Applied Soft Computing 13 (2013) 361–371

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Large margin mixture of AR models for time series classification

B. Venkataramana Kini*, C. Chandra Sekhar

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

ARTICLE INFO

Article history:
Received 18 October 2011
Received in revised form 10 June 2012
Accepted 7 August 2012
Available online 14 September 2012

Keywords:
Large margin autoregressive model
Large margin mixture autoregressive model
Time series classification
Outlier detection
Rejection option
Generative and discriminative hybrid models

ABSTRACT

In this paper, we propose the large margin autoregressive (LMAR) model for classification of time series patterns. The parameters of the generative AR models for different classes are estimated using the margin of the boundaries of AR models as the optimization criterion. Models that use a mixture of AR (MAR) models are considered for representing the data that cannot be adequately represented using a single AR model for a class. Based on a mixture model representing each class, we propose the large margin mixture of AR (LMMAR) models. The proposed methods are applied on the simulated time series data, electrocardiogram data, speech data for E-set in English alphabet and electroencephalogram time series data. Performance of the proposed methods is compared with that of support vector machine (SVM) based classifier that uses AR coefficients based features. The proposed methods give a better classification performance compared to the SVM based classifier. Being generative models, the LMAR and LMMAR models provide a generative interpretation that enables utilization of the rejection option in the high risk classification tasks. The proposed methods can also be used for detection of novel time series data.

1. Introduction

Time series form an important class of data objects in tasks such as speech recognition and medical signal analysis. An important property shared by time series data 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 time series is generated as a linear combination of previous values and a probability density function can be defined on the AR model [1]. The AR model is widely used in the tasks such as time series prediction and parametric spectrum estimation. In the current work, our objective is to utilize AR models for time series classification.

Time series classification is an important problem in pattern recognition. In time series classification, a set of time series with class labels is given. This data is used to build a model for each class (training phase). When a new time series is given, an appropriate label is assigned to it (testing phase).

Pattern recognition methods for classification can be broadly divided into generative and discriminative methods. Generative models learn a joint probability distribution $p(\mathbf{x}, y)$, of the input $\mathbf{x}$ and the label y, and perform classification using Bayes rule, $y = arg\max_i p(y_i|\mathbf{x})$. Discriminative models learn the mapping from the input $\mathbf{x}$ to the label y. The generative model provides a conducive framework for imposing structure and prior knowledge on a given problem. Due to the generative nature of these models, the outliers (data points that do not belonging to any of the known classes) can be detected easily and uncertain classifications can be referred to domain experts. Discriminative methods give a superior performance by focusing on the given task of classification [2]. Discriminative models have been used in diverse time series classification tasks [3–6].

The black-box nature of discriminative models makes incorporation of the structure of the problem difficult. For example, in time series classification using the discriminative models, it is not possible to make use of the temporal correlation present in a time series directly. Discriminative models have to depend on the temporal methods such as AR modeling for feature extraction. This motivates combining these two methods synergistically, leading to a hybrid method that retains the richness of generative models, at the same time providing a superior classification performance. There is an evidence that the models constructed in this manner can capture the subtleties of the time series data being analyzed [7]. In the hybrid method, the parameters of the generative model are estimated to maximize the classification performance rather than maximizing the likelihood of the training data [7–9].

Recently, there has been a significant interest in the discriminative training of generative models for classification tasks, particularly in the area of speech recognition (for a review see [7–11]). Most of these works use hidden Markov models or Gaussian mixture models as the generative models. The current work is on discriminatively training AR models for time series classification.

* Corresponding author.
E-mail addresses: venkataramana.kini@gmail.com (B.V. Kini), chandra@cse.iitm.ac.in (C.C. Sekhar).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.08.027

Also, the probabilistic kernel functions can be used to capture temporal characteristics of time series [12,13]. These methods develop temporal models such as hidden Markov model and linear dynamic model for each time series data and then compute Kullback–Leibler (KL) divergence between these models. These methods cannot be considered to perform discriminative training of generative models. These methods define kernel functions in the space of generative model parameters.

In [14,15] the AR coefficients are extracted from the time series data and the discriminative classifiers are used for classification of time series using AR coefficients based features.

