20001. In general JS20002 is quite reliable in the sense that is competetive with most of our evaluation criteria simultaneously.

Table 18: Selected SOM Sars Y2 results for the imputation of continuous variables.

	variable AGE									
EXP.	Slope	dL1	dL2	dLinf	K-S	K-S_1	K-S_2	m1	m2	MSE
JS20001	0.80	18.79	24.05	90	0.094	0.049	0.00328	4.69175	335	0.14026
JS20002	0.99	9.36	14.92	74	0.203	0.055	0.00527	0.18179	234	0.00129
JS20003	0.71	22.17	29.13	89	0.231	0.097	0.01680	9.20166	574	0.53622
FS20001	0.991	4.63	8.49	79	0.006	0.0019	0.00001	0.14056	6.96	0.00125
FS20002	0.993	4.56	7.63	88	0.011	0.0022	0.00001	0.152	4.85	0.00128
OS20001	0.84	11.25	17.45	95	0.131	0.063	0.00580	6.01962	593	0.23017
XS20001	0.77	26.08	32.16	91	0.003	0.001	0.00000	0.03374	2	0.00108
	variable HOURS									
EXP.	Slope	dL	dL2	dLinf	K-S	K-S_1	K-S_2	m1	m2	MSE
JS20001	1.0	7.1	10	70	0.13	0.05	0.0039	0.59550	152	0.00161
JS20002	0.9	9.9	14	71	0.01	0.00	0.0000	0.19233	38	0.00126
JS20003	1.0	7.4	11	71	0.17	0.03	0.0023	0.61581	134	0.00165
FS20001	0.93	9.91	16	79	0.19	0.027	0.0024	0.00248	6.33	0.00122
OS20001	0.9	16.5	24	90	0.24	0.12	0.0257	10.93863	327	0.13779
XS20001	0.6	27.1	33	90	0.45	0.22	0.0899	20.53781	624	0.48263

The evaluation results for FS20001 and FS20002 seem to be fine. Especially SEX, RELAT, and the easy ones: BATH, INSIDEWC and COBIRTH are well imputed. The problem is that the statistics given do not take into account the monotonous nature of ROOMSNUM. AGE is very well imputed with the very low absolute and squared deviance at the unit level (dL1 and dL2). The Upgrade method for AGE does not show a notable improvement as it did in the development data. HOURS was considered as a very interesting continuous variable and thus was tested carefully. However, the evaluation results do not fully correspond with expectations. But the results are still rather competitive.

2.2.6 Dataset: UK Household Y3 (SARS)

We start with a warning ! We think that imputation results of Sars Y3 data set cannot be taken too seriously. Our reason for this statement is that the results do not reveal the imputation performance of the algorithms. Instead evaluation statistics, especially categorial ones, seem to depend on the number of edits. Maybe the reason for this is the sensitivity of Wald statistics for variables with large number of categories, but there is also an obvious problem: different experiments are done with different number of changes in the data sets. Although the statistics is computed over the whole data set, many descriptors are still sensitive to the number of differences between true and imputed data sets. Thus the **evaluation results are inherently conditionalized over different data sets !** This is wrong.

To illustrate the problem we have made two additional experiments JS30005 and JS30006. Both are real and honest experiments that were done before the true data was presented to us. But they are constructed to get better evaluation statistics than what we did get from our original experiments JS30001, JS30002, JS30003 and JS30004. Especially JS30006 is interesting. All imputed values are the same as in JS30005, execpt that no outliers were detected, and therefore no outliers were imputed in JS30006. Yet JS30006 is very much better than JS30005 according to Wald statistics.

The JS3000x experiments were made with a software version that was under development and did not support different editing parameters for variables under same SOM model. Therefore a total of 26 models (one for each imputed variable) had to be made. We like to note that such a problem does not exist in our current software version and all data can be imputed with about 5 different SOM models. An additional note is required about JS30002 and JS30003, which can be used only for the evaluation of editing, because outliers were marked as missing values and are omitted by the NAG evaluation software.

The experiments are summarized in table 19. First four experiments are rather similar. They use maximum probability class in SOM nodes for categorial variables and Normal pdf model for continuous ones. The main difference is the outlier detection "cut probability" that controls how easily a sample is marked as an outlier. Last two experiments, JS30005 and JS30006, are imputed with random donor in SOM nodes. The editing parameters of JS30005 are the same as in JS30002 and no editing was done in JS30006. As before XS30000, XS30050 and XS30100 are baseline results but only XS30000 is realistic.

We remaind again that only JS30001, JS30004, JS30005 and JS30006 are realistic for the evaluation of imputation performance because in these experiments no missing values were left in the data files.

Table 19: Summary of SOM experiments with the SARS Y3 data set.

Experiment	OK	Description	models	nodes	rules	runtime
JS30001	*	SOM + Posterior+Norm. pdf	26	64-1024	some	26×20 min
JS30002		SOM + Posterior+Norm. pdf	26	64-1024	some	26×20 min
JS30003		SOM + Posterior+Norm. pdf	26	64-1024	some	26×20 min
JS30004	*	SOM + Posterior+Norm. pdf	26	64-1024	some	26×20 min
JS30005	*	SOM+random donor	26	64-1024	some	26×20 min
JS30006	*	SOM+random donor (no edits)	26	64-1024	some	26×20 min
XS30000	Random donor 0% errors detected, marked as outliers and imputed					
XS30050	Random donor 50% errors detected, marked as outliers and imputed					
XS30100	Random donor 100% errors detected, marked as outliers and imputed					

The Sars experiments were made with earlier version of the software, which was under development. It was mainly a problem with the user interface that made us build separate SOM models for each of the variables, thus a total of 26 models. All experiments were then made with the same set of models by changing only the ediing parameters and the imputation method. In addition in all models the

following preprocessing was used.

Categorial variables were dummy coded and imputed using mean imputation within clusters. The final values of imputed categorical variables were picked randomly according to posterior probabilities of the categories.

Continuous variables were robustly min-max equalized using 5% and 95% fractiles.

Pre-edit rules Hard errors, out of bounds and invalid categories, were identified at initial phase, and their error probability was set to 1.0. All hard errors were set to missing data value and imputed.

