Double quantization forecasting method for filling missing data in the CATS Time Series

Geoffroy Simon, John Lee, Michel Verleysen, Marie Cottrell

IJCNN'2004 proceedings – International Joint Conference on Neural Networks
Budapest (Hungary), 25-29 July 2004, IEEE, pp. 1635-1640

Double Quantization Forecasting Method for Filling Missing Data in the CATS Time Series

Geoffroy Simon, John A. Lee, Michel Verleysen
Machine Learning Group
Université catholique de Louvain
Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium
E-mail: simon,lee,verleysen@dice.ucl.ac.be

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

***Abstract*— The double vector quantization forecasting method based on Kohonen self-organizing maps is applied to predict the missing values of the CATS Competition data set. As one of the features of the method is the ability to predict vectors instead of scalar values in a single step, the compromise between the size of the vector prediction and the number of repetitions needed to reach the required prediction horizon is studied. The long-term stability of the double vector quantization method makes it possible to obtain reliable values on a rather long-term forecasting horizon.**

I. Introduction

Time series forecasting can be defined as the problem of determining in advance the future values of a given time series. This problem is of major interest in many fields as, e.g., finance (forecasting returns or stock markets), hydrology (predicting river floods), engineering (estimating future electrical consumption) or even in management (anticipating work load). Many methods have been developed to solve this problem with very different approaches, from statistics to system identification and more recently neural networks. All these methods are usually classified in some general categories: (N)AR(X), (N)ARMA(X), FIR, BJ, etc. Whatever the original problem and the used method may be, the methodological approach is always the same: Given a time series data set, one tries to find a suited model of this series. This model is then used to forecast the future evolution of the series. In this paper a NAR-type model will be used, i.e. a non-liner (N) auto-regressive (AR) model, based on Kohonen maps. The X part is omitted since no exogenous information will be used.

The CATS Competition Data Set is a time series prediction problem. The goal of this competition is to be able to predict 100 values corresponding to five holes of twenty values in the series, the fifth one being at the end. As each hole is preceded by 980 known values one approach to solve the competition problem would be to fit five models, each one being specifically designed to predict the next twenty values. In this paper however, a global model able to predict the 100 missing values will be presented.

The model used here is derived from the double vector quantization method (DVQ) [1]. This method is based on the use of two Kokonen's self-organizing maps (SOM) [2] in parallel. The SOM is a tool usually used in classification or feature extraction tasks, but more scarcely in time series prediction despite a few previous attempts [5], [6], [7], [8], [9], [10]. The DVQ method has been recently proved to be stable for long-term time series prediction [1]. Considering that 20 values is a longer-term horizon than the usual one step ahead prediction framework, the method will be used here to forecast the missing values of the CATS data set.

The DVQ method is also able to predict vectors of values instead of scalar ones: in the temporal context, vectors mean to predict several forecasts in a single operation, rather than repeating scalar forecasts and using predictions to predict the next values. The size of the prediction vectors is a parameter of the method. In the CATS competition context where 20 consecutive values have to be predicted, we have the choice to predict a vector of 20 values, to repeat 20 times a scalar forecast (in this case using the prediction at each step to predict the next one), or any intermediate situation (4 times a vector of 5 values, etc.).

In the following of this paper, we first present an analysis of the CATS data set, the conclusions of this analysis guiding the experiments. In section III we briefly recall some basic concepts about the SOM maps and then present the forecasting method for the scalar case (for the sake of simplicity). Section IV is devoted to the description of the experimental methodology and section V presents the obtained results. A discussion will conclude this paper.

