Nonlinear Time Series Prediction by Weighted Vector Quantization

A. Lendasse[1], D. Francois[1], V. Wertz[1], and M. Verleysen[2]

Université catholique de Louvain
[1] CESAME, av. G. Lemaître 3, B-1348 Louvain-la-Neuve, Belgium
{lendasse, francois, wertz}@auto.ucl.ac.be,
[2] DICE, pl. du Levant 3, B-1348 Louvain-la-Neuve, Belgium
verleysen@dice.ucl.ac.be

Abstract. Classical nonlinear models for time series prediction exhibit improved capabilities compared to linear ones. Nonlinear regression has however drawbacks, such as overfitting and local minima problems, user-adjusted parameters, higher computation times, etc. There is thus a need for simple nonlinear models with a restricted number of learning parameters, high performances and reasonable complexity. In this paper, we present a method for nonlinear forecasting based on the quantization of vectors concatenating inputs (regressors) and outputs (predictions). Weighting techniques are applied to give more importance to inputs and outputs respectively. The method is illustrated on standard time series prediction benchmarks.

1 Introduction

Time series prediction is a problem with applications in various domains such as finance, electrical load and river flood forecasting, etc. The problem consists in predicting the next value of a series known up to a specific time, using the (known) past values of the series, and possibly exogenous data.

Classical methods (AR, ARMA, Box-Jenkins methodology, etc.) have been used with various successes for a while. In some cases, linear models are sufficient to predict series with a reasonable accuracy. In other cases, linear models reveal not sufficient, making the use of nonlinear ones necessary.

The potential increased performances of nonlinear models, when dealing with nonlinear processes, are balanced by their drawbacks what the learning concerns: number of learning parameters to adjust, increased computation times, convergence difficulties, overfitting, etc. There is thus a need to develop simple nonlinear models, with easy learning, a restricted number of user-adjusted parameters, but still showing high performances on datasets that are inherently non linear.

This paper presents a nonlinear regression method, applied in our case to the time series prediction problem, based on the completion of missing values and the quantization of concatenated vectors. Based on this general framework, different schemes are proposed to weight the respective importance of each variable (inputs and output) in the concatenated vectors to quantify.

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2657, pp. 417–426, 2003.

In the following of this paper, we will first present the Vector Quantization (VQ) problem, and show how it can deal with missing values (section 2). Next, we will show how VQ with missing values can be used for regression in general, and time series prediction in particular (section 3), including weighting of the inputs and outputs. In section 4, the weighting technique is applied to another non linear regression model, Radial-Basis Function Networks. The methods presented in the paper are illustrated in running examples inserted in sections 3 and 4 for clarity. Finally, section 5 draws some conclusions.

2 Vector Quantization

2.1 Definition

Vector quantization [3] is a way to summarize the information contained in a large database. Let us define a database through the following example:

Table 1. Database definition

	variable 1	variable 2	…	variable *j*	…	variable *D*
1st data	11	17	…	V_{1j}	…	87
2nd data	12	34	…	V_{2j}	…	7
3rd data	82	-32	…	V_{3j}	…	92
4th data	34	65	…	V_{4j}	…	42
…	…	…	…	…	…	…
*i*th data	V_{i1}	V_{i2}	…	V_{ij}	…	V_{iD}
…			…	…	…	…
*N*th data	-2	34	…	V_{Ni}	…	V_{ND}

Elements (observations, or data, or vectors) of this database are thus lines V_i ($1 \le i \le N$), with elements V_{ij} ($1 \le j \le D$). The principle of Vector Quantization (VQ) is to replace this database by another one, containing fewer vectors C_k of the same dimension D ($1 \le k \le M$, $M < N$). The number M of vectors C_k is a parameter of the method.

