

Forecasting the CATS benchmark with the Double Vector Quantization method

Geoffroy Simon^{a,*}, John A. Lee^{a,2}, Marie Cottrell^b, Michel Verleysen^{a,b,3}

^aUniversité catholique de Louvain, Machine Learning Group-DICE Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium

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

Available online 26 February 2007

Abstract

The Double Vector Quantization (DVQ) method, a long-term forecasting method based on the self-organizing maps algorithm, has been used to predict the 100 missing values of the CATS competition data set. An analysis of the proposed time series is provided to estimate the dimension of the auto-regressive part of this nonlinear auto-regressive forecasting method. Based on this analysis experimental results using the DVQ method are presented and discussed. As one of the features of the DVQ method is its ability to predict scalars as well as vectors of values, the number of iterative predictions needed to reach the prediction horizon is further observed. The method stability for the long term allows obtaining reliable values for a rather long-term forecasting horizon.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Time Series prediction; CATS competition; Double Vector Quantization method

1. Introduction

Time series prediction is a problem encountered in many fields: from engineering (predictive control of industrial processes) to finance (forecasting returns of shares or stock markets) this general problem has already been studied by a large community of researchers. Models and prediction methodologies have been proposed by statisticians, mathematicians and engineers, as well as people from econometrics and more recently from the neural network community. Whatever the time series, whatever the method used to predict the series, the methodology always first consists in constructing a model of the time series. This model is then used to predict the future of the series. In this paper, this general methodology will be applied, using a nonlinear auto-regressive model, namely the Double Vector Quantization (DVQ) method [14].

This paper concerns the prediction of the CATS benchmark proposed as a time series competition at IJCNN 2004. As explained in the descriptive paper [13], the CATS competition prediction problem consists in predicting 100 missing values distributed in five gaps of 20 data points within a time series of 4900 data points. Each gap is preceded by 980 known values. The data set starts with 980 known data and the last gap is at the end of the time series. In this paper, a global model able to predict the 100 missing values is developed. To set the parameters of the model (number of prototypes in the quantization steps, choice of the preprocessing and of the block prediction horizon), an extended cross-validation procedure is used.

The DVQ nonlinear auto-regressive [14] method is based on Kohonen's self-organizing maps (SOMs) [11]. Two SOMs are used and linked by a stochastic model. The two SOMs are used here as a clustering tool for prediction even if the SOM algorithm is usually considered as a classification or feature extraction tool. Nevertheless, some previous attempts have already tried to use the SOM algorithm for time series prediction. For example, Cottrell et al. [3] uses SOMs to create clusters in the regressor space, Walter et al. [18] and Vesanto [16] associate each cluster to a linear local model and Dablemont et al. [7] associates each cluster to a

*Corresponding author. Tel.: +32 10 47 80 61; fax: +32 10 47 25 98.

E-mail address: simon@dice.ucl.ac.be (G. Simon).

URL: <http://www.dice.ucl.ac.be/~gsimon/> (G. Simon).

¹G. Simon is funded by the Belgian F.R.I.A.

²J. A. Lee is a Scientific Research Worker of the Belgian F.N.R.S.

³M. Verleysen is a Research Director of the Belgian F.N.R.S.

nonlinear one. Another approach is to split the problem into the predictions of, respectively, a normalized curve and the curve mean and standard deviation [6]. Using the recursive SOMs approach [17] (and pioneer work on leaky integrators [2]) one tries to learn sequences of data, as applied in [10] for speech recognition problems. Recursive SOMs can be further combined with local linear models, as in [12]. The DVQ method is rather different from these previous works since it aims at predicting long-term trends instead of providing short-term accurate predictions. Furthermore, as long-term predictions are the main concern, a theoretical proof of the DVQ stability at long-term has been proposed recently [14]. Considering that a gap of 20 values is a long-term horizon when compared to the usual one-step-ahead framework, the DVQ method is used to forecast the 100 missing values of the CATS data set.

Another original aspect of the DVQ method is its ability to predict vectors of values in a single step. This generalization from scalar to vector cases will be used and studied on the CATS data set: the DVQ method can recursively predict 20 times a one-step-ahead scalar forecast, or a vector of 20 values in one single prediction step, or any other intermediate situation. Note that, as a consequence, the size of the prediction vectors is a parameter of the DVQ model. This parameter will be optimized by cross-validation in the experiments.

In the following sections of this paper, we first present an analysis of the CATS data set. In this analysis the correlation dimension of the CATS time series is estimated. The goal of this analysis is to determine the size of the regressors for the DVQ nonlinear auto-regressive model. In Section 3 a short reminder on the basic concepts about the SOMs is followed by the description of the forecasting method for the scalar case. Its extension to vectors is briefly sketched, and some general comments on the DVQ method and its use in practice are given. Section 4 is devoted to the description of the experimental methodology that has been specifically developed and used for the CATS series. Section 5 presents the results, before a concluding discussion.

