ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

X-SOM and L-SOM: a Nested Approach for Missing Value Imputation

Paul Merlin[1], Antti Sorjamaa[2], Bertrand Maillet[1] and Amaury Lendasse[2]

1- A.A.Advisors-QCG - Variances and University of Paris-1 (CES/CNRS and EIF)
106 bv de l'hôpital F-75647 Paris cedex 13 - France

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

Abstract. In this paper, a new method for the determination of missing values in temporal databases is presented. This one is based on a robust version of a nonlinear classification algorithm called Self-Organizing Maps and it consists of a combination of two classifications in order to take advantage of spatial as well as temporal dependencies of the dataset. This nested approach leads to a significant improvement of the estimation of the missing values. An application of the determination of missing values for hedge fund return database is presented.

1 Introduction

The presence of missing values in the underlying time series is a recurrent problem when dealing with databases. Because of the absolute need of complete time series for most of the models, a number of methods to handle missing data have been proposed.

Self-Organizing Maps [1] (SOM) aim ideally to group homogeneous individuals through a low-dimensional projection and to highlight the neighborhood structure between the classes. The SOM networks have the ability to be robust, even when some values are missing [2]. SOM-based methods for recovering the missing values have already been proposed, for instance in [3] and [4]. They usually make an intensive use of the spatial correlation and fill the missing values of a time series by the corresponding values of the network neurons after training.

However, one can mention two main drawbacks. First, the dynamics of the time series are not taken fully into account, and secondly, the rebuilding process is discrete. As in [5], we propose, a combination of a transversal (X-SOM) and a longitudinal (L-SOM) classifications allowing us to overcome the above limits and to incorporate spatial, as well as temporal dependencies.

The structure of this paper is as follows. In Section 2, the SOM algorithm and its robust version are presented. The following section is dedicated to present the new algorithm for conditional missing values recovery. In the last section, a financial time series return dataset is used to illustrate the accuracy of the method.

2 Self-Organizing Maps

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, compound in 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, $\mathbf{x}_n$, for $n = [1, 2, ..., N]$. All units of a network can be collected to a weight matrix $\mathbf{m}\left(t\right) = \left[\mathbf{m}_1\left(t\right), \mathbf{m}_2\left(t\right), ..., \mathbf{m}_c\left(t\right)\right]$ where $\mathbf{m}_i\left(t\right)$ 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 neighborhood function $\lambda(\mathbf{m}_i, \mathbf{m}_j, t)$, which defines the shape and the size of the neighborhood at time t.

First the network nodes are initialized randomly from the data sample space. Then, the iterative learning process begins. For a randomly selected sample $\mathbf{x}_{t+1}$, the Best Matching Unit (BMU), which is the neuron whose weights are closest to the sample is calculated as $\mathbf{m}_{BMU(\mathbf{x}_{t+1})} = \arg\min_{\mathbf{m}_i, i\in I}\left\{\|\mathbf{x_{t+1}} - \mathbf{m}_i\left(t\right)\|\right\}$, where $I = [1, 2, ..., c]$ is the set of network node indices, BMU denotes the index of the best matching node and $\|.\|$ is standard Euclidean norm.

If the randomly selected sample includes missing values, the BMU cannot be solved outright. Instead, an adapted SOM algorithm [2] 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

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

where $\mathbf{x}_{t+1,k}$ denotes the k^{th} value of the chosen data vector and $\mathbf{m}_{i,k}(t)$ is the k^{th} value of the i^{th} code vector. k goes through all the indexes in the subset $NM_{\mathbf{x_{t+1}}}$, where values are not missing.

Then the BMU is calculated with

$$\mathbf{m}_{BMU(\mathbf{x}_{t+1})} = \arg\min_{\mathbf{m}_i, i\in I}\left\{\|\mathbf{x_{t+1}} - \mathbf{m}_i\left(t\right)\|_{NM_{\mathbf{x_{t+1}}}}\right\}. \tag{2}$$

When the BMU is found the network weights are updated as

$$\mathbf{m}_i\left(t+1\right) = \mathbf{m}_i\left(t\right) - \varepsilon(t)\lambda\left(\mathbf{m}_{BMU(\mathbf{x}_{t+1})}, \mathbf{m}_i, t\right)\left[\mathbf{m}_i\left(t\right) - \mathbf{x_{t+1}}\right], \forall i \in I, \tag{3}$$

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 $\lambda(., ., .)$.

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 recursive learning procedure is stopped when the SOM algorithm has converged.

Since our method is able to handle missing values by making an intensive use of the SOM algorithm, issues regarding the SOM convergence have a significant impact on the missing value reconstruction quality. One way to ensure the convergence is to use the Robust SOM (RSOM) [6]. The idea is to use a bootstrap process to ensure the convergence.

First, an empirical probability for any pair of individuals to be neighbors in the SOM map is estimated with a resampling technique: 40% of observations and individuals are removed, the SOM learning process is performed and finally the removed individuals are projected onto the map, allowing us to get the whole neighborhood structure. The above technique is repeated several times and the empirical estimate of the probability is calculated.

