ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 63-68

MAP Decomposition of a Mixture of AR Signals Using Multilayer Perceptrons

Christophe Couvreur*

Service de Physique Générale
Faculté Polytechnique de Mons
Rue de Houdain 9, B-7000 Mons, Belgium

Abstract. We consider the problem of classifying multiple simultaneous autoregressive (AR) signals based upon the observation of their sum using a multilayer perceptron network (MLP). We propose a method that allows the training of the classifier to be performed on *separate* AR processes, and uses the Bayesian interpretation of the outputs of a MLP to obtain the maximum *a posteriori* probability decomposition of a mixture of *simultaneous* AR processes. Some simulations results are presented. The extension of our method to a more general class of problems is discussed.

1. Introduction

It is well known that autoregressive (AR) models can represent a wide variety of signals. For example, they have been used successfully in speech processing under the name LPC [1]. However, in certain applications multiple signals each described by an AR model may arise simultaneously. To deal with such situations we define a new model for simultaneous AR signals. A time series $y[n]$ is defined as a mixture of AR signals (noted ΣAR) if $y[n]$ is a sum of c components $x_i[n]$, and the signals $x_i[n]$ are generated by independent AR(p) processes. That is,

$$\begin{aligned} y[n] &= \sum_{i=1}^{c} x_i[n], & (1)\\ x_i[n] &= -\sum_{k=1}^{p} a_i[k]x_i[n-k] + e_i[n], & (2) \end{aligned}$$

where the processes $e_i[n]$ are mutually independent white noise sequences. Figure 1 gives a block diagram of such a process with the conventional notation $A_i(z) = 1 + \sum_{k=1}^{p} a_i[k]z^{-k}$ where z^{-1} denotes the delay operator. It can be shown that such a sum of c independent AR(p) processes is equivalent to an ARMA(cp, cp) process with a particular structure [2].

This work was supported by a Rotary Foundation Ambassadorial Scholarship.

*Christophe Couvreur is a Research Assistant of the Fonds National de la Recherche

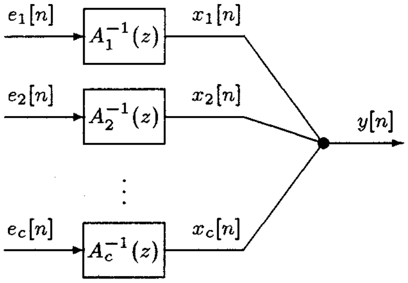

Figure 1: A mixture of c AR processes.

In this paper we consider the problem of classifying the AR components present in a ΣAR signal. That is, we assume that each of the AR models $A_i(z)$ generating the signals $x_i[n]$ belongs to one of M possible classes of AR models, and we want to find the class membership of the AR components present in the ΣAR signal $y[n]$. We further assume that, during the training phase, labeled samples of the signals $x_i[n]$ are available for each of the M possible classes. However, during the classification phase, only $y[n]$ is observable. This problem is typically encountered in environmental sound recognition and in processing of speech in a noisy background.

An autoregressive process is entirely characterized, up to its variance, by its p AR parameters $a_i[1], \ldots, a_i[p]$ or, equivalently, by the p complex roots $z_{i,1}, \ldots, z_{i,p}$ of $A_i(z)$ [3]. These roots are called the *poles* of the AR model. Given L consecutive samples of the AR process $x_i[1], \ldots, x_i[L]$, it is possible to compute an estimate of the AR parameters $a_i[k]$ using well known algorithms such as the autocorrelation method [3]. The poles are then obtained directly by finding the roots of $A_i(z)$. Similarly, a ΣAR process is characterized by the poles of the AR part of its equivalent ARMA process. It is easily seen that these poles are simply the union of all the cp poles of the c AR components [2]. Given L consecutive samples of a ΣAR process $y[1], \ldots, y[L]$, it is possible to compute an estimate of the parameters of the AR part of the equivalent ARMA process using methods initially developed for ARMA models, e.g., the modified Yule-Walker method [3]. The poles are then obtained by finding the roots of the resulting polynomial. If we were only interested in the classification of *one* AR signal into *one* of the M classes based on the observation of the signal, we could simply train a pattern classifier to perform this classification based on the p poles. If we choose to use a multilayer perceptron (MLP) network for this *1 of M* pattern classification problem, the MLP would need $2p$ inputs elements[1] and M output elements. For simultaneous classification of the c AR components of a ΣAR model, we can similarly use the cp poles computed from $y[n]$ as inputs to a classifier. One obvious solution is to train a MLP to recognize directly all the $\binom{M+c-1}{c}$ possible combinations of classes. This requires a network with $2cp$

Scientifique-Belgacom.

[1] We need $2p$ inputs to represent p complex poles. Note that if the AR signals are real, the poles are real or come in complex conjugate pairs, and the number of representative inputs reduces to p.

input elements and $\binom{M+c-1}{c}$ output elements. Even for moderate c and M, the size of the resulting network will preclude efficient training. In the sequel of this paper we propose a method to avoid this problem. Our algorithm allows the training of the classifier to be performed on *separate* AR processes, and uses a combinatorial argument to obtain the classification of *simultaneous* AR processes.