The new vectors C_k, called centroids, are deemed to have a distribution similar to the distribution of the initial V_i ones. Once the centroids are fixed through some algorithm, each vector V_i is then quantized by its "nearest-neighbor" centroid:

$$\mathrm{VQ}(V_i) = \arg\min_k \left(\|V_i - C_k\|^2 \right). \tag{1}$$

The Voronoi region associated to a centroid C_k is the region of the space associated to C_k by equation (1). In other words, it is the region of the space nearest to C_k than to any other centroid.

Determining the "best" set of centroids, i.e. the set that minimizes, in average (on all data contained in the original database), the quantization error defined by the distance in equation (1), is a nonlinear optimization problem. Many algorithms have been proposed in the literature to determine the centroids: batch algorithms, as Lloyd's one, or on-line adaptive ones, as the Competitive Learning method [5].

As an example, competitive learning works as follows. At each iteration, a data V_i is drawn randomly from the initial database. The nearest centroid C_* is determined (according to equation (1)) and moved (adapted) in the direction of data V_i:

$$C_* = C_* + \alpha(V_i - C_*), \tag{2}$$

where α is a time-decreasing adaptation factor. The same operation is repeated several times over the database.

Of course, all VQ algorithms find a set of centroids that correspond to a local minimum of the quantization error criterion, not a global one. Many initialization methods and many adaptation rules have been proposed in the literature, aiming to find a good compromise between the "quality" of the local minimum and the computational complexity (number of itaretions).

2.2 Missing Data

An easy-to-understand, but not so well known and used property of VQ is that databases with missing data can be quantized easily. Let us imagine that the initial database is now of the following form:

Table 2. Database with missing data

	variable 1	variable 2	…	variable *j*	…	variable *D*
1st data	11	17	…	V_{1j}	…	87
2nd data	?	34	…	?	…	7
3rd data	82	?	…	V_{3j}	…	92
4th data	34	65	…	V_{4j}	…	?
…	…	…	…	…	…	…
*i*th data	V_{i1}	V_{i2}	…	V_{ij}	…	V_{iD}
…			…	…	…	…
*N*th data	-2	?	…	V_{Ni}	…	V_{ND}

In Table 2, each '?' corresponds to a value that was not measured, or not memorized, for any reason; the '?'s are the missing values. The VQ process defined by equations (1) and (2) remains valid, if the computation of Euclidean distance in equation (1) is replaced by a distance computed only on the know variables and not on the missing ones (of course, it is assumed that the centroids have no missing values). Similarly, the adaptation of centroids by equation (2) is performed only using the corresponding existing coordinates of data V_i, and not on the missing ones.

Using VQ with missing data has proved to be efficient to find estimations of missing values in databases [4]. A missing value is replaced by the corresponding variable of the centroid nearest to the observation. This technique may be viewed as estimating a missing value with a function of the available variables in the same observation:

$$\hat{V}_{ij} = C_{kj}, \tag{3}$$

where C_k is the nearest centroid from V_i, according to equation (1) where the distance computation is restricted to the known coordinates of V_i. Using the known informa-

tion from V_i is certainly more appropriate than replacing a missing value by the average of the corresponding variable from other observations (average on a column of the database), as it is often the case when dealing with incomplete data. VQ may be viewed as a simple, but still efficient, way to use the information contained in the known coordinates of vector V_i.

3 Prediction with VQ and Missing Data

3.1 Quantization without Weighting

Time series prediction may be viewed as a regression problem when expressed as follows (without exogenous variables):

$$\hat{y}_t = g\left(y_{t-1}, y_{t-2}, ,..., y_{t-N+1}, \theta\right). \tag{4}$$

In this equation, $\hat{y}_t$ is the estimation of the unknown value of the time series at time t, while $y_{t-1},\ldots, y_{t-N+1}$ are the known N-1 past values of the series which form the *regressor*. θ is the set of parameters of the estimator g; the estimator can be linear or nonlinear in the general case.