II. Analysis of the CATS data set

As mentioned in the introduction, the model implemented by the DVQ method is a NAR model. An important question to answer is the choice of the AR part, i.e. the size of the regressor. More formally, having at our disposal a time series of $x(t)$ values with $1 \leq t \leq n$, the prediction problem can be defined as follows :

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

where d is the size of the vector of values to be predicted, f is the model of the data generating process, p is the number of past values to consider, θ are the model parameters and ε_t is the noise. The past values are gathered in a p-dimensional vector called *regressor*. Both p and d must be chosen. As mentioned above, d will result from a compromise between

IJCNN'2004 proceedings – International Joint Conference on Neural Networks
Budapest (Hungary), 25-29 July 2004, IEEE, pp. 1635-1640

scalar predictions with repetitions and a larger vector of values to predict as a whole; in practice the choice of d will be determined by extensive simulations.

Concerning the size p of the regressor, it is possible to have insights about a plausible value (or range of values) by an in-depth examination of the series. In this paper, the search for the regressor size p is based on Grassberger-Proccacia's [11] *correlation dimension*; the procedure is for example summarized in [12].

Grassberger-Proccacia's procedure allows to estimate the correlation dimension D_c [11] of a time series. Then, according to Takens theorem [13], a regressor of size $p = 2 * D_c + 1$ will describe the data in an embedding space containing enough information to allow a correct modeling of the series.

In short, the Grassberger-Procaccia procedure computes the correlation dimension D_c according to:

$$D_c = \lim_{r \to 0} \frac{ln(C_m(r))}{ln(r)}, \tag{2}$$

where $C_m(r)$ is the *correlation integral* [11] defined as:

$$C_m(r) = \lim_{n \to \infty} \frac{2}{n(n-1)} \sum_{1 \leq t < t' \leq n} I(\|x(t) - x(t')\| \leq r). \tag{3}$$

Function $I(.)$ takes a value equal to 1 if its expression into parenthesis is true, and 0 otherwise.

Intuitively, the idea in relation (2) is to count the number of points $x(t')$ in a hyper sphere centred in $x(t)$ with radius r. The limit when n tends to ∞ is taken in relation (3), i.e. the definition is given for an infinite number of data (in the series). Then, the ratio between the log of this number of points and the corresponding radius is observed, as the radius tends to zero. In other words ones tries to count the number of data that are at a distance of at most r one from another, given an infinite number of data, while considering smaller and smaller values for r. In practice of course, we do not have an infinite number of data at our disposal. Therefore the left and right parts of the $ln(C_m(r))$ against $ln(r)$ diagram will not be reliable, so that the most informative slopes between those extremes in the diagram have to be identified.

Figure 1 shows the results obtained with the 4900 known values of the CATS time series. With respect to Grassberger-Proccacia's procedure described above, the data are now the p-dimensional regressors defined in (1). As mentioned in [12] the correlation dimension is given by the slope in the linear part of the curves. When the size of the data space (size p of the regressor) increases, it will reach the dimension where it is effectively possible to compute the correlation dimension (obviously, working in a too low dimensional space does not allow to estimate a large dimension!). As this dimension is unknown, the experience must be carried out for increasing dimensions of the data space; when the required level is reached and above, the estimated correlation dimensions will remain identical (i.e. the curves will be parallel).

Figure 1 shows a plot of $ln(C_m(r))$ against $ln(r)$ for increasing dimensions of the data space (i.e. increasing sizes p of the regressors). The expected saturation effect explained in [12] is clear for values of $ln(r)$ between 5 and 8; the correlation dimension seems to be around 1.

Another representation of the correlation dimension can be given in plotting the estimation of the correlation dimension as in Figure 2. A flat region can be seen around $ln(r) = 6$ to 7 where the correlation dimension is again approximately one. In conclusion, according to Taken's theorem, any regressor for the CATS time series should be at most of size 3.