2. Algorithm

2.1. Formulation of the Problem

Since the autocovariance method and the modified Yule-Walker method are consistent [3], we can assume that, for L large, they provide exact estimates of the poles of the AR and ΣAR processes, respectively. That is, we will assume that we have the true poles of the AR processes $x_i[n]$ during the training phase, and that we have the true poles of the ΣAR process $y[n]$ during the recognition phase. Of course, this assumption is only asymptotically valid for L tending to infinity.

Let $\omega_1, ..., \omega_M$ be the M possible classes for the AR processes $x_i[n]$ and let $P(\omega_j)$ be the *a priori* probability of class ω_j. An AR(p) process is characterized by the set of its p poles $\mathbf{z} = \{z_1, \dots, z_p\}$. Let $p(\mathbf{z}|\omega_j)$ be the class-conditional probability density for $\mathbf{z}$, the probability density function for $\mathbf{z}$ given that the AR process belongs to class ω_j. The class-conditional distributions are to be mutually independent. During the training phase, we have a set $\mathcal{X}$ of N labeled training patterns $\mathcal{X} = \{(\mathbf{z}_1, \ell_1), \dots, (\mathbf{z}_N, \ell_N)\}$, where ℓ_k denotes the class label of $\mathbf{z}_k$. The patterns are drawn from the distributions $p(\mathbf{z}|\omega_j)$ according to the priors $P(\omega_j)$, $j = 1, \dots, M$. During the recognition phase, we are given a set of cp poles $\mathcal{Z} = \{z_1, \dots, z_{cp}\}$. The number of classes c and the AR order p are assumed to be known *a priori*. The poles are obtained by the modified Yule-Walker method *in no particular order*. We want to find the combination of c classes $\Psi = \{\psi_1, \dots, \psi_c; \psi_i \in \{\omega_1, \dots, \omega_M\}\}$, that maximizes the *a posteriori* probability $P(\Psi|\mathcal{Z})$. The maximizer $\Psi_{\text{MAP}} = \arg\max_\Psi P(\Psi|\mathcal{Z})$ gives then the *maximum a posteriori probability* (MAP) decomposition of the ΣAR process $y[n]$ into c AR(p) components.

2.2. Training of the Network

It is well known that the multilayer perceptron, when trained as a *1 of M* classifier using backpropagation, estimates the Bayes *a posteriori* class probabilities, provided that the size of the training sample is large enough and that the backpropagation learning procedure does not get stuck in a local minima [5][6]. Therefore, if a multilayer perceptron with p input elements and M output elements is trained as a classifier on the set of training patterns $\mathcal{X}$, its outputs approximate the *a posteriori* probabilities $P(\omega_k|\mathbf{z})$ for $k = 1, \dots, M$ when a pattern $\mathbf{z}$ is applied to the input. These *a posteriori* probabilities can be used to find the MAP decomposition Ψ_{MAP} as we will now see.

2.3. Maximum *a posteriori* Probability Decomposition

We can rewrite $P(\Psi|\mathcal{Z})$ as

$$P(\Psi|\mathcal{Z}) = P(\{\psi_1,\ldots,\psi_c\}|\mathcal{Z}) = S(\Psi)P([\psi_1,\ldots,\psi_c]|\mathcal{Z}), \tag{3}$$

where $\{\cdot\}$ denotes a set of elements without any particular order, and $[\cdot]$ denotes an ordered set of elements. The factor $S(\Psi)$ is the number of *distinct* orderings of the classes in Ψ, which can be expressed as $S(\Psi) = c!/(s_1!\ldots s_M!)$ where s_i is the number of occurrences of class ω_i in Ψ. Let $\breve{\mathcal{Z}}^{(k)} = [\breve{\mathbf{z}}_1^{(k)},\ldots,\breve{\mathbf{z}}_c^{(k)}]$, where $\breve{\mathbf{z}}_i^{(k)} = \{\breve{z}_{i,1}^{(k)},\ldots,\breve{z}_{i,p}^{(k)}\}$, be an ordered partition of $\mathcal{Z}$ into c subsets of p poles each. Then, we have

$$\begin{aligned} P([\psi_1,\ldots,\psi_c]|\mathcal{Z}) &= \sum_{k=1}^{K} P([\psi_1,\ldots,\psi_c]|\breve{\mathcal{Z}}^{(k)})P(\breve{\mathcal{Z}}^{(k)}) \\ &= \frac{1}{K}\sum_{k=1}^{K} P([\psi_1,\ldots,\psi_c]|\breve{\mathcal{Z}}^{(k)}), \end{aligned} \tag{4}$$

where K is the total number of partitions of $\mathcal{Z}$, $K = (cp)!/(p!)^c$. Since the AR components are independent, it is easy to show that $P([\psi_1,\ldots,\psi_c]|\breve{\mathcal{Z}}^{(k)})$ can be factored as

$$P([\psi_1,\ldots,\psi_c]|\breve{\mathcal{Z}}^{(k)}) = \prod_{i=1}^{c} P(\psi_i|\breve{\mathbf{z}}_i^{(k)}). \tag{5}$$

