X-SOM and L-SOM: A Double Classification Approach for Missing Value Imputation

Paul Merlin[a], Antti Sorjamaa[b], Bertrand Maillet[c], Amaury Lendasse[d]

[a]*A.A.Advisors - Variances and University of Paris-1 (CES/CNRS),*
106 bv de l'hôpital F-75647 Paris cedex 13 - France,
paul.merlin@univ-paris1.fr.
[b]*Helsinki University of Technology - ICS*
P.O. Box 5400, 02015 HUT - Finland,
antti.sorjamaa@hut.fi.
[c]*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,
bmaillet@univ-paris1.fr.
[d]*Helsinki University of Technology - ICS*
P.O. Box 5400, 02015 HUT - Finland,
lendasse@hut.fi.

Abstract

In this paper, a new method for the determination of missing values in temporal databases is presented. It 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 double classification leads to a significant improvement of the estimation of the missing values. An application of the missing value imputation for hedge fund returns is presented.

Key words: Missing Value Completion, Self-organizing Maps, Empirical Orthogonal Function

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 in the literature.

Self-organizing Maps [5] (SOM) aim ideally to group homogeneous individuals through a low-dimensional projection and to highlight the neighborhood structure between classes. The SOM networks have the ability to be robust, even when some values are missing [9]. SOM-based methods for recovering the missing values have already been proposed, in [9] and [12] for instance. They usually make an intensive use of the spatial correlation and fill the missing values of 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. 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 the presentation to the new algorithm for conditional missing value recovery. The following section is recalling various imputation methods. 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 almost no information about the input data is required [5]. 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 denoted $\mathbf{m}(t) = [\mathbf{m}_1(t), \mathbf{m}_2(t), ..., \mathbf{m}_c(t)]$, where $\mathbf{m}_i(t)$ is the T-dimensional weight vector of 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.

First the network nodes are randomly initialized 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 unit of the neuron whose weights are closest to the sample is calculated as $\mathbf{m}_{BMU_{\mathbf{x}_{t+1}}} = \text{Argmin}_{\mathbf{m}_i, i \in I}\{\|\mathbf{x}_{t+1} - \mathbf{m}_i(t)\|\}$, where $I = [1, 2, ..., c]$ is the set of network node indices, $BMU_{\mathbf{x}_{t+1}}$ (or BMU hereafter for the sake of simplicity) 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 [9] is used. For the randomly drawn sample, $\mathbf{x}_{t+1} \in \mathbb{R}^T$, having missing value(s), we split the original set into two subsets $\mathbb{R}^T = NM_{\mathbf{x}_{t+1}} \cup M_{\mathbf{x}_{t+1}}$, 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

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

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, with k going through all the indexes in the subset $NM_{\mathbf{x}_{t+1}}$ when values are not missing.

Then the BMU is calculated with

$$\mathbf{m}_{BMU_{\mathbf{x}_{t+1}}} = \underset{\mathbf{m}_i, i \in I}{\text{Argmin}} \left\{ \|\mathbf{x}_{t+1} - \mathbf{m}_i(t)\|_{NM_{\mathbf{x}_{t+1}}} \right\}. \quad (2)$$

When the BMU is found the network weights are updated as, $\forall i \in I$ and $\forall k \in NM$

$$\mathbf{m}_{i,k}(t+1) = \mathbf{m}_{i,k}(t) - \varepsilon(t)\lambda\left(\mathbf{m}_{BMU}, \mathbf{m}_i, t\right)\left[\mathbf{m}_{i,k}(t) - \mathbf{x}_{t+1,k}\right], \quad (3)$$

where $\varepsilon(t)$ is the adaptation gain parameter, which is $]0,1[$-valued, gradually decreasing 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 starts 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) [4].

The idea is to use a bootstrap process to ensure the convergence. First, an empirical probability for any pair of input variables 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 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. Secondly, the map chosen in the second step is the one that maximizes the likelihood of the neighborhood structure.

3. X-SOM/L-SOM-based Estimation Methodology

SOM-based estimation methods have already been proposed (for instance, [3]). These methods typically classified time-series and then, using peer-group peculiarities (as 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 secondly, the rebuilding process is discrete; missing values are filled with the corresponding values of the neurons to which the time-series is the closest to. Thus, for all series belonging to the same cluster, the estimations are the same.

