Exploitation of Neural Methods for Imputation

Pasi Piela

Statistics Finland
pasi.piela@stat.fi
FIN-00022 STATISTICS FINLAND, FINLAND

Abstract

In this presentation I will discuss modern imputation methods based on the neural nets methodology. The most important method used here is the Tree-Structured Self-Organising Map, or TS-SOM. The TS-SOM is a computationally fast variation of the basic Self-Organising Maps, or SOMs. It is a combination of the SOM, tree-structured clustering and computational speedup techniques. SOM is an iterative method for classification and can thus also be used for finding homogeneous clusters suitable as multivariate imputation classes.
MLP (Multi-Layer Perceptron) and SVM (Support Vector Machines) are considered briefly from the point of view of imputation. Along with many other modern methods, TS-SOM is included in a versatile software program entitled NDA, or Neural Data Analysis, which was created and will be maintained by a research group on Software Engineering and Computational Intelligence of the University of Jyväskylä, Finland. Imputation methods have been implemented into NDA in co--operation with a research group of Statistics Finland. This presentation is based on research conducted under the EUREDIT FP5 project of the European Union.

Keywords: *tree-structured self-organising maps, neural data analysis, imputation classes*.

1. Introduction

Today's growing need to handle missing values is generating increasing interest in imputation methods. Studies have recently been focusing on methods beyond the very basic, conventional ones, such as mean, random donor (hotdeck) and nearest neighbour imputation, or NN. Computationally efficient methods for finding hidden data structures can improve imputation by giving better models according to the observed data. This is not to say that the nearest neighbour imputation method would still not be the best method by far in many cases. Besides using the overall random donor method without classification or clustering as the absolute minimum that can always be improved upon, one should also always consider NN methods as recommended benchmark competitors to it. The versatile family of NN methods is behind some more advanced neural network methods, too.

Many terms of neural network methodology, such as the name itself, are needlessly confusing and mystifying, leading readers to think that they mean something ultra modern and complex. The terms originate from other sciences but from the *statistical learning* perspective any method can in the end be viewed just as a generalisation of common statistical methods – often with Gaussian assumptions. Neural networks are usually regarded as a highly non-parametric class of regression models.

Neural networks can be complex – at least in terms of their formula representations. However, many unsolved, complicated problems, often arising from computing distances between data vectors $||\mathbf{x}_i - \mathbf{x}_j||$, lie behind the easiest iterations. How to calculate distances when there are categorical variables and different numbers of classes? How to equalise continuous variables? Binarisation of categorical variables leads to very obvious and serious problems of monotonousness. This takes us straight to the official statistics perspective.

Official Statistics and Neural Networks

Many neural network methods stem from general physics, biology and engineering. The datasets of official statistics are diversified and contain several types of variables besides continuous ones. Experts often know their data well, but datasets with large numbers of variables and observations can be hard to model efficiently, that is, there can be hidden structures behind the data, which the user may want to take advantage of in the modelling.

This paper is based on a piece of research conducted under the EU's FP5 EUREDIT project (D*evelopment and evaluation of new methods for editing and imputation*). EUREDIT was concluded in March 2003, after three years of intensive study by European experts from both statistical and information processing sciences. All the project papers are to be published shortly. More information can be found at: *http://www.cs.york.ac.uk/euredit/*

The main approaches from modern neural methods that were selected for the EUREDIT project research were neural network methods, such as SVM, CMM, MLP and SOM (Support Vector Machines, Correlation Matrix Memory, Multi-Layer Perceptron networks and Self-Organising Maps). Besides on traditional imputation and editing, Statistics Finland concentrated on the SOM methods in co-operation with the University of Jyväskylä.

The project partners tested their methods on representative datasets from official statistics derived from household surveys, business surveys, censuses, panel surveys, time series and business registers. Upon conclusion of their evaluations, the partners were required to test their carefully selected methods on specific evaluation versions of the datasets without knowing the true values and to send the results/estimates to the Office of National Statistics in the UK for neutral assessment.

2. Tree-structured self-organising maps

Self-organising map [8] is one of the most popular neural network algorithms. SOM can be seen as a multivariate algorithm that models the joint distribution of data, a projection algorithm where the dimension of the latent space is typically two, or the most important perspective for imputation: SOM can be seen as a clustering algorithm [5]. The Tree-Structured Self-Organising Map [9] is a computationally fast variation of the SOM. It is a combination of the self-organising map, tree-structured clustering and computational speedup techniques. *Theoretically, the SOM algorithm can be interpreted as a discretized approximation procedure for computation of principal curves or surfaces* [13].