The model structure (covariates) are desctibed in the following table and are same for all experiments JS30001, JS30002, JS30003,JS30004, JS30005, JS30006. Because of the strong similarity between the models only the main differences are reported here.

variable	covariates (same in all JS3000x experiments)
AGE	AGE, QUALNUM, QUALEVEL, MSTATUS, ECONPRIM, SEX, ISCO1, LTILL
LTILL	LTILL, SEX, ECONPRIM, AGE, ISCO1, QUALEVEL, HOURS, HHSPTYPE, CENHEAT
MSTATUS	MSTATUS, SEX, ECONPRIM, AGE, ISCO1, QUALNUM, ROOMSNUM, PERSINHH
RELAT	RELAT, SEX, ECONPRIM, ROOMSNUM, PERSINHH, MSTATUS, AGE, QUALEVEL
SEX	SEX, RELAT, ECONPRIM, ISCO1, AGE, MSTATUS, HOURS, QUALEVEL
BATH	BATH, HHSPTYPE, INSIDEWC, CENHEAT, QUALEVEL, ISCO1, LTILL, ROOMSNUM, PERSINHH
CENHEAT	CENHEAT, HHSPTYPE, ROOMSNUM, BATH, INSIDEWC, CARS, PERSINHH, QUALEVEL, ISCO1, LTILL
HHSPTYPE	HHSPTYPE, CENHEAT, BATH, INSIDEWC, CARS, ROOMSNUM, PERSINHH, QUALNUM, ISCO1, LTILL
INSIDEWC	INSIDEWC, HHSPTYPE, CENHEAT, BATH, ROOMSNUM, PERSINHH, QUALEVEL, ISCO1, LTILL, HOURS
ROOMSNUM	ROOMSNUM, HHSPTYPE, PERSINHH, CARS, INSIDEWC, CENHEAT, BATH, TENURE
COBIRTH	COBIRTH, QUALNUM, QUALEVEL, AGE, SEX, MSTATUS, LTILL, ECONPRIM, ISCO1
DISTWORK	DISTWORK, WORKPLCE, ECONPRIM, AGE, HOURS, LTILL, MSTATUS, SEX, ISCO1
ECONPRIM	ECONPRIM, WORKPLCE, AGE, HOURS, SEX, ISCO1, QUALEVEL, MSTATUS
HOURS	HOURS, LTILL, WORKPLCE, ECONPRIM, ISCO1, SEX, AGE, MSTATUS
ISCO1	ISCO1, QUALNUM, QUALEVEL, ECONPRIM, SEX, AGE, HOURS, MSTATUS
ISCO2	ISCO2, QUALEVEL, ECONPRIM, SEX, ISCO1, AGE, WORKPLCE, HOURS
QUALEVEL	QUALEVEL, ISCO1, SEX, QUALNUM, HOURS, TERMTIM, MSTATUS, ECONPRIM, WORKPLCE
QUALNUM	QUALNUM, ISCO1, ECONPRIM, SEX, HOURS, MSTATUS, QUALEVEL, WORKPLCE, AGE

variable	covariates
QUALSUB	QUALSUB, ISCO1, SEX, AGE, HOURS, MSTATUS, ECONPRIM, WORKPLCE
WORKPLCE	WORKPLCE, ECONPRIM, DISTWORK, ISCO1, SEX, AGE, HOURS, LTILL, MSTATUS
MIGORGN	MIGORGN, AGE, SEX, TERMTIM, ISCO1, ECONPRIM, HOURS, LTILL, WORKPLCE
RESIDSTA	RESIDSTA, AGE, MSTATUS, QUALEVEL, TERMTIM, ISCO1, ECONPRIM, SEX, LTILL, HOURS
URVISIT	URVISIT, AGE, HOURS, LTILL, QUALEVEL, RESIDSTA, TERMTIM, ISCO1, ECONPRIM
TERMTIM	ECONPRIM, RESIDSTA, SEX, ISCO1, QUALEVEL, MSTATUS, , AGE, HOURS
CARS	CARS, PERSINHH, HHSPTYPE, ROOMSNUM, AGE, ISCO1, , ECONPRIM, SEX, QUALEVEL
TENURE	TENURE, ROOMSNUM, PERSINHH, AGE, MSTATUS, , QUALEVEL, SEX, ISCO1, ECONPRIM, HHSPTYPE

The technical details of the models are given in the following tables.

JS30001 Experiment technical details	
Software	Linux+batch NDA (optimized version)
Hardware	AMD Athlon XP 1900 + 2GB RAM
Set up time	20 minutes (not including data analysis)
run time	26*20 minutes
Remarks a)	Complete run time is for edit/imputation of all incomplete variables
Remarks b)	Long computational time is due to "non intelligent" use of TS-SOM algorithm, ie. models were build for each variable separately. With intelligent use of TS-SOM algorithm the complete run time could easily be reduced to 3600 seconds (or to less).
Model type	SOM + Posterior probability (categories) + Normal Pdf (continuous varaibles)
Remark	Posterior probabilities are node mean values for categories that are trained with 0,1 dummy values.
SOMs	26
SOM for **AGE**	SOM +Normal pdf (age)
Parameters	layer 5 (1024 nodes), Sigma1=4.0, Sigma2=21.0, Edit Cut Pr=0.025.
SOM for **LTILL**	SOM + node mean (Posterior prob.)
Parameters	layer 3 (64 nodes), Edit Cut Pr=0.0, Train Cut Pr=0.2
SOM for **MSTATUS**	SOM + node mean (Posterior prob.)
Parameters	layer 3 (64 nodes), Edit Cut Pr=0.05, Train Cut Pr=0.2
SOM for **RELAT**	SOM + node mean (Posterior prob.)
Parameters	layer 3 (64 nodes), Edit Cut Pr=0.2, Train Cut Pr=0.1
SOM for **SEX**	SOM + node mean (Posterior prob.)
Parameters	layer 3 (64 nodes), Edit Cut Pr=0.2, Train Cut Pr=0.05

JS30001 Experiment technical details (cont.)	
SOM for **BATH**	SOM + node mean (Posterior prob.)
Parameters	layer 3 (64 nodes),Edit Cut Pr=0.2, Train Cut Pr=0.0
SOM for **CENHEAT**	SOM + node mean (Posterior prob.)
Parameters	layer 3, Edit Cut Pr=0.0, Train Cut Pr=0.3
SOM for **HHSPTYPE**	SOM + node mean (Posterior prob.)
Parameters	layer 3,Edit Cut Pr=0.1, Train Cut Pr=0.3
SOM for **INSIDEWC**	SOM + node mean (Posterior prob.)
Parameters	layer 3 ,Edit Cut Pr=0.1, Train Cut Pr=0.2
SOM for **ROOMSNUM**	SOM + node mean (Posterior prob.)
Parameters	layer 5, Sigma1=2.5, Sigma2=0.85, Edit Cut Pr=0.5
SOM for **COBIRTH**	SOM + node mean (Posterior prob.)
Parameters	layer 4, Edit Cut Pr=0.49, Train Cut Pr=0.3
SOM for **DISTWORK**	SOM + node mean (Posterior prob.)
Parameters	layer 4, Edit Cut Pr=0.7, Train Cut Pr=0.3
SOM for **ECONPRIM**	SOM + node mean (Posterior prob.)
Parameters	layer 4, Edit Cut Pr=0.5, Train Cut Pr=0.3
SOM for **HOURS**	SOM + Normal pdf (age)
Parameters	layer 4, Sigma1=6.0, Sigma2=3.0, Edit Cut Pr=0.3
SOM for **ISCO1**	SOM + node mean (Posterior prob.)
Parameters	layer 4, Edit Cut Pr=0.6, Train Cut Pr=0.3
SOM for **ISCO2**	SOM + node mean (Posterior prob.)
Parameters	layer 5, Edit Cut Pr=0.95, Train Cut Pr=0.5
SOM for **QUALEVEL**	SOM + node mean (Posterior prob.)
Parameters	layer 3, Edit Cut Pr=0.225, Train Cut Pr=0.3
SOM for **QUALNUM**	SOM + node mean (Posterior prob.)
Parameters	layer 4, Edit Cut Pr=0.25, Train Cut Pr=0.25
SOM for **QUALSUB**	SOM + node mean (Posterior prob.)
Parameters	layer 3, Edit Cut Pr=0.15, Train Cut Pr=0.3
SOM for **WORKPLCE**	SOM + node mean (Posterior prob.)
Parameters	layer 3, Edit Cut Pr=0.15, Train Cut Pr=0.3
SOM for **MIGORGN**	SOM + node mean (Posterior prob.)
Parameters	layer 3, Edit Cut Pr=0.15, Train Cut Pr=0.3
SOM for **RESIDSTA**	SOM + node mean (Posterior prob.)
Parameters	layer 3, Edit Cut Pr=0.01, Train Cut Pr=0.2
SOM for **URVISIT**	SOM + node mean (Posterior prob.)
Parameters	layer 3, Edit Cut Pr=0.15, Train Cut Pr=0.3
SOM for **TERMTIM**	SOM + node mean (Posterior prob.)
Parameters	layer 4, Edit Cut Pr=0.15, Train Cut Pr=0.3
SOM for **CARS**	SOM + node mean (Posterior prob.)
Parameters	layer 4, Edit Cut Pr=0.05, Train Cut Pr=0.3
SOM for **TENURE**	SOM + node mean (Posterior prob.)
Parameters	layer 5, Edit Cut Pr=0.45, Train Cut Pr=0.5

