Recursive Non-Linear Autoregressive Models (RNAR): Application to Traffic Prediction of MPEG Video Sources

Nikolaos Doulamis, Anastasios Doulamis and Klimis Ntalianis

Electrical & Computer Engineering Department, National Technical University of Athens
9, Heroon Polytechniou str. 15773 Zografou, Greece
e-mail: ndoulam@cs.ntua.gr

ABSTRACT

In this paper, an efficient algorithm for recursive estimation of a Non-linear Autoregression (NAR) model is proposed. In particular, the model parameters are dynamically adapted through time so that a) the model response, after the parameter updating, satisfies the current conditions and b) a minimal modification of the model parameters is accomplished. The first condition is expressed by applying a first-order Taylor series to the non-linear function, which models the NAR system. The second condition implies the solution to be as much as close to the previous model state. The proposed recursive scheme is evaluated for the traffic prediction of real-life MPEG coded video sources.

1. INTRODUCTION

Linear Auto-Regressive models (ARs) have emerged as a useful tool for many applications in signal and/or image processing problems. Examples includes, texture modeling of visual content, speech processing applications, models for future sample prediction and so on [1]. The input-output relationship of an AR model is provided by the following equation

$$x(n)=\sum_{i=1}^{p}\varphi_i\cdot x(n-i)+e(n) \tag{1}$$

where $x(n)$ is the input signal, while the $e(n)$ an i.i.d. (independent and identically distributed) error signal. The φ_i refer to model parameters, while *p* denotes the model order. Parameters φ_i are estimated using the Yale Walker equations, which are obtained by minimizing the error $e(n)$ as described in [1].

As is observed from (1), the parameters φ_i are considered constant for all samples of signal $x(n)$. However, this is valid for stationary signals. Non-stationary process requires adaptive AR filters in which the model parameters are updated through time. Adaptive AR models have been proposed in the literature such as the RLS (Recursive Least Square) algorithm.

Another drawback of AR models is that they linearly relate the signal samples, which is not valid for many real-life applications, where many non-linearities usually appear. To face the aforementioned difficulty, the linear input-output relation of (1) is extended to a non-linear one. However, the difficulty in this case is the estimation of the model parameters. For this reason, usually non-linear functions of specific type have been used, such as the quadratic filters, in which the model parameters can be calculated more effortlessly [2].

Other more complicated approaches use a generic non-linear system implemented by a neural network architecture [3]. However, the aforementioned algorithms are usually trapped to local minima and thus the error $e(n)$ is not minimized, deteriorating the model performance. Another disadvantage is that they are not suitable for non-stationary signals in which the model parameters should be adapted through time. Furthermore, the computational complexity is high so that its implementation to real-life applications is prohibited [4].

To overcome the above-mentioned difficulties, a new efficient algorithm is proposed in this paper for recursive estimation of the parameters of a Non-linear Autoregression model, which is called RNAR in the rest of the paper. Particularly, the proposed scheme recursively updates the model parameters through time so that a) the model response, after the adaptation, satisfies the current conditions as much as possible while simultaneously b) a minimal modification of the model parameters is obtained. These two conditions implies that the system adapts to the current conditions, without however, forgetting the previous model behavior.

The first condition is expressed by analyzing the input-output relation of the non-linear model using a first order Taylor series expansion. The second condition is satisfied by the minimum modification of the model parameters. Combining the two above-mentioned conditions, the optimal parameter increments are uniquely estimated in an efficient and cost effective manner.

The proposed RNAR model is then applied to predict the traffic rates of real-life MPEG coded video sources. Experimental results indicate that a very good prediction of the video traffic is obtained compared to all other linear or non-linear techniques.

2. NON-LINEAR AUTOREGRESSIVE MODELS (NAR)

Similarly to equation (1), an NAR of order *p* will satisfy the following equation

$$x(n)=g(x(n-1),x(n-2),...,x(n-p))+e(n)=$$
$$g(\mathbf{x}(n-p))+e(n)\,,\quad \mathbf{x}(n-p)=[x(n-1)\cdots x(n-p)]^T \tag{2}$$

where the $e(n)$ is the i.i.d. error as in (1) and $g(\cdot)$ a unknown non-linear function, which models the input-output relation. In the following, we denote the model of (2) as NAR(*p*) similar to the notation of the linear case.