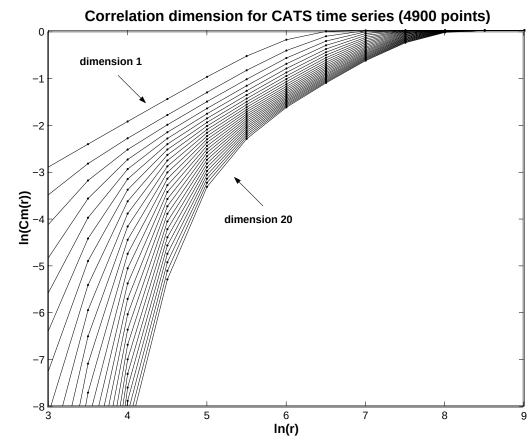

Fig. 1. Estimation of the correlation dimension using the Grassberger-Procaccia procedure, log of the correlation integral $C_m(r)$ against the log of the hyper-sphere radius r.

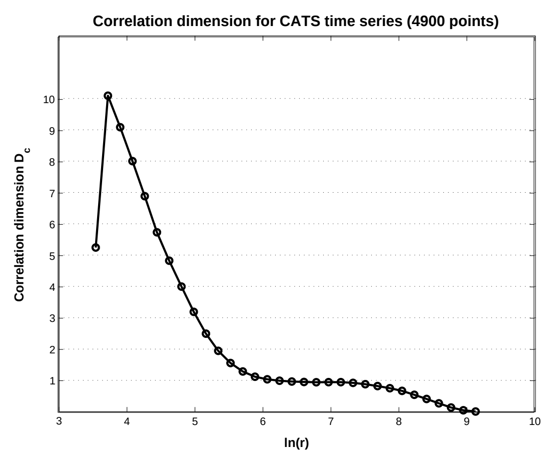

Fig. 2. Correlation dimension obtained for various values of the hyper sphere radius r (in log scale).

Note that this correlation dimension estimation is only a preliminary rough calculation, in order to get a first insight on the series. Indeed, because of the high correlation between successive values in the series (or in other words its 'smoothness'), it may happen that the correlation dimension estimation just catches this correlation, and not the dynamics of the series. On Figure 1, this could be seen in the form of two saturation effects in the slopes, as detailed above: one when the correlation between successive values is catch, the other one when the true dimensionality is. According to the very low value (one) found for the correlation dimension in the CATS series, this risk doesn't have to be underestimated. Nevertheless, as no other reasonable value can be found, we will consider in the following that the value found for the correlation dimension is reliable, and a regressor of size 3 will thus be used.

The problem of a time series with a very low correlation dimension is that each value only depends of the few preceding ones. Any model built according to the above principles is therefore restricted to a very limited amount of information, and the prediction becomes hard and unstable. This is for example the case in financial time series prediction: the high sampling frequency of financial indexes makes them extremely smooth at short term. In such context, one usually model pre-processed series instead of the original ones; the pre-processing can consist in differences, returns, etc. Because of the similarities between the correlation dimension results on such financial series and on the CATS one, the same kind of pre-processing is developed here. In addition to the original series, two pre-processed ones will be used in the experiments: the series obtained by differences and by returns. The difference time series is obtained as:

$$x_d(t) = x(t+1) - x(t), \tag{4}$$

while the return time series is computed as:

$$x_r(t) = \frac{x(t+1) - x(t)}{x(t)}. \tag{5}$$

The correlation dimension of these two new time series can also be computed, using the same Grassberger-Procaccia procedure. It must be mentioned that the results obtained in these cases are not conclusive (no visible saturating slope in the $ln(C_m(r))$ against $ln(r)$ diagrams). The results found on the original series will thus be kept as a rough estimation.

III. The double quantization forecasting method

A. Self-organizing Maps

The Self-Organizing Map (SOM) is an unsupervised classification algorithm introduced in the 80's by Teuvo Kohonen [2]. Since its first description, Self-Organizing Maps have been applied in many different fields to solve various problems. Their theoretical properties are well established [3], [4].