If the past knowledge on the series is concatenated into N-dimensional vectors of the following form:

$$Y_t = \left[y_t \quad y_{t-1} \quad y_{t-2} \quad \cdots \quad y_{t-N+1}\right], \tag{5}$$

the vectors Y_t may be viewed as the vectors V_i from our initial database. VQ can be applied to these vectors. Next, the VQ model may be used on incomplete vectors, where the missing value is precisely the y_t one. Therefore, model g in equation (4) consists in the approximation of the missing value by the corresponding coordinate of the nearest centroid, as given by equation (3).

This method is illustrated in this paper on a well-known time series benchmark, the Mackey-Glass series [7]. The series has been generated according to

$$\frac{dy}{dt} = \beta y(t) + \frac{\alpha y(t-\delta)}{1+y(t-\delta)^{10}} \tag{6}$$

with β = -0.1, α = 0.2 and δ = 17. The series is sampled with a sampling period of 6. A part of the series is illustrated in Fig.1. 10000 points have been generated; the first 5000 ones are used for learning (VQ), the 5000 last ones for validation. Each regressor contains the four last values of the series, as recommended in the literature [8].

In order to validate the method, we define the Normalized Mean Square Error (*NMSE*) on the N_V data in the validation set as follows [9]:

$$NMSE = \sum_{t=1}^{N_V} (y_t - \hat{y}_t)^2 \Big/ \sum_{t=1}^{N_V} (y_t - \bar{y})^2 \tag{7}$$

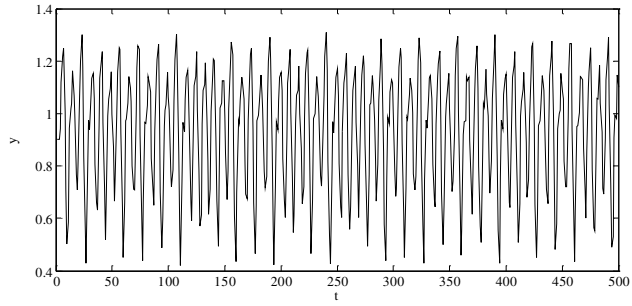

Fig. 1. Mackey-Glass series (see text for parameters).

Remind that the number of centroids is a parameter of the VQ method. The *NMSE* on the validation set with respect to this number of centroids is illustrated in Fig.2.

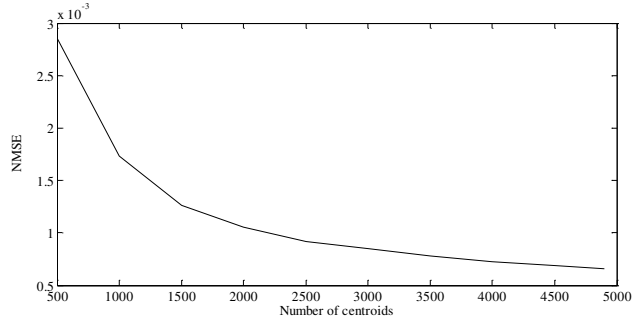

Fig. 2. *NMSE* of the Mackey-Glass series prediction with VQ.

Figure 2 surprisingly shows that the optimal number of centroids is as high as… the number of learning data! This result is not so surprising when considered the fact that data generated by equation (6) are noiseless. With so many centroids, the VQ method becomes equivalent to lazy learning [2], a method which consists in looking in the learning data the one closest from the regressor. Lazy learning has proved to be efficient in noiseless problems.

The same method, applied to the same series with added 0.25 variance noise, gives the results illustrated in Fig.3. The optimal number of centroids is now 20, and the optimal *NMSE* 0.30. For the sake of comparison, a linear model using the same regressor gives *NMSE*=0.36 (using the same learning and validation sets).

As a general comment, the number of centroids decreases with the amount of noise, as the VQ allows to "eliminate" the noise by averaging it in each Voronoi region. Of course, the price paid for this is an increased quantization error on the noiseless equivalent problem. This is nothing else than an illustration of the well-known bias-variance dilemma.