TS-SOM methodology was developed at the University of Jyväskylä by exploiting the Kohonen map technology. The Jyväskylä group was the first to developed the so-called Neural Data Analysis (NDA) software [7], and as the next step the group went on to implement the NDA technology in data editing and imputation. This work was done with assistance of the author of this paper. This integrated system will be abbreviated as NDAEI.

From the imputation perspective, the technology for the NDAEI is analogous to tree-methods or Automatic Interactive Detection [2a] methods. Tree-methodology (classification trees and regression trees) has rarely been used for imputation, although one interesting exception is found in the work done by the so-called EU FP4 AutImp project [3,4]. Piela & Laaksonen [11] present a study and results for the AID methods in imputation partly from this project. The algorithms behind the TS-SOM technology are much more complex and comprehensive than those in standard tree-methodology. The first version was published in [7], and the official statistics point is presented in [12].

The basic self-organising map defines mapping from the input data space $\boldsymbol{R}^n$ onto a latent space consisting typically of a two-dimensional array of nodes or neurons [8]. The original *batch algorithm* starts with a random fixed size sample of initial neuron points. The data are then divided into Voronoi clusters, that is, for every unit of the data the nearest neuron is selected from the set of initial neurons using Euclidean distances. The mean of each cluster defined by the corresponding neuron point is then calculated. The neurons are now moved *towards* the cluster means, and the Voronoi clustering is repeated and new cluster means calculated until no noticeable changes occur.

A distinct feature of the SOM is that the cluster mean is weighted by a neighbourhood function. In other words, the cluster mean is actually the weighted mean of the cluster mean itself and the mean of the clusters in its neighbourhood. The idea is to start with a large neighbourhood and reduce it during the iterations.

Naturally, the definition of neighbourhood depends on the problem in question. The neighbourhood topology in the SOM almost guarantees that clusters near to each other have something in common. This obviously helps graphic visualisation of the data analysis, as well as imputation.

The tree-structured self-organising map is made of several SOMs arranged into a tree structure (see Figure 1). The topmost layer ($L = 0$) has one neuron. Layer 1 has four neurons in a two-dimensional and two neurons in a one-dimensional case. Here we consider the two-dimensional case. Thus, each neuron has its own associated subgroup of data, four subgroups on layer 1, and 16 subgroups on layer 2. The subgroups form a cluster in which the centroid is the weight vector of the best matching unit b, $\mathbf{w}_b$.

The training is repeated layer by layer using knowledge about the neurons of the frozen layer l–1 in the search for the best minimising unit (BMU) on the next layer l. That is, the search for the BMU for layer l is restricted into a small set of neurons: sons and sons of neighbours of the BMU of the previous layer. This reduces considerably the *computational complexity* when compared to the basic SOM.

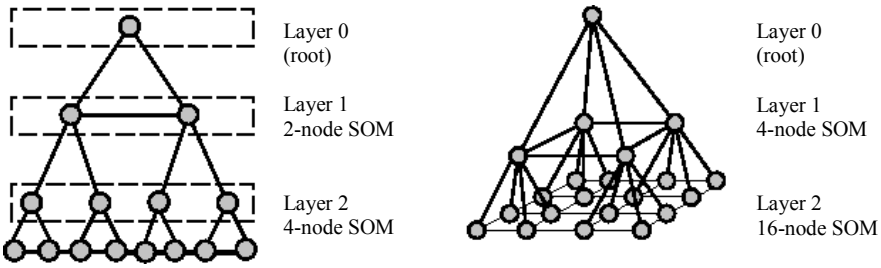


Figure 1. Illustrations of one and two-dimensional TS-SOM structures.

The training is usually made with the batch algorithm. During each epoch, the BMUs are searched for all data vectors using the tree search, and the new centroids $\mathbf{m}_i(t)$ are then the weight vectors $\mathbf{w}_i(t)$ computed using the rule:

$$\mathbf{w}_b(t+1) = \frac{1}{N_b + \sum_{i \in N_c(b)} \alpha N_i} \left(N_b m_b(t) + \sum_{i \in N_c(b)} \alpha N_i \mathbf{m}_i(t) \right),$$

where $N_c(b)$ is a set of indices of neighbours of b, and N_i is the number of data records in the Voronoi region (cluster) i. The smoothing is partially controlled through the parameter $\alpha \in [0..1]$. One side advantage here is that the size of the neighbourhood can be kept constant, and the usual problem with the basic SOM does not, in fact, exist. Moreover, these algorithms have been modified for handling missing data.