In a few words, a SOM map has a fixed number of units quantifying the data space. Those units, also called prototypes or centroids, are linked by predefined neighbourhood relationships that can be represented graphically through a 1- or 2-dimensional grid. After learning, the grid of prototypes has two properties. First, it defines a vector quantization of the input space, as any other vector quantization algorithm. Secondly, because the grid relationships are used in the learning algorithm itself, the grid representation has a topological property: two close inputs will be projected on either the same or close prototypes in the grid. The Kohonen map can thus be seen as an unfolding procedure, or a nonlinear projection from the data space on a 1- or 2-dimensional grid. The prototypes in a Kohonen map can also be seen as representatives of their associated class (the set of data nearer from a specific prototype than from any other one), turning the algorithm into a classification (or at least a clustering) tool. One of the main features of Kohonen maps is their ability to easily project data in a 2-dimensional representation, allowing intuitive interpretations.

B. The double vector quantization method (DVQ)

Though the SOM are usually considered as a classification, feature extraction or recognition tool, there exist a few works where SOM are used in time series prediction problems, as [5], [6], [7], [8], [9], [10]. In most of these situations however, the goal is to reach a reliable one step ahead prediction. In this work we are specifically looking for longer-term ones, and more precisely to 20 steps ahead prediction in the context of the CATS Competition.

A complete description of the DVQ method is given in [1], together with a full proof of the method stability for long-term predictions. A brief description of the method will be given here in the simple case of a scalar time series prediction. Full details for the vector case can be found in [1].

The goal of the method is to extract long-term information or trends of a time series. The method is based on the SOM algorithm used to *characterize* (or *learn*) the past of the series. Afterwards a *forecasting* step allows to predict future values.

1) Characterization: According to the formulation of a nonlinear auto-regressive model (1), the method uses regressors of past values to predict the future evolution of a time series. Having at disposal a scalar time series of n values, the correlation dimension D_c is evaluated, leading to the choice of p-dimensional regressors. The n known values of the time series are then transformed into p-dimensional regressors:

$$x_t = \{x(t-p+1), \ldots, x(t-1), x(t)\}, \tag{6}$$

where $p \leq t \leq n$, and $x(t)$ is the original time series at our disposal. As one may expect $n-p+1$ such regressors are obtained from the original time series.

The original regressors x_t are then manipulated such that other regressors are created, according to:

$$y_t = x_{t+1} - x_t. \tag{7}$$

The y_t vectors are called the *deformation* regressors, or the *deformations* in short. By definition each deformation y_t is associated to a single regressor x_t. Of course, $n-p$ deformations are obtained from a time series of n values.

At this stage of the method there exist two sets of regressors. The first one contains the x_t regressors and is representative of the original space (of regressors). The space containing the y_t deformations is representative of the deformation space. Those two sets of vectors, of the same dimension p, will be the data manipulated by the SOM maps.

Applying the SOM algorithm to each one of these two sets results in two sets of prototypes, denoted respectively $\bar{x}_i$, with $1 \leq i \leq n_1$, and $\bar{y}_j$, with $1 \leq j \leq n_2$. The classes associated to those prototypes are denoted respectively c_i and c'_j.

IJCNN'2004 proceedings – International Joint Conference on Neural Networks
Budapest (Hungary), 25-29 July 2004, IEEE, pp. 1635-1640

Characterizing the two time series through the quantization of the regressors and deformations is a static-only process. The dynamics of the past evolution of the series has to be modelled too. In fact, this is possible because the dynamics is implicitly recorded in the deformations. The issue is thus to build a representation of the existing relations between the original regressors and the deformations. For this purpose, a matrix $f(ij)$ is defined according to:

$$f_{ij} = \frac{\#\{x_t \in c_i \text{ and } y_t \in c'_j\}}{\#\{x_t \in c_i\}}, \tag{8}$$

with $1 \leq i \leq n_1$, $1 \leq j \leq n_2$. Intuitively the probability of having a certain deformation j associated to a given regressor i is approximated by the empirical frequencies (8) measured on the data at disposal. Each row of the $f(ij)$ matrix ($1 \leq j \leq n_2$) in (8) is in fact the conditional probability that y_t belongs to c'_j given the fact that x_t belongs to c_i. Of course, elements f_{ij} ($1 \leq j \leq n_2$) sum to one for each i.