The main difficulty of implementing an NAR model is that function $g(\cdot)$ is actually unknown. For this reason, modeling of function $g(\cdot)$ is required. Using the principles of functional analysis theory, we can find an estimate of $g(\cdot)$, say $\hat{g}(\cdot)$ as follows [3]

$$\hat{g}(\mathbf{x}(n-1)) \approx g(\mathbf{x}(n-1)) = \sum_k v_k \Phi_k \left(\sum_{j=1}^{p} w_{j,k} \cdot x(n-j) \right) \quad (3)$$

In the previous equation, the v_k, $w_{j,k}$ refer to the model parameters, while $\Phi_k(\cdot)$ to appropriate functionals. A common choice for $\Phi_k(\cdot)$ is the sigmoid function,

$$\Phi_k(x) = 1/(1-\exp(-x)) \quad (4)$$

From the above, it is clear that estimation of an NAR(p) model is equivalent to the estimation of the parameters v_k, $w_{j,k}$ of (3) which model the unknown function $g(\cdot)$.

3. RECURSIVE NON-LINEAR AUTOREGRESSIVE MODELS (RNAR)

In this section, we propose an efficient algorithm for recursive estimation of the model parameters of (3). Let us first assume that all model parameters at time m are included in a vector $\mathbf{w}(m)$. That is

$$\mathbf{w}(m) = [\cdots v_k(m) \cdots w_{j,k}(m) \cdots]^T \quad (5)$$

In this case, we have added the index m to indicate that the parameters are time dependent.

Without loss of generality, let us assume that the model parameters are updated at the time instance k. Then, the recursive algorithm is implemented so that the system response satisfies the current condition as much as possible, i.e.,

$$\mathbf{w}(k): \ \hat{g}_{\mathbf{w}(k)}(\mathbf{x}(k-p)) \equiv \hat{x}(k) = x(k) \quad (6)$$

where $\hat{g}_{\mathbf{w}(k)}(\cdot)$ is the estimate of $g(\cdot)$ when $\mathbf{w}(k)$ model parameters are used.

In order to track equation (6), we assume that a small perturbation of the model parameters is sufficient to adapt the model performance to current conditions. Thus, the parameters are related as

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \Delta\mathbf{w} \quad (7)$$

where $\Delta\mathbf{w}$ refers to the small increment of the model parameters.

Equation (7) permits the application of a first order Taylor series expansion to (6). The sufficient conditions for being valid the Taylor series expansion can be found in [5]. In this case, we can prove that the condition of (6) is written as

$$b = \mathbf{a}^T \cdot \Delta\mathbf{w} \quad (8)$$

where the scalar b is the difference of the model response before model updated and the actual response, that is

$$b = x(k) - \hat{g}_{\mathbf{w}(k-1)}(\mathbf{x}(k-p)) \quad (9)$$

and vector $\mathbf{a}$ depends on the previous model parameters as follows

$$\mathbf{a} = [\cdots \Phi_k(1-\Phi_k) \cdot v_k \cdot x(n-j) \cdots, \cdots \Phi_k \cdots, 1]^T \quad (10)$$

However, many $\Delta\mathbf{w}$ can satisfy equation (8) since the number of elements of $\Delta\mathbf{w}$ is greater than one. Uniqueness is imposed by an additional requirement, which takes into consideration the variation of the model parameters. In particular,

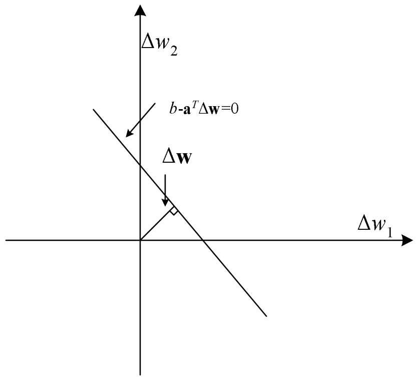


Figure 1: A graphical representation of the optimal small perturbation solution

among all possible solutions the one which causes a minimal modification of the model parameters is selected as the most appropriate. This means that the model adapts to current condition with, a minimum modification of the previous model response. This requirement is achieved by following constraint minimization

$$\text{minimize } \|\Delta\mathbf{w}\|_2 \text{ or equivalently } (\Delta\mathbf{w})^T \cdot \Delta\mathbf{w} \quad (11)$$

$$\text{subject to } b - \mathbf{a}^T \cdot \Delta\mathbf{w} = \mathbf{0} \quad (12)$$