2. Analysis of the CATS data set

As mentioned in the Introduction, the model implemented by the DVQ method is a NAR model. As usual with (N)AR models, the key point is to determine the order of the AR part, i.e. to evaluate the size of the regressor. More formally, having at disposal a time series of $x(t)$ values with $1 \leq t \leq n$, predicting the values for $t > n$ can be defined as follows:

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

where d is the size of the vector of values to be predicted, f is the model of the data generating process (considered here as a nonlinear one), p is the number of past values to be

considered, 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 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 value p of the regressor size, 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–Procaccia’s [9] correlation dimension; the procedure is for example summarized in [1].

Grassberger–Procaccia’s procedure allows estimating the correlation dimension D_c [9] of a time series. Then, according to Takens’ theorem [15], a regressor of size $p = 2 * D_c + 1$ will describe the data in an embedding space containing enough information to allow a modeling of the manifold described by the time series values.

In short, the Grassberger–Procaccia procedure computes the correlation dimension D_c of vector data x_t according to:

$$D_c = \lim_{r \rightarrow 0} \frac{\ln(C(r))}{\ln(r)}, \quad (2)$$

where $C(r)$ is the *correlation integral* [9] defined as follows:

$$C(r) = \lim_{n \rightarrow \infty} \frac{2}{n(n-1)} \sum_{1 \leq t < t' \leq n} I(\|x_t - x_{t'}\| \leq r). \quad (3)$$

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

Intuitively, the idea in relation (3) is to count the number of data $x_{t'}$ in a hypersphere centered in x_t with radius r . This operation is repeated for each data x_t . Then the limit for n tends to ∞ is taken, i.e. the definition is given for an infinite number of data x_t (in the series). Finally, the ratio between the log of this number of data and the log of the corresponding radius is observed in relation (2), as the radius r tends to zero. In other words while estimating the correlation integral (3), one tries to count the number of data $x_{t'}$ that are at most at distance r from x_t , given an infinite number of data. The correlation integral thus expresses the asymptotic proportion of data pairs whose distance is less than r , with respect to the total number $n(n-1)/2$ of pairs. The correlation dimension (2) is therefore obtained as the limit of the correlation integral values obtained when the radius r is decreased to zero. As mentioned in [1] the correlation dimension is given by the slope in the linear part of the curves. This follows the initial assumption by Grassberger and Procaccia [9] that the correlation integral $C(r)$ behaves as a power law of r for small values of r . In other words, $C(r) \approx r^{D_c}$. The correlation dimension is therefore obtained as the slope of the $D_c * \ln(r)$ curve.

In practice of course, we do not have an infinite number of data at our disposal. Therefore, the left and right parts of the $\ln(C(r))$ against $\ln(r)$ diagram will not be reliable, so

that the most informative slopes between those extremes in the diagram have to be identified. When the size of the data space 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 or exceeded, the estimated correlation dimensions will remain identical (i.e. the curves will be parallel, which can be observed on the curves as a saturation effect).

Fig. 1 shows the results obtained with the 4900 known values of the CATS time series. With respect to Grassberger–Procaccia’s procedure described above, the data x_t are now the p -dimensional regressors defined in (1). The figure shows a plot of $\ln(C(r))$ against $\ln(r)$ for increasing dimensions of the data space (i.e. increasing sizes p of the regressors). The expected saturation effect [1] 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 Fig. 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 Takens’ theorem, any regressor for the CATS time series should be at most of size 3.

Note that the correlation dimension estimation is only a preliminary rough calculation, in order to get a first insight into 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. In Fig. 1, this could be seen in the

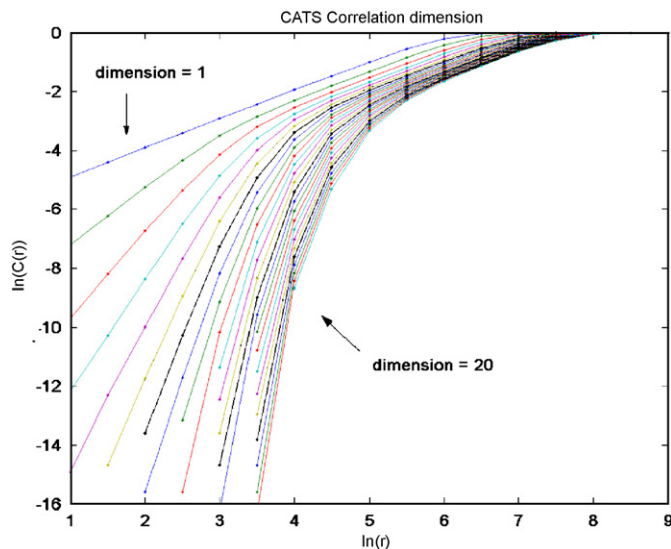


Fig. 1. Estimation of the correlation dimension using the Grassberger–Procaccia procedure; log of the correlation integral $C(r)$ versus the log of the hypersphere radius r .