Then the SOM algorithm is, once again, executed several times, but without resampling. From these maps, we select the one whose neighborhood structure is the closest to the empirical probability obtained at the previous step.

The benefits of such a procedure are double. First, the bootstrap process applied during the step one allows the minimization of the effect of possible outliers present in the database. Second, the map chosen in the second step is the one, which maximizes the likelihood of the neighborhood structure.

3 Nested SOM-based Estimation Methodology

SOM-based estimation methods have ever been proposed (for instance, [7] or [3]). These methods typically classified time series and then, using peer-group specificities like mean of individuals or the code vector itself, estimated candidates for the missing values. However, one can mention two main drawbacks. First, the dynamics of the time series are not taken into account, and second, the rebuilding process is discrete, missing values of the time series are filled by the corresponding values of the neurons to which the time series is closes to. Thus, for all series belonging to the same cluster, the estimations are the same.

Following [5], we propose a double classification to overcome the limits. As previously seen in [7] and [3], the first network, identified by its code vector weights $\mathbf{m}^1$ (each unit corresponding to a T-dimensional weight vector), groups individuals, through a longitudinal classification (denoted L-SOM). Then, for each time series $\mathbf{x}_i$ containing missing values, the weights of the associated BMU are substituted for any missing values

$$\mathbf{x}_{i,k} = \mathbf{m}_{BMU(\mathbf{x_i}),k}, \tag{4}$$

for $k \in M_{\mathbf{x}}$.

Simultaneously, we run another SOM classification $\mathbf{m}^2$, on the transversal dataset $\mathbf{x}'$ (each unit corresponds to an n-dimensional weight vector, where n is the number of time series in $\mathbf{x}$). The second cross classification (denoted X-SOM) no more clusterizes observations but realizations. Estimation of missing values operates exactly as in Equation 4.

We have now, two nonlinear estimations for each missing value $\mathbf{x}_{i,k}$ of the dataset. The first one is accurate when considering spatial dependencies, whereas the second integrates temporal correlations more efficiently. We propose to linearly combine these two candidates according to their distances to their respective BMUs. Let d_1 be the inverse of the distance from the sample $\mathbf{x}_i$ to its associated BMU in $\mathbf{m}^1$, $d_1 = \left(\left\| \mathbf{x_i} - \mathbf{m}^1_{BMU(\mathbf{x_i})} \right\|_{NM_{\mathbf{x_i}}} \right)^{-1}$. We define d_2 equivalently as $d_2 = \left(\left\| \mathbf{x'_k} - \mathbf{m}^2_{BMU(\mathbf{x'_k})} \right\|_{NM_{\mathbf{x'_k}}} \right)^{-1}$.

Then, for each missing value of $\mathbf{x}_{i,k}$, we estimate the missing values contained in the sample through the Nested SOM by

$$\mathbf{x}_{i,k} = d_1/\left(d_1 + d_2\right) \mathbf{m^1}_{BMU(\mathbf{x_i}),k} + d_2/\left(d_1 + d_2\right) \mathbf{m^2}_{BMU(\mathbf{x'_k}),i}. \tag{5}$$

For the Nested SOM, we still have to select the optimal grid sizes c^1 and c^2. This is done by using validation and the same validation sets for all combinations of the parameters c^1 and c^2. The Nested SOM that gives the smallest validation error is used to perform the final completion of the data.

4 Experimental Results

In the following application, we illustrate our imputation method on a dataset of hedge fund returns[1] composed of 120 funds containing 120 monthly returns from a 10-year period.

Since the hedge fund strategies are well diversified, such assets guarantee us that the time series are not (too much) interdependent. The observed correlations between the assets remain reasonable; the mean, minimum and maximum correlations are respectively .10, −.62 and .77. Moreover, since we do not want to favor one of the two classifications (spatial or temporal), we only keep the first 120 funds so that the number of observations remains equal to realizations. Regarding the correlations of the transposed dataset, we find that the mean, minimum and maximum cross correlations are .00, −.75 and .74, respectively.

Figure 1 shows 15 among the 120 rescaled fund values[2]. The fund values are low-correlated time series and there are no missing values originally contained in the database.

Before running any experiments, we randomly removed 7.5 percent of the data to a test set. The test set contains 1 080 values. For the validation, the same amount of data is removed from the dataset. Therefore, for the model selection and learning we have a database with a total of 15 percent missing values.

The Monte Carlo Cross-Validation with 20 folds is used to select the optimal parameters for the L-SOM, the X-SOM and the Nested SOM method. The 20 selected validation sets are the same for each method. The validation errors are shown in Figure 2. In the case of the Nested SOM, the errors shown are the minimum errors after the X-SOM with different L-SOM sizes.

The optimal size of the L-SOM grid is found to be 10×10, which is a total of 100 units. We have roughly as many code vectors in the map as observations

[1] provided by *HFR*.

[2] $v'_t = 100 \prod_{i=1}^{t} (1 + r_i)$, with r_i the return of a fund at the time i.

