Double Quantization of the Regressor Space for Long-Term Time Series Prediction: Method and Proof of Stability

G. Simon[a] *, A. Lendasse[b] M. Cottrell[c], J.-C. Fort[dc], M. Verleysen[a] †

[a]DICE - Université catholique de Louvain
Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium

[b]Helsinki University of Technology - Neural Networks Research Centre
P.O. Box 5400, FIN-02015 HUT, FINLAND
on leave from CESAME - Université catholique de Louvain
Avenue Georges Lemaître, 4, B-1348 Louvain-la-Neuve, Belgium.

[c]SAMOS-MATISSE, UMR CNRS 8595, Université Paris I - Panthéon Sorbonne
Rue de Tolbiac 90, F-75634 Paris Cedex 13, France

[d]Lab. Statistiques et Probabilités, CNRS C55830, Université Paul Sabatier Toulouse 3
Route de Narbonne 118, F-31062 Toulouse Cedex, France

The Kohonen self-organization map is usually considered as a classification or clustering tool, with only a few applications in time series prediction. In this paper, a particular time series forecasting method based on Kohonen maps is described. This method has been specifically designed for the prediction of long-term trends. The proof of the stability of the method for long-term forecasting is given, as well as illustrations of the utilization of the method both in the scalar and vectorial cases.

1. Introduction

Determining in advance the future evolution of a time series is a problem of major interest in many fields of applications as finance (forecasting returns or stock markets), hydrology (predicting river floods), engineering (estimating future electrical consumption), etc.

As this problem can be found in many fields, many methods have been developed with very different approaches, from statistics to system identification and more recently neural networks. Most of the time, the models are linear and perform well on a rather short-term horizon, depending on the complexity of the problem. Their efficiency on a longer term is more questionable. This fact is due to the learning strategy used to fit the model to the data. The goal is usually to optimize the performance at a given term, most often the next time step. There are only a few attempts to explicitly predict values at long term, or at least global trends, as for example [1]. This problem is quite hard since the uncertainty increases with the horizon of prediction.

Another issue generally shared by classical models (such as ARX, ARMAX, ...) is that they are used to predict a single value of a scalar time series. In practice some industrial applications require the prediction of a set of values in one single step instead of several independent values. Forecasting a vector of values requires more complex models able to predict several components together. If the approach is to develop several simple models and combine them to predict a vector, one can lose the correlation information between the vector components. Though each model may perform well, the forecasting accu-

*G. Simon is funded by the Belgian F.R.I.A.
†M. Verleysen is Senior Research Associate of the Belgian F.N.R.S.
The scientific responsibility rests with the authors.

racy could be rather poor when considering the vector of predicted values as a whole. Developing methods able to predict several values at each step, with the same expected performance on each value, should thus be a major concern.

Let us consider now the general problem of forecasting at long term. Despite the fact that long-term predictions in real situations will probably never be very accurate, in some applications there is a need to have at least some ideas about the future of the time series. For example, answers to questions such as "Are there bounds on the future values?"or "What can we expect in average?"or even "Are the confidence intervals on future values large or narrow?"can give some ideas about the time series evolution at long-term.

In this paper, we present a forecasting method specifically designed to perform long-term forecasting in terms of general evolution of the time series. Simulations, which are here defined as one-step ahead predictions performed recursively to enlarge the prediction horizon, are the real goal. Repeating such simulations by a Monte-Carlo procedure enables the observation of their distribution and the computation of classical statistics such as mean, variance, quartiles, confidence intervals, etc. In order to achieve these goals, the method should be stochastic (to allow Monte-Carlo repetitions) and stable (to avoid unrealistic predictions even in the recursive case). The stochastic nature of the method is guaranteed by the use of a conditional probability law model. Its stability is proved in the paper. An attractive feature of the method is that it can be used to predict either scalar or vectorial time series, with the same expected precision for each component in the vectorial case; this will allow reducing the number of recurrences in a long-term prediction.

The general problem of time series forecasting first consists in the development of a model, which is in turn used to predict future values. More formally, given a time series of values $x(t)$ with $1 \leq t \leq n$, the prediction can be defined as:

$$[x(t+1), ..., x(t+d)] = f(x(t), ..., x(t-p+1)) + \varepsilon_t, \quad (1)$$

where d is the size of the vector to be predicted, f is the model of the data generating process, p is the number of past values that influence the future values and ε_t is a centred noise vector. The past values are gathered in a p-dimensional vector called *regressor*. Having at one's disposal n past values $x(t)$ (with $n >> p$ and $n >> d$) means that relation (1) is known for $(n - p - d + 1)$ past time steps. The problem is thus to model relation (1) using the past information contained in the regressors.

The general principle of the method presented in this paper is to segment the space of regressors, in order to build local models. For this step of the method, the Self-Organizing Map (SOM) algorithm [2] is used. This algorithm performs a vector quantization of the data, leading to representatives (*prototypes*) in each portion of the space. The idea of the method is to use two SOMs, one to segment the regressor space, and another one to segment the space of differences between consecutive regressors. These differences are built to include the temporal dependences in the model. Once these two maps are built and their relations characterized, simulations over a long-term horizon can be performed. By repeating these simulations using a Monte-Carlo procedure, it is therefore possible to study their distribution and the statistics that give information on the long-term distribution of the time series.

Though we mainly use the vector quantization property of the SOMs to segment the spaces, we have chosen to use SOMs instead of other vector quantization (VQ) methods, since SOMs are efficient and fast compared to other VQ methods with a similar complexity [3]. Furthermore, they provide an intuitive and helpful graphical representation. As the quantization properties (by contrast to topological properties) of SOMs are similar for one- and two-dimensional maps, the former will be used in this work for simplicity and illustration.

In this paper, we first briefly recall some basic concepts about the SOMs. Then, we describe the principle of the double vector quantization (DVQ) forecasting method. For the sake of simplicity, the method is first presented for scalar time series prediction (i.e. $d = 1$ in (1)) and detailed later on for vector forecasting. We define the method

stability and give in full details the proof that the method is stable according to his definition. We also show illustrative examples for both scalar and vector predictions.

2. The Kohonen Self-Organizing Maps

The Self-Organizing Map (SOM) is an unsupervised classification algorithm introduced in the 80's by Teuvo Kohonen [2]. Self-Organizing Maps have often been used in many different applications since their first description. Their theoretical properties are well established [4], [5].

A Self-Organizing Map places a fixed number of prototypes in the data space, performing a rough approximation of the data density. These prototypes are linked by neighbourhood relationships using a predefined 1- or 2-dimensional grid. During the learning stage, the prototypes are moved within the data space according to the location of the considered data and to constraints given by the neighbourhood relationships. After the learning stage the set of prototypes has established a vector quantization of the data. Each prototype is associated to a region of the space, namely a Voronoi zone or a *cluster*, where data share some similar features. Furthermore, the prototypes preserve the topology: two similar data belong either to the same cluster or to two neighbouring ones (on the grid). The SOM obtained after learning also allows a graphical representation that can be interpreted intuitively.

Though the SOMs are usually considered as a classification or recognition tool, there are a few works where SOMs were used in forecasting tasks. For example, some authors [6] use SOMs to create clusters in the regressor space, eventually associating each cluster to a linear local model [7], [8] or a nonlinear one [9]. Other VQ algorithms like Neural Gas [10] are also used in combination with local linear models. Another way is to split the problem into the prediction of a normalized curve, and the prediction of the curve mean and standard deviation [11]. Recursive SOMs [12] (and pioneer work on leaky integrators [13]) try to learn sequences of data, as applied in [15] for speech recognition problems. RSOMs can be further combined with local linear models [14]. By contrast to these works on short-term forecasts, the method presented in this paper uses SOMs to build a stochastic model specifically designed to provide long-term predictions.

3. The double quantization method

The goal of the method presented in this paper is to extract long-term information or behaviour from a time series. The method is based on the SOM algorithm used to characterize (or to *learn*) the series. In a further *forecasting* step, the model previously learned is used to perform a prediction of the long-term evolution of the time series.