Following [10], we propose a double classification to overcome these limits. As previously seen in [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

$$\mathbf{x}_{i,k} = \mathbf{m}_{BMU_{\mathbf{x}_i},k}, \quad (4)$$

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 clusters 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 $\mathbf{m}^1$, $d_1^{-1} = \left\|\mathbf{x}_i - \mathbf{m}^1_{BMU_{\mathbf{x}_i}}\right\|_{NM_{\mathbf{x}_i}}$. We define d_2 equivalently as $d_2^{-1} = \left\|\mathbf{x}'_k - \mathbf{m}^2_{BMU_{\mathbf{x}'_k}}\right\|_{NM_{\mathbf{x}'_k}}$.

Then, for each missing value of $\mathbf{x}_{i,k}$, we estimate the missing values contained in the sample through the double classification (denoted X-SOM/L-SOM) by

$$\mathbf{x}_{i,k} = d_1/(d_1+d_2)\,\mathbf{m}^1_{BMU_{\mathbf{x}_i},k} + d_2/(d_1+d_2)\,\mathbf{m}^2_{BMU_{\mathbf{x}'_k},i}. \quad (5)$$

For the X-SOM/L-SOM, we still have to select the optimal grid sizes c^1 and c^2. This is done by using a cross-validation principle. The X-SOM/L-SOM that gives the smallest validation error is used to perform the final completion of the data.

4. About Imputation Methods

In this section, we briefly review some alternative methods to deal with missing values. We begin with the Expectation Maximization (EM) Algorithm. Indeed, it is very common to apply this algorithm when dealing with financial asset returns. We then summerize the Empirical Orthogonal Functions methods and show how this factorial decomposition can be applied for missing value estimation. Finally, we present a method that also attempts to capture special time-series structure of datasets thought the combination of the SOM and EOF (see [11]).

4.1. Expectation Maximization Methods

The EM algorithm, presented by Dempster, Laird and Rubin in [2], is a technique to find maximum likelihood estimates in a

missing data situation. Since the estimates of the mean and the covariance matrix of an incomplete dataset depend on the unknown missing values, and, conversely, estimates of the missing values depend on the unknown statistics of the data. This estimation problem is nonlinear and has to be done iteratively.

The EM algorithm consists of two steps:

1. E-step calculates the expectation of the complete data sufficient statistics given the observed data and current parameter estimates.
2. M-step updates the parameter estimates through the maximum likelihood approach based on the current values of the complete sufficient statistics.

The algorithm proceeds in an iterative manner until the difference between the last two consecutive parameter estimates converges to a specified *criterion*. The final E-step computes the expectation of each missing value given the final parameter estimates and the observed data. This result will be used as the imputation value.

For each iteration (t), the E-step consists of

$$Q\left(\theta\left|\theta^{(t)}\right.\right) = E\left[L\left(\theta\left|Y\right.\right)\left|Y_{obs}, \theta^{(t)}\right.\right], \tag{6}$$

where

$$\begin{cases} L\left(.\left|Y\right.\right) \text{ denotes the likelihood function conditionally to Y,} \\ \theta \text{ the vector of parameter to be estimated,} \\ Y_{obs} \text{ the non-missing values,} \\ Y \text{ the sample,} \\ \theta^{(t)} \text{ the last vector of estimated parameter.} \end{cases}$$

Thus, the $(t+1)^{th}$ M-step finds $\theta^{(t+1)}$ that maximizes $Q\left(\theta\left|\theta^{(t)}\right.\right)$ such that

$$Q\left(\theta^{(t+1)}\left|\theta^{(t)}\right.\right) = \max_{\theta} Q\left(\theta\left|\theta^{(t)}\right.\right). \tag{7}$$

The main drawback of the EM algorithm is when the M-step is not in a closed form. In this case, the M-step could be difficult to perform.

Meng and Rubin [7] proposed an alternative algorithm called the Expectation Conditional Maximization (ECM) to solve this problem. The M-step is decomposed to multiple conditional maximizations. Consider $\theta = [\theta_1, \theta_2, ..., \theta_k]$ a k-dimensional vector of parameters. Then the Conditional M-step consists of k successive maximizations, for $i = 1, ..., k$ (with previous notations)

$$Q\left(\theta^{(t+1)}\left|\theta^{(t)}\right.\right) = \max_{\theta_i} Q\left(\theta\left|\theta^{(t)}\right.\right). \tag{8}$$

Otherwise, the ECM algorithm performs in the same way than the EM algorithm presented above.