We propose an approach to discriminatively train the generative AR models using the large margin method [17], that retains the rich interpretation of AR models. This interpretation enables the model to utilize the rejection option in cases of high risk classification tasks and the detection of outliers when all classes are not covered in the training data set. The model uses the large margin concepts, that are utilized in support vector machines (SVM). However, unlike SVMs, large margin AR (LMAR) models retain generative interpretation. Similar to other discriminative methods, temporal dependencies cannot be directly captured in the SVMs. Therefore, AR modeling or other temporal methods are used for feature extraction.

In certain time series classification problems, some classes may not be adequately represented using a single AR (MAR) model. In such cases, it is useful to consider a mixture of AR models. We propose to build a mixture of AR models [24] for each class and further train these MAR models using the large margin method (LMMAR model).

The mixture of AR models presented in our paper is related to a mixture of ARMA models [24], which is used for time series clustering. It should be noted that in [25] a single time series is modeled using a mixture of AR models for segmenting the time series into homogeneous parts. However, in our case a set of multiple time series is modeled using a mixture of AR models.

In the next section, we present the AR model based methods for time series classification. We first present the method that uses a single AR model for each class and the large margin method for estimation of parameters of AR models. Then we present the method that uses a mixture of AR models for each class trained using the large margin method. In Section 3, we present our studies using the proposed methods for time series classification on different data sets. We compare the performance of the proposed method with that of the SVM based classifiers that use AR coefficients based features.

2. AR model based methods for time series classification

An AR process models the linear dependency that may exist in a given time series. It models the signal as the output of a linear system driven by white noise of zero mean and unknown variance. Autoregressive moving average (ARMA) model regresses on noise as well. However, there exists an equivalent higher order AR model. Hence, without loss of generality, AR models are considered in this paper.

Let the time series training data be: $\mathbf{X} = \{< \mathbf{x}_1\, y_1 >, < \mathbf{x}_2\, y_2 > , \ldots, < \mathbf{x}_n\, y_n >, \ldots, < \mathbf{x}_N\, y_N >\}$, $\mathbf{x}_n \in M\Re$ *and* $y_n \in \{1, 2, \ldots, C\}$. Here $\mathbf{x}_n = [x_n(1), x_n(2), \ldots, x_n(M)]^T$ is the nth time series of length M, y_n is the corresponding class label and C is the number of classes.

Using an AR model with order P, the value of time series $\mathbf{x}_n$ at discrete time t can be represented as:

$$x_n(t) = -\sum_{p=1}^{P} a_{np} x_n(t-p) + e_n(t) = \hat{x}_n(t) + e_n(t) \quad (1)$$

where $e_n(t) \sim \mathcal{N}(0, \sigma^2)$ is the zero mean white noise with σ^2 as variance, and $\mathbf{a}_n = [a_{n1}, a_{n2}, \ldots, a_{nP}]^T$ are the AR coefficients.

The autocorrelation function (ACF) of $\mathbf{x}_n$ at lag p is estimated using $r_{np} = \sum_t x_n(t)x_n(t+p)$, $p = 1, \ldots, P$ and represented as $\mathbf{r}_n = [r_{n1}, \ldots, r_{nP}]^T$. The temporal characteristic of a time series can be captured using its ACF [1]. The variance of time series, r_{n0}, estimated using $\sum_t x_n(t)x_n(t)$ gives its instantaneous characteristic.

Since $e_n(t) \sim \mathcal{N}(0, \sigma^2)$, the probability density function (pdf) of $\mathbf{x}_n$ can be written as [1,18]:

$$p(\mathbf{x}_n|\mathbf{a}_n, \sigma^2) = (2\pi\sigma^2)^{-M/2} exp(-0.5\sigma^{-2}\sum_{t=1}^{M} e_n^2(t)) = (2\pi\sigma^2)^{-M/2} exp(-0.5\sigma^{-2}\mathbf{a}_n^T \Sigma_n \mathbf{a}_n) \quad (2)$$

where the autocorrelation matrix, $\mathbf{\Sigma}_n$, is defined as

$$\Sigma_n = \begin{pmatrix} 1 & r_1 & r_2 & \ldots & r_{P-1} \\ r_1 & 1 & r_1 & \ldots & r_{P-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{P-1} & r_{P-2} & \ldots & r_1 & 1 \end{pmatrix}_n \quad (3)$$

Using Yule–Walker (Y–W) equations [1], the AR coefficients $\mathbf{a}_n$ can be derived from the autocorrelation function $\mathbf{r}_n$ and the autocorrelation matrix $\mathbf{\Sigma}_n$ as $\mathbf{a}_n = {\mathbf{\Sigma}_n}^{-1}\mathbf{r}_n$.