As explained in the introduction, this method can be applied to scalar time series as well as to vectorial ones. The method will first be described in the scalar case. Though the generalization to the vectorial case is straightforward, some additional details will be provided. The method will be applied to various time series in section 5.

3.1. Characterization (scalar case)

According to the general formulation of a non-linear auto-regressive model (1), the method uses regressors of past values to predict the future evolution of a time series. Those regressors are created as follows. Having at one's disposal a scalar time series of n values, the latter are converted into p-dimensional vectors, according to:

$$\mathbf{x}_{t-p+1}^{t} = \{x(t-p+1), \ldots, x(t-1), x(t)\}, \quad (2)$$

where $p \leq t \leq n$, and $x(t)$ are the values of the original time series at our disposal. The $\mathbf{x}_{t-p+1}^{t}$ are called regressors, and $n-p+1$ of them are obtained from the original time series. In the $\mathbf{x}_{t-p+1}^{t}$ notation, $t-p+1$ and t respectively denote the first and last time indices of the regressor; this notation will be used throughout Sections 3 and 5 of this paper. Note that the order of the regressor p is supposed to be optimal, i.e. it is supposed to contain all the information that can be obtained from the past evolution of the time series. More considerations on the determination of an optimal regressor in the context of non-linear prediction can be found for example in [16].

The regressors $\mathbf{x}_{t-p+1}^{t}$ are then manipulated so

that other vectors are created, according to:

$$\mathbf{y}_{t-p+1}^{t} = \mathbf{x}_{t-p+2}^{t+1} - \mathbf{x}_{t-p+1}^{t}. \quad (3)$$

The $\mathbf{y}_{t-p+1}^{t}$ vectors are called *deformations*. By definition each deformation $\mathbf{y}_{t-p+1}^{t}$ is associated to a single regressor $\mathbf{x}_{t-p+1}^{t}$; $n-p$ deformations can be obtained from a time series with n values.

To clarify the situation, the space containing the $\mathbf{x}_{t-p+1}^{t}$ regressors will be called in the following the *original space*, while the one containing the $\mathbf{y}_{t-p+1}^{t}$ deformations will be called the *deformation space*.

The main step of the method is the application of the SOM algorithm to each of the two spaces. The SOM algorithm performs a vector quantization of the original and deformation spaces respectively. The first SOM, in the original space, results in n_1 prototypes $\bar{\mathbf{x}}_i$ ($1 \leq i \leq n_1$). Clusters containing all regressors associated respectively to each prototype $\bar{\mathbf{x}}_i$ are denoted C_i with $C_i \in \mathcal{C}$, where $\mathcal{C}$ is the set of prototypes in the original space such that $\#\mathcal{C} = n_1$. In the deformation space n_2 prototypes $\bar{\mathbf{y}}_j$ ($1 \leq j \leq n_2$) are obtained by a second SOM and the corresponding clusters containing the deformations are noted $C'_j \in \mathcal{C}'$; $\mathcal{C}'$ is the set of prototypes in the deformation space such that $\#\mathcal{C}' = n_2$.

The double quantization of regressors and deformations only gives a characterization of the past evolution of the time series. This characterization is static and does not reflect the dynamics of this past evolution. However, the dynamics may be found in the associations between the deformations $\mathbf{y}_{t-p+1}^{t}$ and their corresponding regressors $\mathbf{x}_{t-p+1}^{t}$, i.e. in the characterization of how the series has evolved between a regressor and the next one.

To model the dynamics of the time series it is thus necessary to build a representation of the relations between the regressors and their deformations. This representation is a matrix f_{ij} that contains the relations between the $\mathbf{x}_{t-p+1}^{t}$ and the $\mathbf{y}_{t-p+1}^{t}$ with respect to their clusters (C_i and C'_j respectively). Each row of the f_{ij} matrix ($1 \leq j \leq n_2$) is in fact the conditional probability that $\mathbf{y}_{t-p+1}^{t}$ belongs to C'_j given the fact that $\mathbf{x}_{t-p+1}^{t}$ belongs to C_i. In practice, these probabilities are estimated by the empirical frequencies:

$$f_{ij} = \frac{\#\{\mathbf{x}_{t-p+1}^{t} \in C_i \text{ and } \mathbf{y}_{t-p+1}^{t} \in C'_j\}}{\#\{\mathbf{x}_{t-p+1}^{t} \in C_i\}}, \quad (4)$$

with $1 \leq i \leq n_1$, $1 \leq j \leq n_2$. As expected with such a definition, elements f_{ij} ($1 \leq j \leq n_2$) sum to one for a fixed i. This matrix will be called the transition matrix in the following.

3.2. Forecasting (scalar case)

Having at one's disposal two sets of prototype vectors, respectively in the original and deformation spaces, together with the transition matrix it is now possible to use this characterization of the time series to perform a long-term evolution forecasting. Defining horizon $h = 1$ as the next value, i.e. $t+1$ for instant t, the goal is to forecast evolutions of the series for horizons $h > 1$.

Consider a value $x(t)$ for instant t. The corresponding regressor is $\mathbf{x}_{t-p+1}^{t}$. Using the prototypes $\bar{\mathbf{x}}_i$ in the original space, the cluster of $\mathbf{x}_{t-p+1}^{t}$ is determined, for example k (this operation consists in finding the nearest neighbour $\bar{\mathbf{x}}_k$ from regressor $\mathbf{x}_{t-p+1}^{t}$ according to the Euclidean distance used in the SOM algorithm). A deformation prototype $\bar{\mathbf{y}}_l$ is then chosen at random among the $\bar{\mathbf{y}}_j$ according to the conditional probability distribution defined by row k of the transition matrix, i.e. according to $f_{kj}, 1 \leq j \leq n_2$. The prediction for instant $t+1$ is obtained according to relation (3):

$$\hat{\mathbf{x}}_{t-p+2}^{t+1} = \mathbf{x}_{t-p+1}^{t} + \bar{\mathbf{y}}_l, \quad (5)$$

where $\hat{\mathbf{x}}_{t-p+2}^{t+1}$ is the estimate of $\mathbf{x}_{t-p+2}^{t+1}$ given by our model.

Note that the result $\hat{\mathbf{x}}_{t-p+2}^{t+1}$ obtained here is a vector of size p. The components of $\hat{\mathbf{x}}_{t-p+2}^{t+1}$ are in fact estimations for instants from $t-p+2$ to $t+1$. In the scalar case, and values from $t-p+2$ to t being known, the scalar prediction $\hat{x}(t+1)$ is extracted from $\hat{\mathbf{x}}_{t-p+2}^{t+1}$. The procedure is then iterated, plugging in $\hat{x}(t+1)$ for $x(t)$ in (2) to compute $\hat{\mathbf{x}}_{t-p+3}^{t+2}$ by (5) and extracting $\hat{x}(t+2)$. The same is done for $\hat{x}(t+3)$, $\hat{x}(t+4)$, ..., $\hat{x}(t+h)$. These iterations up to horizon h are called a simulation of the time series.

As the goal of the method is to provide some ideas about the possible evolution of the series, we are interested in the distribution of the simulations. The whole simulation procedure above has thus to be repeated. Since the random choice of deformation according to the conditional probability distributions given by the rows of the transition matrix is stochastic, the simulation procedure is repeated using a Monte-Carlo procedure. The observation of all these simulations makes it possible to estimate their distribution, and infer global information about the time series such as the evolution of its mean, its variance, confidence intervals, etc.

3.3. Generalization: vector forecasting

Suppose that the studied problem requires the prediction of a vector of values in one step instead of (several) single value(s). For example when forecasting an electrical consumption, the problem may be to forecast hourly values for a whole day instead of predicting each value separately. The problem is thus to predict vectors $\mathbf{x}_{t+1}^{t+d}$ of future values of the time series $x(t)$, where $\mathbf{x}_{t+1}^{t+d}$ is defined as:

$$\mathbf{x}_{t+1}^{t+d} = \{x(t+1), x(t+2), ..., x(t+d)\}. \quad (6)$$

In such applications d is determined according to some a priori knowledge about the series (for example the 24 hourly values in the electrical consumption problem).

