Combination of SOMs for Fast Missing Value Imputation

Antti Sorjamaa[1], Amaury Lendasse[1] and Eric Severin[2]

1- Helsinki University of Technology - ICS
P.O. Box 5400, 02015 HUT - Finland

2- Department GEA
University of Lille 1 - France

Abstract. This paper presents a methodology for missing value imputation. The methodology is based on a combination of Self-Organizing Maps (SOM), where combination is achieved by Nonnegative Least Squares algorithm. Instead of a need for validation as when using traditional SOMs, the combination proceeds straight into final model building. Therefore, the methodology has very low computational time. The combination of SOMs also increases accuracy at the same time. The performance is demonstrated using a database from corporate finance field.

1 Introduction

The presence of missing values in the underlying time series is a recurrent problem when dealing with databases. Number of methods have been developed to solve the problem and fill the missing values.

In this paper, we focus on Self-Organizing Maps [1] (SOM), which aim to ideally group homogeneous individuals, highlighting a neighborhood structure between classes in a chosen lattice. The SOM algorithm is based on unsupervised learning principle where the training is entirely stochastic, data-driven. No information about the input data is required. Recent approaches propose to take advantage of the homogeneity of the underlying classes for data completion purposes [2]. Furthermore, the SOM algorithm allows projection of high-dimensional data to a low-dimensional grid. Through this projection and focusing on its property of topology preservation, SOM allows nonlinear interpolation for missing values.

But how to find optimal SOM size and shape? One of the typical machine learning paradigms is about finding the model that best fits the given data, in terms of test or validation. Searching for such a model can be very time consuming: finding the model class that best suits the type of data, optimizing the possible hyper-parameters, and finally training the model once all details of the model structure have been selected. This procedure can lead to a rather good model, which fits the data and avoids the pitfalls of overfitting.

On the other hand, creating a combination of less good models might achieve better performance, while alleviating the problem of extensive validation procedure. Even faster model building is achieved through parallel computation, which is easy to implement when several different models are built.

The goal is then to weight each model so that the overall output of a linear combination of models has the best possible performance. Several ensemble techniques have been proposed, out of which two kinds can be distinguished [3]: the variable weights approach and the "average" ones. Traditionally, average weights ensemble techniques are used and simply take an average of all the built models. While this obviously has the advantage of having immediately the weights of all models, it yields suboptimal results. The variable weights ensemble techniques try to optimize the weight of each model in the ensemble according to a criterion. Techniques such as the Genetic Algorithm have been recently used for such optimization [4] but are very time consuming.

This paper describes a new method, which combines several SOMs in order to enhance the accuracy of the nonlinear interpolation. The combination is achieved with a classical constrained linear solution the Nonnegative Least Squares and it improves the accuracy of the imputation as well as speeds up the process by removing the need for validation.

The global methodology is presented in the next section, including all the methods combined in the global methodology. The Section 3 demonstrates the accuracy of the methodology by using an example from corporate finance.

2 Global Methodology

The global methodology is summarized in Figure 1.

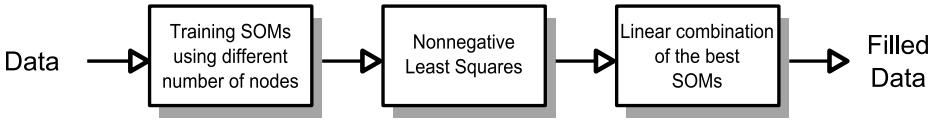

Fig. 1: Global methodology summarized.

The core of the methodology is the Self-Organizing Map (SOM). Several SOMs are trained using different number of nodes and the imputation results of the SOMs are linearly combined. The linear coefficients are computed using Nonnegative Least Squares (NNLS) algorithm. The SOM imputation methodology and the combination are explained more deeply in the following.