Therefore, (2) can be written as:

$$p(\mathbf{x}_n|\mathbf{r}_n) \propto \exp\left(-\frac{1}{2}\mathbf{r}_n^T {\Sigma_n}^{-1}\mathbf{r}_n\right) \quad (4)$$

The autocorrelation matrix Σ_n represents the temporal structure of one time series $\mathbf{x}_n$. That is, the inherent assumption in Eq. (4) is that AR process is ergodic. We can relax this assumption by using multiple time series from the same class for estimating the ensemble autocorrelation matrix $\mathbf{\Sigma}_c$ for class c. The ensemble average provides a robust estimate of the AR process compared to parameters estimated with ergodic assumption. We propose that each class is an AR process represented by the AR model design matrix $\mathbf{\Sigma}_c$ and the quantity $\mathbf{r}_n^T \mathbf{\Sigma}_c^{-1}\mathbf{r}_n$ is similar to the squared Mahalanobis distance. Using such models for different classes, $c \in \{1, 2, \ldots, C\}$, the classification of a new time series $\mathbf{x}$ with $\mathbf{r}$ as its ACF can be performed using the following decision rule:

$$y = arg\min_c \{\mathbf{r}^T \mathbf{\Sigma}_c^{-1}\mathbf{r}\} \quad (5)$$

It is to be noted that the autocorrelation matrix $\mathbf{\Sigma}_c$ has the Toeplitz structure and can be characterized only by the ACFs for the examples of the class. In the next subsection, we develop the large margin AR model for classification of time series data.

2.1. Large margin AR model

In the large margin AR (LMAR) model, we propose to maximize the margin (distance of the nearest training example from the decision boundary) by optimizing the parameters involved, i.e., the ACFs representing each class. The ACFs are found not only to classify the training data correctly, but also to place the decision boundaries optimally. We propose to constrain each time series in the training data to be at least one unit distance away from the decision boundary of each of the competing classes (similar to what is done in case of large margin Gaussian mixture model [10]). For a time series $\mathbf{x}_n$ with its class label as y_n, the constraints are as follows:

$$\mathbf{r}_n^T(\mathbf{\Phi}_c - \mathbf{\Phi}_{y_n})\mathbf{r}_n \geq 1, \quad \forall c \neq y_n \quad (6)$$

where $\mathbf{\Phi}_c = \mathbf{\Sigma}_c^{-1}$ and $\mathbf{\Phi}_{y_n} = \mathbf{\Sigma}_{y_n}^{-1}$. The model is regularized by imposing the smallest parameter constraint, i.e., by minimizing the sum of the traces of the matrices $\mathbf{\Sigma}_c^{-1} = \mathbf{\Phi}_c$, $c = 1, 2, \ldots, C$. The optimization problem is

$$\begin{aligned} &\text{Minimize} \quad \sum_c trace(\mathbf{\Phi}_c) \\ &\text{subject to,} \\ &\quad 1 + \mathbf{r}_n^T(\mathbf{\Phi}_{y_n} - \mathbf{\Phi}_c)\mathbf{r}_n \le 0, \quad \forall c \ne y_n, n = 1, 2, \ldots, N \\ &\quad \text{and} \;\; \mathbf{\Phi}_c \succ 0, \quad c = 1, 2, \ldots, C \end{aligned} \tag{7}$$

The positive definiteness constraint imposed on $\mathbf{\Phi}_c$ ensures that it represents a valid AR model. The optimal parameters of AR models for different classes are estimated by solving this optimization problem. However, it may not be feasible to classify all the training examples correctly, due to the presence of outliers. To handle such cases, as in SVMs, we introduce the non-negative slack variables ξ_{nc} into the optimization problem:

$$\begin{aligned} &\text{Minimize} \quad \sum_n \sum_c \xi_{nc} + \gamma \sum_c trace(\Phi_c) \\ &\text{subject to,} \\ &\quad 1 + \mathbf{r}_n^T(\mathbf{\Phi}_{y_n} - \mathbf{\Phi}_c)\mathrm{r}_n \le \xi_{nc}, \quad \forall c \ne y_n, \quad n = 1, 2, \ldots, N \\ &\quad \xi_{nc} \ge 0, n = 1, 2, \ldots, N \text{ and } c = 1, 2, \ldots, C \\ &\quad \text{and} \;\; \mathbf{\Phi}_c \succ 0, \; c = 1, 2, \ldots, C \end{aligned} \tag{8}$$