Besides graphic visualisation possibilities, the reason for using the SOM as the imputation model is its effective way of creating imputation cells with small variances from complex data. Many kinds of imputation methods can, naturally, be used within these cells or groups. The TS-SOM gives the special advantage whereby one can choose a donor from the upper levels (parent nodes) of the TS-SOM as well as from the neighbouring clusters if a good real donor is not available in the same cluster.

3. Other neural network methods

Support Vector Machines, SVM
Support Vector Machines, introduced by Vapnik [14], are tools for non-linear regression and classification. They are semi-parametric techniques offering the efficient training characteristics of parametric techniques but having the capability to learn non-parametric dependencies.

SVM is a non-linear generalisation in the following way: the covariate data are first projected onto a higher dimensional feature space and then inserted into the linear algorithm. However, the parameters learned from the feature space are obviously non-linear in the input data space.

SVM is a prediction algorithm, not a probabilistic model, and avoidance of density-estimation is, indeed, seen to underlie the success of the algorithm. The taken non-parametric regression approach is straightforward, with the predicted values generated by the SVM model used as the imputations for the missing data [10].

Multi-Layer Perceptron networks, MLP
Multi-Layer Perceptron is among the most widely applied neural network models and is generally well known [1]. It is composed by a set of elementary units (neurons) linked by weighted connections. These processing units are arranged in layers:

an input layer, one or more hidden layers and an output layer. Training is carried out by making adjustments to the weights whenever generated prediction fails.

MLP can be seen as a regression process that has to be performed step by step for each variable. The target variable is the variable to be imputed, and MLP is trained on those records for which the target value is not missing, and the networks thus generated are applied for imputing missing values [5]. That is, imputed values are obtained by simply using the network to generate predictions for records with missing values.

4. Evaluation results

This case example presents the results for an anonymised sample of the 1991 UK Census data, SARs. The data on age comprise 47,594 units in the Yorkshire and Humber area. The rate of missingness is 7.6%. The intervals of AGE are 0–90, 91, 93 and 95.

Table 1 shows imputation results for the 3,623 missing observations from nearest neighbour and random imputations within the clusters created by the TS-SOM. The TS-SOM has been trained using seven variables of the data, both categorical and continuous, including logarithmic AGE. Four of the training variables - primary economic position, relationship to household head, household space type, marital status - have been selected for the nearest neighbour hot decking by Euclidean distances. If a donor has not been available then a centroid of a cluster has been used.

By preserving the distribution well, overall random donor imputation without any auxiliary information indicates that missingness is not clearly skewed. However, the TS-SOM nearest neighbour imputation fails. Figure 2b clearly shows the problematic part; it seems to be very hard to select the NN explanatory variables that would contain enough information. For example, the TS-SOM nearest neighbour at the fourth level (256 imputation classes) gives a DL1 measure as high as 14.17 (average imputation error = 14.17 years).

However, by only selecting training variables and by allowing imputations to be done randomly, the results are very satisfactory. The DL1 is very small, and distributions are also very well preserved at the aggregate levels (see Fig 2a). Table 1 suggests here that the best tree for imputation is a large one with 256 to 1,024 terminal imputation clusters. Moreover, Figure 3 interestingly shows that the group of young people aged 16 or below, on the other hand, is separable but hardly imputable due to lack of background imputation for this generation. Economic position is “not applicable” only for this group.

In the final evaluation data of the EUREDIT project, the TS-SOM methods performed reasonably well indeed. In this particular case of imputation of AGE, the SVM methods gave the best results, but MLP and SOM gave good results as well. Overall, TS-SOM worked very well with many datasets of the project – also for the editing part.

5. Conclusion

Robust TS-SOM models for automatic editing and imputation in a wide variety of survey data applications were developed during the EUREDIT project. TS-SOM can be considered as a system that includes techniques from both statistical data modelling and from neural net modelling. In this context, it is possible to even understand TS-SOM, for example, as a clustering technique for local MLP models or for just traditional regression models. It also gives powerful visualisation tools that are particularly important in the editing aspects.