4.2. Empirical Orthogonal Functions

Empirical Orthogonal Functions (EOF, [8]) allows us for factorial decomposition. EOF are here used as a denoising tool and for finding the missing values at the same time [1].

The EOF are calculated using standard and well-known Singular Value Decomposition (SVD) such as

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{*} = \sum_{k=1}^{K} \rho_k \mathbf{u}_k \mathbf{v}_k, \tag{9}$$

where $\mathbf{X}$ is the $(T \times N)$ data matrix, $\mathbf{U}$ and $\mathbf{V}$ are collections of singular vectors $\mathbf{u}$ and $\mathbf{v}$ in each dimension respectively, $\mathbf{D}$ is a diagonal matrix with the singular values ρ in its diagonal and K is the smaller dimension of $\mathbf{X}$ (or the number of nonzero singular values if $\mathbf{X}$ is not full rank). The singular values and the respective vectors are sorted in decreasing order.

When EOF are used to denoise the data, not all singular values and vectors are used to reconstruct the data matrix. Instead, it is assumed that the vectors corresponding to larger singular values contain more information with respect to the noise than the ones corresponding to smaller values [8]. Therefore, it is logical to select the q largest singular values and their corresponding vectors and reconstruct the denoised data matrix only using them.

In the case where $q < K$, the reconstructed data matrix is obviously not the same as the original one. The smaller the q, the more different is the new data, but the less noisy it is. The optimal q is selected using validation methods, as for example in [6].

EOF (or SVD) cannot be directly used with databases including missing values. The missing values must be replaced by some initial values in order to use the EOF. This replacement can be for example the mean value of the whole data matrix $\mathbf{X}$ or the mean in one direction, row wise or column wise. The latter approach is more logical when the data matrix has some temporal or spatial structure in its columns or rows.

After the initial value substitution, the EOF process begins by performing the SVD and the selected q singular values and vectors are used to build the reconstruction. In order not to lose any information, only the missing values of $\mathbf{X}$ are replaced with the values from the reconstruction. After their replacement, the new data matrix is again broken down into singular values and vectors with the SVD and rebuilt again. The procedure is repeated until a convergence *criterion* is fulfilled.

The procedure is summarized in Table 1.

4.3. SOM+EOF

The combinaition of the last two methodologies (see [11]) leads to an improvement of the accuracy of the rebuilt process. The SOM algorithm for missing values is first ran through performing a nonlinear projection for finding the missing values. Then, the result of the SOM estimation is used as initialization for the EOF method. This methodology is summarized in Figure 1.

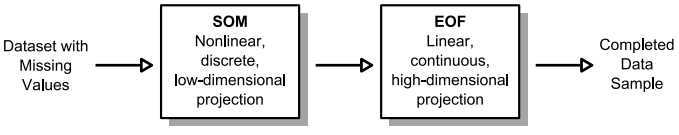

Figure 1: Summary of the (SOM+EOF) Global Methodology.

Table 1: Summary of the EOF method for finding missing values.

1	Initial values are substituted into missing values of the original data matrix $\mathbf{X}$
2	For each q from 1 to K
2.1	SVD algorithm calculates q singular values and eigenvectors
2.2	A number of values and vectors are used to make the reconstruction
2.3	The missing values from the original data are filled with the values from the reconstruction
3	The q with the smallest validation error is used to reconstruct the final filling of the missing values in $\mathbf{X}$

For the SOM, we must select the optimal grid size c and for the EOF the optimal number of singular values and vectors q to be used. This is done using the same validation set for all combinations of the parameters c and q. Finally, the combination of SOM and EOF that gives the smallest validation error is used to perform the final completion of the data.

Even the SOM as well as the EOF are able to fill the missing values alone, the experimental results demonstrate that together the accuracy is better. The fact that these two algorithms suit well together is not surprising. Two approaches are indeed complementarity.