JS30002 Experiment technical details	
Software	Linux+batch NDA (optimized version)
Hardware	AMD Athlon 1900+ processor + 2GB RAM
Set up time	20 minutes
Other processing run time	NA (data preprocess/TS-SOM models build)
Complete run time	20min*26 = 520 minutes
Remarks	Run times are for building 26 TS-SOM models and editing/imputing 26 variable groups
AGE	TS-SOM layer 5, Edit Cut Pr=0.01 Sigma1=4.2, Sigma2=21.0
LTILL	TS-SOM layer 4, Edit Cut Pr=0.0, Train Cut Pr=0.2
MSTATUS	TS-SOM layer 5, Edit Cut Pr=0.05, Train Cut Pr=0.02
RELAT	TS-SOM layer 4, Edit Cut Pr=0.1, Train Cut Pr=0.2
SEX	TS-SOM layer 3, Edit Cut Pr=0.05, Train Cut Pr=0.2
BATH	TS-SOM layer 4, Edit Cut Pr=0.0, Train Cut Pr=0.2
CENHEAT	TS-SOM layer 4, Edit Cut Pr=0.0, Train Cut Pr=0.3
HHSPTYPE	TS-SOM layer 4, Edit Cut Pr=0.1, Train Cut Pr=0.3
INSIDEWC	TS-SOM layer 4, Edit Cut Pr=0.1, Train Cut Pr=0.2
ROOMSNUM	TS-SOM layer 4, Edit Cut Pr=0.35,Train Cut Pr=0.2
COBIRTH	TS-SOM layer 5, Edit Cut Pr=0.49, Train Cut Pr=0.3
DISTWORK	TS-SOM layer 5, Edit Cut Pr=0.7, Train Cut Pr=0.3
ECONPRIM	TS-SOM layer 5, Edit Cut Pr=0.6, Train Cut Pr=0.4
HOURS	TS-SOM layer 4, Edit Cut Pr=0.4, Sigma1=6.0, Sigma2=3.0
ISCO1	TS-SOM layer 5, Edit Cut Pr=0.7, Train Cut Pr=0.6
ISCO2	TS-SOM layer 6, Edit Cut Pr=0.95, Train Cut Pr=0.85
QUALEVEL	TS-SOM layer 4, Edit Cut Pr=0.225, Train Cut Pr=0.3
QUALNUM	TS-SOM layer 5, Edit Cut Pr=0.25, Train Cut Pr=0.25
QUALSUB	TS-SOM layer 4, Edit Cut Pr=0.15, Train Cut Pr=0.3
WORKPLCE	TS-SOM layer 4, Edit Cut Pr=0.15, Train Cut Pr=0.3
MIGORGN	TS-SOM layer 5, Edit Cut Pr=0.15, Train Cut Pr=0.3
RESIDSTA	TS-SOM layer 4, Edit Cut Pr=0.01, Train Cut Pr=0.2
URVISIT	TS-SOM layer 4, Edit Cut Pr=0.15, Train Cut Pr=0.3
TERMTIM	TS-SOM layer 4, Edit Cut Pr=0.1, Train Cut Pr=0.2
CARS	TS-SOM layer 4, Edit Cut Pr=0.1, Train Cut Pr=0.3
TENURE	TS-SOM layer 5, Edit Cut Pr=0.5, Train Cut Pr=0.5

JS30003 Experiment technical details	
Software	Linux+batch NDA (optimized version)
Hardware	AMD Athlon XP 1900 + 2GB RAM
Set up time	20 minutes (no data analysis)
Other processing run time	NA seconds (data preprocess/TS-SOM model build)
Complete run time	26*20 minutes
AGE	TS-SOM layer 5Sigma1=4.2, Sigma2=21.0, Edit Cut Pr=0.01
LTILL	TS-SOM layer 3, Edit Cut Pr=0.0, Train Cut Pr=0.2
MSTATUS	TS-SOM layer 4, Edit Cut Pr=0.025, Train Cut Pr=0.2
RELAT	TS-SOM layer 3, Edit Cut Pr=0.1, Train Cut Pr=0.2
SEX	TS-SOM layer 3, Edit Cut Pr=0.05, Train Cut Pr=0.2
BATH	TS-SOM layer 3, Edit Cut Pr=0.0, Train Cut Pr=0.2
CENHEAT	TS-SOM layer 3, Edit Cut Pr=0.5, Train Cut Pr=0.4
HHSPTYPE	TS-SOM layer 3, Edit Cut Pr=0.1, Train Cut Pr=0.3
INSIDEWC	TS-SOM layer 3, Edit Cut Pr=0.1, Train Cut Pr=0.2
ROOMSNUM	TS-SOM layer 4, Edit Cut Pr=0.9975, Train Cut Pr=0.2
COBIRTH	TS-SOM layer 5, Edit Cut Pr=0.49, Train Cut Pr=0.3
DISTWORK	TS-SOM layer 5, Edit Cut Pr=0.7, Train Cut Pr=0.3
ECONPRIM	TS-SOM layer 5, Edit Cut Pr=0.6, Train Cut Pr=0.4
HOURS	TS-SOM layer 4, Sigma1=3.0, Sigma2=2.0, Edit Cut Pr=0.65
ISCO1	TS-SOM layer 5, Edit Cut Pr=0.65, Train Cut Pr=0.4
ISCO2	TS-SOM layer 6, Edit Cut Pr=0.95, Train Cut Pr=0.85
QUALEVEL	TS-SOM layer 3, Edit Cut Pr=0.1, Train Cut Pr=0.3
QUALNUM	TS-SOM layer 4, Edit Cut Pr=0.25, Train Cut Pr=0.25
QUALSUB	TS-SOM layer 4, Edit Cut Pr=0.15, Train Cut Pr=0.3
WORKPLCE	TS-SOM layer 3, Edit Cut Pr=0.15, Train Cut Pr=0.3
MIGORGN	TS-SOM layer 5, Edit Cut Pr=0.15, Train Cut Pr=0.3
RESIDSTA	TS-SOM layer 3, Edit Cut Pr=0.01, Train Cut Pr=0.2
URVISIT	TS-SOM layer 4, Edit Cut Pr=0.175, Train Cut Pr=0.45
TERMTIM	TS-SOM layer 3, Edit Cut Pr=0.1, Train Cut Pr=0.2
CARS	TS-SOM layer 4, Edit Cut Pr=0.1, Train Cut Pr=0.3
TENURE	TS-SOM layer 4, Edit Cut Pr=0.1, Train Cut Pr=0.3

