

# Large Margin AR Model for Time Series Classification

Venkataramana Kini B

Honeywell Technology Solutions Lab,  
Bannerghatta Road  
Bangalore 560076  
INDIA

Email: venkataramana.bantwal@honeywell.com

C. Chandra Sekhar

Dept of Computer Science and Engg.  
Indian Institute of Technology Madras  
Chennai 600036  
INDIA

Email: chandra@cse.iitm.ac.in

**Abstract**—In this paper we propose a new method for time series pattern classification. It is based on the generative modeling using Autoregressive(AR) model and optimizing the boundaries between these models using the large margin concepts. The developed model captures the correlations in the time series data. Multi-class classification can be performed directly without performing binary classification. The optimization is performed using genetic algorithm for obtaining global optimal parameters. The developed method is applied on simulated and ECG data and found to perform better than the methods which utilize the AR coefficients as the features for the classification.

## I. INTRODUCTION

Dynamic behavior of real world systems can be represented by measurements along temporal dimension (time series). Time series forms an important class of data objects in tasks such as speech recognition and medical signal analysis. In most of these practically occurring time series, one important property shared by them is that the neighboring values in time series are similar (temporal correlation). Therefore, the current value of the time series can be expressed as a finite, *linear aggregate of previous values of the series* and noise (Autoregressive model).

The Autoregressive (AR) model is a generative model in the sense that the current value of the time series is generated as a linear combination of previous values (Figure 1), and a probability distribution function (equation 3) can be defined on the same. The AR model is widely used in the tasks such as time series prediction and parametric spectrum estimation. However, in current work our objective is to utilize AR models in time series classification.

Time series data being omnipresent, time series classification is an important problem in pattern recognition. In time series classification, given a set of time series with class labels, we develop a model for each class. When a new time series is given, a label is assigned to it based on the model (in our case, the AR model) that generates the time series with the highest likelihood. This is essentially generative modeling for time series classification.

However, statistical pattern recognition algorithms for time series classification can be broadly divided into generative and discriminative algorithms. The generative model provides a rich framework for imposing the structure and prior knowledge

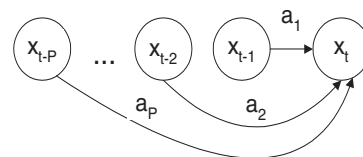


Fig. 1. Graphical representation of an AR model of time series model

on a given problem. However, the recent progress in discriminative learning has demonstrated that a superior classification performance can be obtained by avoiding generative modeling and focusing on the given task of classification. However, the black-box nature of these discriminative models makes incorporation of the structure of the problem difficult. For example, in the current problem of time series classification using discriminative models, we can not capitalize on temporal dependency present in the time series. This motivates to synergistically combine these two kinds of models leading to a hybrid model which retains the richness of generative models, at the same time providing the superior performance. There is an evidence that the models constructed in this manner can capture more subtleties of the signal being analyzed [10]. It is also demonstrated previously that the classification performance on test data will be better when obtained by estimating the parameters of the models to maximize the classification performance on the training data, rather than maximizing the likelihood of the training data.

In other words, generative models essentially learn a joint probability distribution:  $p(\mathbf{x}, y)$ , of the inputs  $\mathbf{x}$  and the label  $y$ , and make their prediction using Bayes rule,  $y = \arg \max_{y_i} p(y_i | \mathbf{x})$ . Discriminative models find the  $p(y | \mathbf{x})$  directly, or learn the direct map from inputs  $\mathbf{x}$  to the labels  $y$ . One of the compelling reasons for choosing the discriminative models over the generative models for classification was pointed by Vapnik: One should solve the (classification) problem directly and never solve a more general problem (e.g. density estimation of each class). Consequently, there has been a significant recent interest in the training of discriminative versions of generative models for classification tasks.

In literature, there are many attempts towards discriminatively training the generative models, particularly in the area of speech recognition (for a review see [10], [7] [11] [14]). Most of these works rely on hidden Markov models or related models as the underlying generative models. However, the AR model is not utilized for generative modeling and consequent discriminative training. The current work is based on discriminatively training AR models for the purpose of time series classification.

The AR modeling has been extensively used for the purpose of feature extraction and hence to classify time series [1] [12]. However, all of these works mostly utilize the capability of AR modeling as a feature extraction technique, in terms of AR coefficients and apply a discriminative classification method on these features. This combination in true sense is not an optimal combination of generative and discriminative models.