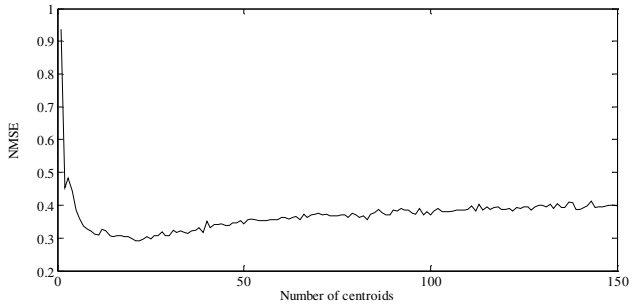

Fig. 3. *NMSE* of the Mackey-Glass with additive noise series prediction with VQ.

3.2 Quantization with Weighting of the Output

The method illustrated in the previous section has an obvious limitation: if the size of the regressor is large, the relative weight given to the y_t predictions in the quantization of vectors Y_t (equation (5)) decreases (it counts for one coordinate, with regards to N-1 coordinated for the regressor).

A straightforward extension of the method is then to give a different weight to the quantization of the y_t predictions by building the Y_t vectors as follows:

$$Y_t = [k y_t \quad y_{t-1} \quad y_{t-2} \quad \dots \quad y_{t-N+1}]. \tag{8}$$

Increasing k means that the VQ is biased to give more importance to the prediction. Parameter k may be determined in the same way as the number M of centroids in the VQ: by optimization on a validation set. Nevertheless, as the optimization is realized in practice by scanning the range of parameters, a double optimization can be time-consuming. It is therefore suggested to optimize the two parameters consecutively, possibly in an iterative way. This way of optimizing the parameters does not guarantee to find a global minimum for both of them, but is a reasonable compromise between the quality of the optimization and the computational load.

On the noisy Mackey-Glass series, starting with M=20 (the result found in the previous section for k=1), an optimization on k gives k=0.5. This value is then used for another optimization on M, which results in M=60 (see Fig. 4). Further optimizations do not improve the result anymore.

3.3 Quantization with Weighting of the Inputs

The VQ process (3) implements a particular nonlinear model as defined in equation (4). Nevertheless, compared to other nonlinear models, such as MLP (Multi-Layer Perceptrons), equation (3) has the drawback that all inputs (all coordinates of Y_t vectors (5) but y_t) are weighted equally. This is evidently not optimal. A further step is thus to weight also the inputs, building the Y_t vectors as follows:

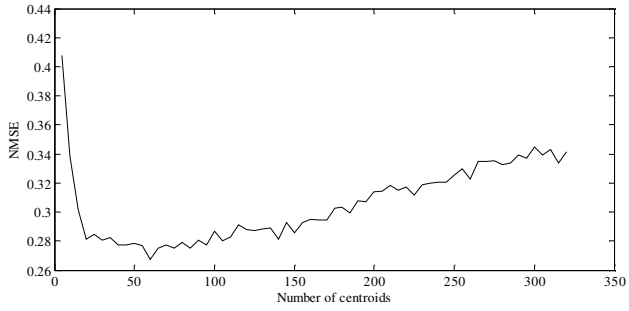

Fig. 4. *NMSE* of the Mackey-Glass with additive noise series prediction with weighted VQ, k=0.5.

$$Y_t = \begin{bmatrix} k y_t & k_1 y_{t-1} & k_2 y_{t-2} & \dots & k_{N-1} y_{t-N+1} \end{bmatrix}. \tag{9}$$

However, the optimization of all k_i parameters is heavy. A full search is often not practicable, while a gradient descent (computed by finite differences on the validation set results) may be very slow. For this reason, we suggest the use of the weights given by the following method.

3.4 Quantization with Weighting of the Inputs by a Linear Model

When building a linear model

$$\hat{y}_t = a_1 y_{t-1} + a_2 y_{t-2} + \dots + a_{N-1} y_{t-N+1}, \tag{10}$$

the a_i coefficients may be considered as the importance (weighting) that variable y_{t-i} plays on the output. In other words, it is the first partial derivative of the output with respect to a specific input. The a_i coefficients can therefore be considered as a first-order approximation of the k_i coefficients needed in (9). Of course, the a_i resulting from a linear hypothesis, they will not be optimum in the sense defined in the previous section. Nevertheless, again, we are looking for a good compromise between an impracticable full search and an inefficient a priori choice such as k_i=1. The coefficients given by linear model (10) appear to be efficient regarding this compromise.