As above, regressors of this kind of time series can be constructed according to:

$$\mathbf{x}_{t-p+1}^{t} = \{\mathbf{x}_{t-p+1}^{t-p+d}, \mathbf{x}_{t-p+d+1}^{t-p+2d}, \ldots, \mathbf{x}_{t-d+1}^{t}\}, \quad (7)$$

where p is considered to be a multiple of d for simplicity.

Relation (7) can be illustrated using the electrical consumption example. Suppose that $d = 24$ hourly values and that values from three days are needed in the regressors. The latter thus contain $p = 72$ values. The regressor at time t is thus given by:

$$\begin{aligned}\mathbf{x}_{t-71}^{t} = \quad & \{\underbrace{x(t-71), \ldots, x(t-48)}_{\mathbf{x}_{t-71}^{t-48}}, \\ & \underbrace{x(t-47), \ldots, x(t-24)}_{\mathbf{x}_{t-47}^{t-24}}, \\ & \underbrace{x(t-23), \ldots, x(t-1), x(t)}_{\mathbf{x}_{t-23}^{t}}\}.\end{aligned} \quad (8)$$

In other words, regressors $\mathbf{x}_{t-p+1}^{t}$ are constructed as the concatenation of p/d d-dimensional vectors of past values of the time series, similarly as it is the concatenation of scalar ($d = 1$) past values in the scalar case. Note again that the order p of this regressor is supposed to be a multiple of d for simplicity, although this is not compulsory.

Deformation regressors can also be defined for the vectorial case, using a generalization of (3):

$$\mathbf{y}_{t-p+1}^{t} = \mathbf{x}_{t-p+d+1}^{t+d} - \mathbf{x}_{t-p+1}^{t}. \quad (9)$$

The algorithm can therefore be generalized in a natural way. The SOM algorithm is applied on both spaces, classifying respectively the vectorial regressors $\mathbf{x}_{t-p+1}^{t}$ and the vectorial deformations $\mathbf{y}_{t-p+1}^{t}$. We obtain n_1 prototypes $\bar{\mathbf{x}}_i$ in the original space associated to the clusters $C_i \in \mathcal{C}$, with $1 \leq i \leq n_1$. In the deformation space, we obtain n_2 prototypes $\bar{\mathbf{y}}_j$ associated to the clusters $C'_j \in \mathcal{C}'$, $1 \leq j \leq n_2$.

Relation (4) can be generalized straightforwardly to the vectorial case: the vectorial definition of the f_{ij} uses the same notations even though $\mathbf{x}_{t-p+1}^{t}$ and $\mathbf{y}_{t-p+1}^{t}$ now design vectors of vectors instead of vectors of scalars.

The simulation forecasting procedure can be generalized too:

- consider the vectorial regressor $\mathbf{x}_{t-p+1}^{t}$;
- find the corresponding vectorial prototype $\bar{\mathbf{x}}_k$;
- choose a vectorial deformation prototype $\bar{\mathbf{y}}_l$ among the $\bar{\mathbf{y}}_j$ according to the conditional distribution given by elements f_{kj} of row k;

- compute the forecast as:

$$\hat{\mathbf{x}}_{t-p+d+1}^{t+d} = \mathbf{x}_{t-p+1}^{t} + \bar{\mathbf{y}}_l; \tag{10}$$

- as $\hat{\mathbf{x}}_{t-p+d+1}^{t+d}$ is a p-dimensional vector, extract the vector $\hat{\mathbf{x}}_{t+1}^{t+d} = (\hat{x}(t+1), \hat{x}(t+2), ..., \hat{x}(t+d))$ from its d last columns;

- repeat h/d times until horizon h.

As for the scalar case, a Monte-Carlo procedure is used to repeat the whole simulation procedure. Then the simulation distribution and its statistics can be observed, which in turn gives information about the long term of the time series.

3.4. Comments

The DVQ method, as any other forecasting one, assumes that the time series data satisfy the relationships of the underlying model. In the case of the DVQ method, the relationships correspond to the lines of the transition matrix, i.e. to conditional probability laws. In other words, it is first assumed that the knowledge of a regressor is sufficient to build a conditional model of prediction. Secondly, it is assumed that using discrete laws instead of continuous ones does not penalize the prediction results.

The first assumption is common in any regressor-based prediction model: even in AR models, it is assumed that the regressor includes sufficient information to perform a prediction. This question is closely related to the application-dependent choice of the regressor size. The series to be modelled also have to be stationary (in mean and variance), at least over a reasonable window size. Short-term trends or periodicities may be taken into account by using sufficiently large regressors (it is the stationarity of the series of regressors that is important, not the one of individual values in the time series). More precisely, regressors should be of a size multiple of the short-term trends or periodicities. However long-term trends and periodicities should be removed from the series before applying the method. Nevertheless, as it will be seen in the experiments, this point may reveal not critical. Indeed long-term variations in the series may be taken into account by a higher number of clusters in the regressor (and deformation).

The second assumption may be seen as a specific implementation of the bias-variance dilemma, when a finite number of data is available: the number of lines and columns in the transition matrix may be increased to reduce the bias or decreased to reduce the variance of the model. As it will be detailed in the experimental part of this paper, the proposed methodology suggests fixing this compromise by cross-validation.

Compared to the use of a single SOM where the regressor and the deformation vectors would be concatenated (even through some weighting), the use of two SOMs enables to build a transition matrix which is a discrete approximation of the conditional law between regressors and deformations; a stochastic model is obtained in each cluster in the regressor space. Thus it is possible to repeat simulations and compute statistics on the results (mean, variance, confidence intervals) which are the real goal of the procedure.

Furthermore using the SOM to quantize the $\mathbf{x}_{t-p+1}^{t}$ and $\mathbf{y}_{t-p+1}^{t}$ vectors helps the method to reach easily the goal of forecasting vectors with the same expected precision for each component. Indeed while looking deeper in the SOM algorithm, it can be noticed that all components of the $\mathbf{x}_{t-p+1}^{t}$ and $\mathbf{y}_{t-p+1}^{t}$ vectors are used in an identical way in all computations of the algorithm. In other words, none of the $\mathbf{x}_{t-p+1}^{t}$ or $\mathbf{y}_{t-p+1}^{t}$ components have a greater importance than other components, for example in the adaptation of prototypes, etc. If equation (1) is a reasonable model of the series, i.e. if the prediction of d values after time t is deemed to be feasible based on p values until time t, then the performances on each of the d components to be predicted are expected to be similar.

As mentioned in the description of the method in the vectorial case, the regressor order p is supposed to be a multiple of d. This is also the case for scalar predictions, as any p is a multiple of 1. In both the scalar and vectorial cases the regressors were supposed to be constituted by successive values of the series (see equations (1), (2), (6), (7)). However, according to some knowledge about the time series or to some validation proce-

dure, it could be advantageous to consider only a limited number of past values in the regressor. In other words, if we consider the scalar case for illustration, the regressor may contain only specific past values. For example, we could take:

$$\mathbf{x}_{t-p+1}^{t} = \{x(t-5), x(t-3), x(t-2), x(t)\}, \quad (11)$$

instead of:

$$\mathbf{x}_{t-p+1}^{t} = \begin{array}{c} \{x(t-5), x(t-4), x(t-3), \\ x(t-2), x(t-1), x(t)\}, \end{array} \quad (12)$$

omitting possibly unnecessary values $x(t-1)$ and $x(t-4)$ in the regressor. This comment also applies to the vectorial case.

Another comment concerns an immediate extension of the method. The method has been presented here in a scalar time series context, both in scalar and vectorial model cases. Nevertheless, if a problem requires the prediction of several scalar series simultaneously (in other words a vectorial series), the same vectorial method could be applied in a straightforward way. The only changes would concern the definition of the regressors that would now become vectors of spatially correlated values instead of temporally successive ones.