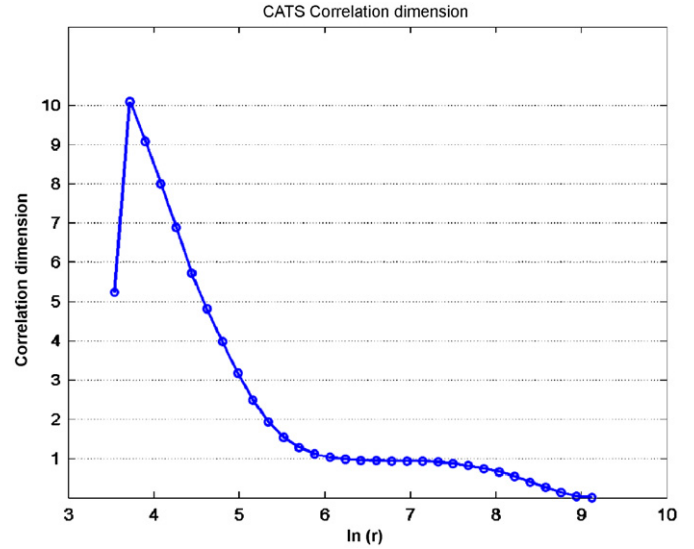


Fig. 2. Correlation dimension obtained for various values of the hypersphere radius r (in log scale) for the CATS time series.

form of two saturation effects in the slopes, as detailed above: one when the correlation between successive values is caught, the other one when the true dimensionality of the series is reached. According to the very low value (one) found for the correlation dimension in the CATS series, this risk does not 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 on 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 models preprocessed series instead of the original ones; the preprocessing 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 preprocessing is developed here. In addition to the original series, two preprocessed ones will be used in the experiments: the series obtained by differences and by returns. The difference time series is obtained as follows:

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

while the return time series is computed as follows:

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

The correlation dimension of these two new time series can also be computed, using the same Grassberger–Procaccia procedure. The results are presented in Figs. 3 and 4.

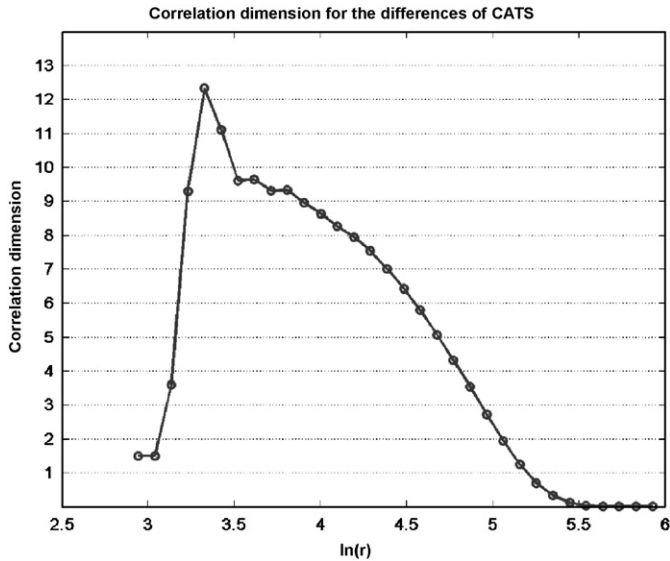


Fig. 3. Correlation dimension obtained for various values of the hypersphere radius r (in log scale) for the difference time series.

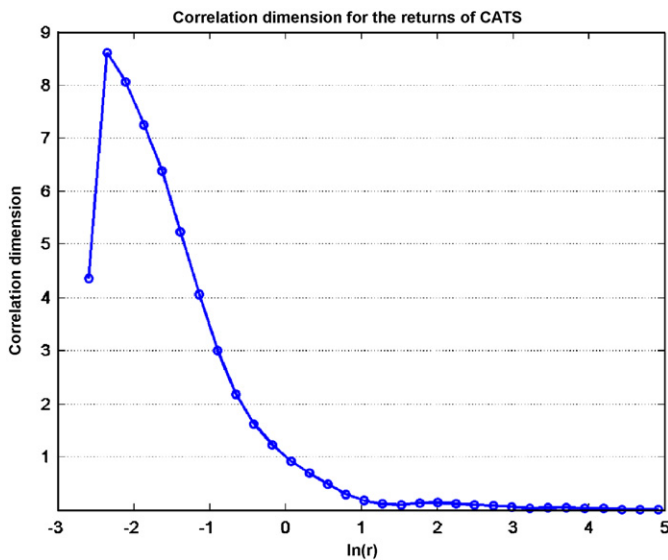


Fig. 4. Correlation dimension obtained for various values of the hypersphere radius r (in log scale) for the return time series.

Unfortunately, in both cases, the results are not conclusive. Indeed there is no visible saturating slope in the $\ln(C(r))$ against $\ln(r)$ diagrams, contrary to the saturation in Fig. 2. The results found on the original series will thus be kept in the following as a rough estimation of the correlation dimension of the CATS time series.

3. The double quantization forecasting method

3.1. Self-Organizing Maps

The SOM is an unsupervised classification algorithm introduced in the 1980s by Kohonen [11]. Since their first description, SOMs have been applied in many different

fields to solve various problems. Their theoretical properties are well established [5,4].