A linear model with a four-dimensional regressor on our previous example, the noisy Mackey-Glass series, gives roughly identical a_i coefficients. As a consequence, the use of these coefficients does not improve the previously presented results. To illustrate the weighting of the VQ by the coefficients of a linear model, we thus use another well-known time series prediction benchmark, the SantaFe A series. Noise with variance=80 has been added. A part of this series is illustrated in Fig. 5. As for the previous example, 5000 points have been used for learning and 5000 for validation. The regressor size is 6. Fig.6 shows the *NMSE* resulting from the weighting with the coefficients of a linear model; the improvement is significant. The k weighting of the output y_t has been obtained with the method described in section 3.2.

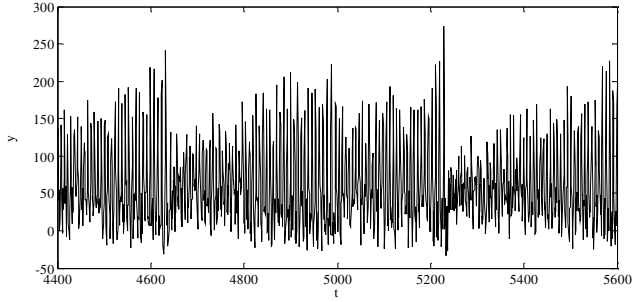

Fig. 5. SantaFe A series.

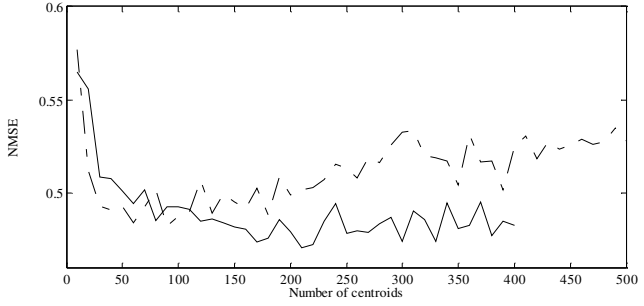

Fig. 6. *NMSE* obtained on the prediction of the noisy Santa Fe A series with a VQ model; dashed: without weighting of the inputs; plain: with weighting of the inputs by the coefficient of a linear model.

4 Radial-Basis Function Networks (RBFN) with Weighted Inputs

Radial-Basis Function Networks (RBFN) are another class of nonlinear approximators, defined by:

$$\hat{y}_t = \sum_{i=1}^{M} \lambda_i e^{-\left\| \frac{x_t - C_i}{\sqrt{2}\sigma_i} \right\|^2} . \tag{11}$$

The coefficients of the model are the centers C_i (often resulting from vector quantization of the inputs), the standard deviations σ_i, and the multiplying factors λ_i. Learning of RBFN models may be achieved in several ways; see for example [1]. In our time series prediction problem, the vector inputs x_t are replaced by the regressors.

RBFN are not so different from VQ methods. In both cases, the underlying principle is to divide the space spanned by the inputs into Voronoi regions. Next, there are two differences between VQ and RBFN.

1. VQ applies a specific and exclusive model in each of the *M* Voronoi regions, while the models around each centroid in the RBFN equation (the Gaussian functions in (11)) are overlapping between adjacent Voronoi zones.
2. The region-specific model in VQ is a constant equal to the first coordinate of the centroid (see equations (3) and (5)), while the equivalent in RBFN is a Gaussian functions with σ_i and λ_i parameters.

With regard to these two comments, RBFN can be viewed as a generalization of VQ models, with smoother approximation capabilities.