In this paper, we propose a discriminatively trained generative AR model, which retains rich interpretation of AR models. The model is motivated from large margin concepts of the support vector machines (SVM), which is an established method in different classification benchmark problems. However, the SVMs do not have generative interpretation. They are inherently designed for solving two class classification problems. The SVMs generally handle the AR coefficients as features. The temporal dependencies can not be naturally captured in the SVMs. The multi-class classification problem can not be naturally handled using SVMs.

In the next section, we develop large margin AR model for time series classification. First we introduce the generative AR model for time series classification, then we develop the proposed large margin AR model. In section 3, we discuss the experimental results of time series classification using the developed model. First we discuss results on the simulated data and then on the ECG time series data which is one of the benchmark time series data.

## II. AR MODEL BASED METHODS FOR TIME SERIES CLASSIFICATION

### A. AR model for Classification

An AR process models the linear dependency that may exist in the signal. It essentially models the signal as the output of a linear system driven by white noise of zero mean and unknown variance. Let  $\mathbf{x}_n = [x_n(1), x_n(2), \dots, x_n(M)]^t$  be the  $n^{\text{th}}$  time series of length  $M$  (training data:  $\mathbf{X} = \{ \langle \mathbf{x}_1 y_1 \rangle, \langle \mathbf{x}_2 y_2 \rangle, \dots, \langle \mathbf{x}_N y_N \rangle \}$   $\mathbf{x}_n \in \mathfrak{R}^n$  and  $y_n \in \{1, 2, \dots, C\}$ ). Here  $C$  is number of classes. The AR model describing a time series can be represented as:

$$x_n(t) = - \sum_{p=1}^P a_p x_n(t-p) + w(t) \quad (1)$$

where,  $w(t) \sim N(0, \sigma^2)$

Here  $w(t)$  is zero mean white noise and the  $a_p$  are AR coefficients.

For the first order AR model, the above equation can be written as (to simplify expressions, we give expressions for the case of first order AR model in this paper, seam-less generalization to higher order is possible) :

$$w(t) = x_n(t) + a_1 x_n(t-1) \quad (2)$$

Since  $w(t) \sim N(0, \sigma^2)$ , the pdf of  $\mathbf{x}_n$  can be written as:

$$\begin{aligned} p_{\mathbf{x}}(\mathbf{x}_n | a_1, \sigma^2) &= p_{\mathbf{w}}(\mathbf{w}_n) \\ &= (2\pi\sigma^2)^{-M/2} \exp(-0.5\sigma^{-2} \sum_{t=1}^M w^2(t)) \\ &= (2\pi\sigma^2)^{-M/2} \exp(-0.5\sigma^{-2} \sum_{t=1}^M (x_n(t) + a_1 x_n(t-1))^2) \end{aligned} \quad (3)$$

where  $\mathbf{w}_n$  is the noise vector associated with  $\mathbf{x}_n$ .

The AR model can be represented alternatively by using the autocovariance matrix  $\Sigma$ , which is characterized by AR coefficients:

$$p_{\mathbf{x}}(\mathbf{x}_n | a_1, \sigma^2) = (2\pi)^{-1/2} |\sigma^2 \Sigma|^{-1/2} \exp(-0.5\sigma^{-2} \mathbf{x}_n^t \Sigma^{-1} \mathbf{x}_n) \quad (4)$$

where  $\Sigma$  is defined as:

$$\Sigma = \begin{pmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{M-1} \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{M-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \rho_{M-1} & \rho_{M-2} & \dots & \rho_1 & 1 \end{pmatrix} \quad (5)$$

The  $\rho_k$  is the autocorrelation function at lag  $k$  defined as:

$$\rho_k = a_1^k \quad (6)$$

The autocovariance matrix represents the covariance matrix of the AR process, and the quantity  $\mathbf{x}_n^t \Sigma^{-1} \mathbf{x}_n$  is Mahalanobis distance. With the unimodal AR process assumption, a particular class can be represented by the ensemble average of such covariance matrices of individual time series belonging to a particular class. Using such models for different classes  $y \in \{1, 2, \dots, C\}$  the classification can be performed using rule:

$$y = \arg \min_c \{ \mathbf{x}^t \Sigma_c^{-1} \mathbf{x} \} \quad (7)$$

It is to be noted that the autocovariance matrix of AR model  $\Sigma$  has Toeplitz structure (which simplifies the computation of its inverse) and is characterized by only the AR coefficients. In the next subsection, we develop the large margin AR model starting from this classification rule (7).

