Laboratory of Data Analysis
University of Jyväskylä

EUREDIT - WP6 reports

Getting Started with NEAT-DATA: SOM Editing and Imputation Software Implementation

Pasi P. Koikkalainen and Ismo Horppu
University of Jyväskylä

Contents

1 Introduction **3**

1.1 System requirements . . . 3

2 An overview of the program **3**

3 How to use the software **4**

3.1 Loading data . . . 6

3.2 Specification of special values (optionally) . . . 7

3.3 Specification of the SOM model . . . 8

3.3.1 Specification of variables . . . 9

3.4 Saving results . . . 11

3.5 Bootstrapping . . . 11

3.6 Saving your work for later use . . . 11

4 Analysis of results **12**

4.1 Scatterplot . . . 12

4.2 Distributions . . . 13

4.3 Evaluation of error detection . . . 14

5 APPENDIX: Some tools for data-analysis **15**

5.1 Data view . . . 15

5.2 Distribution . . . 15

5.3 Scatterplot . . . 16

5.4 Inspect values . . . 17

1 Introduction

This document is a start up guide for SOM based editing software that was developed by the University of Jyväskylä (JyU) in the Euredit project. The software has been build over Neural Data Analysis (NDA) applications platform which has been developed by JyU and is being used in many other applications of neural networks as well.

The SOM implementation is based on the Tree Structured SOM algorithm and it has been modified to work with incomplete and erroneous data.

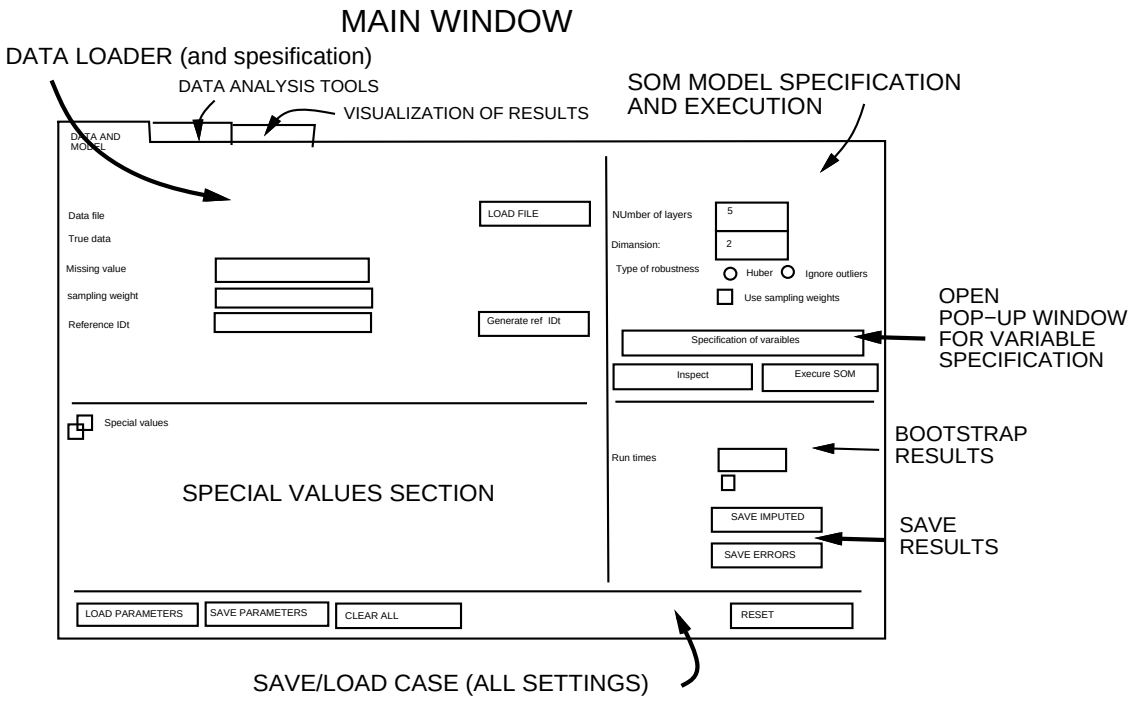
1.1 System requirements

The application works with Windows 2000/XP operating systems. There is not support for Windows 95, Windows 98, or Windows ME. The application may work with Windows NT 3.51 and Windows NT 4 (not tested). A Pentiun class processor is required. In practise a mouse is required to use the software. The amount of memory required depends on size of the dataset and complexity of used Tree Structured Self-Organizing Map (TS-SOM) model. Usually 256 MB of memory is enough.

2 An overview of the program

The main window controls most of the tools of the software and is shown in the figure 1. Other, optional tool sets are made for data-analysis and the visualization of results and are discussed briefly in the later part of this document.

Figure 1: The main window for SOM based editing and imputation.



The main window has five sections, the most important of which are:

Data loader that reads tabulated or separated raw ascii data. The user must load the "bad" data set, which will be imputed and (possibly) edited with the SOM. Additionally, if true data is

available for evaluation purposes, it can be loaded as well. True data is used for comparative evaluation of the editing/imputing performance only and is not required for model building.

SOM model specification and execution allows the user to change the SOM specification and to defined to variables that are used with the SOM model. The minimal requirement is that the user specifies required variables, their types and the method that is used for imputation and editing, if any. The actual specification of variables is done with an additional variable window.

The three other sections allow the user to save or load results and SOM specifications for later use, and define spcial values of data that require nonstandard handling with the SOM model:

Special values section allows the user to to select a special procedure to handle values that indicate some special situation like "not applicable" or "zero".

Bootstrap and save results section allows the user to save imputed data. There is also a possibility to define bootstrap procedure for multiple runs for editing and imputation. Bootstrapped values can be used for evaluation purposes, for example.