2) Forecasting: Now that the past evolution of the time series has been modelled, predictions can be performed. Let us define the last known value $x(t)$ at time t, with corresponding regressor x_t. The prototype $\bar{x}_k$ closest to x_t in the original space is searched. According to the conditional probability distribution defined by row k, a deformation prototype $\bar{y}_l$ is then chosen randomly among the $\bar{y}_j$, according to the f_{kj} probability law. The prediction for instant $t+1$ is finally obtained according to relation (7):

$$\hat{x}_{t+1} = x_t + \bar{y}_l, \tag{9}$$

where $\hat{x}_{t+1}$ is the estimate of x_{t+1} given by the model. In fact $\hat{x}_{t+1}$ is a p-dimensional vector, and only one of its components corresponds to a prediction $\hat{x}(t+1)$ at time $t+1$; this value is thus extracted from the $\hat{x}_{t+1}$ vector and taken as the prediction.

Once a one step ahead prediction (horizon $h = 1$) is computed, the whole procedure can be repeated to obtain predictions for higher values of h. In practice, prediction $\hat{x}(t+1)$ is used to compute $\hat{x}_{t+2}$ through its corresponding regressor $\hat{x}_{t+1}$. $\hat{x}(t+2)$ is then extracted from $\hat{x}_{t+2}$, and so on up to horizon h. This recursive procedure is the standard way to obtain long-term forecasts from a one step ahead method. The whole procedure up to horizon h is called a *simulation*.

3) Comments: The goal of the DVQ method is to provide insights over the possible long-term evolution of a series, and not necessarily a single accurate prediction. The long-term (horizon h) simulations are then repeated using a Monte-Carlo procedure. The simulations distribution can be observed, and statistical information such as variance, confidence intervals, etc can be determined too. The obtained long-term predictions have been proven to be stable [1].

Another important comment is that the method can easily be generalized to the prediction of vectors. With respect to the procedure described in the previous subsection, the only difference is that deformations (7) must be computed by differences of d-spaced values:

$$y_t = x_{t+d} - x_t, \tag{10}$$

a direct generalization of the $d = 1$ case in (7). Then, d scalar values have to be extracted from the $\hat{x}_{t+d}$ vector, and so on. For example, two values could be extracted (corresponding to $\hat{x}(t+1)$ and $\hat{x}(t+2)$). In this case, repeating the procedure means to inject $\hat{x}(t+1)$ and $\hat{x}(t+2)$ to predict $\hat{x}(t+3)$ and $\hat{x}(t+4)$. More details about the vector case can be found in [1].

A third comment concerns the numbers n_1 and n_2 of prototypes respectively in the regressor and deformation spaces. The major concern is that different values of n_1 (n_2) lead to different segmentations of the regressor and the deformation spaces which in turn lead to different models of the time series. Many possibilities can therefore be considered for constants n_1 and n_2 and only an optimal one, in terms of model adequateness with regards to the time series, should be kept.

Finally, since the only property of the SOM used here is the vector quantization, any other vector quantization method could have been chosen to implement the above procedure. The SOM maps have been chosen since they seem more efficient and faster compared to other VQ methods despite a limited complexity [14]. Furthermore, they provide an intuitive and helpful graphical representation. Note that in practice any kind of SOM map could be used, but that one-dimensional maps, or strings, are preferred here.

IV. Methodological aspects of the double quantization for the CATS data set

As mentioned in section III-B the goal of the DVQ method is to provide insights over the possible long-term evolution of a series, and not necessarily a single accurate prediction. In this section the methodology for the experiments will be described having in mind that the method has now to predict accurate values.

A. Scalar and vector predictions

From section II we know that regressor x_t for nonlinear models should contains at most 3 past values:

$$x_t = \{x(t-2), x(t-1), x(t)\}. \tag{11}$$

As this expression has the same form as relation (6), it allows a direct application of the DVQ method on those regressors to predict $x(t+1)$. This direct application of the method is an illustration of the scalar prediction with the DVQ method.