JS30004 Experiment technical details	
Software	Linux+batch NDA (optimized version)
Hardware	AMD Athlon 1900+ processor + 2GB RAM
Set up time	20 minutes
Other processing run time	NA (data preprocess/TS-SOM models build)
Complete run time	20min*26 = 520 minutes
Remarks	Run times are for building 26 TS-SOM models and editing/imputing 26 variable groups
AGE	TS-SOM layer 3, Sigma1=4.0, Sigma2=21.0, Edit Cut Pr=0.01
LTILL	TS-SOM layer 3, Edit Cut Pr=0.0, Train Cut Pr=0.2
MSTATUS	TS-SOM layer 4, Edit Cut Pr=0.025, Train Cut Pr=0.2
RELAT	TS-SOM layer 3, Edit Cut Pr=0.1, Train Cut Pr=0.2
SEX	TS-SOM layer 3, Edit Cut Pr=0.05, Train Cut Pr=0.15
BATH	TS-SOM layer 3, Edit Cut Pr=0.0, Train Cut Pr=0.2
CENHEAT	TS-SOM layer 4, Edit Cut Pr=0.0, Train Cut Pr=0.25
HHSPTYPE	TS-SOM layer 3, Edit Cut Pr=0.1, Train Cut Pr=0.3
INSIDEWC	TS-SOM layer 3, Edit Cut Pr=0.1, Train Cut Pr=0.2
ROOMSNUM	TS-SOM layer 4, Edit Cut Pr=0.33, Train Cut Pr=0.2
COBIRTH	TS-SOM layer 5, Edit Cut Pr=0.49, Train Cut Pr=0.23
DISTWORK	TS-SOM layer 5, Edit Cut Pr=0.7, Train Cut Pr=0.23
ECONPRIM	TS-SOM layer 5, Edit Cut Pr=0.6, Train Cut Pr=0.34
HOURS	TS-SOM layer 4, Sigma1=6.0, Sigma2=3.0, Edit Cut Pr=0.38
ISCO1	TS-SOM layer 5, Edit Cut Pr=0.65, Train Cut Pr=0.4
ISCO2	TS-SOM layer 6, Edit Cut Pr=0.95, Train Cut Pr=0.85
QUALEVEL	TS-SOM layer 4, Edit Cut Pr=0.2, Train Cut Pr=0.3
QUALNUM	TS-SOM layer 4, Edit Cut Pr=0.25, Train Cut Pr=0.25
QUALSUB	TS-SOM layer 4, Edit Cut Pr=0.15, Train Cut Pr=0.3
WORKPLCE	TS-SOM layer 4, Edit Cut Pr=0.125, Train Cut Pr=0.3
MIGORGN	TS-SOM layer 5, Edit Cut Pr=0.125, Train Cut Pr=0.3
RESIDSTA	TS-SOM layer 4, Edit Cut Pr=0.005, Train Cut Pr=0.2
URVISIT	TS-SOM layer 4, Edit Cut Pr=0.135, Train Cut Pr=0.3
TERMTIM	TS-SOM layer 4, Edit Cut Pr=0.05, Train Cut Pr=0.2
CARS	TS-SOM layer 4, Edit Cut Pr=0.05, Train Cut Pr=0.3
TENURE	TS-SOM layer 5, Edit Cut Pr=0.45, Train Cut Pr=0.5

JS30005 Experiment technical details	
Software	Linux+batch NDA (optimized version)
Hardware	AMD Athlon 1900+ processor + 2GB RAM
Set up time	40 minutes
Imputation run time	900 seconds
Complete run time	32100 seconds
Remarks	a) Run times are for building 26 edit TS-SOM models and 11 imputation TS-SOM models b) long computational time is due to "non intelligent" use of TS-SOM algorithm, ie. models were build for each variable separately. With intelligent use of TS-SOM algorithm the complete run time could easily be reduced to 3600 seconds (or to less).
TRAIN :	All train parameters were same as in experiment JS30004
EDIT MODEL:	All edit parameterswere same as in experiment JS30004.
IMPUTATION MODEL:	Random donor in SOM nodes for all variables = 1214563 imputations (missing + outlies) (680000 were missing of 492472 records)

JS30006 Experiment technical details	
Software	Linux+batch NDA (optimized version)
Hardware	AMD Athlon 1900+ processor + 2GB RAM
Set up time	40 minutes
Edit run time	NA
Imputation run time	900 seconds
Other processing run time	NA (data preprocess/TS-SOM models build)
Complete run time	32100 seconds
Remarks	a) Run times are for building 26 edit TS-SOM models and 11 imputation TS-SOM models b) long computational time is due to "non intelligent" use of TS-SOM algorithm, ie. models were build for each variable separately. With intelligent use of TS-SOM algorithm the complete run time could easily be reduced to 3600 seconds (or to less).
TRAIN :	All train parameters were same as in experiment JS30004
EDIT MODEL:	**Only trivial errors** (out of bounds and invalid categories) were edited
IMPUTATION MODEL:	Random donor in SOM nodes for all variables = 705512 imputations (missing + outlies) (680000 were missing of 492472 records)

SOM results for ABI Y3

Selected results of JyU experiments for Sars Y3 are shown in tables 20, 21, 22 and 23. As before extra experiments XS30000, XS30050 and XS30100 are used as baselines about imputation performance. If our imputation model is any good we should do better than naive random donor without any editing (XS30000). In practice, however, we see that XS30000 is quite competetive against JS30001 and JS30004.

Table 20: Selected results of the SOM categorial editing for Sars Y3 data.