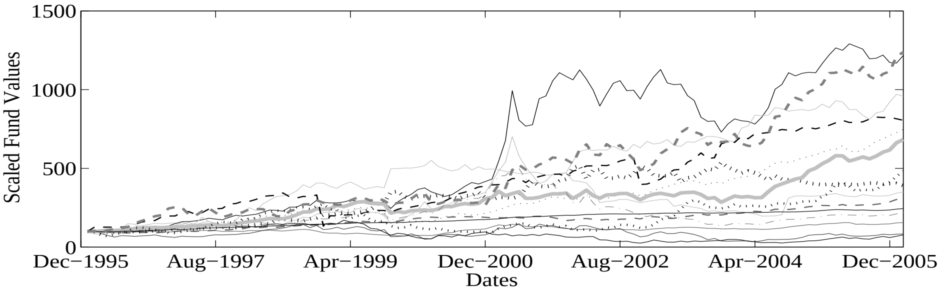

Fig. 1: Rescaled asset values of 15 funds present in the database.

(120). Regarding the cross classification, the X-SOM, we find an optimal size of the grid to be 6×6. It means that we have a nonlinear interpolation between observations and a better approximation of the missing values with more units than data.

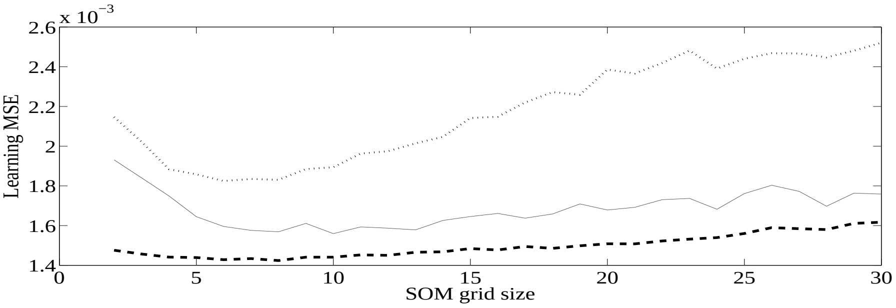

Fig. 2: Validation errors against SOM grid sizes. The L-SOM validation error is the solid line, X-SOM is the dotted line and The Nested SOM is the dashed line.

The smallest error achieved with the Nested SOM method is with the L-SOM grid size 17×17 and with the X-SOM grid size 8×8. The number of neurons is larger for both methods when combined than the L-SOM or X-SOM classification alone. It suggests that the local approximations reduce errors from both L-SOM and X-SOM estimations and enable finer interpolations. From the Figure 2 it is clearly notable that with every SOM size the Nested SOM method gives lower validation error than either L- or X-SOM alone.

Table 1 contains the validation and test errors of all three methods. We can see that the Nested SOM outperforms the L-SOM and the X-SOM reducing the validation error by 19 and 28 percent, respectively, and the test error by 23 and 31 percent.

Table 1: Learning and Test Errors for the L-SOM, the X-SOM and the Nested SOM.

10^{-4}	Learning Error	Test Error
L-SOM	1.6	1.7
X-SOM	1.8	1.9
Nested SOM	1.3	1.3

5 Conclusion

In this paper, we have proposed a new Nested SOM-based method for finding missing values. The L-SOM classification provides efficient missing value estimations that respect spatial dependency structures, whereas the estimations obtained through the X-SOM integrate efficiently the temporal correlations. The combination of these two approaches allows us to overcome the main drawback of the SOM-based imputation methods: the fact that the missing value estimations are discrete. Indeed, considering the distance between series and their associated Best Matching Units make it possible to obtain local continuous approximations of the missing values. As we have shown in the experiments, the combined approach provided estimations that are more accurate than those obtained with each of the methods applied individually.

For further work, the rule driving the local interpolation step may be (easily) upgraded. We also plan to provide a heuristic to find the optimal sizes of the SOMs. The methodology will then be tested on various datasets, and compared to various methodologies to handle missing data. One can also easily think to adopt a conditional approach applying the X-SOM on each L-SOM peer-group to take advantage of the time varying characteristics of the clusters.

References

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

[2] Tariq Samad and Steven Harp. Self-organization with partial data. *Network*, 3(2):205–212, 1992.

[3] 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.

[4] Antti Sorjamaa, Bertrand Maillet, Paul Merlin, and Amaury Lendasse. Som+eof for finding missing values. pages 115–120. European Symposium on Artificial Neural Networks, Bruges, Belgium, 25-27 April, 2007.

[5] Geoffroy Simon, Amaury Lendasse, Marie Cottrell, Jean-Claude Fort, and Michel Verleysen. Double som for long-term time series prediction. Workshop on Self-Organizing Maps, Kitakyushu, Japan, 11-14 September.

[6] Christiane Guinot, Bertrand Maillet, and Patrick Rousset. Understanding and reducing variability of som neighbourhood structure. *Neural Networks*, 19(6):838–846, 2006.

[7] Françoise Fessant and Sophie Midenet. Self-organising map for data imputation and correction in surveys. *Neural Computing & Applications*, 10(4):300–310, 2002.