2.1 Imputation using SOM

The SOM algorithm is based on an unsupervised learning principle, where training is entirely data-driven and no information about the input data is required [1]. Here we use a 2-dimensional network, composed of c units (or code vectors) shaped as a square *lattice*. Each unit of a network has as many weights as the length T of the learning data samples, $\mathbf{x}_n$, $n = 1, 2, ..., N$. All units of a network can be collected to a weight matrix $\mathbf{m}(t) = [\mathbf{m}_1(t), \mathbf{m}_2(t), ..., \mathbf{m}_c(t)]$ where $\mathbf{m}_i(t)$ is the T-dimensional weight vector of the unit i at time t and t represents the steps of the learning process. Each unit is connected to its neighboring units

through a neighborhood function $\lambda(\mathbf{m}_i, \mathbf{m}_j, t)$, which defines the shape and the size of the neighborhood at time t. The neighborhood can be constant through the entire learning process or it can change in the course of learning.

The learning starts by initializing the network node weights randomly. Then, for a randomly selected sample $\mathbf{x}_{t+1}$, we calculate the Best Matching Unit (BMU), which is the neuron whose weights are closest to the sample. The BMU calculation is defined as

$$\mathbf{m}_{BMU(\mathbf{x}_{t+1})} = \arg\min_{\mathbf{m}_i, i \in I} \left\{ \left\| \mathbf{x_{t+1}} - \mathbf{m}_i\left(t\right) \right\| \right\}, \tag{1}$$

where $I = [1, 2, ..., c]$ is the set of network node indices, the BMU denotes the index of the best matching node and $\|.\|$ is a standard Euclidean norm.

If the randomly selected sample includes missing values, the BMU cannot be solved outright. Instead, an adapted SOM algorithm, proposed by Cottrell and Letrémy [5], is used. The randomly drawn sample $\mathbf{x_{t+1}}$ having missing value(s) is split into two subsets $\mathbf{x}_{t+1}^T = NM_{\mathbf{x_{t+1}}} \cup M_{\mathbf{x_{t+1}}}$, where $NM_{\mathbf{x_{t+1}}}$ is the subset where the values of $\mathbf{x_{t+1}}$ are not missing and $M_{\mathbf{x_{t+1}}}$ is the subset, where the values of $\mathbf{x_{t+1}}$ are missing. We define a norm on the subset $NM_{\mathbf{x_{t+1}}}$ as

$$\left\| \mathbf{x_{t+1}} - \mathbf{m}_i\left(t\right) \right\|_{NM_{\mathbf{x_{t+1}}}} = \sum_{k \in NM_{\mathbf{x_{t+1}}}} \left(\mathbf{x}_{t+1,k} - \mathbf{m}_{i,k}(t) \right)^2, \tag{2}$$

where $\mathbf{x}_{t+1,k}$ for $k = [1, ..., T]$ denotes the k^{th} value of the chosen vector and $\mathbf{m}_{i,k}(t)$ for $k = [1, ..., T]$ and for $i = [1, ..., c]$ is the k^{th} value of the i^{th} code vector.

Then the BMU is calculated with

$$\mathbf{m}_{BMU(\mathbf{x}_{t+1})} = \arg\min_{\mathbf{m}_i, i \in I} \left\{ \left\| \mathbf{x_{t+1}} - \mathbf{m}_i\left(t\right) \right\|_{NM_{\mathbf{x_{t+1}}}} \right\}. \tag{3}$$

When the BMU is found the network weights are updated as

$$\begin{aligned} &\mathbf{m}_i\left(t+1\right) = \ldots \\ &\quad \mathbf{m}_i\left(t\right) - \varepsilon(t) \lambda\left(\mathbf{m}_{BMU(\mathbf{x}_{t+1})}, \mathbf{m}_i, t\right) \left[\mathbf{m}_i\left(t\right) - \mathbf{x_{t+1}}\right], \\ &\qquad \forall i \in I, \end{aligned} \tag{4}$$

where $\varepsilon(t)$ is the adaptation gain parameter, which is $]0, 1[$-valued, decreasing gradually with time. The number of neurons taken into account during the weight update depends on the neighborhood function $\lambda(\mathbf{m}_i, \mathbf{m}_j, t)$. The number of neurons, which need the weight update, usually decreases with time.