Now it should be explained how the method can predict vectors in the particular case of the CATS data set. The natural approach is thus to consider that each value should be predicted using its last 3 past values. As a consequence if one wants to predict, for example, a vector of $d = 2$ values, namely $\{\hat{x}(t+1), \hat{x}(t+2)\}$, the following regressors should be used:

$$\{x(t-2), x(t-1), x(t)\} \text{ to predict } \hat{x}(t+1), \tag{12}$$

$$\{x(t-1), x(t), x(t+1)\} \text{ to predict } \hat{x}(t+2). \tag{13}$$

In order to make possible the use of a vector prediction method as DVQ, it is suggested to merge the two regressors and use:

$$\{x(t-2), x(t-1), x(t), x(t+1)\} \tag{14}$$

IJCNN'2004 proceedings – International Joint Conference on Neural Networks
Budapest (Hungary), 25-29 July 2004, IEEE, pp. 1635-1640

to predict $\{\hat{x}(t+1), \hat{x}(t+2)\}$. Of course this is impossible as $x(t+1)$ is unknown at time t. Using the vector prediction property of DVQ, the size (four) of regressor (14) will be kept, but the last four known values will be used:

$$\{x(t-3), x(t-2), x(t-1), x(t)\} \tag{15}$$

instead of (14) to predict $\{\hat{x}(t+1), \hat{x}(t+2)\}$ as a single vector. All these operations can be performed easily using the DVQ method. Indeed it only suffices to compute y_t according to:

$$\begin{aligned} y_t &= x_{t+d} - x_t \\ &= \{x(t-1), x(t), x(t+1), x(t+2)\} \\ &\quad -\{x(t-3), x(t-2), x(t-1), x(t)\}. \end{aligned} \tag{16}$$

The above description illustrates the $d = 2$ case; vectors of predictions of size $d > 2$ can of course be considered too. Of course, as the series is known until time t only, equation (16) is only applied $t - d$ times.

To summarize, the DVQ method is directly applicable in the scalar case. Some care must be taken in the vector case: if vectors of d values have to be predicted then the corresponding regressors have to be merged into a single vector. Only then, the DVQ method in vector case can be applied.

B. 20 step ahead prediction strategies

As the DVQ method can be applied in the vector case the influence of the prediction time horizon in the particular case of the CATS data set can be observed. At least two alternatives can be depicted: the *recursive strategy* and the *bloc strategy*. The first one is the usual strategy that allows to predict recursively the values until the final horizon h, using the last prediction $\hat{x}(t+k)$ to predict the next one $\hat{x}(t+k+1)$. The second one allows to predict all the h future values in one single vector. A mixed approach would be a *recursive-bloc* strategy, where blocs of intermediate size d are predicted through a limited recursive procedure of h/d steps (where h is supposed to be a multiple of d for simplicity).

C. Number of prototypes

As mentioned in section III-B, numbers n_1 and n_2 of prototypes in respectively the regressor and the deformation spaces have to be fixed. A cross-validation procedure is therefore used. This cross-validation procedure mimics the competition problem. Fifteen new holes of length 20 are created randomly in the available data. As the true values are known for those 300 new missing values they can serve as validation set for models learned on the remaining values. Twenty such validation sets are constructed to avoid any bias that could appear due to the random choice of the validation data.

To compare the different models that will be learned on the various learning sets a mean square error MSE validation criterion is used. This criterion is comparable to the one proposed in the CATS competition and is defined as:

$$MSE = \frac{\sum_{y_t \in VS} (y_t - \hat{y}_t)^2}{\#VS}, \tag{17}$$

where VS represents a validation set of 300 new missing values. The best model will be the one which has the lowest average MSE over the 20 validations sets.