Finally, note that in practice any kind of SOM can be used, but it is assumed here that one-dimensional maps (or strings) are more adequate in this context, for example for illustration purposes. More specifically, remind that each row of the transition matrix corresponds to a conditional probability law. If the series contains some regularity (as assumed when building a model), all possible values of the deformations (columns in the matrix) will not occur for a specific cluster in the regressor space; in other words, the transition matrix will be sparse. Furthermore, because of the topological property of Kohonen maps, non-zero elements corresponding to similar deformations will appear close to one another on a row. This last property will be less visible if two-dimensional maps are used instead of one-dimensional ones. Indeed in this case the index of the transition matrix columns (for example) should span the two-dimensional indices of the Kohonen map, resulting in close indices (on the map) being separated in the column index. This is the reason why one-dimensional maps are preferred in the following of this paper, insisting on the fact that nothing prevents the use of other maps besides this visualization property.

4. Method stability

Looking at the predictions obtained by the model described in the previous sections, the predicted values could either be contained in the range of the learning set or exceed this range, in particular at long term. In the first case, the series of predicted values is said to be stable while it is said to be unstable in the other case. We will prove in this section that the method presented in this paper is stable according to this definition. Stability is indeed a necessary condition to ensure that long-term forecasts will give some useful information about the future of the series.

To improve the readability of the proof simplier notations will be used. In the following of this section, for a fixed d and a fixed p, notation X_t will represent the vector $\mathbf{x}_{t-p+1}^{t}$. The last known the regressor will be denoted X_0. The prototype of a cluster C'_j of deformations will be noted Y_j. Finally, hats will be omitted for simplicity as all regressors X_t are estimations, except for $t = 0$.

The stability property is intuitively not surprising. As the model will produce predictions that are random choices according to an observed probability law, these predictions will remain in the range of observed data. If, for some reason, the prediction tends to exceed this range during the simulation, the next deformations will then drive the predictions back inside the range, at least with high probability. The following of this section is intended to give a technical proof of this intuitive result. Indeed as the stability of the method is a primary concern, it is necessary to prove that the method will not be unstable at long term, even with a low probability.

The proof consists in two steps: it is first shown that the series generated by the model is a Markov chain; secondly, it is demonstrated that this particular type of Markov chain is stable.

To prove that the series is a Markov chain, we consider the starting regressor X_0 of the simulation, and C_0 its corresponding SOM cluster in the

initial space. The deformation that is applied to X_0 at this stage is Y_0. Then the next values of the series are given by:

$$\begin{aligned} X_1 &= X_0 + Y_0, \\ X_2 &= X_1 + Y_1 = X_0 + Y_0 + Y_1, \\ &\dots \end{aligned} \tag{13}$$

with Y_0, Y_1, ... drawn at random among the deformation code vectors, according to the transition matrix, for clusters C_0, C_1, ... respectively. The series of predicted X_t, with $t > 0$, is therefore a Markov chain, homogeneous in time (the transition distributions are not time dependent), irreducible and defined over a numerable set (the initial X_t are in finite number, and so are the deformations).

To show the stability of this Markov chain and thus the existence of a stationary probability distribution, Foster's criterion [17] is applied. Note that Foster's criterion in fact proves a stronger result: the Markov chain will be proved to be ergodic. This stronger condition will be satisfied; consequently the Markov chain defined by relation (13) has a unique stationary (limiting) distribution.

A necessary and sufficient condition for an irreducible numerable chain to be ergodic (and therefore stable) is that there exists a positive function $g(.)$, a positive ε and a finite set Ω such that:

$$\forall x \in \Omega : \mathrm{E}\left(g(X_{t+1}) \mid X_t = x\right) < \infty, \tag{14}$$

$$\forall x \notin \Omega : \mathrm{E}\left(g(X_{t+1}) \mid X_t = x\right) - g(x) \leq -\varepsilon. \tag{15}$$

The proof is done here for a two-dimensional case, but can easily be generalized to other dimensions.

In the following proof, the function $g(.)$ is chosen to be $g(.) = \|.\|^2$ in (14) and (15).

Since the Markov chain is homogenous, it is sufficient to observe transition Y_0 from X_0 to X_1. The same development is also valid for any other transition.

Before going on in further details, let us remark that, if we consider a SOM with at least three prototypes in general position, cluster C_0 covers strictly less than half a plane. This fact can easily be observed for any vector quantization problem,

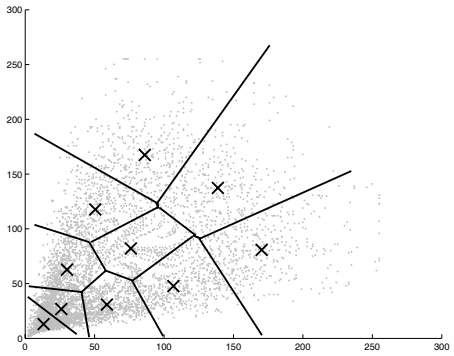


Figure 1. Two-dimensional regressors (dots), prototypes (crosses) and clusters from the Santa Fe A time series.

and in particular for any quantization of the regressor space in a time series context. In Figure 1 for example, the regressors of the Santa Fe A time series [18] (with $p = 2$) are plotted in their respective clusters in the $\mathbb{R}^2$ plane, with the corresponding prototypes.

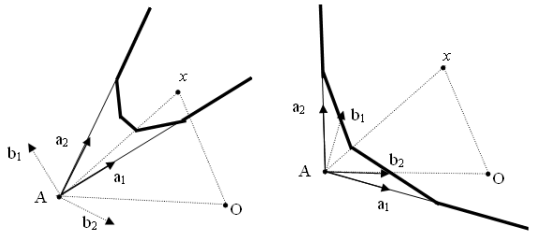
To prove Foster's criterion, we distinguish two cases. The first one is when $\|X_0\| < R_0$, where R_0 can be any constant. In this case we have by triangular inequality:

$$\begin{aligned} \mathrm{E}(\|X_1\|) &< R_0 + \|Y_0\| \\ &\leq R_0 + \max_j(\|Y_j\|). \end{aligned} \tag{16}$$

As the deformations Y_j are in finite number, the maximum of their norm is finite. This proves equation (14) in a straightforward and obvious way in the case of a finite norm of X_0 (i.e. bounded cluster case).

The other case to be considered is therefore when $\|X_0\| \to +\infty$. This can only happen in unbounded clusters; in Figure 1 for example, all clusters are unbounded except two. The unbounded cluster case is much more technical to prove.

Looking at Figure 2, we see that each unbounded cluster is included in a cone with vertex A and delimited by the normalized vectors a_1 and a_2. These vectors delimiting the border of the cone are chosen in the direction of the two infinite



(a) A Cluster within an acute angle (b) A Cluster within an obtuse angle

Figure 2. Notations for the cone containing an unbounded cluster of a SOM; see text for details.

segments at the borders of the cluster. There are two possibilities: a_1 and a_2 form either an acute angle or an obtuse one, as shown in Figure 2(a) and Figure 2(b) respectively.

In order to prove that Foster criterion can be applied, we first prove three technical lemma (Properties 1., 2. and 3.).

Property 1.
Denoting

$$lim_{\|x\|\to\infty}\frac{x}{\|x\|}\cdot a_i = \delta_i, \tag{17}$$

we have δ_1 and δ_2 both strictly positive in the acute angle case, while either δ_1 or δ_2 is positive for an obtuse angle.

Indeed, consider the origin O. Vector $\overrightarrow{Ox}$ is given by:

$$\overrightarrow{Ox} = \overrightarrow{OA} + \overrightarrow{Ax}. \tag{18}$$

Using (18) in (17), we have:

$$\begin{aligned}\frac{x}{\|x\|}\cdot a_i &= \frac{\overrightarrow{Ox}}{\|x\|}\cdot a_i \\ &= \left(\frac{\overrightarrow{OA}}{\|x\|}+\frac{\overrightarrow{Ax}}{\|x\|}\right)\cdot a_i \\ &= \frac{\overrightarrow{OA}\cdot a_i}{\|x\|}+\frac{\overrightarrow{Ax}\cdot a_i}{\|x\|} \\ &= \frac{\overrightarrow{OA}\cdot a_i}{\|x\|}+\frac{\overrightarrow{Ax}}{\|\overrightarrow{Ax}\|}\frac{\|\overrightarrow{Ax}\|}{\|x\|}\cdot a_i.\end{aligned}$$

Considering $\|x\| \to +\infty$, we obtain:

$$\frac{x}{\|x\|}\cdot a_i = \underbrace{\frac{\overrightarrow{OA}\cdot a_i}{\|x\|}}_{\to 0} + \frac{\overrightarrow{Ax}}{\|\overrightarrow{Ax}\|}\underbrace{\frac{\|\overrightarrow{Ax}\|}{\|x\|}}_{\to 1}\cdot a_i. \tag{19}$$

This proves relation (17) since the second term of the right member can be bounded below by a strictly positive constant.