After the weight update the next sample is randomly drawn from the data matrix and the procedure is started again by finding the BMU of the sample. The learning procedure is stopped when the SOM algorithm has converged.

Once the SOM algorithm has converged, we obtain some clusters containing our data. Cottrell and Letrémy proposed to fill the missing values of the dataset

by the coordinates of the code vectors of each BMU as natural first candidates for the missing value completion:

$$\pi_{(M_{\mathbf{x}})}(\mathbf{x}) = \pi_{(M_{\mathbf{x}})}\left(\mathbf{m}_{BMU(\mathbf{x})}\right), \tag{5}$$

where $\pi_{(M_{\mathbf{x}})}(.)$ replaces the missing values $M_{\mathbf{x}}$ of sample $\mathbf{x}$ with the corresponding values of the BMU of the sample. The replacement is done for every data sample and then the SOM has finished filling the missing values in the data.

The procedure is summarized in Table 1. There is a toolbox available for performing the SOM algorithm in [6].

Table 1: Summary of the SOM algorithm for finding the missing values.

1. SOM node weights are initialized randomly
2. SOM learning process begins
   (a) Input $\mathbf{x}$ is drawn from the learning data set $\mathbf{X}$
      i. If $\mathbf{x}$ does not contain missing values, BMU is found according to Equation 1
      ii. If $\mathbf{x}$ contains missing values, BMU is found according to Equation 3
   (b) Neuron weights are updated according to Equation 5
3. Once the learning process is done, for each observation containing missing values, the weights of the BMU of the observation are substituted for the missing values

2.2 Combination of Multiple SOMs

The aim is to find the optimal weights α_i for the SOM maps M_i. Each SOM map has different number of nodes and, thus, gives different imputation results for the missing values in the database. For each missing value, every SOM in the combination is giving an estimation and the final estimation of the missing value is the linear combination of the individual SOM estimates. The Procedure is summarized in Figure 2.

In order not to exaggerate the errors of each model in the combination, the α_i is set to be nonnegative. Assuming that each SOM is unbiased, the combination can be made unbiased by having $\sum \alpha_i = 1$.

For the determination of the weights α_i, a classical constrained optimization method called Non-Negative constrained Least-Squares (NNLS) algorithm [7] is used to compute the solution. For the computation, a small set of the data is removed and used as a *calibration set*. The size of the calibration set has to be selected according to the number of missing values in the database.

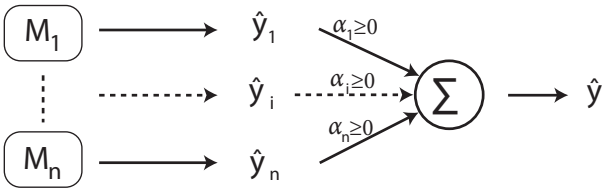

Fig. 2: Illustrative scheme of the combination of SOMs.

Then, the coefficients α_i are solved from the constrained linear system, shown in Equation (6), using the NNLS algorithm.

$$\arg\min_{\boldsymbol{\alpha}} \left\| \boldsymbol{y}^{\mathrm{Cal}} - \sum_{i=1}^{n} \alpha_i \hat{\boldsymbol{y}}_i^{\mathrm{Cal}} \right\|^2 \quad \text{s.t.} \quad \alpha_i \geq 0\,. \tag{6}$$

After solving the system on linear equations above, the combination of SOMs can be used to fill the missing values on the whole database.

The way of combining the SOMs described above, removes the need for lengthy and time consuming Cross-Validation procedure needed in traditional SOM imputation. Each SOM size needs to be trained only once and the most accurate ones are combined using the NNLS and the small calibration set. Furthermore, the linear combination by the NNLS is known to converge in $\frac{1}{2}n$ steps [7] and the result is notably more accurate than any individual SOM map used in the combination.