D. Final predictions

Once the optimal n_1 and n_2 numbers are found, a new learning stage is done using now all the available data (i.e. combining the previous learning and validation sets). To avoid problems due to the random initialisation of the prototypes, several learnings are performed, and the best one is selected according to the validation sets, even if using the latter may led to a small amount of overfitting. Simulations at horizon $h = 21$ are then repeated 100 times, and the mean is computed.

To refine this first result, a specific heuristic is developed by reversing the time series. Indeed, for the four blocks of length 20 inside the series, the prediction can be performed from right to left (decreasing values of time). For those four blocks, the CATS Competition is a missing value problem rather than a forecasting one. Again, simulations at horizon $h = 21$ are repeated 100 times, and the mean value is taken.

The final predictions are derived from the two sets of simulations. For the first four blocks of 20 missing values, predicting up to horizon $h = 21$ (instead of $h = 20$) makes it possible to compare the 21st value to the true (known) one. As some error in long-term trend of the prediction is unavoidable, this error can be compensated at first order through a linear correction of the simulations making the 21st value equal to the true one. This is done both for the original and reverse order simulations. Finally, the mean of the two sets of linearly corrected values, in original and reverse order, is taken. This constitutes the final prediction.

V. Experimental results

According to the ’financial-like’ behaviour of the CATS time series, as discussed in section 2, three time series are considered in all our experiments: the initial CATS, the differences and the returns time series. Furthermore, this ’financial-like’ behaviour already suggests that a recursive strategy may behave poorly for a time horizon of 20 values. Consequently, in addition to the recursive strategy, where predictions are repeated 20 times, a bloc strategy is used, with blocs of size 2, 5, 10 and 20. The time horizon of 20 values therefore correspond to predict 10 blocs of size $d = 2$, 4 blocs of size $d = 5$, etc.

For each one of the three studied time series, for each one of the considered bloc size, a cross-validation using the 20 validation sets has been performed. For comparison purposes the new missing values in the 20 validation sets are the same for each experiment (i.e. each time series and each bloc size). Models with n_1 and n_2 both ranging from 5 to 100 by incremental step of 5 are learned in each experiment. The MSE criterion (17) has been used to estimate the models generalization ability on the validation sets.

Table I gives a summary of the experiments. For each time series, for each bloc size, n_1 and n_2 corresponding to the best model in average are given, together with its average

MSE. For the differences and returns series the MSE is of course computed by coming back to the original values (the inverse transformations are applied on the predictions before computing the MSE on the validation sets).

Time series	# step(s) ahead	n_1	n_2	MSE
Initial	1	90	5	1.66 10^3
	2	50	5	1.32 10^3
	5	25	5	1.36 10^3
	10	20	15	1.70 10^4
	20	65	10	2.98 10^4
Differences	1	25	5	2.59 10^3
	2	90	5	1.90 10^3
	5	80	5	1.86 10^4
	10	55	5	4.67 10^4
	20	55	60	7.43 10^4
Returns	1	10	5	3.39 10^5
	2	5	5	2.04 10^5
	5	55	5	1.83 10^5
	10	15	95	2.67 10^{10}
	20	45	55	4.01 10^{10}

TABLE I

EXPERIMENTS SUMMARY: n_1 AND n_2 FOR THE BEST MODEL IN AVERAGE OVER THE 20 CROSS-VALIDATIONS AND CORRESPONDING MSE.

From this table it seems clear that a model learned on the initial time series is adequate. This rather surprising fact could be explained as follows. The deformations computed according to relation (7) are in fact the differences time series. Thanks to its definition this method thus allows in one single computation to model both the initial time series and the differences time series. None of the two pre-processing used here has proved to be relevant in this particular case of the CATS data set.

Furthermore it is obvious that there exists a compromise between the 20 repetitions of a one step ahead prediction (recursive strategy) and a single prediction of a vector containing the 20 next values (bloc strategy). This compromise seems to be somewhere between 10 predictions of blocs of 2 values and 4 predictions of blocs of 5 values. Nowadays the MSE criterion is the lowest for blocs of size $d = 2$. The corresponding model, with 50 prototypes in the regressors space and 5 in the deformations space, is selected to give the final prediction of the 100 missing values of the CATS Competition according to the heuristic described in section IV-D.