Finally, we have

$$P(\Psi|\mathcal{Z}) = \frac{S(\Psi)}{K}\sum_{k=1}^{K}\prod_{i=1}^{c} P(\psi_i|\breve{\mathbf{z}}_i^{(k)}). \tag{6}$$

As suggested previously, the *a posteriori* class probabilities $P(\psi_i|\breve{\mathbf{z}}_i^{(k)})$ can be estimated using the multilayer perceptron trained on $\mathcal{X}$. The MAP decomposition is then straightforwardly obtained by maximizing (6),

$$\Psi_{\mathrm{MAP}} = \arg\max_{\Psi} P(\Psi|\mathcal{Z}). \tag{7}$$

2.4. Computational Complexity

Our method requires the training of a network with $2p$ inputs and M outputs. Once the network has been trained, the decomposition of a ΣAR signal implies the evaluation of the outputs of the network for all K partitions of the set of poles. The maximization of $P(\Psi|\mathcal{Z})$ is then a simple combinatorial problem which requires the evaluation of the $\binom{M+c-1}{c}$ possible combinations of AR classes. The "direct" approach would require the training of a network with $2cp$ inputs and $\binom{M+c-1}{c}$ outputs. The recognition would then be simply performed by evaluating the network for the set of poles of the ΣAR signal and selecting the output with the maximal value. We see that our approach trades

Table 1: Experimental and theoretical probabilies of classification.

	Both correct	One correct	Both incorrect
Experimental	61.6%	26.3%	12.1%
Theoretical	63.0%	32.7%	4.3%

a reduction of the complexity during training for an increase of the computational load during recognition. Since training is generally the stumbling block of a neural network application, the advantages of reduced training complexity will generally overcome the inconvenience of the additional computations during the recognition phase. Moreover, our algorithm is particularly suited to a parallel implementation, which could eventually solve the computational problem.

3. Preliminary Results

To verify the algorithm and illustrate its performance, we conducted some simulations. The algorithm was implemented in MATLAB on a Sun workstation. In a first experiment, we considered ΣAR processes composed of $c = 2$ simultaneous AR(2) processes with complex conjugates poles. There were $M = 5$ different classes of AR processes with equal priors. The complex poles in each class were distributed on a ring inside the unit circle in the z-plane according to a mixture of Gaussian and uniform probability distributions. The neural network used had two inputs elements representing the real and complex part of the poles, and five outputs elements, one for each class. A multilayer perceptron with one hidden layer of 50 elements was found experimentally to give good results. The network was trained on 10 000 training patterns (2000 training patterns per class) using backpropagation with an adaptive step size and momentum of 0.6. We then generated samples $[y[1], \ldots, y[L]]$ of length $L = 8096$ for 15 000 different ΣAR processes (1000 for each of the 15 possible combinations). The poles of the ΣAR processes were estimated using the modified Yule-Walker method, and the MAP decomposition was obtained using the algorithm described above. The results are summarized in Table 1. This first experiment was deliberately kept simple to allow the analytical derivation of the Bayes error rate for the optimal classifier [4]. The performances of our algorithm matches the theoretical limit reasonably. Other experiments on more complex data show similar results.

4. Concluding Remarks

An original approach to the problem of classifying multiple simultaneous autoregressive signals from the observation of their sum using a multilayer perceptron network has been proposed. The approach is based on the ability a MLP adequately trained to estimate Bayesian a posteriori probabilities for separate AR signals. The method offers substantial gains in terms of training complexity at the expense of increased computations during the recognition phase.

A multilayer perceptron was used in this paper. However, other neural architectures can also be used to estimate the *a posteriori* class probabilities, e.g., radial basis function networks (RBF) or high-order polynomial networks (see [5]).

The approach proposed here can be extended to a more general class of problems. We introduced the algorithm of section 2 as a means of recognizing combinations of poles of AR processes. It can be easily extended to other situations involving the detection of simultaneous components from an unordered set. Examples of applications can be found, for example, in image processing. Necessary conditions for the applicability of our approach are: the different components must be independently distributed, and a separate representative training set of samples must be available for each class of interest.

Acknowledgment

The author wishes to thank Prof. Yoram Bresler, from the University of Illinois at Urbana-Champaign, and Prof. Kurt Hornik, from Technishe Universität Wien, for their suggestions.

References

[1] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] C. Couvreur and Y. Bresler, Decomposition of a Mixture of Gaussian AR Processes, to appear in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Detroit, May 1995.

[3] B. Porat, *Digital Processing of Random Signals, Theory and Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Willey & Son, New York, 1973.

[5] M. Richard and R. Lippman, Neural Network Classifiers Estimate Bayesian *a posteriori* Probabilities, *Neural Computation*, Vol. 3, pp. 461–483, 1991.

[6] W. Ruck, et al., The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function, *IEEE Trans. Neural Networks*, Vol. 1, No. 4, pp. 296–298, Dec. 1990.