3 Experiments

In the following section, the dataset used in the experiments is introduced. Then the SOM methodology is compared with the combined one. Finally, in the end, conclusions are stated.

3.1 Financial Dataset

The financial dataset used in the experiments represents a corporate finance field and it collects information about companies and their performance. The information is completely numerical and it inherently includes 14 percent of missing values.

The source of the data is Thomson One Banker and it includes almost 6000 French and British companies. Each company is represented by 45 yearly key numbers from years 1999 to 2006, including three binary variables for the operative field. All companies are either registered in Paris Stock Exchange or London Stock Exchange and most of the companies are medium sized (51-500 employees) or large (more than 500 employees).

In the 45 key numbers, some characteristics such as assets, current assets, total debt or total equity are taken into account. The objective in this dataset is to build indicators able to explain the variable long term debt (Y) (i.e. long

term debt/total debt). 7 variables are built. Each variable is an indicator to explain Y. The main indicators are Market value of shares/Book value of shares, variation of sales, Altman's score, size, corporate performance (EBITDA/Total assets), industrial sector and the characteristics of the legal system (creditor oriented or common law system for UK firms and debtor oriented or civil law système for French firms). The following Table 2 shows a small piece of the data.

Table 2: Corporate finance dataset. Companies are on the rows and key numbers in columns. The three first columns are in binary format and represent the sector where the company operates. Empty cells are representing missing values.

	Sector							
Company 1	0	0	1	176	201	266	-1395	
2	0	0	1	65451	174			0,580
3	0	0	1	65579	131			1,571
4	1	0	0	53880	128	55		1,396
5	1	0	0	59575	124	46		1,554
6	0	1	0	1195	17		1	
7	0	1	0	8951	41		0,137	

3.2 Results

Before the filling process is started, we need to remove the test set and the calibration set from the data. Test set is removed in order to estimate the accuracy of the methodology and the calibration set is needed for the estimation of the linear combination of the SOM maps. The sets are selected randomly, but with restrictions that no column or row should be left completely empty and that the calibration and test sets do not overlap.

In the experiments, when combining the SOM maps, a total of 50 SOMs were trained from which the combination was created. Each SOM has different amount of nodes aligned into a two-dimensional lattice using hexagonal neighborhood. The size of the SOM was defined as described in [1] and in [6] in *SOM algorithm implementation in SOM Toolbox*. According to the heuristic, the SOM map sizes ranged from 2×3 to 51×116. All SOMs were trained with default settings.

In order to study the influence of the amount of calibration data removed, the filling procedure was repeated 5000 times with different amount of calibration points ranging from 50 to 2000. The Figure 3 shows the average errors in calibration and in test when using the combination of SOMs. In each repetition, the test was selected also randomly and it included 2000 points. The error measure used in the experiments is Normalized Mean Square Error (NMSE).

Naturally, the calibration error goes lower all the time when the calibration set is increased, but the test error levels out after 1200 points. This suggests that the amount of calibration points should be at least 1200 points or higher.

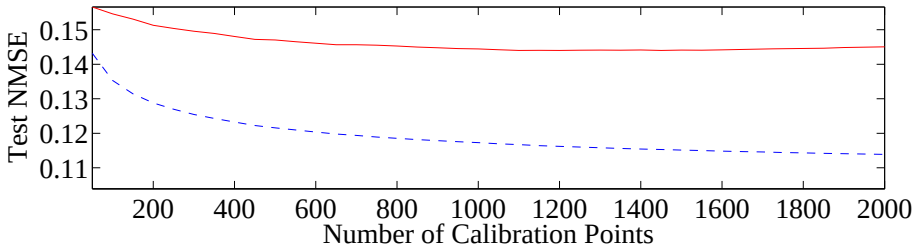

Fig. 3: Test NMSE with respect to different amounts of calibration points in the combination of SOMs. The dashed blue line denotes the calibration error and the solid red line the test error.

Comparing the test performance of individual SOMs and the combination of them, Figure 4 summarizes the results.