As shown in Figure 2, this property is true for both $i = 1$ and 2 for an acute angle $\sphericalangle(a_1, a_2)$ and for at least $i = 1$ or $i = 2$ for an obtuse angle.

Property 2.
We define b_1 such that the angle $\sphericalangle(a_1, b_1)$ is $+\frac{\pi}{2}$. Similarly b_2 is defined such that the angle $\sphericalangle(b_2, a_2)$ is also $+\frac{\pi}{2}$. Then, for both the acute and obtuse angle cases, we have:

$$\inf_{x\in C}\frac{\overrightarrow{Ax}}{\|x\|}\cdot b_i = r_i > 0, \tag{20}$$

where C is the considered cone which has border vectors a_1 and a_2.

Indeed, we can rewrite the first term of (20) as:

$$\inf_{x\in C}\frac{\overrightarrow{Ax}}{\|x\|}\cdot b_i = \inf_{x\in C}\frac{\overrightarrow{Ax}}{\|\overrightarrow{Ax}\|}\frac{\|\overrightarrow{Ax}\|}{\|x\|}\cdot b_i. \tag{21}$$

Since $\dfrac{\|\overrightarrow{Ax}\|}{\|x\|} \to 1$ when $\|x\| \to +\infty$, we have:

$$\inf_{x\in C}\frac{\overrightarrow{Ax}}{\|\overrightarrow{Ax}\|}\underbrace{\frac{\|\overrightarrow{Ax}\|}{\|x\|}}_{\to 1}\cdot b_i > 0. \tag{22}$$

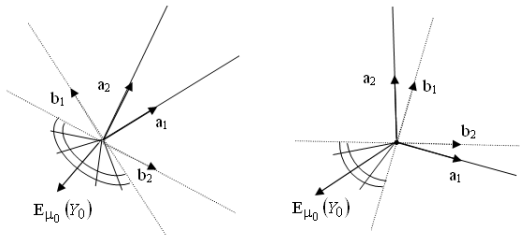
This property is valid for $i = 1$ and $i = 2$ both in the acute and obtuse angle cases.

Property 3.
Assume for the moment that:

$$\mathrm{E}_{\mu_0}(Y_0)\cdot a_1 < 0 \text{ and } \mathrm{E}_{\mu_0}(Y_0)\cdot a_2 < 0, \tag{23}$$

where μ_0 is the empirical distribution corresponding to an unbounded cluster C_0 in the transition matrix. Let us denote:

$$\mathrm{E}_{\mu_0}(Y_0)\cdot a_i = -\gamma_i < 0, \tag{24}$$



(a) Acute angle case. (b) Obtuse angle case.

Figure 3. Third geometrical property, see text for details.

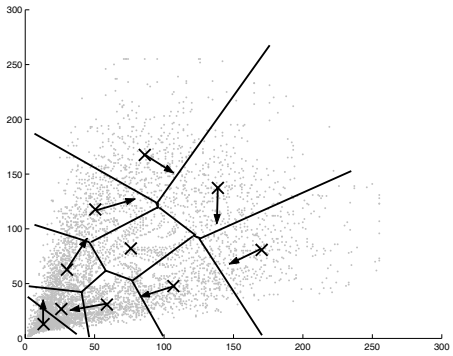


Figure 4. Two-dimensional regressors (dots), prototypes (crosses) and clusters from the Santa Fe A time series, and expected deformations in each unbounded cluster (arrows).

with $\gamma_i > 0$. As shown in Figure 3, we have that:

$$\mathrm{E}_{\mu_0}(Y_0) \cdot b_i < 0, \tag{25}$$

for at least $i = 1$ or $i = 2$ in case of an acute angle (Figure 3(a)) and for both $i = 1$ and $i = 2$ in the obtuse case (Figure 3(b)).

To be convinced of the initial assumption (23), it suffices to realize that, in average, the deformations will point to the interior of the distribution, as illustrated in Figure 3. This can easily be verified numerically too. In particular it has been verified for the Santa Fe A time series example, the result being shown in Figure 4.

Foster's criterion

Now we can apply Foster's criterion in the case of an unbounded cluster; remember that the case for bounded clusters has already been solved in the discussion after (16). Considering cluster C_0 such that its prototype is the nearest from data X_0, and considering its corresponding transition distribution, with $g(.) = \|.\|^2$, we have:

$$\begin{aligned}
&\mathrm{E}(g(X_1) \mid X_0 = x) - g(x) \\
&= \mathrm{E}(g(X_0 + Y_0) \mid X_0 = x) - g(x) \\
&= \mathrm{E}(\|X_0 + Y_0\|^2 \mid X_0 = x) - \|x\|^2 \\
&= \mathrm{E}_{\mu_0}(\|X_0 + Y_0\|^2) - \|X_0\|^2 \\
&= \|X_0\|^2 + 2\mathrm{E}_{\mu_0}(X_0 \cdot Y_0) + \mathrm{E}_{\mu_0}(\|Y_0\|^2) - \|X_0\|^2 \\
&= 2X_0 \cdot \mathrm{E}_{\mu_0}(Y_0) + E_{\mu_0}(\|Y_0\|^2).
\end{aligned}$$

Thus we obtain:

$$\begin{aligned}
&\mathrm{E}(g(X_1) \mid X_0 = x) - g(x) = \\
&2\|x\| \left[\frac{x \cdot \mathrm{E}_{\mu_0}(Y_0)}{\|x\|} + \frac{\mathrm{E}_{\mu_0}(\|Y_0\|^2)}{2\|x\|} \right].
\end{aligned} \tag{26}$$

First let us have a look at the second of the two terms between the brackets in (26). Since $\|Y_0\|^2$ is finite, we have:

$$\sup_{x \in C_0} \mathrm{E}_{\mu_0}(\|Y_0\|^2) < M_0 < \infty.$$

Thus, for $\alpha_0 > 0$ and $\|x\| > \dfrac{M_0}{\alpha_0}$, we have:

$$\frac{1}{\|x\|} \mathrm{E}_{\mu_0}(\|Y_0\|^2) < \alpha_0. \tag{27}$$

Now we still have to cope with the first term between the brackets of (26). We choose either $i = 1$ or $i = 2$ such that:

$$\begin{cases} \lim_{\|x\| \to +\infty} \dfrac{x}{\|x\|} \cdot a_i = \delta_i > 0, \\ \mathrm{E}_{\mu_0}(Y_0) \cdot b_i < 0. \end{cases} \tag{28}$$

In the case of an unbounded cluster, it is always possible to find either $i = 1$ or $i = 2$ such that those two conditions are fulfilled, according to Properties 1. and 3.

We suppose for now that $i = 2$ satisfies those two conditions (28). Looking closer to the first term between the brackets of relation (26), we can decompose $\mathrm{E}_{\mu_0}(Y_0)$ in the (b_2, a_2) basis as:

$$\mathrm{E}_{\mu_0}(Y_0) = (\mathrm{E}_{\mu_0}(Y_0) \cdot a_2) a_2 + (\mathrm{E}_{\mu_0}(Y_0) \cdot b_2) b_2, \tag{29}$$

thus obtaining, for $i = 2$:

$$\begin{aligned}\frac{x}{\|x\|} \cdot \mathrm{E}_{\mu_0}(Y_0) \\ = (\mathrm{E}_{\mu_0}(Y_0) \cdot a_2) \left(\frac{x}{\|x\|} \cdot a_2 \right) \\ + (\mathrm{E}_{\mu_0}(Y_0) \cdot b_2) \left(\frac{x}{\|x\|} \cdot b_2 \right).\end{aligned} \tag{30}$$

According to Property 1., we know that: $\lim_{\|x\| \to +\infty} \frac{x}{\|x\|} \cdot a_2 = \delta_2 > 0$, and thus:

$$\frac{x}{\|x\|} \cdot a_2 > \frac{\delta_2}{2}, \tag{31}$$

for $\|x\|$ sufficiently large.