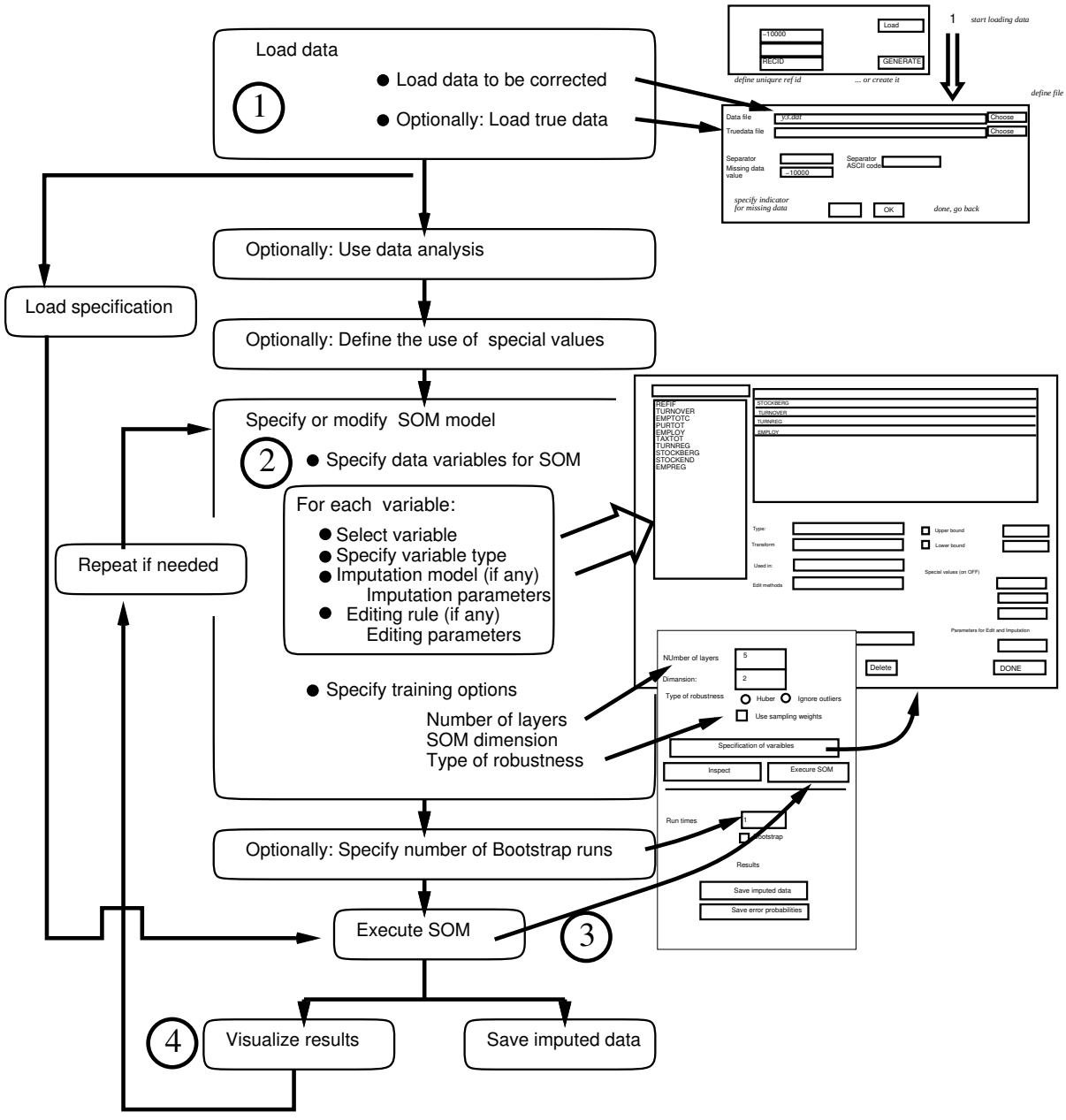
Case section allows the user to save all settings of the software or load old setting from file. There is also an exit button for the whole software.

3 How to use the software

The process of editing or imputing an erroneous and incomplete data set requires a couple of steps, as illustrated in figure 2. For any data set one needs to

1. Load data using file load button. Note that for each data set there must be unique record index variable, which should be selected at this time as well. If such a variable does not exist, it can be created in "create ref ID" button.
2. Select variables to be used in the SOM model. This is done by clicking the "specification of variables button" in the SOM section. This opens a new window. Then for each variable one needs to specify:
   (a) **Variable type** that is either **continuous** or **categorial**. Note that categorial variables are dummy coded with new indicator variables. This creates one new variable for each category. Therefore you might like to check that there are not too many (say less than 20) categories.
   (b) **Imputation method** (if any)
   (c) **Edit method** (if any)
   (d) If variable is or is not used in SOM and/or an additional nearest neighbor model.
   (e) Change parameters for editing process (if needed)
3. TS-SOM training parameters:
   (a) Number of SOM layers (typically 2-5).
   (b) The dimension of the latent SOM surface (typically 2).
   (c) Number of Bootstrap runs (typically 1)
4. Execute SOM model.