	CENHEAT				**INSIDEWC**			
EXPERIMENT	alpha	beta	delta	Dcat	alpha	beta	delta	
JS30001	1.0	0.000	0.036	0.036	0.7261	0.0019	0.03624	
JS30002	1.0	0.051	0.085	0.036	0.4369	0.0066	0.02709	
JS30003	1.0	0.051	0.085	0.036	0.4369	0.0066	0.02709	
JS30004	1.0	0.051	0.085	0.036	0.4369	0.0066	0.02709	
JS30005	1.0	0.043	0.078	0.036	0.4369	0.0066	0.02709	
JS30006	1.0	0.000	0.036	0.036	1.0000	0.0000	0.04741	
XS30000	1.0	0.000	0.036	0.036	1.0000	0.0000	0.04741	
XS30050	0.8	0.000	0.029	0.029	0.5042	0.0000	0.02391	
XS30100	0.6	0.000	0.021	0.021	0.0098	0.0000	0.00047	
	HHSPTYPE				**TENURE**			
EXPERIMENT	alpha	beta	delta	Dcat	alpha	beta	delta	
JS30001	0.930	0.017	0.040	0.023	0.0	0.0057	0.0057	
JS30002	0.904	0.041	0.063	0.023	0.0	0.0057	0.0057	
JS30003	0.904	0.041	0.063	0.023	0.0	0.0057	0.0057	
JS30004	0.904	0.041	0.063	0.023	0.0	0.0057	0.0057	
JS30005	0.905	0.041	0.063	0.023	0.0	0.0055	0.0055	
JS30006	1.000	0.000	0.025	0.025	0.0	0.0000	0.0000	
JXS30000	1.000	0.000	0.025	0.025	0.0	0.0000	0.0000	
JXS30050	0.604	0.000	0.015	0.015	0.0	0.0000	0.0000	
JXS30100	0.203	0.000	0.005	0.005	0.0	0.0000	0.0000	
	RELAT				**SEX**			
EXPERIMENT	alpha	beta	delta	Dcat	alpha	beta	delta	Dcat
JS30001	0.198	0.047	0.056	0.0125	0.114	0.00086	0.0082	0.0074
JS30002	0.184	0.054	0.063	0.0116	0.113	0.00246	0.0096	0.0073
JS30003	0.184	0.054	0.063	0.0116	0.113	0.00246	0.0096	0.0073
JS30004	0.184	0.054	0.063	0.0116	0.113	0.00246	0.0096	0.0073
JS30005	0.184	0.054	0.063	0.0117	0.113	0.00052	0.0078	0.0073
JS30006	1.000	0.000	0.063	0.0632	0.114	0.00000	0.0074	0.0074
XS30000	1.000	0.000	0.063	0.0632	1.000	0.00000	0.0649	0.0074
XS30050	0.521	0.000	0.033	0.0330	0.526	0.00000	0.0341	0.0055
XS30100	0.043	0.000	0.002	0.0027	0.056	0.00000	0.0036	0.0036
	ECONPRIM				**ISCO2**			
EXPERIMENT	alpha	beta	delta	Dcat	alpha	beta	delta	Dcat
JS30001	0.967	0.0083	0.016	0.0085	0.98	0.002	0.0101	0.0078
JS30002	0.967	0.0083	0.016	0.0085	0.56	0.043	0.0475	0.0045
JS30003	0.967	0.0083	0.016	0.0085	0.56	0.043	0.0475	0.0045
JS30004	0.967	0.0083	0.016	0.0085	0.56	0.043	0.0475	0.0045
JS30005	0.968	0.0078	0.016	0.0085	0.68	0.041	0.0470	0.0055
JS30006	1.000	0.0000	0.008	0.0088	1.00	0.000	0.0080	0.0080
XS30000	1.000	0.0000	0.008	0.0088	1.00	0.000	0.0080	0.0080
XS30050	0.543	0.0000	0.004	0.0048	0.49	0.000	0.0039	0.0039
XS30100	0.087	0.0000	0.000	0.0007	0.01	0.000	0.0001	0.0001

Categorial editing statistics for selected variables is shown for some selected variables in table 20. The only conclusion is that editing performance with SOM and simple rules varies a lot between different variables. For example editing was not successful for CENHEAT and HHSTYPE but it did work quite well for TENURE, RELAT and SEX. In general editing works better for variables that have only a small number of categories.

Continuous editing statistics for variables AGE and HOURS is shown in table 21. We see from the results that about 50 % of errors can be detected using the SOM based editing method. We note again that extra experiments XS30000, XS30050 and XS30100 are done by using the knowledge of true data, and are not comparable for editing purposes. Also many JS3000x experiments were based on same or very similar editing parametes, which explains why the editing numbers are similar.

Table 21: Selected results for the editing of continuous variables for Sars Y3 data.

	variable AGE								
EXP.	alpha	beta	delta	RAE	RRASE	RER	tj	AREm1	AREm2
JS30001	0.80	0.00825	0.063	0.00485	0.00028	5.2	17.1	0.00516	0.014
JS30002	0.58	0.05746	0.094	0.01210	0.00026	5.1	50.2	0.00086	0.008
JS30003	0.58	0.05746	0.094	0.01210	0.00026	5.1	50.2	0.00086	0.008
JS30004	0.58	0.05746	0.094	0.01210	0.00026	5.1	50.2	0.00086	0.008
JS30005	0.58	0.04767	0.085	0.01209	0.00026	5.1	50.1	0.00346	0.012
JS30006	0.92	0.00000	0.064	0.01030	0.00035	5.2	30.2	0.01064	0.026
XS30000	1.00	0.00000	0.069	0.04660	0.00102	24.4	47.5	0.04728	0.346
XS30050	0.50	0.00000	0.035	0.02233	0.00070	24.4	32.8	0.02028	0.189
XS30100	0.00	0.00000	0.000 -	0.00007	0.00003	4.3	-2.0	0.00670	0.038
	variable HOURS								
EXP.	alpha	beta	delta	RAE	RRASE	RER	tj	AREm1	AREm2
JS30001	0.52	0.063	0.074	-0.00334	0.00025	3.29	-13.6	0.11	0.066
JS30002	0.47	0.102	0.111	-0.00373	0.00024	3.29	-15.8	0.17	0.106
JS30003	0.47	0.102	0.111	-0.00373	0.00024	3.29	-15.8	0.17	0.106
JS30004	0.47	0.102	0.111	-0.00373	0.00024	3.29	-15.8	0.17	0.106
JS30005	0.47	0.099	0.108	-0.00373	0.00024	3.29	-15.8	0.17	0.103
JS30006	0.76	0.000	0.018	-0.00753	0.00032	3.29	-24.2	0.05	0.031
XS30000	1.00	0.000	0.024	0.07675	0.00197	4.79	41.8	0.03	0.446
XS30050	0.50	0.000	0.012	0.04017	0.00142	4.79	30.3	0.01	0.235
XS30100	0.00	0.000	0.000	-0.00004	0.00002	1.54	-2.7	0.07	0.002

The imputation results of continuous variables are shown in table 22. Experiments JS30002 and JS30003 are omitted from this table because most of the outliers were replaced with missing value indicator, which makes the NAG software to ignore them from computations. Thus JS30002 and JS30003 are not comparable with other experiments.

As one can see the differences between experiments and their distances from the baselines (XS30xxx) are not very significant. It seems that SOM assisted random donor is better than other methods but one can argue that the improvement from naive random donor is not large enough that one should put this much effort for SOM based modelling. What is good about these results is that statistics seems to be independent of the number of edits (XS30xxx results are almost equally good).