In short, a SOM map has a fixed number of units quantifying the data space. Those units, also called prototypes or centroids, are linked by predefined neighborhood relationships that can be represented graphically through a one- or two-dimensional grid. The grid of prototypes has two properties. First, it defines a vector quantization (VQ) of the input space, as any other VQ 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 either on the same prototype or on two close ones in the grid. The Kohonen map can thus be seen as an unfolding procedure or as a nonlinear projection from the data space on a one- or two-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 two-dimensional representation, allowing intuitive interpretations.

3.2. The double vector quantization (DVQ) method

Though the SOM is usually considered as a classification, feature extraction or recognition tool, there exist a few works where the SOM algorithm is used in time series prediction problems, as [3,6,7,12,16,18]. 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 [14], together with a full proof of the method stability for long-term predictions. A brief description of the method is given here in the simple case of a scalar time series prediction. Full details for the vector case can be found in [14]. 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 predicting future values.

3.2.1. Characterization

According to the formulation of a nonlinear autoregressive 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

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

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

$$y_t = x_{t+1} - x_t. \quad (7)$$

The y_t vectors are called the deformation regressors, or the deformations in short. By definition each deformation y_t is associated with 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 as, 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 by, respectively, c_i and c'_j .

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 modeled 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 such that its ij element, denoted f_{ij} , is obtained as follows:

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

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 .

3.2.2. Forecasting

Now that the past evolution of the time series has been modeled, 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):

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

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*.

3.2.3. 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 [14].

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:

$$y_t = x_{t+d} - x_t, \quad (10)$$

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 [14].

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. The values of n_1 and n_2 should therefore be optimized, for example, by resampling procedures (cross-validation, bootstrap) on a predefined error criterion, as the one-step-ahead error.

Finally, since the only property of the SOM used here is the VQ, any other VQ 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 [8]. 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.

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

As mentioned in Section 3.2 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 due to the competition context.

4.1. Scalar and vector predictions

From Section 2 we know that regressor x_t for nonlinear models should contain at most three past values:

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

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

The use of the correlation dimension to set the size of the regressor should however be taken with care when predicting vectors instead of scalars, as it will be the case for the CATS series. As an example, if one wants to predict a vector of $d = 2$ values, namely $\{\hat{x}(t+1), \hat{x}(t+2)\}$, the following two regressors should be, respectively, used:

$$\begin{aligned} \{x(t-2), x(t-1), x(t)\} & \text{ to predict } \hat{x}(t+1), \\ \{x(t-1), x(t), x(t+1)\} & \text{ to predict } \hat{x}(t+2). \end{aligned} \quad (12)$$

In order to use the DVQ method, it is suggested to merge the two regressors and use:

$$\{x(t-2), x(t-1), x(t), x(t+1)\} \quad (13)$$

to predict $\{\hat{x}(t+1), \hat{x}(t+2)\}$ together. Of course this is impossible since value $x(t+1)$ is unknown at time t . Therefore, the regressors that will be used in the following have the same size as in Eq. (13) but they are shifted such that their last component now corresponds to $x(t)$

$$\{x(t-3), x(t-2), x(t-1), x(t)\}. \quad (14)$$

In this example, deformations y_t are then computed according to

$$\begin{aligned} y_t &= x_{t+d} - x_t \\ &= \{x(t-1), x(t), x(t+1), x(t+2)\} \\ &\quad - \{x(t-3), x(t-2), x(t-1), x(t)\}. \end{aligned} \quad (15)$$

The same procedure can be extended for values of d greater than 2.

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 which may only contain known values. Only then, the DVQ method in vector case can be applied.

4.2. 20 step ahead prediction strategies

As the CATS competition requires predicting block of 20 successive unknown values, several strategies can be used to reach this prediction horizon. The first one, called the *recursive strategy*, is probably the most common way to obtain a long-term prediction. Predictions are obtained recursively until the final time horizon h ; i.e. the one-step-ahead predictions are included one by one in the regressor to obtain the next one-step-ahead prediction. Formally, the last prediction $\hat{x}(t+k)$ is used to predict the next one $\hat{x}(t+k+1)$ as part of the regressor for $t+k+1$ ($1 \leq k \leq d-1$).

The second approach, called the *block strategy*, consists in predicting all the h future values in one single vector. This is made possible by the use of a vector prediction method, as the DVQ one.

A mixed approach is a *recursive-block strategy*, where blocks of intermediate size d are predicted through a limited number of h/d recursive steps (where h is supposed to be a multiple of d for simplicity).

4.3. Number of prototypes

As mentioned in Section 3.2, the 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 gaps of length 20 have been created randomly in the available data. As the true values for those 300 new missing values are known they can serve as validation set for models learned on the remaining values. Note that the random selection was slightly constrained: the new gaps could not overlap between them and with the existing ones. The whole validation procedure has been repeated 20 times, to average the dependencies to the choice of the 300 new gaps.

To compare the different models that will be learned on the 20 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 follows:

$$MSE = \frac{\sum_{y_t \in VS} (y_t - \hat{y}_t)^2}{300}, \quad (16)$$

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

4.4. Final predictions

Once the optimal n_1 and n_2 numbers are found, a new learning stage is done now using all available data (i.e. the 4900 data of the original CATS time series).