Furthermore, from Property 3. we have $\mathrm{E}_{\mu_0}(Y_0) \cdot a_2 = -\gamma_2 < 0$ (23), and $\mathrm{E}_{\mu_0}(Y_0) \cdot b_2 < 0$, from (28). The last parenthesis in (30) can be developed as:

$$\frac{x}{\|x\|} \cdot b_2 = \left(\frac{\overrightarrow{OA}}{\|x\|} + \frac{\overrightarrow{Ax}}{\|x\|} \right) \cdot b_2. \tag{32}$$

For x sufficiently large in the unbounded cluster, that is for $\|x\| \to +\infty$, $\frac{\overrightarrow{OA}}{\|x\|} \cdot b_2 \to 0$.

Furthermore, by Property 2., we have:

$$\frac{\overrightarrow{Ax}}{\|x\|} \cdot b_2 \geq r_2 > 0. \tag{33}$$

Equation (32) can thus be simplified in:

$$\frac{x}{\|x\|} \cdot b_2 \geq \frac{r_2}{2} > 0, \tag{34}$$

for $\|x\|$ sufficiently large.

Replacing all those results in (30) we obtain:

$$\begin{aligned}\frac{x}{\|x\|} \mathrm{E}_{\mu_0}(Y_0) \\ \leq \underbrace{(\mathrm{E}_{\mu_0}(Y_0) \cdot a_2)}_{=-\gamma_2 \text{ by (24)}} \underbrace{\left(\frac{x}{\|x\|} \cdot a_2 \right)}_{>\frac{\delta_2}{2} \text{ by (31)}} \\ + \underbrace{\mathrm{E}_{\mu_0}(Y_0) \cdot b_2}_{<0 \text{ by (28)}} \underbrace{\left(\frac{x}{\|x\|} \cdot b_2 \right)}_{\geq \frac{r_2}{2} \text{ by (34)}},\end{aligned}$$

which finally results in:

$$\frac{x}{\|x\|} \mathrm{E}_{\mu_0}(Y_0) < -\gamma_2 \frac{\delta_2}{2}, \tag{35}$$

when $\|x\|$ is large enough, denoted here $\|x\| > L_0$.

The above development has been made under the hypothesis that $i = 2$ in (28). If on the contrary $i = 1$ satisfies the two conditions (28), we obtain by the same development:

$$\frac{x}{\|x\|} \mathrm{E}_{\mu_0}(Y_0) < -\gamma_1 \frac{\delta_1}{2}, \tag{36}$$

when $\|x\| > L_0'$.

Using relations (27) and (35 or 36), we can now rewrite (26) as:

$$\begin{aligned}&\mathrm{E}(g(X_1) \mid X_0 = x) - g(x) \\ &= 2\|x\| \left[\frac{x \cdot \mathrm{E}_{\mu_0}(Y_0)}{\|x\|} + \frac{\mathrm{E}_{\mu_0}(\|Y_0\|^2)}{2\|x\|} \right] \\ &< 2\|x\| \left[-\alpha_0 + \tfrac{1}{2}\alpha_0 \right] \\ &= -2\|x\| \tfrac{\alpha_0}{2},\end{aligned} \tag{37}$$

where $\|x\| > K_0 = \max(L_0, L_0')$ and α_0 in (27) is chosen such that $\alpha_0 = \min(\frac{\gamma_1 \delta_1}{2}, \frac{\gamma_2 \delta_2}{2})$.

This whole development has been done for one cluster C_0. The value α_0 depends on this cluster C_0, as well as vectors a_1, b_1, a_2, b_2 and values M_0, L_0, K_0. If we now consider all possible unbounded clusters C_i, taking $\alpha = \inf_{C_i} \alpha_i$ and $K = \sup_{C_i} K_i$, we have:

$$\begin{aligned}&\forall \|x\| \geq K : \\ &\frac{x \mathrm{E}_{\mu_0}(Y_0)}{\|x\|} + \frac{\mathrm{E}_{\mu_0}(\|Y_0\|^2)}{2\|x\|} < -\frac{\alpha}{2} < 0.\end{aligned} \tag{38}$$

By (26) and (38), we finally obtain:

$$\mathrm{E}(g(X_1) \mid X_0 = x) - g(x) < -\alpha \|x\|. \tag{39}$$

The right member of this inequality tends to $-\infty$ for $\|x\|$ increasing to $+\infty$.

To conclude, let us now define precisely the set Ω that must be used in Foster's criterion i.e. relations (14) and (15):

$$\Omega = \Big(\bigcup_{1 \leq j \leq n_2} C_j' \Big) \bigcup \{X_0 |\ \|X_0\| < K\}, \tag{40}$$

where C'_j are the bounded clusters as discussed in the introduction to the proof. With this definition, the above developments prove Foster's criterion (14) and (15). Thus the chain defined in relation (13) is ergodic, and admits a unique stationary distribution.

5. Experimental results

In this section, results obtained with the DVQ method are described. The method is illustrated on two time series. The first one is the well-known Santa Fe A benchmark presented in [18]. This application of the method illustrates the scalar case. The second time series is the Polish electrical consumption [20] from 1989 to 1996. This real-world problem requires the prediction of vectors of 24 hourly values and illustrates the vectorial case.

5.1. Methodology

The choice of constants n_1 and n_2, representing the number of prototypes in each SOM, has not been discussed yet. Different values of n_1 (n_2) lead to different segmentations of the regressors (deformation) space. This means in fact different models of the time series since the conditional distribution in the transition matrix are constructed according to the segmentations obtained with the two SOMs.

In the following applications we use a simple validation procedure to fix n_1 and n_2. For that purpose, we divide the set of data in three parts: a learning, a validation and a test set. The learning set is used to fix the values of the model parameters, such as the prototypes in the SOMs and the frequencies in the transition matrix. The validation set is used to fix meta-parameters, such as n_1 and n_2. The test set aims at seeing how the model behaves on unused data that mimic real conditions. More elaborated validation procedures such as cross-validation, leave-one-out or bootstrap (all discribed in [19]) could be used to fix the meta-parameters, but no significant difference was observed in these cases on the two examples illustrated in this paper.

The selection of n_1 and n_2 must be made according to an error criterion. We have chosen a one-step ahead mean square error criterion computed over the validation set:

$$e_{MSE} = \sum_{x(t+1)\in ValidSet} (x(t+1) - \hat{x}(t+1))^2. \tag{41}$$

Note that once numbers n_1 and n_2 have been set, a new learning is done on the reassembled learning and validation sets. This new learning is only performed once for the selected model with optimal n_1 and n_2.

5.2. Scalar forecasting: Santa Fe A

The Santa Fe A time series [18] has been obtained from a far-infrared-laser in a chaotic state. This time series has become a well-known benchmark in time series prediction since the Santa Fe competition in 1991. The completed data set contains 10 000 data. This data set has been divided here into three parts: the first 6000 data in the learning set, the 2000 following ones in the validation set, and the 100 next ones in the test set. Though a larger training set is used here, the same time horizon ($h = 100$) as in the competition is used as it is a rather long-term horizon for this time series [18].

According to previous experiments on this series [18], the regressors have been constructed according to:

$$\begin{aligned} x_{t-6}^t = \quad & \{x(t), x(t-1), x(t-2), x(t-3), \\ & x(t-5), x(t-6)\}. \end{aligned} \tag{42}$$

In other words, in this example $d = 1$, $p = 6$ and $h = 100$. Note that $x(t-4)$ is missing from the regressor.