Figure 2: Main steps of the SOM editing and imputation procedure

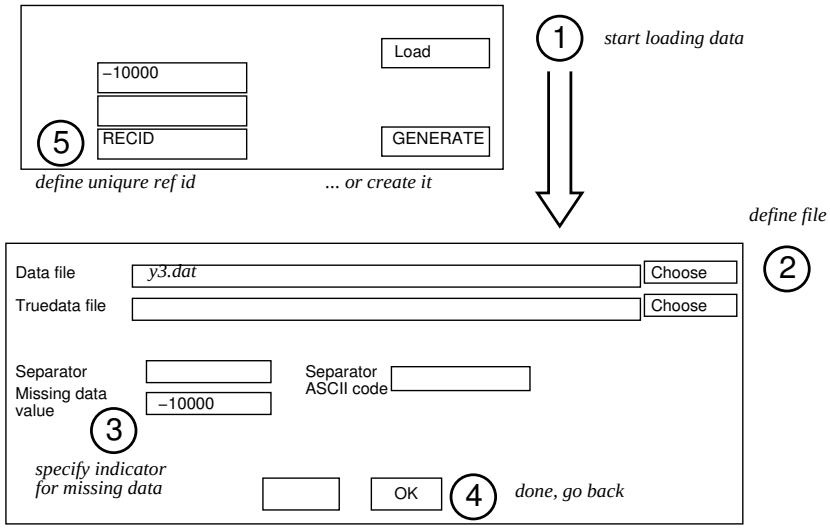


3.1 Loading data

To load a dataset do the following steps, as depicted in figure 3.

1. Push `"Load"` button (the screen changes).
2. To specify the incomplete and/or erroneous dataset file push upper `"Choose file"` button, then locate file and press button `"Open"`. (To specify the true dataset file, use the lower `"Choose file"` button.)
3. For non NDA datafile: select separator (TAB, COMMA, or CUSTOM). For the CUSTOM separator you must define ASCII code of the separator character. Specify also missing data value (which is -10000 by default).
4. Finally, press `"OK"` button and the dataset(s) should be loaded (message will appear to denote success or failure).

Figure 3: Steps that are required to load a new data set.



REMARKS:

a) filepath/name must not contain spaces.

b) load options for non NDA datafiles are “shared”, same for both the incomplete data set and the true data set..

c) before changing to a completely new dataset (containing different variables than current dataset) it is best to push ”Clear all” button from the case controls area.

After loading **the missing data value** and **the reference ID variable** must be specified.

The reference ID variable must be unique for each record. If the dataset has not a reference ID variable then push ”Generate refID” button . The button executes an action which tries to create a row number variable named REF or REFID. After creating it the variable is selected automatically. REMARK: generated reference ID variable is not saved to a file (therefore it must be recreated when loading the same dataset later).

One can also select a sampling weight variable if such is available and it is required. **Usually we do not recommed the use of sampling weights with SOM.** Examples of sampling weights and their meanings are: value of 1.0 means ”single” weight, value 2.0 means ”double” weight, and 0.5 means ”half” weight.

3.2 Specification of special values (optionally)

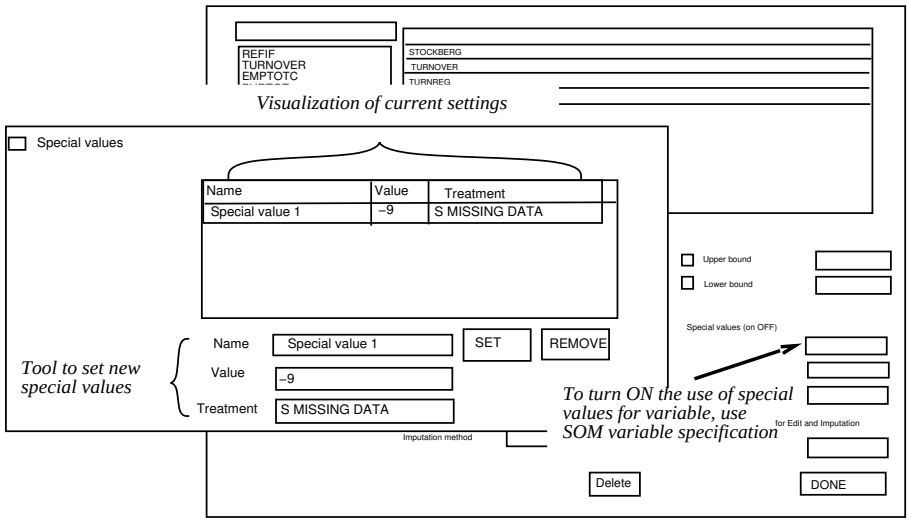
The dataset can have special values (for example, non applicable values) which can require special treatment. There are currently two ways to handle special values: by treating as missing data or by coding as a category. By selecting the former option the special values are changed to missing data values, and after processing the original values are restored. The latter option specifies that a category is created for the special value. The original missing data pattern is copied to the category too. The corresponding special values of the original variable are changed to missing data values.

A continuous variable is build in postprocessing phase using posterior probabilities of special value categories (if there are such) and continuous "value" category. A maximum probability category is picked. For a categorial variable special values are coded as categories if special value treatment(s) are not defined.

To specify a treatment for a special value do the following steps (see figure 4):

1. Switch to `"Data and model"` view.
2. Make sure that `"Special values"` checkbox is enabled (controls for special values should be visible).
3. Select a name for the special value. Currently there can be at maximum 3 special value treatments, names are "Special value 1", "Special value 2", and "Special value 3". These names correspond to Special value 1-3 ON/OFF options in specification of variables view.
4. Enter a data value next to `"Value"` label.
5. Select a treatment for the special value.

Figure 4: A tool to define new special values.