Weighting the output in a RBFN model has no sense, as appropriate output weights are computed through the optimization of λ_i, which is a standard procedure in RBFN learning. However, RBFN suffer from the same drawback as VQ what concerns the a priori identical weighting given to all inputs. In the RBFN context, this is illustrated by the use of a scalar σ_i standard deviation, instead of a variance/covariance matrix.

The principle of weighting the inputs, presented in the context of VQ, may however be applied to RBFN [6]. Fig. 7 shows the results of a RBFN model trained on the noisy SantaFe A series; inputs have been weighted according to the coefficients of a linear model. The training of the RBFN model has been realized according to [1]. Again, the improvement due to weighting is clear.

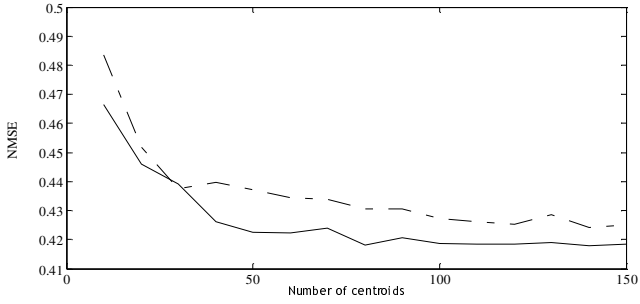

Fig. 7. *NMSE* obtained on the prediction of the noisy Santa Fe A series with a RBFN model; dashed: without weighting of the inputs; plain: with weighting of the inputs by the coefficient of a linear model.

5 Conclusion

This paper presents a nonlinear time series prediction method based on the quantization of vectors with missing data. These vectors concatenate inputs (regressors) and outputs (predictions). Estimating missing values is here used to predict the outputs.

The method presented in this paper is not restricted to times series prediction, and may be applied to regression problems.

Weighting the regressor values and/or the predictions is proposed to improve the quality of the estimation. The same weighting scheme is used to give adequate importance to the inputs of a RBFN model.

The method is illustrated on standard time series prediction benchmarks; the results show clear improvements, decreasing the prediction error on validation sets.

Acknowledgements. Michel Verleysen is Senior research associate at the Belgian National Fund for Scientific Research (FNRS). The work of A. Lendasse, D. François and V. Wertz is supported by the Interuniversity Attraction Poles (IAP), initiated by the Belgian Federal State, Ministry of Sciences, Technologies and Culture. The scientific responsibility rests with the authors.

References

1. Benoudjit, N., Archambeau, C., Lendasse, A., Lee, J., Verleysen, M.: Width optimization of the Gaussian kernels in Radial Basis Function Networks, ESANN 2002, European Symposium on Artificial Neural Networks, Bruges (2002) 425–432.
2. Bontempi, G.L., Birattari, M., Bersini, H.: Lazy Learning for Local Modeling and Control Design, International Journal of Control, 72(7/8), (1998) 643–658.
3. Gray, R.: Vector Quantization, IEEE Mag., Vol. 1,(1984) 4–29.
4. Ibbou, S., Classification, analyse des correspondences et methodes neuronales, PhD Thesis, Paris (1998).
5. Kohonen, T.: Self-Organizing Maps, Springer Series in Information Sciences, Vol. 30, third edition (2001).
6. Lendasse, A., Lee, J., de Bodt, E., Wertz, V., Verleysen, M.: Approximation by Radial-Basis Function networks – Application to option pricing, Accepted for publication in Connectionist Approaches in Economics and Management Sciences, C. Lesage ed., Kluwer academic publishers.
7. Mackey, M. C. , Glass, L.: Oscillations and Chaos in Physiological Control Systems, Science, (1977) 197–287.
8. Vesanto, J.: Using the SOM and Local Models in Time Series Prediction, Proceedings WSOM'97: Workshop on Self-Organizing Maps, 1997.
9. Weigend, A. S., Gershenfeld, N. A.: Times Series Prediction: Forecasting the future and Understanding the Past, Addison-Wesley Publishing Company (1994).