Kohonen strings of 1 to 200 prototypes in each space have been used. All the 40 000 possible models have been tested on the validation set. The best model among them has 179 prototypes in the regressor space and 161 prototypes in the deformation space. After re-learning this model on the joined learning and validation sets, 1000 simulations were performed on a horizon of 100 values. Then, the mean and confidence interval at 95% level were computed, giving information on the possible long-term evolution of the time

series. Figure 5 shows the mean of the 1000 simulations compared to the true values contained in the test set, together with the confidence interval at 95% level. Figure 6 shows a zoom on the first 30 values. In Figure 7, we can see 100 simulations picked up at random for the same 30 values. Note the stability obtained through the replications. For a simpler model with $n_1 = 6$ and $n_2 = 8$ (used for illustrations purposes), Figure 8 shows the code vectors and regressors (resp. deformations) in each cluster; Table 1 shows the corresponding transition matrix. As expected (see details in section 3.4), the transition matrix is sparse and most non-zero elements on a row are grouped together.

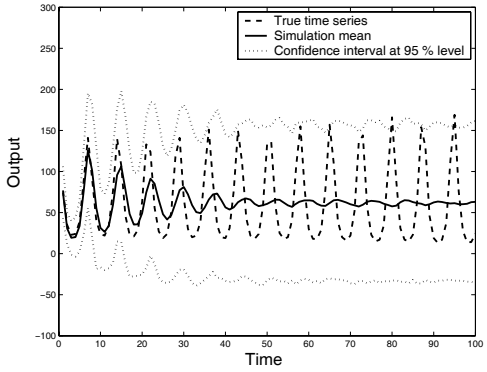


Figure 5. Comparison between the mean of the 1000 simulations (solid) and the true values (dashed), together with confidence intervals at 95% level (dotted).

From Figure 6, it should be noted that the method gives roughly the first 25 values of the time series, a result that is not so far from those usually obtained with other neural network models [18]. One exception is the winner of the Santa Fe A competition, a specifically designed Time Delay Neural Network [18].

Although the confidence intervals give less information at long term, Figure 5 illustrates the conclusion of the stability proof: despite the fact

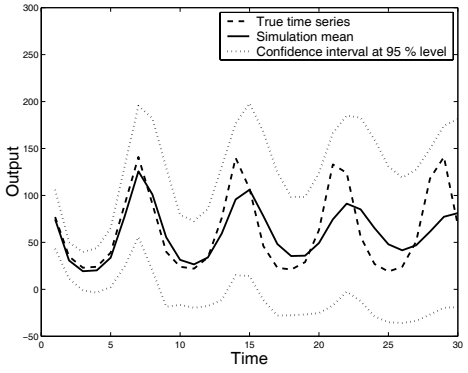


Figure 6. Comparison for the first 30 values between the mean of the 1000 simulations (solid) and the true values of the test set (dashed), together with confidence intervals at 95% level (dotted).

that at some horizon this (as any) model will not provide any useful information anymore, the prediction will remain stationary and within the scope of the original series.

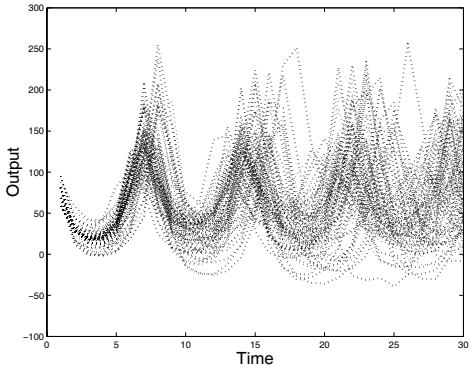


Figure 7. 100 simulations picked out at random from the 1000 simulations made for the Santa Fe A long-term forecasting.

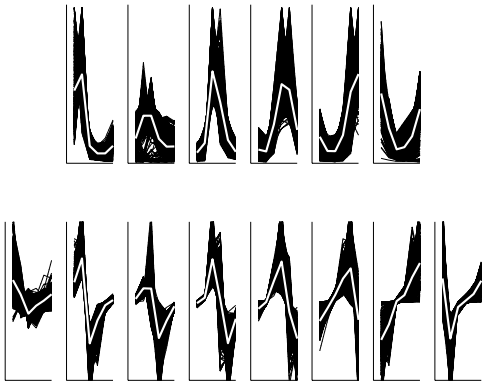


Figure 8. The code vectors and associated curves in the regressor (top) and deformation (bottom) spaces (when $n_1 = 6$ and $n_2 = 8$). The code vectors are represented in white as 6-dimensional vectors (according to (42)). Regressors (resp. deformations) belonging to each cluster are shown in black.

0.12	0	0	0	0	0	0.23	0.66
0.67	0.30	0	0	0	0	0.02	0.01
0.05	0.55	0.40	0	0	0	0	0
0.03	0	0.30	0.54	0.13	0	0	0
0	0	0	0	0.50	0.48	0.02	0
0.06	0	0	0	0.0	0.34	0.56	0.04

Table 1
Example of transition matrix, here with $n_1 = 6$ and $n_2 = 8$ as in Figure 8. Note that in each row, the frequency values sum to one.

5.3. Vector forecasting: the Polish electrical consumption

This second example concerns the Polish electrical consumption time series [20]. This series contains hourly consumption values from 1989 to 1996. The whole dataset contains about 72 000 hourly data and is plotted in Figure 9. Due to the daily periodicity of the time series, we are interested in daily predictions. This is a vectorial case with $d = 24$, since it seems natural to forecast the 24 next values (the next day) in one step.

The 3000 x_{t-23}^t data of dimension 24, considered without any preprocessing, have been divided in three sets as follows: we use the first 2000 data the learning set, the next 800 ones the validation set and the last 200 the test set. Since the optimal regressor is unknown, many different regressors were tried, using intuitive understanding of the process. The final regressor that has been selected is:

$$x_{t-p+1}^t = \{x_{t-23}^t, x_{t-47}^{t-24}, x_{t-71}^{t-48}, x_{t-167}^{t-144}, x_{191}^{t-168}\}. \quad (43)$$

This regressor contains the 24 hourly values of today, yesterday, of two, six and seven days ago. While there is no proof that this regressor could be the optimal one, it is the one that makes the lowest error (41) on the validation set. Since the regressor contains 5 data of dimension $d = 24$, we work in a 120-dimensional space. The algorithm is run on the learning set with values for n_1 and n_2, each varying from 5 to 200 prototypes by steps of 5. The lowest error is made by a model with $n_1 = 160$ and $n_2 = 140$ respectively.

As above, a new learning is performed for the model with $n_1 = 160$ and $n_2 = 140$ with a new learning set now containing 2000+800 data. Then 1000 complete forecasts are performed with this model. Figure 10 presents the mean of the 1000 simulations obtained with 24-dimensional vectors and with horizon h limited to 40 days (a sin-

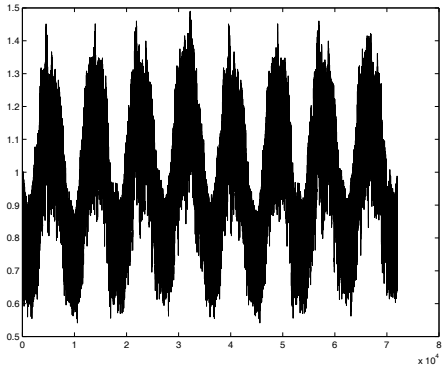


Figure 9. The Polish electrical consumption time series, between 1989 and 1996.

gle plot of the whole 24 * 200 predicted values becomes unreadable). For convenience, Figure 11 shows a zoom and a comparison between the mean of those 1000 long-term predictions and the real values. A confidence interval at 95% level is also provided.

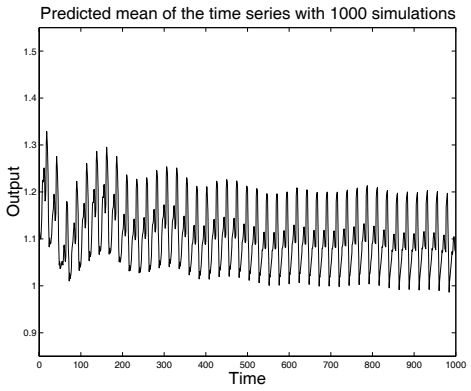


Figure 10. Mean of the 1000 simulations at long term ($h = 40$).