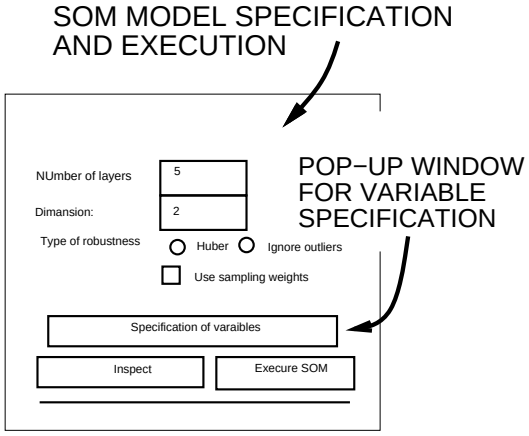


The use (or not use) of special values is defined for each variable separately. To assign a treatment for a variable you must enable corresponding special value (set its state to `"ON"`) from the **SOM model specification of variables**. There can be multiple (up to 3) treatments for a same data value, however one variable can use only one treatment at one time.

3.3 Specification of the SOM model

The specification of the SOM model includes

- setting the complexity of the SOM model (number of layers and dimension of the latent SOM surface),
- definition of robustness type to be used in training (if there are suspected errors in data)
- Definition of all variables for the SOM model. This is done in a separate window.



There are two options for robust SOM training: Huber and "ignore outliers".

With the latter option outliers are ignored (the influence function value of them is zero) when building the TS-SOM model. With the former option outliers are moved to
STD (standard deviation) $\times$ **sigma1** (Edit parameter) in every SOM node.

REMARK: by selecting "EDIT NONE" for a variable in SOM model (defined in specification of variables window) the variable is modelled without robustness.

The total number of neurons (nodes, clusters) in the SOM model can be computed from the formula:

$$neurons = (2^{dimension})^{nol-1},$$

in which *nol* denotes number of layers (whose minimum value is 2).

From the formula one can see that the number of neurons grows exponentially when increasing the number of layers, and the order of growing is defined by the SOM dimension.

By pushing `"Inspect model"` button this becomes visible. The form contains SOM layer count, robustness type, sampling weight variable, list of training variables. To see process complexity (= data record count, SOM neuron count, data dimension) push `"Complexity"` button. The data dimension computes a k-categorial variable as k-dimension (the categorial variables are dummy coded).

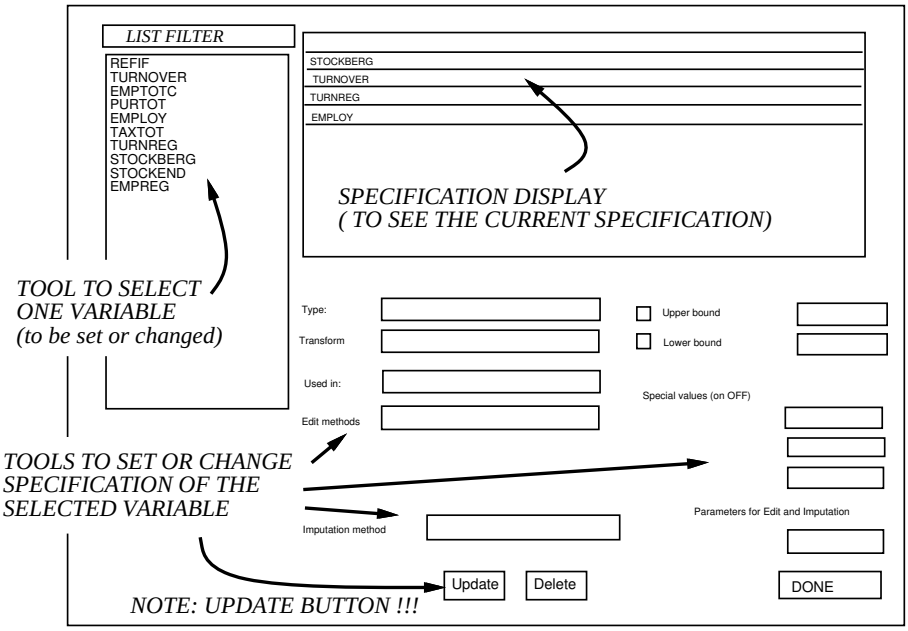
By pushing `"Execute SOM"` button the data production process is started (assuming that a dataset and a SOM specification are available). After the process is completed a message is shown. REMARK: the process cannot be cancelled by the application and during the process the software does not respond to user actions.

By pushing `"Specification of variables"` opens a new window for variable specification. This is explained in the next section.

3.3.1 Specification of variables

Specification of variables consists of five parts which are list filter, specification, list of variables, edit controls, and action buttons (bottom row), as shoen in figure 5.

Figure 5: A window for the specification of SOM variables.



On the left part of the window a tool with **a list of variables** is provied for variable selection. Before updating variable settings, or removing a variable from specification make sure that you have selected the correct variable. The selection is shown as blue background color. The tools has also a list filter with options `"ALL"` and `"SPECIFICATION SET"`. The former option shows all variables of the dataset. The latter option shows only the variables which are currently used in the specification.

Specification display contains a table of variables in the current specification. The current parameters for the variables are also displayed. This can be use to check that the specification is as the user has intended.

A number of controls can be used to set the variable parameters. The variable type can be either continuous or categorial. Selecting the type affects also to set of visible edit controls. Edit controls are:

Type: Variable type, continuous or categorial. REMARK: categorial variable with many categories can take require much memory (each category is coded as own dummy 0/1 variable). In post-processing categorial variables are build by debinarizing the dummy 0/1 variables according to their posterior probabilities.

Transform: Transform, none or logarithmic. REMARK: for logarithmic transform the values of variable must be non negative. However, special values can be negative, assuming that they are specified.

Used in: Defines whether variable is used in SOM training, in nearest neighbour imputation, or in both. If a variable is imputed using random donor or nearest neighbour method then it has to be specified to be used in SOM. With nearest neighbour imputation the variable must be specified to be used in nearest neighbour.