First, the SOM algorithm allows us for a nonlinear projection. In that sense, even for dataset with complex and nonlinear structure, the SOM code vectors will succeed to capture the nonlinear characteristics of the inputs. However, the projection is done on a low-dimensional grid (in our case two-dimensional) with the possibility of losing some of the intrinsic information of the data.

Secondly, the EOF method is based on a linear transformation using the Singular Value Decomposition. Because of the linearity of the EOF approach, it will not reflect the non-linear structures of the dataset, but the projection space can be as high as the dimension of the input data and remains continuous.

At this stage, one can wonder about the efficiency of an EOF method for missing values initialized with X-SOM/L-SOM candidates (the method is straight forward and doesn't need to be presented here). The method will be tested in the experiment section but we expect a low improvement since spatial and temporal structure are already take into account by the X-SOM/L-SOM.

5. Experimental Results

In the following application, we illustrate our imputation method on a dataset of hedge fund returns[1] composed of 120 funds containing 121 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. Regarding the correlations of the transposed dataset, we find that the mean, minimum and maximum cross-correlations are .00, −.75 and .74, respectively.

Figure 2 shows some of the fund performances[2]. We observe that the fund returns under studies are low-correlated time-series. This highlights the fact that numbers of underlying factors (exchange rates, interest rates, credit grades, stocks on various markets, country and sector bets, styles drifts, uses of derivatives...) are considered in hedge fund industry. This also leads to a more difficult completion task, due to the heterogeneity of implicit risk factors.

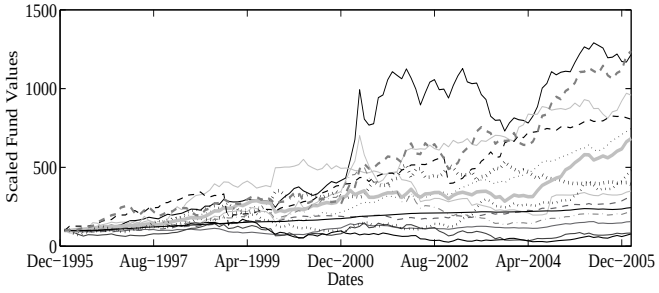

Figure 2: Rescaled asset values of 15 funds present in the database.

It is to be noticed that the initial database does not contain any missing values. For the following experiments, we artificially remove some values from the original dataset. More precisely, we generate two sets of data with different kinds of deletion. The first set contains missing values that appear randomly whereas the second set contains missing values that appears at the beginning of time-series. This second set is more realistic of a practical financial missing value issue, since fund return time-series often suffer from a lack of reporting due to a too recent inception date.

For the first test set, we randomly removed 7.5 percent of the data. 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, the EOF, the SOM+EOF and the X-SOM/L-SOM method. The 20 selected validation sets are the same for each method. The validation errors are shown in Figure 3. In the case of the X-SOM/L-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, representing a total of 100 units. We have roughly as many

[1] this database can be download at http://www.cis.hut.fi/projects/tsp/downloads/Finance2.tar.gz

[2] for the sake of simplicity, we rescaled the initial performances to 100 such as $v'_t = 100 \prod_{i=1}^{t} (1 + r_i)$, with r_i the return of a fund at the time i.

code vectors in the map as observations (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.

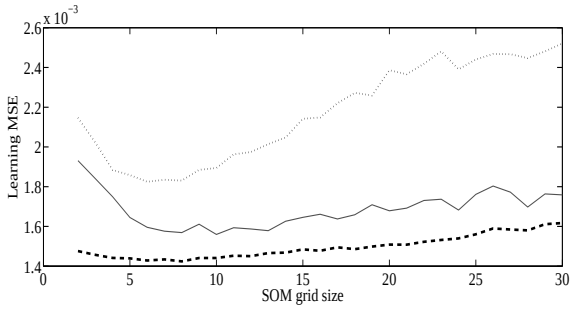

Figure 3: Validation errors against SOM grid sizes. The L-SOM validation error is the solid line, X-SOM is the dotted line and the X-SOM/L-SOM is the dashed line.

The smallest error achieved with the X-SOM/L-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 combining 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 3, it is clearly noticeable that with any SOM size the X-SOM/L-SOM method gives lower validation error than either L- or X-SOM alone.

Table 2: Learning and Test Root Mean Squared Errors for the ECM, the L-SOM, the X-SOM, the EOF, the SOM+EOF, the X-SOM/L-SOM and the X-SOM/L-SOM+EOF.