The final and most interesting results for Sars Y3 are the evaluation tables about imputation performance, some of which are summarized in table 23. The first and most striking observation is the difference between experiments JS30005 and JS30006. These are essentially same experiments, except that only very simple edits were made in JS30006, while much more outliers were marked as errors in JS30005. As shown in edit statistics, there were almost no false alarms, which supports a conclusion that **Wald statistics, W, is highly sensitive to the number of edits !** Thus we believe that

Table 22: Selected results for the imputation of continuous variables for Sars Y3 data.

	variable AGE									
EXPERIMENT	Slope	dL1	dL2	dLinf	K-S	K-S_1	K-S_2	m1	m2	MSE
JS30001	1.12	12	16	84	0.176	0.076	0.00836	3.73	544	0.035
JS30004	1.07	13	17	85	0.159	0.080	0.00898	3.08	514	0.001
JS30005	0.98	9	16	95	0.009	0.004	0.00002	0.11	13	0.221
JS30006	0.98	8	15	95	0.012	0.004	0.00003	0.39	36	0.176
XS30000	0.73	26	32	95	0.009	0.003	0.00002	0.30	29	2.774
XS30050	0.73	26	32	95	0.011	0.004	0.00003	0.38	35	0.590
XS30100	0.73	26	32	95	0.011	0.004	0.00003	0.38	33	0.005
	variable HOURS									
EXPERIMENT	Slope	dL1	dL2	dLinf	K-S	K-S_1	K-S_2	m1	m2	MSE
JS30001	0.64	29	35	90	0.59	0.30	0.138	27	1076	8.36
JS30004	1.01	28	34	90	0.69	0.29	0.134	26	1101	14.60
JS30005	0.72	26	32	90	0.45	0.20	0.064	18	598	6.66
JS30006	0.75	24	32	90	0.43	0.21	0.078	19	617	0.78
XS30000	0.67	27	34	90	0.46	0.23	0.094	21	647	0.30
XS30050	0.67	27	34	90	0.46	0.23	0.096	21	661	0.09
XS30100	0.68	27	34	90	0.46	0.23	0.096	21	667	1.49

data files with different number of edits cannot be compared with each other. Best results according **these descriptors** are then obtained if one does no edits at all. Also baseline results support this as clearly shown by HHSPTYPE for XS30000,XS30050 and XS30100. Although errors were detected correctly using true data, results of XS30100 are worse than results of XS30000. The problem is the use of Wald statistics. According some other type of statistics the situation might look totally different. For example, in the development phase we used an information type of criteria that measures the divergence between two categorial variables, and the results were indeed quite different from Wald type of statistics. Correct editing was improving the performance of variables.

Table 23: Selected results of the SOM categorial editing for Sars Y3 data.

	CENHEAT			**INSIDEWC**		
EXPERIMENT	W	D	Eps	W	D	Eps
JS30001	11988	0.30	0.29	969	0.025	0.015
JS30004	32400	0.52	0.50	3075	0.067	0.058
JS30005	13586	0.60	0.00	3075	0.067	0.058
JS30006	151	0.44	0.43	129	0.004	0.000
XS30000	45	0.44	0.44	101	0.014	0.003
XS30050	29	0.44	0.00	99	0.011	0.001
XS30100	19	0.43	0.00	99	0.009	0.000

	HHSPTYPE			**TENURE**		
EXPERIMENT	W	D	Eps	W	D	Eps
JS30001	16780	0.815	0.810	17258	0.82	0.820
JS30004	35985	0.830	0.826	17258	0.82	0.820
JS30005	20099	0.836	0.832	1268	0.62	0.612
JS30006	995	0.711	0.704	126	0.57	0.570
XS30000	32	0.744	0.738	46	0.68	0.677
XS30050	74	0.735	0.729	46	0.68	0.677
XS30100	143	0.729	0.723	46	0.68	0.677
	DISTWORK			**LTILL**		
EXPERIMENT	W	D	Eps	W	D	Eps
JS30001	18591	0.999	0.999	6748.00	0.119	0.000
JS30004	32878	1.000	1.000	6767.00	0.120	0.000
JS30005	21375	0.924	0.921	3.75	0.174	0.000
JS30006	7096	0.853	0.847	4.41	0.173	0.000
XS30000	7433	0.930	0.926	0.34	0.211	0.202
XS30050	9340	0.935	0.931	0.30	0.206	0.198
XS30100	11186	0.936	0.932	1.61	0.204	0.000
	RELAT			**SEX**		
EXPERIMENT	W	D	Eps	W	D	Eps
JS30001	33715	0.68	0.67	10561	0.454	0.000
JS30004	28815	0.69	0.68	427	0.449	0.000
JS30005	25899	0.50	0.49	1	0.348	0.000
JS30006	1495	0.28	0.27	0	0.346	0.000
XS30000	11	0.69	0.69	2	0.498	0.491
XS30050	22	0.69	0.68	1	0.489	0.000
XS30100	30	0.68	0.00	3	0.484	0.000
	ECONPRIM			**ISCO2**		
EXPERIMENT	W	D	Eps	W	D	Eps
JS30001	9842.6	0.745	0.737	16348	0.915	0.91
JS30004	9842.6	0.745	0.737	38335	0.989	0.98
JS30005	2.0	0.710	0.701	8	0.969	0.96
JS30006	0.7	0.600	0.587	9461	0.943	0.93
XS30000	2062.1	0.832	0.824	8686	0.964	0.96
XS30050	2501.4	0.821	0.813	9520	0.964	0.96
XS30100	3061.8	0.814	0.807	10311	0.964	0.96

Final conclusions are about SOM methodology for the editing and imputation of categorial variables. As noted several times already, the SOM imputation methodology for categorial variables is not fully developed. Therefore it seems that SOM assisted random donor is safe choise and it overperforms all other SOM based imputation methods. Unfortunately the naive random donor (XS30000), which was used as a baseline seems quite competetive as well. The SOM is at its best when the number of categories is relatively small and it seems to fail when there are many, almost equal probability categories.

2.2.7 Dataset: Swiss Environment Protection Expenditures Y2 (EPE)

The Swiss Environment Protection Expenditure data set is the most difficult for SOM based editing. The data set is relatively small (1040 records), high dimensional (71 variables), and there are many functional relationships between the variables. This is almost classical example of data that is hard to learn from examples. The correct SOM imputation procedure for EPE would require the modelling of functions as a part of SOM training, which is quite laborous to do.

In our experiments, JE20001 and JE20002, we have tested if any kind of results can be achieved by using the SOM. First experiment is done without editing rules and the second is done with them. The extra experiment XE20000 is a baseline with naive random donor imputation, which usually does not work well with this type of data. These experiments are summarized in table 24.

Table 24: Summary of SOM experiments with the EPE Y2 data set.

Experiment		Description	models	nodes	edit rules	runtime
JE20001		SOM + Normal pdf + posterior prob.	9	16-64	NO	45 Sec
JE20002		SOM + Normal pdf + posterior prob.	9	16-64	YES	45 Sec
XE20000		full random donor	-	-	NO	-

The data contains large amount of zero values. Therefore the data was partitioned to subgroups, according to variables netinv, curexp, receipts and subsid. The subgroups correspond to: some net investments, no net investments, some expenditures, no expenditures, some receipts, no receipts, some subsidies and no subsidies. Investments, expenditures, receipts and subsidiers variables were imputed in corresponding subgroups. In the experiments the following preprocessing operations were done:

Categorial variables were dummy coded

Continuous variables (all the rest) All continuous variables were log transformed and MIN-MAX equalized.