Edit method: “none” or “TS-SOM”. None says that outliers are not tried to be detected, and thus

TS-SOM model is not robust for the variable. Edit parameters for a continuous variable are sigma1, sigma2, and edit cut probability. Sigma1 controls the outlier decision boundary, if a observation is more than Sigma1 times standard deviation from the neuron centroid then it considered to be outlier. Default value for sigma1 is 3. Sigma2 is a scaling parameter for the normal distributions (which are used within clusters to provide error probabilities). Lowering sigma2 will increase error probabilities, and raising will decrease. The default value for sigma2 is 1.0. REMARK: sigma2 affects to normal pdf imputation too (it controls how widely the imputed values are spread). Edit cut probability defines the outlier "boundary" probability. Values with error probabilities above the edit cut probability are considered to be outliers.

For a categorial variable the edit parameters are training cut probability and edit cut probability. Both of the parameters must be between 0 and 1. Value 1 means that no observation is considered to be outlier, whereas value 0 means that all observation are considered to be outliers.

Imputation method: REMARK: There must be at least one variable with imputation method other than NONE ! This menu selects the imputation method, which can be mean, normal pdf (**p**robability **d**ensity **f**unction), uniform pdf, nearest neighbour, or random donor. In mean imputation the neuron centroids are used to fill missing data values. With normal and uniform pdf methods a corresponding density function is used to draw missing data values. REMARK: with normal pdf imputation the imputed values are cut to be at most $2 \times sigma2 \times std$ from the neuron centroid.

In nearest neighbour donators are searched using the nearest neighbour variables as distance data. In random donor donators are randomly picked from the observed records. REMARK: it is possible that nearest neighbour or random donor will fail within a cluster (if there are no donators available), in such case mean imputation is done within the "failing" clusters.

REMARK: normal pdf and uniform pdf methods make "sense" only with continuous variables, therefore the software does not allow selection of them for a categorial variable.

Lower bound: (continuous variables only). A lower boundary for a variable can be specified. If the `Lower bound` checkbox is checked then all values below the boundary are marked as missing data (before doing modeling, edit, or imputation). After the process the boundary is used to clamp out of bound values to the bound. REMARK: special values can (and should) be outside the boundary, special values must be specified so that they are not edited by the procedure.

Upper bound: (continuous variables only). Upper bound works similarly as lower bound. REMARK: (same as with lower bound).

Special value1/Special value2/Special value3: specifies whether a special value 1/2/3 treatment is used for the selected variable. See 3.2 for details of defining treatment of a special value.

Action buttons can be used to update parameters of variable and to add or remove a variable from specification. Pushing `"Done"` button will close the form. Basic actions are:

Add/update variable:

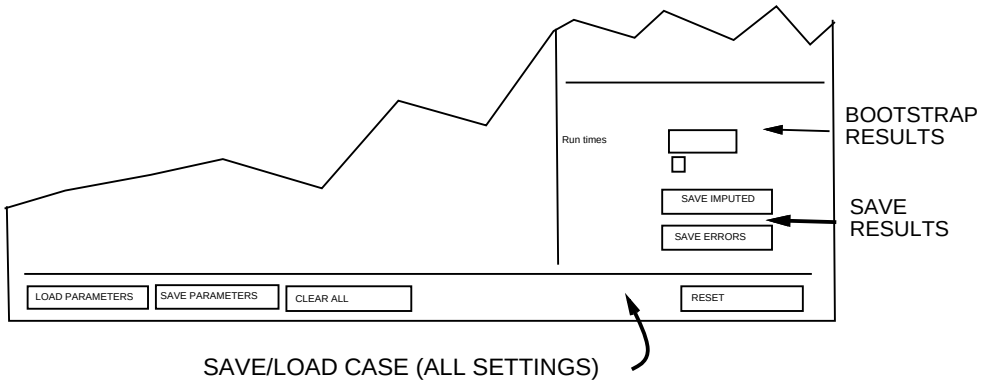
1. Select variable from the list of variables.
2. Set variable parameters.
3. Push `"Update"` button (the variable comes first in the specification table.)

Remove variable:

1. Select variable from the list.
2. Push "Delete" button (the variable should disappear from the specification table, and from the list of variables if you have selected `"SPECIFICATION SET"` filter.

There are possibilities to save the work that was done by using the tools of figure 6.

Figure 6: Tools for saving the results and the parameters of the SOM model (including the specification of variables).



3.4 Saving results

After you have executed the SOM model you can save the results.

`"Save imputed data"` button allows saving of the imputed dataset to a file (select target file and then press `"Save"` button).

Error probabilities (if the dataset was edited) can be saved to a file by pressing `"Save error probabilities"` button, and specifying the file and pressing `"Save"` button.

REMARKS:

a) the fileformat is TAB separated.

b) error probabilities file has reference ID variable and error probability variables for variables in the SOM specification.

3.5 Bootstrapping

The whole editing and imputing process can be rerun several times with bootstrap.

When the number of bootstrap runs is bigger than 1, the procedure samples the original data set with replacement, trains the SOM model and creates an instance of evaluation statistics. After several runs the evaluation statistics of the data is provided (given in the evaluation of results window) with mean values and with scatter.

Naturally the runtime of the algorithm takes as long times the normal as there are specified reruns.