10^{-3}	Learning Error	Test Error
ECM	2.8	3.7
L-SOM	1.6	1.7
X-SOM	1.8	1.9
EOF	1.6	1.7
SOM+EOF	1.4	1.6
X-SOM/L-SOM	1.3	1.4
X-SOM/L-SOM+EOF	1.3	1.5

Table 2 contains the validation and test errors of all three methods. We can see that the X-SOM/L-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. The EOF initialized with a X-SOM/L-SOM does not outperform the estimation with a X-SOM/L-SOM.

For the second test set, we also removed 7.5 percent of the data. We choose a constrained randomisation process in order to get missing value at the begining of time-series and to ensure that at least one third of the time-series remains complete. The same cross-validation procedure has been followed.

Regarding the second experiment, the qualities of rebuilt processes are slightly downgraded. The concentration of missing values at the beginning of time-series especially penalizes the

Table 3: Learning and Test Root Mean Squared Errors for the ECM, the L-SOM, the X-SOM, the EOF, the SOM+EOF, the X-SOM/L-SOM and the X-SOM/L-SOM+EOF.

10^{-3}	Learning Error	Test Error
ECM	3.0	3.8
L-SOM	1.6	1.8
X-SOM	2.0	2.0
EOF	1.9	2.0
SOM+EOF	1.7	1.8
X-SOM/L-SOM	1.5	1.6
X-SOM/L-SOM+EOF	1.6	1.7

L-SOM and the SOM+EOF. In this second experiment, the EOF applied to the X-SOM/L-SOM does not lead to any improvement. We performed additional tests[3] onto some other databases and our results remain consistent with those presented in Tables 2 and 3.

6. Conclusion

In this paper, we have proposed a new X-SOM/L-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 accurated than those obtained with others missing value estimation methods.

References

[1] J. Boyd, E. Kennelly, P. Pistek, Estimation of EOF expansion coefficients from incomplete data, Deep Sea Research 41 (1994) 1479-1488.

[2] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via EM algorithm, Journal of the Royal Statistical Society 39 (1977) 1-38.

[3] F. Fessant, S. Midenet, Self-Organising map for data imputation and correction in surveys, Neural Computing & Applications 10 (2002) 300-310.

[4] C. Guinot, B. Maillet, P. Rousset, Understanding and reducing variability of SOM neighbourhood structure, Neural Networks 19 (2006) 838-846.

[5] T. Kohonen, Self-organizing Maps, Springer-Verlag (Germany, Berlin, 1995).

[3] The other database for which the completion algorithms were tested was containing the weekly net asset value times-series of 100 long-only stock funds investing on the American market on the period 2003-2007. The same ranking of the compared methodologies is observed, and the same magnitude was obtained regarding the improvement provided by the X-SOM/L-SOM method compared to the traditional ECM one. Detailled results are available from authors on demand.

[6] A. Lendasse, V. Wertz, M. Verleysen, Model selection with cross-validations and bootstraps - Application to time series prediction with RBFN models, in Proc. ICANN'03, Lecture Notes in Computer Science, Vol. 2714 (Springer-Verlag, Berlin, 2003) 573-580.
[7] X. Meng, D. Rubin, Maximum likelihood estimation via the ECM algorithm, Biometrika 80 (1993) 267-278.
[8] R. Preisendorfer, Principal Component Analysis in Meteorology and Oceanography, Elsevier Science Pub. Co. (U.S.A, New York, 1988).
[9] T. Samad, S. Harp, Self-organization with partial data, Network 3 (1992) 205-212.
[10] G. Simon, A. Lendasse, M. Cottrell, J.-C. Fort, M. Verleysen, Double SOM for long-term time series prediction, Proceedings of the Workshop on Self-Organizing Maps (Japan, Kitakyushu, 2003), 35-40.
[11] A. Sorjamaa, B. Maillet, P. Merlin, A. Lendasse, SOM+EOF for finding missing values, Proceedings of the European Symposium on Artificial Neural Networks (Belgium, Brugges, 2007) 115-120.
[12] S. Wang, Application of self-organising maps for data mining with incomplete data sets, Neural Computing and Applications 12 (2003) 42-48.