VI. CONCLUSION

In this paper the double vector quantization method, based on the SOM maps, has been applied to the CATS data set.

An analysis of the data shows some interesting aspects of the time series. Its correlation dimension seems to be as low as one. To take into account this particular aspect potentially limiting for nonlinear models other time series have been defined, i.e. the differences and the returns of the initial CATS series.

These three time series have been modelled using various size for blocs of predictions corresponding to longer-time horizons, in order to take the most of the vector prediction ability of the double vector quantization method.

The number of units in the SOM maps has been discussed and selected using a cross-validation procedure on new holes created randomly on the CATS data set. This procedure, together with the chosen validation criterion, has been implemented to select the best model in average in conditions as close as possible to the Competition ones.

An heuristic specifically designed in the CATS Competition context is also described.

ACKNOWLEDGEMENTS

G. Simon is funded by the Belgian F.R.I.A., M. Verleysen is Senior Research Associate of the Belgian F.N.R.S.

REFERENCES

[1] G. Simon, A. Lendasse, M. Cottrell, J.-C. Fort, M. Verleysen, ”Double SOM for Long-term Time Series Prediction”, in *Proc. of WSOM’03*, Kitakyushu (Japan), pp. 35-40, 2003.

[2] T. Kohonen, *Self-organising Maps*, 2nd ed, Springer Series in Information Sciences, Vol. 30, Springer, Berlin, 1995.

[3] M. Cottrell, J.-C. Fort, G. Pagès, ”Theoretical aspects of the SOM algorithm”, *Neurocomputing*, 21, pp. 119-138, 1998.

[4] M. Cottrell, E. de Bodt, M. Verleysen, ”Kohonen maps versus vector quantization for data analysis”, in *Proc. of ESANN’97*, Bruges (Belgium), D-Facto pub. (Brussels), pp. 187-193, 1997.

[5] J. Walter, H. Ritter, K. Schulten, ”Non-linear prediction with self-organising maps”, in *Proc. of IJCNN*, San Diego, CA (USA), pp. 589-594, July 1990.

[6] J. Vesanto, ”Using the SOM and Local Models in Time-Series Prediction”, in *Proc. of WSOM’97*, Espoo (Finland), pp. 209-214, 1997.

[7] T. Koskela, M. Varsta, J. Heikkonen, K. Kaski, ”Recurrent SOM with Local Linear Models in Time Series Prediction”, in *Proc. of ESANN’98*, Bruges (Belgium), D-Facto pub. (Brussels), pp. 167-172, 1998.

[8] 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.

[9] M. Cottrell, B. Girard, P. Rousset, ”Forecasting of curves using a Kohonen classification”, *Journal of Forecasting*, Vol. 17, pp. 429-439, 1998.

[10] S. Dablemont, G. Simon, A. Lendasse, A. Ruttiens, F. Blayo, M. Verleysen, ”Time series forecasting with SOM and local non-linear models - Application to the DAX30 index prediction”, in *Proc. of WSOM’03*, Kitakyushu (Japan), pp. 340-345, 2003.

[11] P. Grassberger, I. Procaccia, ”Measuring the strangeness of strange attractors”, *Physica*, Vol. D9, pp. 189-208, 1983.

[12] F. Camastra, A. M. Colla, ”Neural Short-Term Prediction Based on Dynamics Reconstruction”, *Neural Processing Letters*, n. 9, Vol. 1, pp. 45-52, 1999.

[13] F. Takens, ”Detecting strange attractors in turbulence”, in *Dynamical Systems and turbulence*, Lecture Notes in Mathematics **898**, Springer-Verlag, 1981.

[14] E. de Bodt, M. Cottrell, P. Letremy, M. Verleysen, ”On the use of Self-Organizing Maps to accelerate vector quantization”, accepted for publication in *Neurocomputing*, Elsevier.