3.6 Saving your work for later use

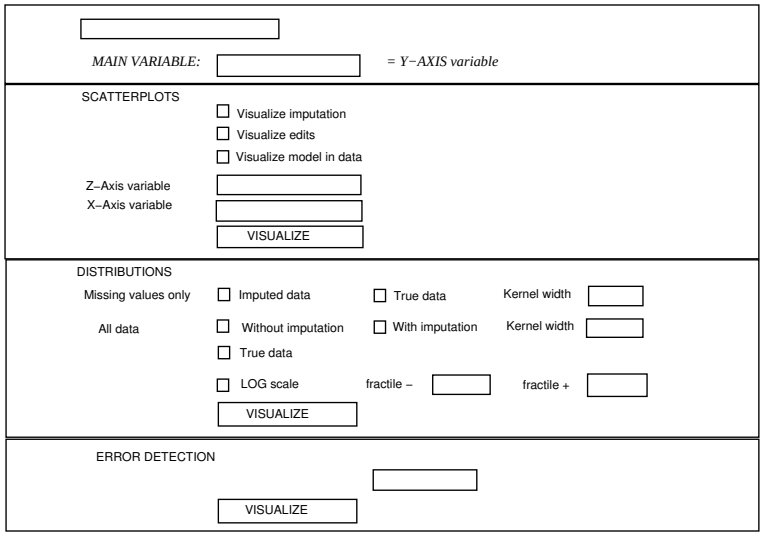
All the parameters required for edit and imputation process can be saved to file by pushing `"Save parameters"` button, and selecting a target file, and finally pushing `"Save"` button. There is also `"Load parameters"` button for loading the parameters. `"Clear all"` button clears all parameters, there is no undo for the operation (unless you have the parameters in a file).

REMARK: edit and imputation specification is saved to a file which has same name as selected target file but has an attached `.spec` extension.

4 Analysis of results

After training the SOM model it is possible to visualize the results, either with or without true data. If true data is available for evaluation purposes, more possibilities for visualization are available.

Figure 7: Tools to visualize results of editing and imputation.



The analysis of results is currently supported with three types of displays: "SCATTERPLOT", "DISTRIBUTIONS", and "ERROR DETECTION". Tools to define these visualizations are shown in figure 7. In addition to visual displays a numerical evaluation of results is possible if there is true data for comparison.

NOTE ! Visualization requires "complete data", which is either true data or imputed data. No missing values should remain in the data sets after SOM execution. Thus, if there are missing values, they must be imputed.

4.1 Scatterplot

To visualize edit and/or imputation scatterplot(s) do the following:

1. Select the main variable (this is Y-axis).
2. Select with check puttons: visualize imputation and/or edit.
3. Select X-axis variable.
4. (Optionally: Select Z-axis variable.)
5. Push `"Visualize"` button.

If the true **dataset is not available** then only **imputed missing** data values, **imputed outliers**, and **found outliers** can be visualized.

In the case of editing, when the true dataset is available, also **true errors**, **not found errors**, **detected but not true errors** , and **found errors** (correct detections) can be visualized.

In the case of imputation, when true data is available, **true values of missing data** and **true values of imputed outliers** can be visualized.

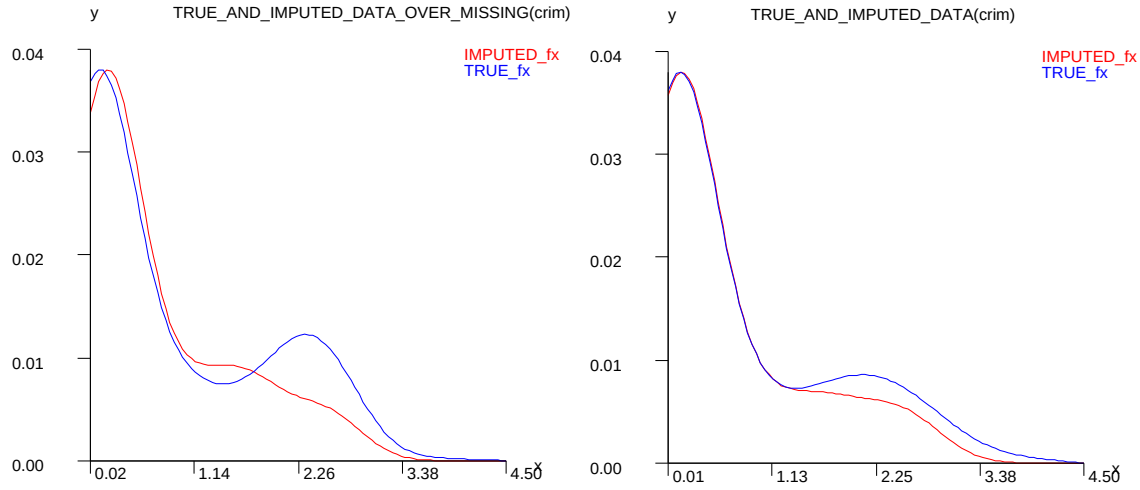
With true data known, also the imputation performance (of missing data values and/or outliers) can also be visualized by true vs. imputed scatterplot. The graphic contains a pdf estimate using contours.

The more diagonal the graphic the better. However, symmetric scatter among the diagonal means preservation of distribution.

4.2 Distributions

With this tool it is possible to visualize the distribution of imputed values and observed values. If the true dataset is available then it is possible to visualize the change in distribution over missing data values or over all data, as shown in figure 8.

Figure 8: Parzen estimators of imputed and true data.



Visualization is done with a couple of selections.

1a) Visualizing over missing data:

1. Set `"Imputed data"` checkbox.
2. (If the true dataset is available then check `"True data"` checkbox, which is next to `"Imputed data"` checkbox.)

or 1b) Instructions for visualizing imputed data:

1. Set `"Data without imputation"` checkbox.

or 1c) Instructions for visualizing over all data:

1. Set `"Data with imputation"` checkbox.
2. (If the true dataset is available then check `"True data"` checkbox, which is below `"Data without imputation"` checkbox.)