This paper focuses mainly on the TS-SOM methods, but other neural network methods that were evaluated and developed further during the project also performed reasonably well. The new techniques that emerged as promising were neural net methodology, such as MLP for imputation of missing data across a wide range of situations; fast automatic clustering algorithms, like Correlation Matrix Memory (CMM) for handling imputation in very large datasets with minimal user intervention; and Support Vector Machine (SVM) methodology for missing data imputation with categorical data [6].

Naturally, many failures also occurred and all the methods do, in fact, need a lot of further developing to make them reasonably suitable for editing and imputation, especially by the non-expert user.

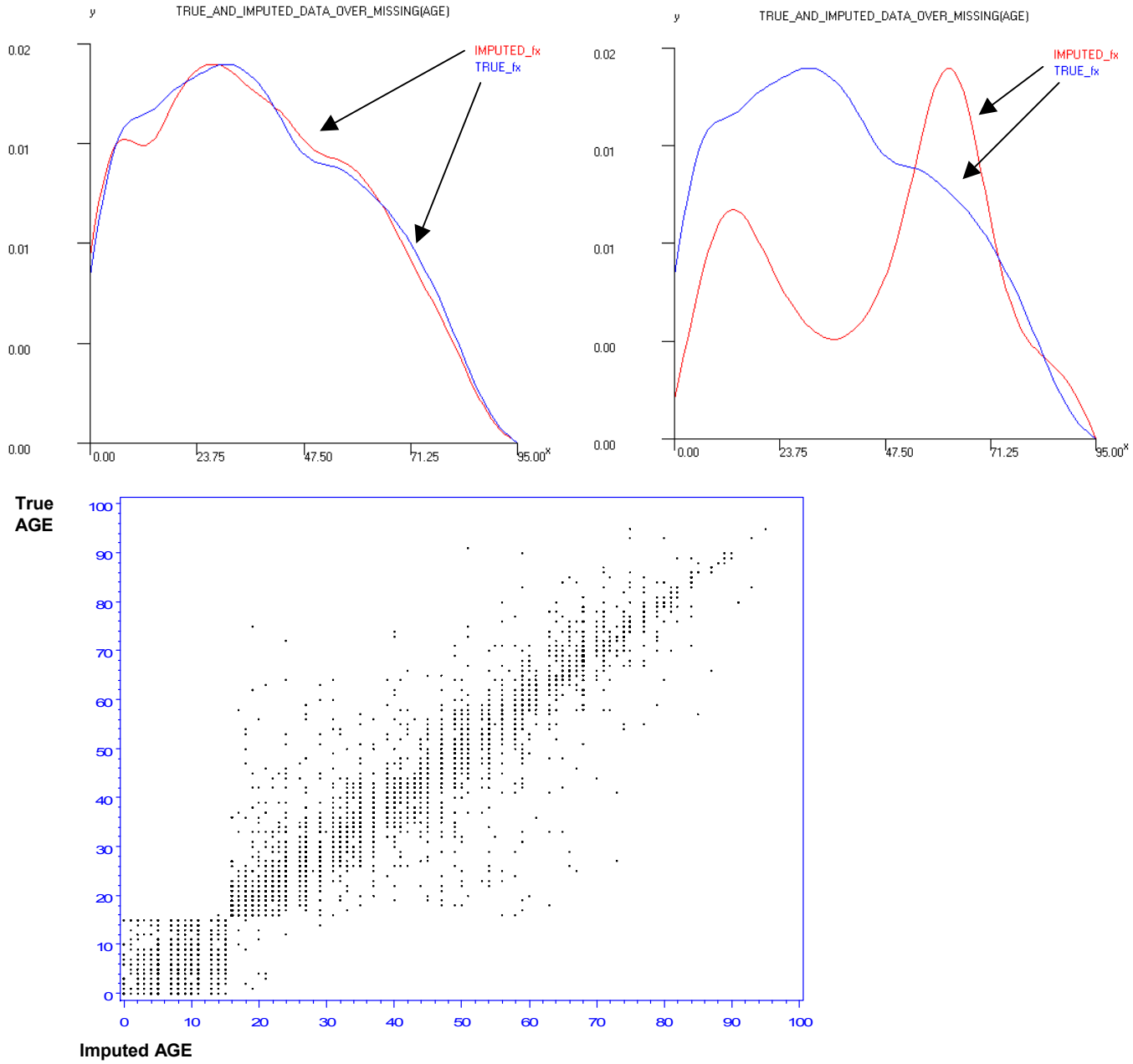
References

[1] Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, United Kingdom.
[2] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984). *Classification and Regression Trees.* Wadsworth, Belmont, C.A.
[3] Chambers, R.L., Hoogland, J., Laaksonen, S., Mesa, D.M., Pannekoek, J., Piela, P., Tsai, P. and de Waal, T. (2001a). The AUTIMP-project: Evaluation of Imputation Software. Research Paper 0122. Statistics Netherlands.
[4] Chambers, R.L., Crespo, T., Laaksonen, S., Piela, P., Tsai, P. and de Waal, T. (2001b). The AUTIMP-project: Evaluation of WAID. Research Paper 0121. Statistics Netherlands.
[5] Chambers, R. (2003). Overview of the methods investigated by the EUREDIT project. *http://www.cs.york.ac.uk/euredit/*
[6] Charlton, J. (2003). Recommendations towards Edit And Imputation Strategies.
[7] Häkkinen, E. (2001). *Design, Implementation and Evaluation of the Neural Data Analysis Environment*. PhD thesis. Jyväskylä University Library, Jyväskylä, Finland.
[8] Kohonen, T. (1997). *Self-Organizing Maps*. Springer, Berlin, Heidelberg.
[9] Koikkalainen, P. and Oja, E. (1990). Self-Organizing Hierarchical Feature Maps. In *Proc. IJCNN-90-Wash-DC, Int. Joint Conf. on Neural Networks*, Volume II, pp. 279-285, Piscataway, NJ., IEEE Service Center.
[10] Mallinson, H. & Gammerman, A. (2003). Imputation Using Support Vector Machines. *http://www.cs.york.ac.uk/euredit/*