### B. Large margin AR model

In the large margin AR (LMAR) model, we propose to maximize the margin (distance from the decision boundary) by optimizing the parameters involved, i.e., the AR coefficients representing each class. The AR coefficients are found not only to classify the training data correctly, but also to place

the decision boundaries optimally. We propose to constrain each data point to be at least one unit distance away from the decision boundary to each competing class (similar to what is done in case of large margin Gaussian mixture model [14])

$$\mathbf{x}_n^t (\boldsymbol{\Sigma}_c^{-1} - \boldsymbol{\Sigma}_{y_n}^{-1}) \mathbf{x}_n \geq 1, \quad \forall c \neq y_n \quad (8)$$

As in SVMs, we impose the smallest parameter constraint (which provides good regularization for the resulting solution) by minimizing the trace of the matrix  $\boldsymbol{\Sigma}_c^{-1} = \boldsymbol{\Phi}_c$ :

$$\begin{aligned} & \min \sum_c \text{trace}(\boldsymbol{\Phi}_c) \\ \text{s.t.}, & \quad 1 + \mathbf{x}_n^t (\boldsymbol{\Phi}_{y_n} - \boldsymbol{\Phi}_c) \mathbf{x}_n \leq 0, \quad \forall c \neq y_n, n = 1, 2, \dots, N \\ & \text{and } \boldsymbol{\Phi}_c \succ 0, c = 1, 2, \dots, C \end{aligned} \quad (9)$$

There may be outliers in the training data and it may not be feasible to classify all the training data points correctly. To handle such cases, as in SVMs, we introduce the nonnegative slack variables  $\xi_{nc}$ :

$$\begin{aligned} & \min \sum_{nc} \xi_{nc} + \gamma \sum_c \text{trace}(\boldsymbol{\Phi}_c) \\ \text{s.t.}, & \quad 1 + \mathbf{x}_n^t (\boldsymbol{\Phi}_{y_n} - \boldsymbol{\Phi}_c) \mathbf{x}_n \leq \xi_{nc}, \\ & \quad \xi_{nc} \geq 0, \forall c \neq y_n, n = 1, 2, \dots, N \\ & \quad \boldsymbol{\Phi}_c \succ 0, c = 1, 2, \dots, C \end{aligned} \quad (10)$$

Similar to the case of SVMs, the  $\gamma$  is set through cross-validation. The optimization involves AR coefficients as decision variables, which appear in nonlinear form (equation 6). Therefore, we utilize genetic algorithms to ensure that the global minimum is obtained (Table II).

### III. EXPERIMENTAL RESULTS

#### A. Experimental results on simulated data

Two classes of time series data: The zero mean AR(1) time series is generated with AR coefficients uniformly distributed in the ranges  $(0.30 \pm .01)$  and  $(0.60 \pm .01)$ , respectively. The noise variance is 0.01 for both classes. From each class, we generate 200 time series (100 for training and 100 for test). As expected, the AR model and LMAR model classified all the time series correctly from test data set. Now, as the AR coefficients are moved closer and closer, the classification performance of the AR model and LMAR model is studied. From Figure 2 we infer that LMAR model is robust compared to AR model.

#### B. Abnormal ECG pattern diagnosis

The MIT-BIH ECG database is the on-line component of the research resource for complex cardiac and other physiological signals. For our studies, the data is obtained from the MIT-BIH arrhythmia database, MIT-BIH ventricular arrhythmia database and MIT-BIH supraventricular database. These together have following kinds of ECG signals: Normal Sinus Rhythm (NSR), Atrial Premature Contraction (APC), Premature Ventricular Contraction (PVC), Supra Ventricular Contraction (SVT), Ventricular Tachycardia (VT), and Ventricular Fibrillation (VF).