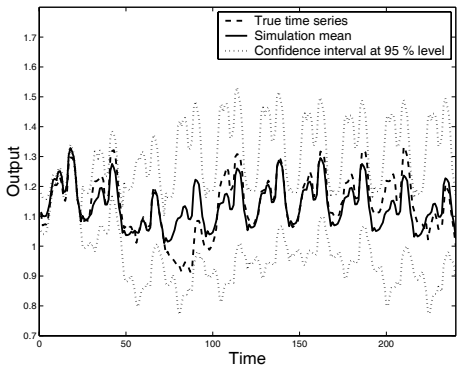


Figure 11. Comparison between the true values (dashed), the mean of the predictions (solid) and the confidence interval at 95% level (doted).

From Figure 11, it is clear that the mean of the predictions at long term will show the same periodicity as the true time series and that the values will be contained in a rather narrow confidence interval. This fact denotes a probable low variability of the series at long term. This also extends the good results for short-term forecasting obtained in [1], [20] or [21], for a single day ahead, to a much longer term.

Figure 12 shows 100 predictions picked up at random from the Monte-Carlo procedure. It is visible that most simulations have almost the same shape; this is a major argument for having some ideas about the long-term evolution of the series.

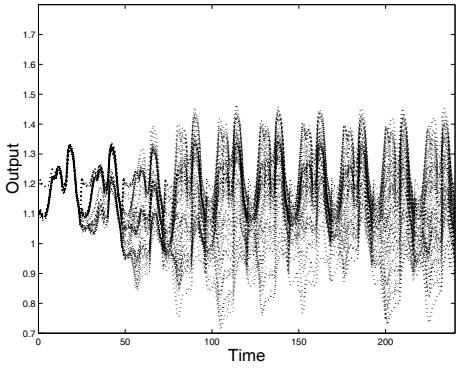


Figure 12. Plot of 100 simulations chosen at random from the 1000 simulations performed on the series.

6. Conclusion

In this paper, we have presented a time series forecasting method based on a double use of the SOM algorithm respectively in the original space (containing the regressors of the time series) and in the deformation space (containing the deformation regressors). The links between the two SOMs are characterized by a transition matrix. The stochastic behaviour of the method

in the forecasting stage allows repeating the simulations using a Monte-Carlo procedure. These repetitions make it possible to compute statistics (mean, variance, confidence intervals, etc.) on the long-term predictions.

The utilization of the SOM helps the method to reach the goal of forecasting a vector of values with the same expected accuracy on each of its components, making the method applicable to vectorial time series forecasting. Spatial and temporal vectors are possible; it is possible to use the vectorial method to predict several scalar series together or to predict several future values of a scalar series in one block.

A stability concept is defined for long-term forecasts. According to this definition, this paper includes a detailed proof of the method stability.

This method could be used in many contexts. This paper illustrates its utilization on a standard benchmark (Santa Fe A series) and on a real-world problem of electrical load forecasting. The method can easily be used on other applications like financial series. Future work includes experiments on series of spatial vectors (interest rates, biomedical signals such as electroencephalograms, etc.). From a methodological point of view, the properties of the transition matrix may be further studied using the theory of Markov chains. Finally, the compromise between the length of the predicted vectors and the number of recurrences needed to reach a specific time horizon may be studied both theoretically and experimentally.

Acknowledgements

We would like to thank Professor Osowsky from Warsaw Technical University for providing us the Polish Electrical Consumption data used in our example. The authors are grateful to the reviewers for their helpful comments and in-depth reading of the manuscript.

REFERENCES

1. M. Cottrell, B. Girard, P. Rousset, *Long Term Forecasting by Combining Kohonen Algorithm and Standard Prevision*, in Proc. of International Conference on Artificial Neural Networks ICANN'97, October 1997, Lausanne (Switzerland), W.Gerstner, A.Germond, M.Hasler, J.D.Nicoud Eds., Lecture Notes in Computer Science, n 1327, Springer, p. 993-998.
2. T. Kohonen, *Self-organising Maps*, Springer Series in Information Sciences, Vol. 30, Springer, Berlin, 1995.
3. E. de Bodt, M. Cottrell, P. Letremy, M. Verleysen, *On the use of Self-Organizing Maps to accelerate vector quantization*, Neurocomputing, Vol. 56, pp. 187-203, Elsevier, 2003.
4. M. Cottrell, J.-C. Fort, G. Pagès, *Theoretical aspects of the SOM algorithm*, Neurocomputing, Vol. 21, pp. 119-138, Elsevier, 1998.
5. M. Cottrell, E. de Bodt, M. Verleysen, *Kohonen maps versus vector quantization for data analysis*, in Proc. of European Symp. on Artificial Neural Networks, April 1997, Bruges (Belgium), D-Facto pub. (Brussels), pp. 187-193.
6. M. Cottrell, E. de Bodt, Ph. Grégoire, *Simulating Interest Rate Structure Evolution on a Long Term Horizon: A Kohonen Map Application*, in Proc. of Neural Networks in The Capital Markets, Californian Institute of Technology, World Scientific Ed., Pasadena, 1996.
7. J. Walter, H. Ritter, K. Schulten, *Non-linear prediction with self-organising maps*, in Proc. of IJCNN, San Diego, CA, 589-594, July 1990.
8. J. Vesanto, *Using the SOM and Local Models in Time-Series Prediction*, in Proc. of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, pp. 209-214, 1997.
9. S. Dablemont, G. Simon, A. Lendasse, A. Ruttiens, F. Blayo, M. Verleysen, *Time series forecasting with SOM and local nonlinear models - Application to the DAX30 index prediction*, in Proc. of Workshop on Self-Organizing Maps WSOM'03, September 2003, Kitakyushu (Japan), pp. 340-345.
10. T. Martinetz, S. Berkovich, K. Schulten, *Neural-gas network for vector quantization and its application to time-series prediction*, IEEE Trans. on Neural Networks, Vol. 4, No. 4, pp. 558-569, 1993.

11. M. Cottrell, B. Girard, P. Rousset, *Forecasting of curves using a Kohonen classification*, Journal of Forecasting, Vol. 17, pp. 429-439, 1998.
12. T. Voegtlin, *Recursive Self-Organizing Maps*, Neural Networks, Vol. 15, No. 8-9, pp. 979-991, 2002.
13. G. J. Chappell, G. J. Taylor, *The temporal Kohonen map*, Neural Networks, Vol. 6, pp. 441-445, 1993.
14. T. Koskela, M. Varsta, J. Heikkonen, K. Kaski, *Recurrent SOM with Local Linear Models in Time Series Prediction*, in Proc. of European Symp. on Artificial Neural Networks, April 1998, Bruges (Belgium), D-Facto pub. (Brussels), pp. 167-172.
15. J. Kangas, *On the Analysis of Pattern Sequences by SelfOrganizing Maps*, Ph. D. thesis, Helsinki University of Technology, 1994
16. M. Verleysen, E. de Bodt, A. Lendasse, *Forecasting financial time series through intrinsic dimension estimation and non-linear data projection*, in Proc. of International Workconference on Artificial and Natural Neural networks (IWANN'99), Springer-Verlag Lecture Notes in Computer Science, n 1607, pp. **II**596-**II**605, June 1999.
17. G. Fayolle, V. A. Malyshev, M. V. Menshikov, *Topics in constructive theory of countable Markov chains*, Cambridge University Press, 1995.
18. A. S. Weigend, N. A. Gershenfeld, *Times Series Prediction: Forecasting the future and Understanding the Past*, Addison-Wesley Publishing Company, 1994.
19. B. Efron, R. J. Tibshirani, *An introduction to the bootstrap*, Chapman & Hall, 1993.
20. S. Osowski, K. Siwek, L. Tran Haoi, *Short term load forecasting using neural networks*, in Proc. of III Ukrainian-Polish Workshop, Aluszta, Krym 2001, pp. 72-77.
21. A. Lendasse, M. Cottrell, V. Wertz, M. Verleysen, *Prediction of Electric Load using Kohonen Maps - Application to the Polish Electricity Consumption*, in Proc. of American Control Conference ACC 2002, June 2002, Anchorage, Alaska (USA), pp. 3684-3689.