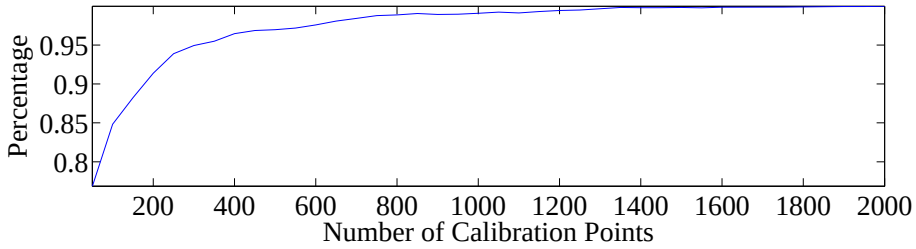

Fig. 4: Percentage of times the combination of SOMs outperforms the best individual SOM with respect to the amount of calibration points.

From Figure 4, it can be seen that already with 400 calibration points the combination outperforms the best individual SOM in more than 95 percent of the cases. The 99 percent limit is reached with 1000 calibration points and the 2000 points gives us 99.98 percent.

Figure 5 shows the normalized validation and test errors for the individual SOMs and the test errors for the combination of SOMs.

Figure 5 clearly shows that the combination outperforms the individual SOMs in terms of the test NMSE. Whereas the best individual SOM achieves a test error of 0.1949, the combination gives us more than 14 percent lower test error, 0.1668.

4 Conclusion

This paper demonstrates many benefits of using a combination of SOM maps instead of traditional SOM methodology. The combination achieves better performance than any individual SOM, based on the obtained test errors.

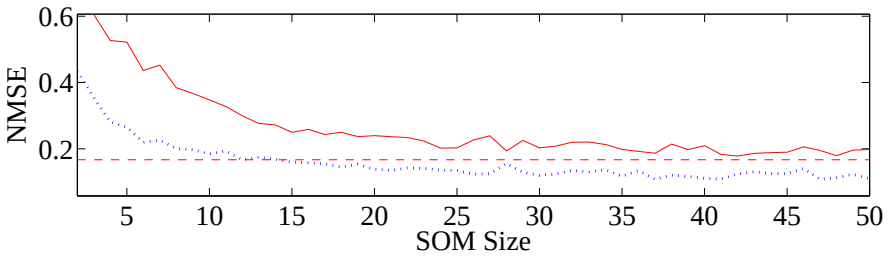

Fig. 5: Validation and test NMSE with respect to different SOM sizes. The validation NMSE is presented by the blue dotted line and the test errors by red solid line. The red dashed horizontal line denotes the test error of the obtained combination of SOMs.

Wheres in traditional SOM algorithm, one has to make carefully sure that the trained SOM has converged correctly and is viable to be used for the filling of the missing values, the combination methodology select the valid SOMs automatically upon the filling procedure. This makes the combination robust and reliable filling methodology.

Finally, the computational time is lower, when using the combination of SOMs than traditional SOM method. The SOMs need to be trained only once, instead of a lengthy cross-validation procedure for the size and other parameters.

References

[1] Teuvo Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 1995.

[2] Shouhong Wang. Application of self-organising maps for data mining with incomplete data sets. *Neural Computing and Applications*, 12(1):42–48, 2003.

[3] T. G. Dietterich. *Handbook of brain theory and neural networks*. Cambridge MA: MIT Press, 2nd edition, 2002. Chapter: Articles: Ensemble Learning.

[4] Z. Zhou, J. Wu, and W. Tang. Ensembling neural networks: many could be better than all. *Artif. Intell.*, 137(1-2):239–263, 2002.

[5] Marie Cottrell and Patrick Letrémy. Missing values: Processing with the kohonen algorithm. pages 489–496. Applied Stochastic Models and Data Analysis, Brest, France, 17-20 May, 2005.

[6] SOM Toolbox: http://www.cis.hut.fi/projects/somtoolbox/.

[7] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. SIAM Classics in Applied Mathematics, 3rd edition, 1995.