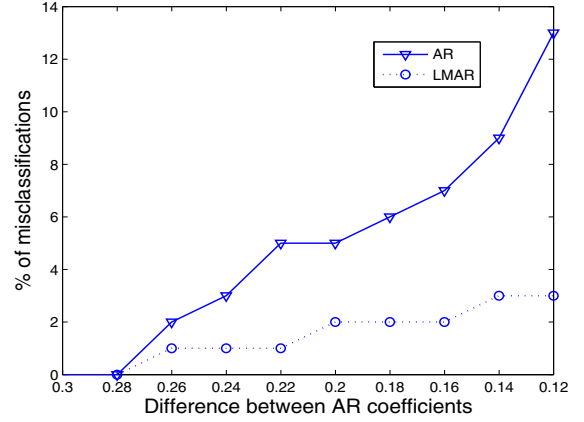


Fig. 2. The comparison of time series classification performance for LMAR model and generative AR model on simulated data

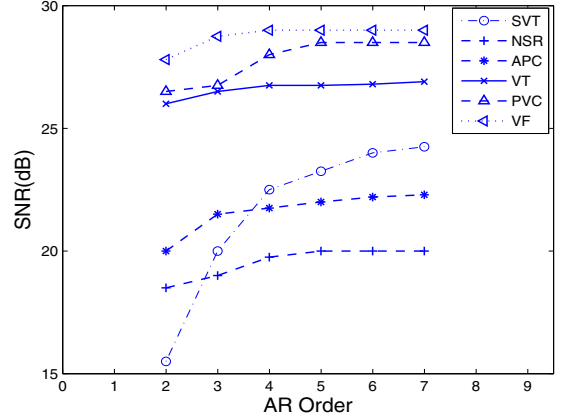


Fig. 3. Signal to noise ratio for different class time series

1) *Preprocessing*: The sampling frequencies of these databases are different: arrhythmia database: 360 Hz, Ventricular database: 250 Hz and supraventricular database: 128 Hz. Therefore, the arrhythmia and supraventricular databases were re-sampled to 250 Hz. It is important to remove the noise from the ECG time series data due to the power interference, respiration, spike, etc., and detect the R peaks in the ECG signals. The R peaks of ECG were detected using the Tompkin's algorithm. The time series data length in AR modeling should be carefully selected, and the length should cover at least one cardiac cycle. The cardiac cycle lengths vary for different arrhythmias and normal cardiac signals. In normal ECG of an adult, the heart rate ranges between 60 – 100 beats per minute. In APC the RR interval is shorter and in VF it is shorter compared to NSR. Therefore, in the current study 300 sample length time series is used in AR modeling.

Also, we perform differencing (third order) of the ECG signal to accommodate the non-stationary nature of the ECG signals. The effective length of the time series after differencing

TABLE I  
CLASSIFICATION PERFORMANCE OF LMAR MODEL

Classes(↓) (→)	NSR	APC	PVC	SVT	VF	VT
NSR	98.88	0.39	0.33	0.28	0.06	0.06
APC	0.25	97.13	0.61	1.8	0.11	0.1
PVC	0.11	0.31	98.06	0.55	0.55	0.42
SVT	0.22	0.03	0.8	98.77	0.13	0.05
VF	0.20	0	0.01	0.02	97.66	2.11
VT	0.14	0	0.02	0.17	2.44	97.23

ing is 297.

2) *Parameters*: The appropriate AR model order  $P$  is crucial for capturing the dependencies in the signal. The higher value of  $P$  will lead to capturing of noise and lower value of  $P$  could lead to insufficient capturing of signal characteristics. The Signal to Noise Ratio (SNR) is utilized for deciding the appropriate order of the AR modeling. The SNR is defined to be:

$$SNR = 10 \log \frac{\sum_1^M (x(t))^2}{\sum_1^M (x(t) - \hat{x}(t))^2} \quad (11)$$

3) *Abnormal ECG pattern diagnosis*: In this problem of abnormal ECG pattern diagnosis we model different classes using AR model and optimize the boundary between them for minimum number of misclassifications. Genetic algorithm is applied for optimizing the parameters. The classification performance of diagnosis of different ECG patterns shown in the Table I. These results are significantly better than the results published in [1] where AR coefficients are used as features (93 to 96 %).

#### IV. CONCLUSION

In this work we proposed a new time series classification method. The proposed method is inspired from discriminative training of generative models, which has richness of generative models and performance of discriminative methods. An AR model is proposed to represent each class. The boundary between them is further optimized using the large margin concepts. The proposed method is applied on ECG data and found to perform better than the methods which make use of AR modeling as a feature extraction step.

However, in certain cases it may not be possible to represent each class using a single AR model reliably. In such cases, it is required to represent each class using mixture of AR models [15]. Therefore, in future we propose to utilize mixture AR model to represent each class. Also, we plan to apply LMAR model for classifying other benchmark time series datasets.