To avoid problems due to the random initialization of the prototypes, several learning procedures are performed,

and the best one is selected according to the validation sets, even if using the latter may lead to a small amount of overfitting (as the validation sets are now part of the new learning set). Simulations at final horizon are then repeated 100 times, and the mean is computed.

To refine this direct application of the DVQ method to the CATS prediction problem, some specific heuristics have been developed. The first one is to reverse the time series. Indeed, for the four blocks inside the series, the prediction can be performed from right to left (decreasing values of time) as well as it is performed from left to right (chronological order). For those four blocks of length 20, the CATS Competition is a missing value problem rather than a forecasting one. The DVQ method will therefore be applied in both directions.

The second heuristic considers a prediction horizon up to $h = 21$ (instead of $h = 20$). Indeed, as the 21st value is known for the four first gaps, predicting to horizon $h = 21$ allows a comparison between the true value and the 21st predicted one. As some error in long-term trends of the prediction is unavoidable, the comparison between these true and predicted values leads to an error that can be compensated at first order through a linear correction of the simulated predictions. This correction is performed such that the 21st predicted value is made equal to the true one.

The final predictions that were sent to the competition were obtained by the DVQ method combined with the two heuristics explained above. More precisely, for the four first blocks of 20 missing values, a prediction to horizon $h = 21$ was performed in both directions, resulting from 100 DVQ simulations. As the true value for the 21st data is known in both directions for these four cases, a linear correction has been applied. The final predictions were then obtained as the average of the two linearly corrected predictions.

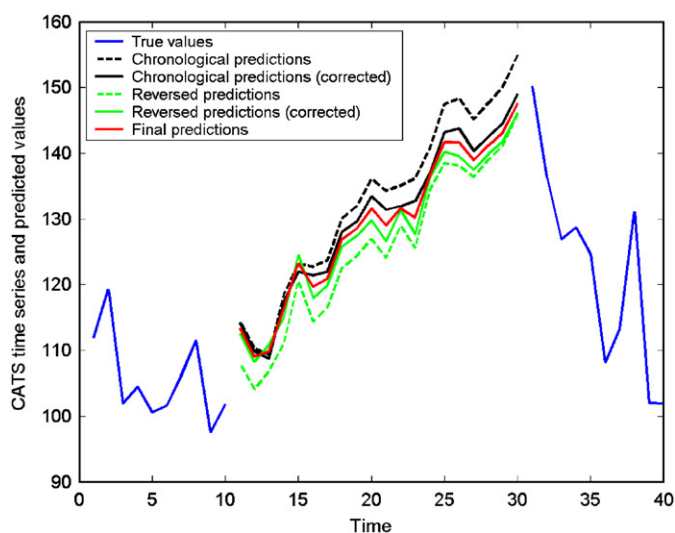


Fig. 5. Corrections applied to obtain the final predictions using the two heuristics (first block of missing values; data 981–1000). See text for details.

Of course, as the true 21st value is unknown for the fifth block of missing data, the above strategy cannot be applied. The final predictions for this case were obtained as the mean of 100 DVQ simulations.

Fig. 5 shows the various steps leading to the final predictions for the first block of missing values. The outermost curves are the mean of the 100 DVQ simulations (top of the figure is the chronological order, bottom is the reversed order). The two inner curves are the linearly corrected values using the comparison between the 21st true and predicted values. The fifth curve in the middle represents the final predictions, i.e. the mean of the two (linearly corrected) inner curves.

5. Experimental results

According to the ‘financial-like’ behavior of the CATS time series, as discussed in Section 2, three time series are considered in all our experiments: the initial CATS, the difference and the return time series. Furthermore, this ‘financial-like’ behavior already suggests that a recursive strategy may behave poorly for a time horizon of 20 values. Consequently, in addition to the recursive strategy, where one-step-ahead predictions are repeated 20 times, a recursive-block strategy is used, with blocks of size 2, 5 and 10, and finally a block strategy is used with a bloc size equal to the time horizon $h = 20$. In the recursive-block strategy, the time horizon of 20 values corresponds to predict 10 blocks of size $d = 2$, 4 blocks of size 5, etc.

For each one of the three time series and for each one of the block sizes, a cross-validation using the 20 validation sets has been performed as described in Section 4.3. For comparison purposes the new missing values in the 20 validation sets are identical in all experiments. Models with n_1 and n_2 both ranging from 5 to 100 by incremental steps of 5 are learned in each experiment. The *MSE* criterion (16) has been used to estimate the models generalization ability on the validation sets.

Table 1 gives a summary of the experiments. For each time series, for each block size, n_1 and n_2 corresponding to the best model in average are given, together with the average *MSE*. For the difference and return time series the *MSE* is of course computed by first applying the inverse transformations on the predictions, in order to obtain *MSE* values that can be compared between the series.

From this table it seems clear that a model learned on the initial time series is adequate; none of the two preprocessing methods suggested in (4) and (5) reveals anything interesting. Indeed the *MSE* values obtained on the validation sets are in both cases larger than those obtained with the initial time series.