Special values. There was special information of some enterprises not filling in the questionnaire (exp93 is 2 and variables equal to zero), such records were handled as missing data when training models and computing imputation statistics.

Edit rules. In second experiment JAE0002 edit rules were used to derive formulas for deterministic imputation. The deterministic imputation was used to impute approximate 20% of missing data values.

Technical details of the experiments are given in the following tables.

JE20001 technical details	
Software	Windows+NDA
Hardware	Intel Celeron/700MHz + 256MB RAM
Set up time	20 minutes
Imputation run time	10 seconds
Other processing run time	35 seconds (data preprocess/TS-SOM models build)
Complete run time	45 seconds
Remarks	Run times are for building 9 TS-SOM models and imputing 9 variable groups
Model type	SOM + normal pdf (continuous var.) + Posterior prob. (caregorial var.) only simple (out of bounds) edit rules used
Number of SOMs	9
SOM 1: To impute	layer 3, sigma2=1 act, lang, deliv, emp
Train with	act, lang, deliv, emp, totinvto, totexpto, subtot, rectot
SOM 2: To impute	layer 3, sigma2=1 totexpto
Train with	totexpto, emp, totexpot, taxexptot
SOM 3: To impute	layer 3, sigma2=1 curexpwp, curexpwm, curexpap, curexpnp, curexpot, taxexpwp, taxexpwm, taxexpap, taxexpnp, taxexpot, totexpwp, totexpwm, totexpap, totexpnp, totexpot
Train with	same as above (to impute)
SOM 4: To impute	layer 3, sigma2=1 totinvto
Train with	totinvto, emp, eopinvtot, pininvtot, othinvtot
SOM 5: To impute Train with	layer 3, sigma2=1 eopinvwp, eopinvwm, eopinvap, eopinvnp, pininvwp, pininvwm, pininvap, pininvnp, othinvwp, othinvwm, othinvap, othinvnp, totinvwp, totinvwm, totinvap, totinvnp, eopinvot, eopinvtot, pininvot, pininvtot, othinvot, othinvtot, totinvot same as above + emp, totinvto
SOM 6: To impute	layer 3, sigma2=1 rectot
Train with	rectot, emp, totinvto, totexpto, subtot
SOM 7: To impute	layer 3, sigma2=1 recwm, recap, recot, recwp, emp, rectot
Train with	same as above
SOM 8: To impute	layer 2, sigma2=1 subtot
Train with	subtot, emp, totinvto, totexpto, rectot
SOM 9: To impute	layer 2, sigma2=1 subwm, subap, subot
Train with	subwm, subap, subot, subtot, emp

JE20002 technical details	
Software	Windows+NDA
Hardware	Intel Celeron/700MHz + 256MB RAM
Set up time	20 minutes
Imputation time	10 seconds
Other time	35 seconds (data preprocess/TS-SOM models build)
Complete time	45 seconds
Remarks	Run times are for building 9 TS-SOM models and imputing 9 variable groups
Model type	Same as JE20001 except that edit rules 1-7,13-28 are used (numbers refer to EPE editrules defined in edit rules.doc)
Number of SOMs	9
EDIT RULES:	
1:	(exp93=1 AND (totinvto+totexpto) != 0) OR (exp93=2 AND (totinvto+totexpto)=0) OR exp93=3
2:	XOR(netinv,totinvto=0) OR exp93=2
3:	XOR(curexp,totexpto=0) OR exp93=2
4:	XOR(subsid,subtot=0) OR exp93=2
5:	XOR(receipts,rectot != 0) OR exp93=2
6: totinvwp	=eopinvwp+pininvwp+othinvwp
7: totinvwm	=eopinvwm+pininvwm+othinvwm
13: othinvtot	=othinvwp+othinvwm+othinvap+othinvnp+othinvot
14: totinvto	=totinvwp+totinvwm+totinvap+totinvnp+totinvot
15: totinvto	=eopinvtot+pininvtot+othinvtot
16: totinvto	=eopinvwp+eopinvwm+eopinvap+eopinvnp+eopinvot +pininvwp+pininvwm+pininvap+pininvnp+pininvot +othinvwp+othinvwm+othinvap+othinvnp+othinvot
17: totexpwp	=curexpwp+taxexpwp
18: totexpwm	=curexpwm+taxexpwm
19: totexpap	=curexpap+taxexpap
20: totexpnp	=curexpnp+taxexpnp
21: totexpot	=curexpot+taxexpot
22: curexptot	=curexpwp+curexpwm+curexpap+curexpnp+curexpot
23: taxexptot	=taxexpwp+taxexpwm+taxexpap+taxexpnp+taxexpot
24: totexpto	=totexpwp+totexpwm+totexpap+totexpnp+totexpot
25: totexpto	=curexptot+taxexptot
26: totexpto	=curexpwp+curexpwm+curexpap+curexpnp+curexpot +taxexpwp+taxexpwm+taxexpap+taxexpnp+taxexpot
27: subtot	=subwp+subwm+subap+subnp+subot
28: rectot	=recwp+recwm+recap+recnp+recot

SOM results for EPE Y2

Selected results of the EPE Y2 data set are reported in the tables 25 and 26. As before, two donor methods, nearest neighbors OE20001 and naive random donor XE20000 are used as baselines. We must note, however, that in the case of EPE data donor methods are not good in general. Therefore, although SOM seems competetive with respect to these baselines, we can't really say that SOM is the best method to handle this type of data. The only situations where the SOM based editing could be an option are those where imputation must be done quickly and the requirements for perfect results are not too severe. Thus the SOM could be used as a first approximation of what can be achieved by imputation.

Another reason why it is difficult to evaluate the performance of SOM is that only a couple of experiments were done with EPE data by different partners of the Euredit project. For example table 25 is all we know about categorial imputations.

Table 25: Selected results of categorial SOM imputation for the EPE Y2 data set.

	DELIV			**LANG**		
EXPERIMENT	W	D	Eps	W	D	Eps
JE20001	3.0	0.75	0.25	2.0	0.6666	0.0
JE20002	3.0	0.75	0.25	2.0	0.6666	0.0
OE20001	3.0	0.75	0.25	1.0	0.3333	0.0
XE20000	4.0	1.00	1.00	0.0	0.6666	0.0

For continuous variables there were also CBS experiments with carefully build edit rules. Although we do not present those results here, we note that in almost all cases experiment CE20001 was showing better results than the SOM based methods. For the benefit of SOM we can say that SOM results were never to very bad, rather JE20002 was usually between CE20001 and OE20001.

Table 26: Selected SOM EPE Y2 results for the imputation of continuous variables.