#### REFERENCES

- [1] D Ge , N Srinivasan and S M Krishnan: Cardiac arrhythmia classification using autoregressive modeling. BioMedical Engineering OnLine: <http://www.biomedical-engineering-online.com/content/1/1/5>, 2002.
- [2] Coast DA, Stren RM, Cano GG, Briller SA: An approach to cardiac arrhythmia analysis using hidden Markov models. IEEE Trans Biomed Eng, 37:826-836, 1990.
- [3] Ham FM, Han S: Classification of cardiac arrhythmias using fuzzy ARTMAP. IEEE Trans Biomed Eng, 43:425-430, 1996.
- [4] Guvenir HA, Acar B, Demiroz G, Cekin A: A supervised learning algorithm for arrhythmia analysis. Comp Cardiol, 24:433-436, 1997.

TABLE II  
GENETIC ALGORITHM FOR OPTIMIZING THE AR COEFFICIENTS

```

begin
  t ← 0;
  initialize Ξ(t);
  evaluate Ξ(t);
  find the best and the worst chromosome of Ξ(t);
  while t < a
    predetermined number of iteration do
      select two parents from Ξ(t);
      generate two offspring by crossover and mutation;
      evaluate the offspring;
      if the offspring is fitter than the worst parent then
        select a parent except excluding the best one and replace it;
      find the best and the worst chromosome of Ξ(t);
      t ← t + 1;
    end
  end

```

- [5] Mukhopadhyay S, Sircar P: Parametric modelling of ECG signal. Med Biol Eng Comp, 34:171-173, 1996.
- [6] Pinna GD, Maestri R, Cesare AD: Application of time series spectral analysis theory: Analysis of cardiovascular variability signals. Med Biol Eng Comp, 34:142-148,1996.
- [7] Oksana Yakhnenko, Adrian Silvescu and Vasant Honavar: Discriminatively Trained Markov Model for Sequence Classification, International Conference on Data Mining, 498-505, 2005.
- [8] Arnold M, Miltner WHR, Witte H: Adaptive AR modeling of nonstationary time series by means of Kalman filtering. IEEE Transactions on Biomed Eng , pp. 45:553-562, 1998.
- [9] Silipo R, Marchesi C: Artificial neural networks for automatic ECG analysis. IEEE Trans Signal Processing, pp. 46:1417-1425, 1998.
- [10] M.J.F. Gales: Discriminative Models for Speech Recognition, Information Theory and Applications Workshop, UCSD, California, USA 2007.
- [11] B. Taskar and C. Guestrin and D. Koller: Max-Margin Markov Networks, Advances Neural Information Processing Systems - 16, pp.25-32, 2003.
- [12] Nai-Jen Huan and Ramaswamy Palaniappan: Neural network classification of autoregressive features from electroencephalogram signals for braincomputer interface design, Journal of Neural Eng, 142-150, 2004.
- [13] Song M H, Lee J, Sung P C, Kyoung J L, Sun K Y: Support Vector Machine Based Arrhythmia Classification Using Reduced Features. Int Journal of Control, Automation and Systems vol. 3, no. 4, pp.571-579, 2005.
- [14] Sha, F; Saul, L.K.: Large margin hidden Markov models for automatic speech recognition, IEEE Transactions on Audio, Speech, and Language Processing, Vol. 14, No. 5, pp.1584-1595, 2006.
- [15] Yimin Xiong, Dit-Yan Yeung: Time series Clustering with ARMA Mixtures, Journal of Pattern Recognition vol. 37, iss. 8, pp. 1675-1689, 2004.

#### APPENDIX

By eliminating slack variables the optimization function involved in large margin AR model (Eqn. 10) can be rewritten as: ( $\mathbf{a} = \langle a_1, \dots, a_C \rangle$  are decision variables)

$$\Xi(\mathbf{a}) = \sum_{n, c \neq y_n} \max(0, 1 + \mathbf{x}_n^t (\Phi_{y_n} - \Phi_c) \mathbf{x}_n) + \gamma \sum_c \text{trace}(\Phi_c) \quad (12)$$

The  $\Phi_c$  is solely characterized using ensemble AR coefficients ( $a_c$ ), initialized based on coefficients of the ensemble models for each class. The Table II provides genetic algorithm for optimizing the above function.