Furthermore, it is obvious that there exists a compromise between 20 repetitions of a one-step-ahead prediction (recursive strategy) and a single prediction of a vector containing the 20 next values (block strategy). This

Table 1

Experiment summary: n_1 and n_2 for the best model in average over the 20 cross-validations and corresponding MSE

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}

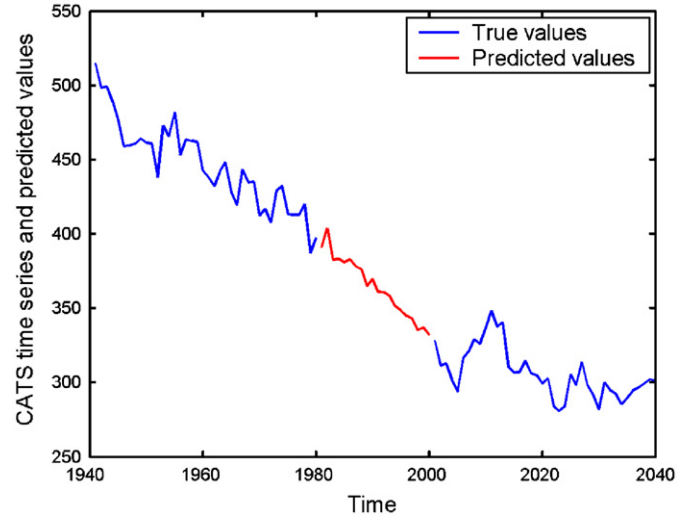


Fig. 7. True values and final predictions, second gap.

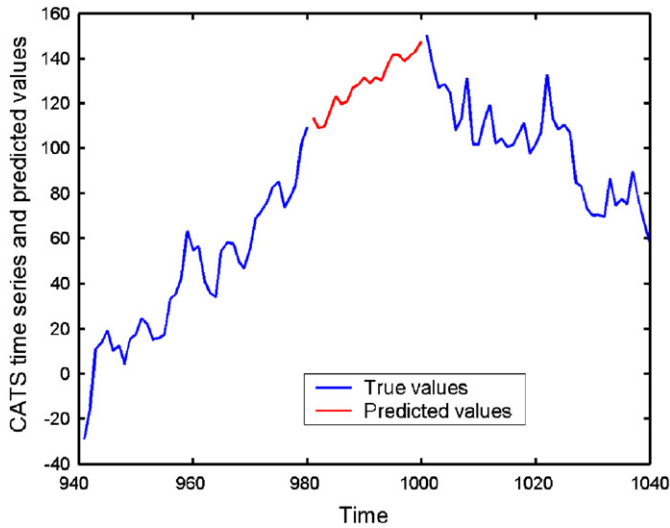


Fig. 6. True values and final predictions, first gap.

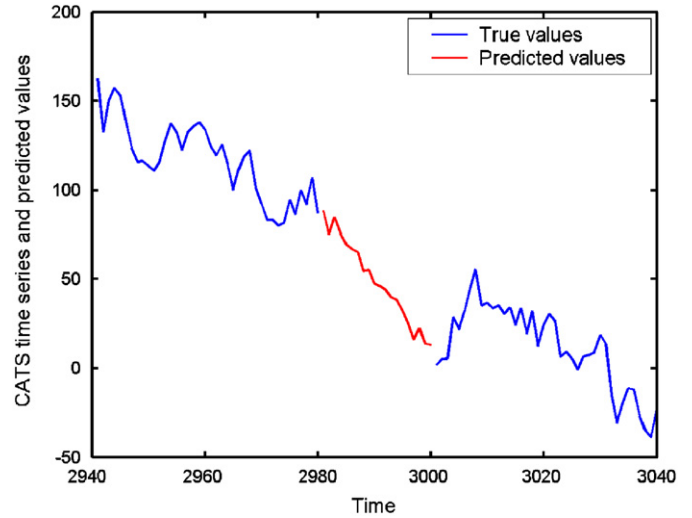


Fig. 8. True values and final predictions, third gap.

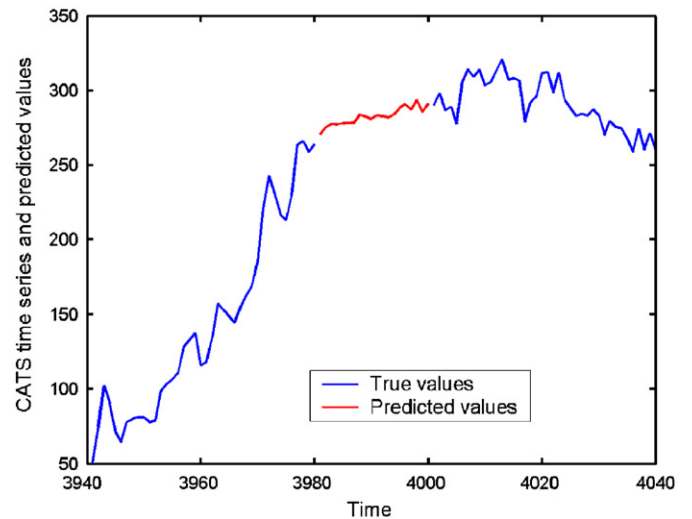


Fig. 9. True values and final predictions, fourth gap.

compromise seems to be somewhere between 10 predictions of blocks of 2 values and 4 predictions of blocks of 5 values. Nowadays the MSE criterion is the lowest for blocks of size $d = 2$. The corresponding model, with 50 prototypes in the regressor space and 5 in the deformation space, is selected to give the final prediction of the 100 missing values of the CATS Competition according to the heuristic described in Section 4.4. Figs. 6–10 show how the DVQ method predicts the missing values for the five 20-values blocks of the CATS competition.