	Slope	dL1	dL2	dLinf	K-S	K-S_1	K-S_2	m1	m2	MSE
EXP.	**variable totinvwp**									
JE20001	0.433	55	146	442	0.23	0.009	0.000	22.2	19941	4.185
JE20002	0.157	58	171	524	0.76	0.015	0.005	31.7	10916	4.798
OE20001	0.671	46	136	327	0.77	0.015	0.004	39.2	20721	5.199
XE20000	0.082	133	234	2765	0.53	0.034	0.006	55.7	17340	12.384
EXP.	**variable totinvwm**									
JE20001	0.036	60	160	922	0.69	0.019	0.003	5.4	14606	8.884
JE20002	0.474	28	79	373	0.76	0.028	0.007	25.2	5861	9.020
OE20001	0.143	36	79	489	0.71	0.023	0.004	18.4	5621	8.511
XE20000	0.039	77	145	1982	0.71	0.046	0.008	24.1	9737	12.549
EXP.	**variable othinvtot**									
JE20001	2.217	27	74	148	0.70	0.029	0.007	27.7	5731	1.626
JE20002	1.001	27	53	273	0.87	0.030	0.011	27.1	2935	1.619
OE20001	1.944	27	72	195	0.67	0.028	0.006	26.4	5595	1.605
XE20000	0.565	29	84	156	0.65	0.023	0.005	20.5	3136	1.481

	Slope	dL1	dL2	dLinf	K-S	K-S_1	K-S_2	m1	m2	MSE
EXP.	**variable totinvto**									
JE20001	0.170	169	534	2608	0.19	0.007	0.000	54.4	177812	115.296
JE20002	0.604	100	376	1219	0.33	0.007	0.000	16.6	2938	95.963
OE20001	0.256	127	323	1270	0.64	0.012	0.002	50.7	59022	105.634
XE20000	0.032	481	950	16057	0.42	0.049	0.009	298.8	764749	630.817
EXP.	**variable totexptot**									
JE20001	0.014	79	266	2586	0.55	0.021	0.003	6.2	42853	4.763
JE20002	0.165	79	181	1375	0.33	0.024	0.004	35.5	11194	6.943
OE20001	0.155	71	139	2099	0.24	0.022	0.003	22.6	1532	5.314
XE20000	0.568	40	121	522	0.66	0.017	0.003	32.8	14751	4.330
EXP.	**variable totexpwp**									
JE20001	0.012	72	275	3212	0.55	0.020	0.001	33.1	71466	19.198
JE20002	0.174	44	75	1010	0.58	0.033	0.006	12.0	1223	5.434
OE20001	0.627	21	55	402	0.31	0.012	0.001	9.9	1468	3.773
XE20000	0.194	30	63	461	0.76	0.020	0.005	15.7	3189	4.587
EXP.	**variable totexpap**									
JE20001	0.038	53	274	666	0.75	0.009	0.001	8.3	75640	1.208
JE20002	0.398	32	66	424	0.28	0.018	0.002	7.7	721	1.027
OE20001	0.530	26	85	235	0.53	0.007	0.001	1.6	8565	0.754
XE20000	0.183	31	70	312	0.86	0.024	0.006	19.3	3958	1.241
EXP.	**variable totexpnp**									
JE20001	4.495	7	14	46	0.88	0.058	0.019	7.4	246	0.011
JE20002	1.928	6	12	50	0.71	0.053	0.013	6.2	207	0.008
OE20001		8	15	60	0.99	0.061	0.023	8.0	250	0.012
XE20000	0.020	8	16	60	0.98	0.058	0.022	7.5	231	0.010
EXP.	**variable totexpot**									
JE20001	5.393	20	67	197	0.93	0.018	0.006	19.9	4603	10.567
JE20002	0.848	17	58	134	0.32	0.008	0.000	2.8	1693	10.405
OE20001	0.130	21	68	197	0.92	0.017	0.005	18.5	4416	10.544
XE20000	10.093	20	67	197	0.98	0.018	0.006	20.4	4610	10.554
EXP.	**variable totexpto**									
JE20001	0.023	241	934	2362	0.17	0.019	0.000	164.6	742597	587.7247
JE20002		7	44	306	0.05	0.000	0.000	3.0	18469	31.127
OE20001	0.838	39	117	1327	0.16	0.003	0.000	4.2	40644	34.1216
XE20000	0.140	113	349	2934	0.67	0.007	0.001	20.8	48835	39.325
EXP.	**variable subtot**									
JE20001		7	11	15	0.48	0.000	0.000	7.6	168	0.010
JE20002		7	11	15	0.48	0.000	0.000	7.6	168	0.010
OE20001		1	2	2	0.48	0.480	0.230	1.4	4	0.010
XE20000		1	2	2	0.47	0.479	0.229	1.4	4	0.010
EXP.	**variable rectot**									
JE20001	0.033	85	135	1384	0.37	0.113	0.026	56.3	15008	0.982
JE20002	0.027	65	137	1348	0.33	0.056	0.008	34.9	14955	0.593
OE20001	0.070	41	132	477	0.84	0.012	0.002	1.8	13263	0.276
XE20000	3.436	21	52	130	0.89	0.052	0.014	21.4	2875	0.288

3 Conclusions

The following comments represent the viewpoint of Statistics Finland. The current NDA software includes functional editing and imputation system. The strengths of NDA in generally are its very wide graphical visualization facilities, sophisticated algorithms and especially its adaptability. The userinterface for editing and imputation is continuously under serious development from easy-to-use perspective. Its main weakness, however, is the complexity of use outside this E&I system.

The strengths of the TS-SOM algorithm are very low computational complexity, effectiveness and the main target: getting behind the hidden structures of the data. It should be noted that the algorithm for imputation is mainly used for creating imputation classes or clusters. After that, any imputation method can be selected, and actually the method can be called as TS-SOM imputation at least when centroids of the clusters are used as donors.

The strong weaknesses are among the standardization and scaling techiques. How to scale continuous and categorical equally? How to binarize equally categorical variables with unequal number of classes? How to take into account an importance between variables and especially their possible monotony? There are yet many possible solutions for these problems and they, of course, depend on the case in question. But the solutions are not necessarily always satisfactory or the problem is just very hard to solve.

Statistics Finland prefers strongly techniques in which practical know-how and so called classical methods are working together with the computationally more advanced ones such as the SOM based automatic methods. The very naive example of this co-operation in imputation is the one where part of the variables are imputed by other method and the other, harder, part by TS-SOM. These kind of solutions are nonetheless far from naive when the editing problematics are in question.

4 Bibliography

References

(1) Kohonen, T., Self-organized formation of topologically correct feature maps, Biological Cybernetics 43, 1982, 59-69.

(2) Koikkalainen, P., 1992, Artificial Intelligence Without Symbols, In proc. STeP-92 New Directions in Artificial Intelligence, Volume 2, 202-211.

(3) Piela, P. (2002)] Introduction to self-organizing maps modelling for imputation Ű techniques and technology. Research in Official Statistics, 2, 5-19.

(4) T. Hastie and W. Stuezle, Principal Curves, JASA 406 (1989), 502–516.

(5) M. LeBlanc and R. Tibshirani, Adaptive Principal Surfaces, JASA 425 (1994), 53–64.

(6) H. Ritter, T. Martinetz and K. Schulten, Neural Computation and Self-Organizing Maps (1992), Addison-Wesley.