[11] Piela, P. & Laaksonen S. (2001). Automatic Interaction Detection for Imputation – Tests with the WAID Software Package. Contributed Paper for the *Federal Committee on Statistical Methodology Research Conference*, Washington, DC Area.
[12] Piela, P. (2002). Introduction to Self-Organizing Maps Modelling for Imputation – Techniques and Technology. Research in Official Statistics, 2, 5-19.
[13] Ritter, H., Martinez, T., and Schulten, K. (1992). *Neural Computation and Self-Organizing Maps: An Introduction*. Addison-Wesley, Reading, MA.
[14] Vapnik, V.N. (1995). *The Nature of Statistical Learning Theory.* Springer, New York.

Table 1. Test results for the imputation variable AGE in the UK SARs data. L = TSSOM clustering, at level L. $L = l$ means that the data have been divided into 4^l clusters/subclasses for imputation. DL1 is the average difference between true $\hat{Y}$ and corresponding imputed values Y^*:

$$d_{L1}(\hat{\mathbf{Y}}, \mathbf{Y}^*) = \sum_{i=1}^{n} w_i \left| \hat{Y}_i - Y_i^* \right| \Big/ \sum_{i=1}^{n} w_i \,, \quad \text{where: } w_i = 1 \; \forall \; i \in \mathbf{N}.$$

Method	Mean	Std. Dev.	25% Quantile	Median	75% Quantile	95% Quantile	DL1
True values (N = 3623)	37.27	23.06	19	35	55	76	0
Overall random donor	37.37	22.93	19	35	55	76	26.62
TS-SOM L=4, nearest neighbour	45.88	26.51	21	49	66	90	14.17
TS-SOM L=5, nearest neighbour	45.96	26.30	22	49	65	90	13.27
TS-SOM L=1, random donor	37.49	21.87	21	35	54	75	14.51
TS-SOM L=2, random donor	38.09	22.25	21	37	56	75	6.10
TS-SOM L=3, random donor	37.34	22.57	20	35	56	75	4.83
TS-SOM L=4, random donor	36.83	22.79	19	35	54	75	4.59
TS-SOM L=5, random donor	**37.27**	22.39	19	35	54	75	**4.52**
TS-SOM L=6, random donor	36.71	23.11	19	35	54	75	5.32



Figures 2a, 2b and 3. Success of the TS-SOM L=5 imputation for AGE in the SARs data, at data and unit levels. Clockwise from left: **2a)** Estimated distributions of randomly imputed observations and corresponding true observations using Partzen windows and Gaussian weighting; **2b)** Estimated distribution for NN imputed observations; **3)** Scatterplot for randomly imputed AGE against corresponding true values.