3.1. Model Parameters Estimation

Equations (11) and (12) are minimized using Lagrange multipliers. In this case (11) and (12) are equivalent to [6]

$$(\Delta\mathbf{w})^T \cdot \Delta\mathbf{w} + \boldsymbol{\lambda}^T \cdot (b - \mathbf{a}^T \cdot \Delta\mathbf{w}) \quad (13)$$

where the elements of vector $\boldsymbol{\lambda}$ corresponds the Lagrange multipliers. Differentiating equation (13) with respect to $\Delta\mathbf{w}$ and $\boldsymbol{\lambda}$ and setting the results equal to zero, we obtain the optimal parameter perturbation

$$\Delta\mathbf{w} = \frac{\mathbf{a}^T}{(\mathbf{a}^T \cdot \mathbf{a})} \cdot b \quad (14)$$

From equation (14), it can be seen that the optimal parameters increments $\Delta\mathbf{w}$ are directly calculated with respect to vector **a** and scalar b. As a result, a recursive implementation of a non-linear autoregression model is accomplished which is denoted as RNAR in the following. The physical meaning of equation (14) is presented in Figure 1. In this case we have considered that only two parameters are required for modeling the NAR system. As can be seen from Figure 1, the optimal solution is obtained as the perpendicular from the origins to the line defined by the constraint.

3.2. Activation of Parameter Updating

While, in theory, the recursive estimation of model parameters can be performed at every new incoming sample of the signal, in practice there is no reason to activate the adaptation algorithm in case that the model accuracy is satisfactory. For this reason, the deviation of the actual signal samples and the model response is evaluated.

$$A = \begin{cases} 1 & \text{if } D = |\hat{x}(n) - x(n)| > T \\ 0 & \text{if } D = |\hat{x}(n) - x(n)| \leq T \end{cases} \quad (15)$$

In case that the deviation exceeds an unacceptable threshold the model parameters are updated (*A*=1). Otherwise, the model parameters remain the same (*A*=0).

4. APPLICATION TO TRAFFIC PREDICTION OF MPEG VIDEO SOURCES

In this section, we apply the proposed RNAR model for traffic prediction of real-life MPEG coded video sources. Video traffic prediction is very useful for network management algorithms and congestion control schemes, which prevent the communication systems from possible overload [7], [8].

In our simulations, a long duration MPEG coded video source has been evaluated. Thus, scene changes, high variation of motion activity, camera zooming and panning and changes of luminosity conditions are encountered. For the MPEG coded scheme, three different types of frames are presented; the I (Intraframe), the P (Predicted) and the B (Bi-directional). These types of frames are predicted individually since they present different statistical properties [7].

Figure 2 presented the traffic rate of *I*, *P* and *B* frames of the MPEG sequence over a time window of 250 frames versus the frame number. In this figure, the solid line corresponds to the actual data, while the dotted line refers to the predicted data. For traffic prediction, the proposed RNAR model has been applied. In our case, model adaptation is performed each time a prediction error greater than 10% has been encountered. As is observed, in all cases, the prediction accuracy is very high, even at time instances of highly fluctuated frame rates.

An alternative way to indicate the good performance of the proposed model as traffic-rate predictor is to plot the predicted data versus the actual ones. Figures 3 present the results for the three types of frames (*I*, *P* and *B*) of the sequence. In these figures, the solid line represents the perfect fit.

4.1. Comparison with other Linear and Non-linear Traffic Prediction Models

In the following, the performance of the proposed RNAR model as traffic rate predictor is compared with three other methods. The first uses a Recursive implementation of a linear AR model (RAR) [1]. The second a recurrent neural network architecture as in [9], which simulates a Non-linear ARMA (NARMA) system. Finally, in the third method an non-adaptable NAR model [10] has been used.