After the CATS Competition was closed, the results of the 17 classified competitors out of the 24 submissions were made available [13]. It can be seen in [13] that the DVQ method was ranked to the fourth position on the problem

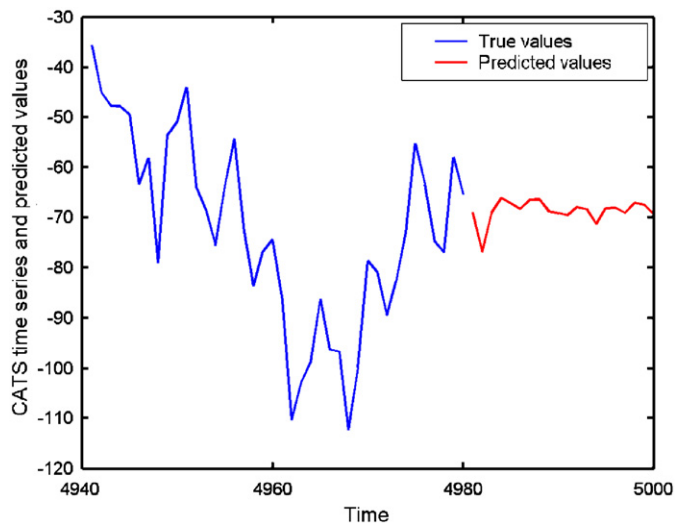


Fig. 10. True values and final predictions, fifth gap.

of predicting the first four gaps of 20 missing data. The value of the E_2 criterion in [13] (average sum of squares of errors on first 80 missing data) is 351. Besides the efficiency of the forecasting method, this is probably, and not surprisingly, a consequence of the fact that taking into account the first known value after each gap indeed improves the prediction accuracy. This information is not available for the fifth block of 20 missing data. The E_1 criterion in [13] (average sum of squares of errors on the 100 missing data) takes into account this very different problem. On this criterion, the DVQ method performs slightly worse with a result of 653, and is ranked in seventh position.

6. Conclusion

In this paper the results obtained with the DVQ method, based on the SOM maps, applied to the CATS data set are presented.

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 modeled using various sizes of prediction blocks corresponding to longer time horizons, in order to make the most of the vector prediction ability of the DVQ method.

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

Heuristics specifically designed in the CATS competition context are also described.

Finally, the predictions obtained using the best selected model and the heuristics are illustrated graphically.

Acknowledgments

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.

References

- [1] F. Camastra, A.M. Colla, Neural short-term prediction based on dynamics reconstruction, *Neural Process. Lett.* 9 (1) (1999) 45–52.
- [2] G.J. Chappell, G.J. Taylor, The temporal Kohonen map, *Neural Networks* 6 (1993) 441–445.
- [3] M. Cottrell, E. de Bodt, Ph. Grégoire, Simulating interest rate structure evolution on a long term horizon: a Kohonen map application, in: *Proceedings of Neural Networks in The Capital Markets*, Californian Institute of Technology, World Scientific, Pasadena, 1996.
- [4] M. Cottrell, E. de Bodt, M. Verleysen, Kohonen maps versus vector quantization for data analysis, in: *Proceedings of European Symposium on Artificial Neural Networks*, Bruges, Belgium, April 1997, D-Facto Pub, Brussels, pp. 187–193.
- [5] M. Cottrell, J.-C. Fort, G. Pagès, Theoretical aspects of the SOM algorithm, *Neurocomputing* 21 (1998) 119–138.
- [6] M. Cottrell, B. Girard, P. Rousset, Forecasting of curves using a Kohonen classification, *J. Forecast.* 17 (1998) 429–439.
- [7] 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: *Proceedings of Workshop on Self-Organizing Maps WSOM'03*, Kitakyushu, Japan, September 2003, pp. 340–345.
- [8] E. de Bodt, M. Cottrell, P. Letremy, M. Verleysen, On the use of self-organizing maps to accelerate vector quantization, *Neurocomputing* 56 (2004) 187–203.
- [9] P. Grassberger, I. Procaccia, Measuring the strangeness of strange attractors, *Physica D9* (1983) 189–208.
- [10] J. Kangas, On the analysis of pattern sequences by self organizing maps, Ph.D. Thesis, Helsinki University of Technology, Helsinki 1994.
- [11] T. Kohonen, *Self-organising Maps*, Springer Series in Information Sciences, vol. 30, Springer, Berlin, 1995.
- [12] T. Koskela, M. Varsta, J. Heikkonen, K. Kaski, Recurrent SOM with local linear models in time series prediction, in: *Proceedings of European Symposium on Artificial Neural Networks*, Bruges, Belgium, D-Facto Pub, Brussels, April 1998, pp. 167–172.
- [13] A. Lendasse, E. Oja, O. Simula, M. Verleysen, Time series prediction competition: the CATS benchmark, in: *Proceedings of IJCNN'2004*, IEEE, Budapest, Hungary, July 2004, pp. 1615–1620.
- [14] G. Simon, A. Lendasse, M. Cottrell, J.-C. Fort, M. Verleysen, Double quantization of the regressor space for long-term time series prediction: method and proof of stability, *Neural Networks* 17 (8–9) (2004) 1169–1181.
- [15] F. Takens, Detecting strange attractors in turbulence, in: *Dynamical Systems and Turbulence—Warwick, 1980*, Lecture Notes in Mathematics vol. 898, Springer, Berlin, 1981.
- [16] J. Vesanto, Using the SOM and local models in time-series prediction, in: *Proceedings of Workshop on Self-Organizing Maps (WSOM'97)*, Espoo, Finland, 1997, pp. 209–214.
- [17] T. Voegtlin, Recursive self-organizing maps, *Neural Networks* 15 (8–9) (2002) 979–991.
- [18] J. Walter, H. Ritter, K. Schulten, Non-linear prediction with self-organising maps, in: *Proceedings of IJCNN*, San Diego, CA, July, 1990, pp. 589–594.