2. SET GRAPH PARAMETERS :

1. (Optionlly: set lower and upper fractiles.)
2. Set kernel variance (= gaussian kernel width). The variance is specified as data variance within specified fractiles.
3. (Optionally: `Use LOG scale`, affects to X-axis scale; REMARK: affects to scale of kernel variance too.)
4. Push `"Visualize"` button.

REMARK: special values are automatically removed from dataset(s) before computing an estimate of distribution.

4.3 Evaluation of error detection

The error detection tool can be used to evaluate the error detection performance and to decide the edit cut probability (assuming that the true dataset is available).

To visualize the error detection following steps are required:

1. Select variable.
2. Specify visualization cut probability. The parameter specifies to which probability the X-axis is stopped (right edge).
3. Press `"Visualize"` button.

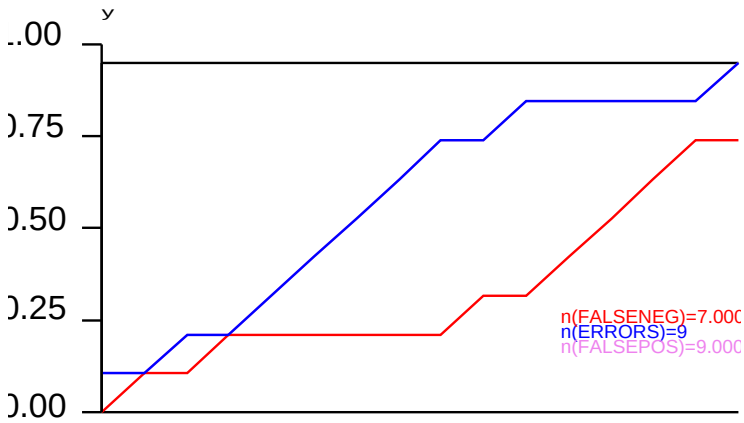
The error detection graphic consists of four line graphs which are: error probability of value, proportion of false negative errors, proportion of false positive errors, and proportion of true errors 9. The X-axis is sorted according to error probabilities from highest to smallest probability. The Y-axis is probability for the error probability line graph, and proportion for the other line graphs. The graphic contains as text amount of false negative, false positive, and true errors (in the upper right corner).

REMARK: The false negative errors line graph is cut to 1.00. Therefore if there are more false negative errors than true errors then the line graph cannot be used directly to read the amount. However, one can read the exact amount of false negative errors from `n(FALSEPOS)` label in upper right corner of the graphic.

Instructions for deciding edit cut probability (after a ”good” SOM model has been found):

1. Set edit cut probability for the variable to 0.0 (in the specification of SOM).
2. Execute SOM.
3. Visualize the error detection graphic.
4. Find a X-axis position (if such exists) in figure in which the proportion of false positive errors is higher than proportion of false negative errors.
5. Select the X-axis position if the amount of false negative errors is not too high. Otherwise move to left in X-axis until there are not too many false negative errors.
6. Read Y-axis value, use the value as edit cut probability.

Figure 9: Graphs for the evaluation of error detection.



5 APPENDIX: Some tools for data-analysis

The graphical user interface (GUI) of the NEAT-DATA algorithm provides currently only limited support for data analysis. There is support for advantaged methods in the NDA software platform, but the use of these methods requires deeper understanding of the NDA macro programming language.

The GUI supported data-analysis consists of four tools which are

- data view,
- distribution,
- scatterplot, and
- inspect values.

The tool is selected by clicking `"Analysis tool:"` combobox and selecting a tool (by pressing mouse left button).

With all the tools the user must know that the incomplete and/or erroneous dataset is named as `DPP_Data`, whereas the true dataset is named as `DPP_TrueData`.

5.1 Data view

To view the incomplete and/or erroneous dataset as grid view type `/dpp/DPP_Data` after `Name:` label. For viewing the true dataset type `/dpp/DPP_TrueData`. In case you have run the process then you can view the imputed dataset by typing `/dpp/DPP_ImputedData`.

The grid view shows at most 1000 rows from the dataset. You can use range controls to select start and end rows (row indices start from 0).

By enabling the checkbox `Edit` you can change the dataset. To do this select a data value (by clicking mouse left button over it) and type a new value. REMARK: there is no option to restore the old value (except by re-editing the value).

5.2 Distribution

The tool can be used to visualize an estimation of distribution of a variable. There is a traditional histogram estimation method available and three kernel estimators (which provide continuous estimate), see figures 10.

Do the following steps to visualize a distribution:

1. Select data -> DPP_Data (or DPP_TrueData if available).
2. Select variable.
3. Select method (histogram, parzen window, parzen window with gaussian weighting or K-nearest neighbours).
4. Select amount of samples.
5. Set method specific parameter (if there is such).
6. (Optionally: set / uncheck LOG scale, affects to X-axis scale.)
7. (Optionmally: specify lower and upper fractiles.)

The method specific parameters are:

Parzen window : window length, defines the kernel window length. The length is specified as proportion of data range (= maximum value-minimum value) within specified fractiles.

Parzen window with gaussian weighting : kernel variance which controls the shape (= width) of gaussian kernel. The variance is specified as proportion of data variance within specified fractiles.