As an objective criterion for evaluating the prediction accuracy in our case, the relative prediction error with respect to the actual data E is use,

$$E = \frac{1}{N} \sum_{n=1}^{N} \frac{|x(n) - \hat{x}(n)|}{x(n)} \times 100 \quad (16)$$

where N is the total number of samples of the signal, while $x(n)$, $\hat{x}(n)$ the *n*th sample of the actual and predicted signal.

Table I presents the prediction accuracy results obtained for the *I, P, B* frame streams using the three aforementioned methods. As is observed, the proposed model provides the best prediction performance in all cases, while the NARMA approach [9] the second one. This is due to the fact that the first method uses a linear model (although its recursive implementation) to predict the MPEG video traffic, while the third does not update model parameters during prediction. To clearly illustrate the differences of the proposed method from the second best technique (NARMA), a rate by rate comparison is depicted in Figures 4. As can be seen, the NARMA cannot track with high accuracy the highly fluctuated traffic rates. Furthermore, it presents an unstable behavior especially when applied for long traffic periods.

5. REFERENCES

[1] S. Haykin *Adaptive Filter Theory.* Prentice Hall 1996.

[2] G.L. Sicuranza, "Quadratic filters for signal processing," *Proc. of the IEEE*, Vol. 80, No. 8 , Aug. 1992, pp 1263 -1285.

[3] J. Connor, D. Martin and L. Altas, "Recurrent Neural Networks and Robust Time Series Prediction," *IEEE Trans. on Neural Networks*, vol. 5, no. 2, pp. 240-254.

[4] S. Haykin. *Neural Networks: A Comprehensive Foundation.* New York: Macmillan, 1994.

[5] A. Doulamis, N. Doulamis and S. Kollias, "On Line Retrainable Neural Networks: Improving the Performance of Neural Network in Image Analysis problems," *IEEE Trans. on Neural Networks,* Vol. 11, No. 1, pp. 137-155, January 2000.

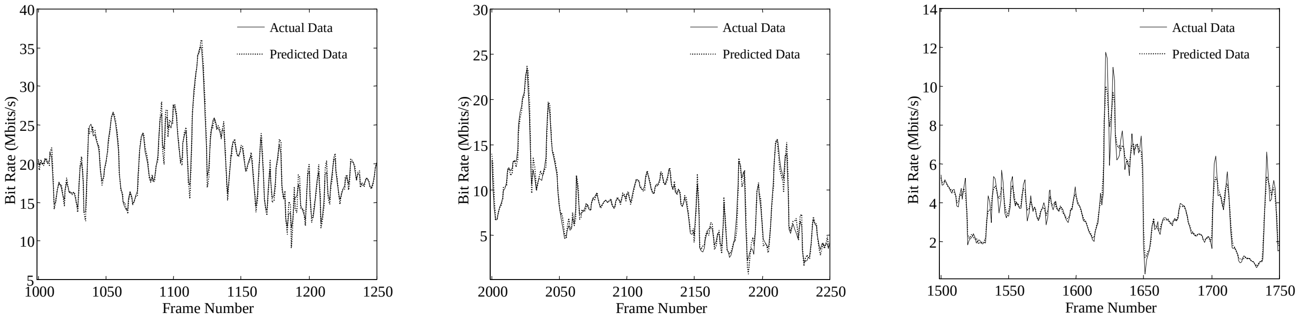
[6] D. J. Luenberger. *Linear and non Linear Programming.* Addison-Wesley 1984.

[7] N. Doulamis, A. Doulamis, G. Konstantoulakis and G. Stasinopoulos, "Efficient Modeling of VBR MPEG-1 Video Sources," *IEEE Trans on Circuits and Systems for Video Technology,* vol. 10, no. 1, pp. 93-112, Feb. 2000.

[8] A. Doulamis, N. Doulamis and S. Kollias, "Modeling and Queuing Analysis of Variable-Bit-Rate MPEG Video Streams," Journal of Electronics and Communications, Vol. 38, No. 1, pp.1-8, 1998.