Geoffrey Simon was born in 1978 in Dinant, Belgium. He received the M.Sc. degree in Computer Sciences in 2002 from the Facultés Universitaires Notre Dame de la Paix (Namur, Belgium). He is now working as Ph. D. student at the Microelectronic Laboratory of the Electrical Engineering Department of the Université catholique de Louvain (UCL). His research topics cover nonlinear time series analysis, artificial neural networks, nonlinear statistics and self-organization applied to time-series forecasting problems. He is currently member of the UCL Machine Learning Group. His work is funded by a grant from the Belgian F.R.I.A. (Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture).



John Aldo Lee was born in 1976 in Brussels, Belgium. He received the M.Sc. degree in Applied Sciences (Computer Engineering) in 1999 and the Ph.D. degree in Applied Sciences (Machine Learning) in 2003, both from the Université catholique de Louvain (UCL, Belgium). His main interests are nonlinear dimensionality reduction, intrinsic dimensionality estimation, independent component analysis, clustering and vector quantization. He is a former member of the UCL Machine Learning Group and is now a Scientific Research Worker of the Belgian F.N.R.S. (Fonds National de la Recherche Scientifique). His current work aims at developing specific image enhancement techniques for Positron Emission Tomography in the Molecular Imaging and Experimental Radiotherapy department of the Saint-Luc University Hospital (Belgium).



Marie Cottrell was born in 1943 in Béthune, France. She was a student at the Ecole Normale Supérieure de Sèvres, and received the Agrégation de Mathématiques degree in 1964 (with 8th place), and the Thèse d'Etat (Modélisations de réseaux de neurones par des chaînes de Markov et autres applications) in 1988. From 1964 to 1967, she was a High School Teacher. From 1967 to 1988, she was successively an Assistant and an Assistant Professor at the University of Paris and

at the University of Paris-Sud (Orsay), except from 1970 to 1973, during which she was a Professor at the University of Havana, Cuba. From 1989, she is a full Professor at the University Paris 1—Panthéon-Sorbonne. Her research interests include stochastic algorithms, large deviation theory, biomathematics, data analysis and statistics. Since 1986, her main works deal with artificial and biological neural networks, Kohonen maps and their applications in data analysis. She is the author of about 80 publications in this field. She is in charge of the SAMOS research group at the University Paris 1. She is regularly solicited as referee or international conference program committee member. In 2005, she organized the Fifth WSOM Conference in Paris.



Michel Verleysen was born in 1965 in Belgium. He received the M.S. and Ph.D. degrees in electrical engineering from the Université catholique de Louvain (Belgium) in 1987 and 1992, respectively. He was an Invited Professor at the Swiss E.P.F.L. (Ecole Polytechnique Fédérale de Lausanne, Switzerland) in 1992, at the Université d'Evry Val d'Essonne (France) in 2001, and at the Université Paris 1—Panthéon-Sorbonne in 2002, 2003 and 2004. He is now a Research Director of the Belgian F.N.R.S. (Fonds National de la Recherche Scientifique) and Lecturer at the Université catholique de Louvain. He is editor-in-chief of the Neural Processing Letters journal, chairman of the annual ESANN conference (European Symposium on Artificial Neural Networks), associate editor of the IEEE Transactions on Neural Networks journal, and member of the editorial board and program committee of several journals and conferences on neural networks and learning. He is author or co-author of about 200 scientific papers in international journals and books or communications to conferences with reviewing committee. He is the co-author of the scientific popularization book on artificial neural networks in the series "Que Sais-Je?", in French. His research interests include machine learning, artificial neural networks, self-organization, time-series forecasting, nonlinear statistics, adaptive signal processing, and high-dimensional data analysis.