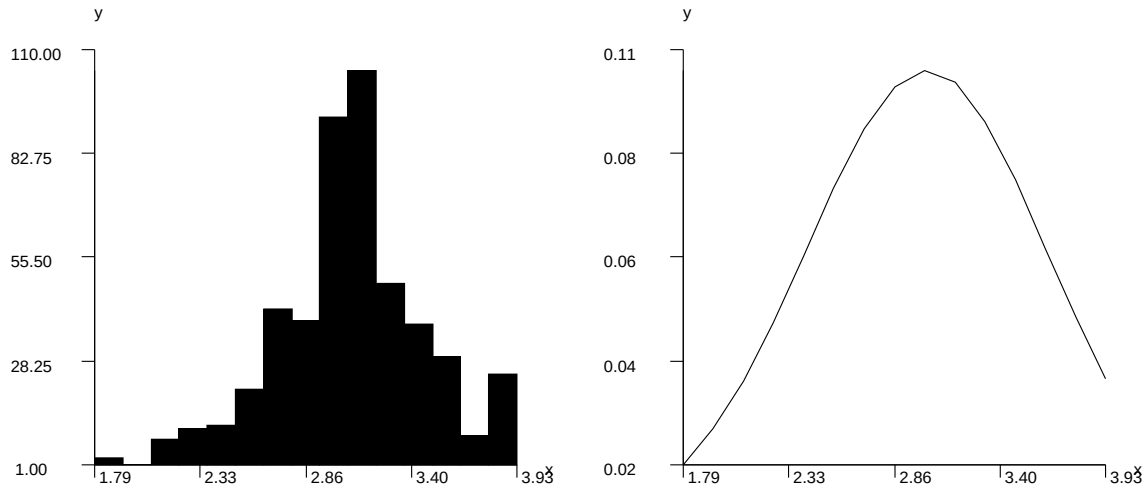
K-nearest neighbours : nearest neighbour count which controls how many neighbours are locally used to estimate the distribution.

REMARK:

a) Y-axis of histogram estimate is frequency, whereas with kernel methods the Y-axis is proportion/density.

b) special values are automatically removed from dataset(s) before computing an estimate of distribution.

Figure 10: Visualization of univariate distributions.

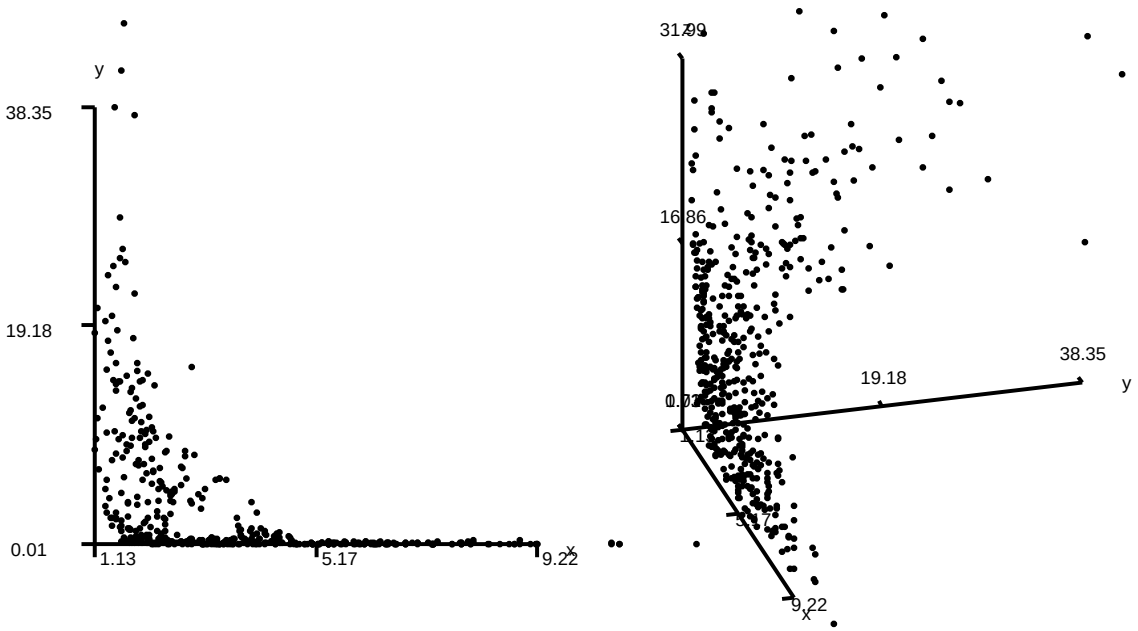


5.3 Scatterplot

The tool can be used to visualize a scatterplot of variables as shown in 11. To draw a scatterplot following steps have to be done:

1. Select data set (to display all variables)
2. 1. Select X-axis variable from variable list, 2. push axis `"X"` button.
3. (Optionally: set scaling function, and scaling range.)
4. 1. Select Y-axis variable, 2. push axis `"Y"` button.
5. (Optionally: set scaling function, and scaling range.)
6. (Optionally: set Z-axis variable, for 3D scatterplot.)
7. (Optionally: select color.)
8. Push button `"Show"`

Figure 11: Visualization of scatter plots



Instructions for rotating 3D scatterplot:

1. Set X, Y, Z variables, and press button `"Show"`.
2. Press right mouse button over the graph window and select `Toolbar`.
3. Press 3D rotate control button which is next to `Show` button in the new window (in toolbar).
4. Check `Auto` checkbox and start dragging the axes in the new 3D rotate window.

5.4 Inspect values

This tool can be used to compute proportions of special values in the dataset (including missing data value). The tool is useful for deciding treatment of special value(s).

Instructions for using the tool are:

1. Select data.
2. Select variable(s), multi selection is done by pressing CTRL key and clicking mouse left button over variables to be selected.
3. Enter values to be inspected next to `"Values:"` label. REMARK: the values must be separated by space, for example ”0 -9 -10000”.
4. Push `Update` button, and the summary table refreshes.