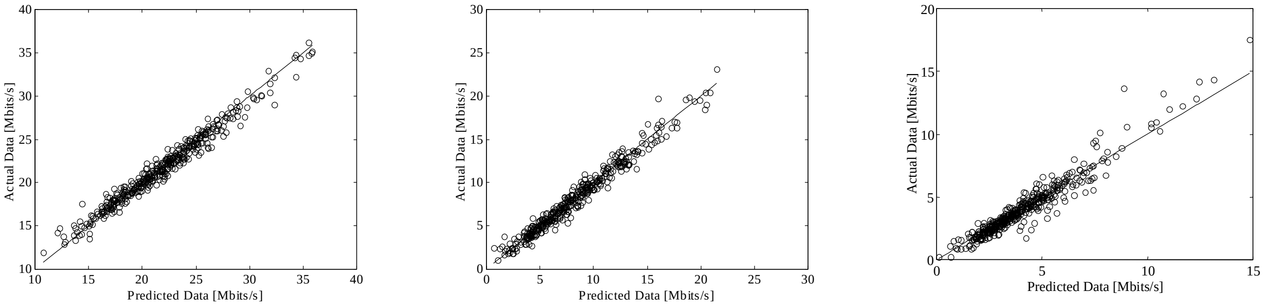
[9] P.-R. Chang and J.-T. Hu, "Optimal Nonlinear Adaptive Prediction and Modeling of MPEG Video in ATM Networks Using Pipelined Recurrent Neural Networks," *IEEE journal of Selected Areas in Comm.*, vol. 15, no. 6, Aug. 1997.

[10] J. Chong, S.Q. Li and J. Ghosh, "Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real Time VBR over ATM," *IEEE journal of Selected Areas in Comm.,* vol. 13, pp. 12-33, Jan. 1995.



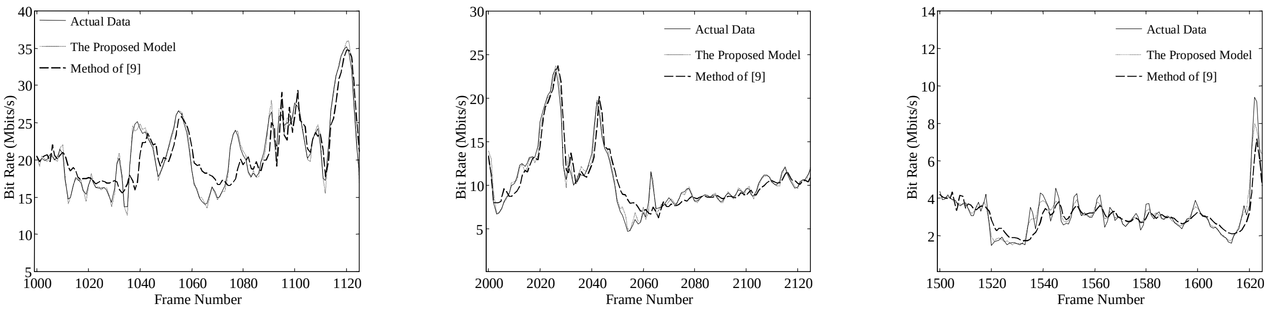
(a) (b) (c)

Figure 2: The actual and the predicted traffic rate using the proposed model over a time window of 250 frames (a) for *I* frames. (b) for *P* frames and (c) for the B frames.



(a) (b) (c)

Figure 3: The actual data versus the predicted ones. (a) *I* frames. (b) *P* frames. (c) *B* frames.



(a) (b) (c)

Figure 4: A rate by rate comparison of the proposed model with an RARMA model implemented as in [9] over a time window of 250 frames in case of (a) *I* frames, (b) for *P* frames and (c) *B* frames.

	Prediction Error		
Sequences	*I* - Frame (%)	*P*- Frame (%)	*B* - Frame (%)
The proposed RNAR Model	1.12	1.89	2.81
NARMA Model	7.12	8.55	10.02
NAR Model	9.36	10.87	11.88
RAR Model	12.58	13.42	14.75

Table I: Comparison of frame losses for the actual data of Source3 and the proposed model at different buffer sizes and utilization degrees.