This constrained optimization can be reduced to unconstrained optimization:

$$\Xi(\mathbf{r}_{c=1,\ldots,C}) = \sum_{n, c \ne y_n} max(0, 1 + \mathbf{r}_n^T(\Phi_{y_n} - \Phi_c)\mathbf{r}_n) + \gamma \sum_c trace(\Phi_c) \tag{9}$$

and the optimization problem is solved using quadratic optimization solvers [9,19–21]. The Φ matrices are ensured to be positive definite by performing eigen decomposition and reconstructing Φ by using only positive eigen values after every optimization step. The value of trade-off parameter γ is determined empirically.

The LMAR model based classification of time series proposed here is different from the large margin Gaussian model based classification of i.i.d. data in [22]. The LMAR method explicitly captures the autocorrelation information in time series data and optimizes boundary between ensemble autoregressive models. It can handle variable length and nonstationary time series data (after performing differencing operation). The LMAR method is useful when it is possible to represent each class by the autocorrelation structure of time series data of that class.

2.2. Mixture of AR (MAR) models

When the data of a class cannot be adequately represented using a single AR model, it is necessary to represent it using a mixture of AR models. In this section, we present the maximum likelihood (ML) method for estimation of parameters of a mixture of AR models. Let $\mathbf{X}_c = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_c}\}$ be a set of time series data from class c. Let us assume that the time series data is generated by K different AR models, which correspond to the K mixture components. The responsibility of generating $\mathbf{x}_n$ by kth component is given by $p(\mathrm{x_n}|\mathbf{a}_k, \sigma_k^2)$. For the entire mixture, it is given by a convex combination of such responsibilities:

$$p(\mathrm{x_n}) = \sum_{k=1}^{K} \alpha_k p(\mathrm{x_n}|\mathbf{a}_k, \sigma_k^2) \tag{10}$$

where α_ks are the mixture coefficients satisfying the condition $\sum_{k=1}^{K} \alpha_k = 1$. The ML estimates of the parameters $\hat{\mathbf{a}}_k, \hat{\sigma}_k^2$ for $k = 1, 2, \ldots, K$ can be obtained by maximizing the log likelihood of the data of class c:

$$L_c(X_c) = \sum_{n=1}^{N_c} ln \sum_{k=1}^{K} \alpha_k p(\mathbf{x}_n|\mathbf{a}_k, \sigma_k^2) \tag{11}$$

This parameter estimation problem does not have a closed form solution. However, the above problem can be solved using the expectation–maximization (EM) method. In this method, the component labels are the latent variables. The mixture coefficients and the AR model parameters for each component are estimated by maximizing the conditional expectation of log-likelihood of the joint density of data and component labels, given the data and the current estimates of the parameters.

Below, we formulate the EM framework for estimation of parameters of an MAR model.

2.2.1. Estimation of parameters of MAR model using EM algorithm

The latent variable $\mathbf{z}_n$ for a time series $\mathbf{x}_n$ is defined as a 1-of-K vector $\mathrm{z_n} = [z_n^1, z_n^2, \ldots, z_n^K]^T$ such that $z_n^k = 1$ and $z_n^j = 0$, for $j \ne k$. This indicates that $\mathbf{x}_n$ was generated by the component k. The complete log-likelihood function for the missing data $\mathbf{Z}_c = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{N_c}\}$ and the dataset $\mathbf{X}_c$ can be written as:

$$L_c((\mathbf{X}_c, \mathbf{Z}_c)|\boldsymbol{\alpha}, \boldsymbol{\theta}) = ln \prod_{n=1}^{N_c} p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\alpha}, \boldsymbol{\theta}) = \sum_{n=1}^{N_c} \sum_{k=1}^{K} z_n^k ln(\alpha_k p(\mathbf{x}_n|\boldsymbol{\theta}_k)) \tag{12}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]^T$, $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_K\}$, and $\theta_k = \{\mathbf{a}_k, \sigma_k^2\}$.

The EM algorithm consists of two steps. The first step is referred to as E-step which is an evaluation of the conditional expectation of L_c given the data X_c and the current estimates of the parameters. The second step, called as M-step, constitutes estimating the unknown parameters by maximizing the conditional expectation. The two steps are repeated until the convergence is achieved.

E-step: The conditional expectation $Q(\boldsymbol{\alpha}, \boldsymbol{\theta}|\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}})$ of L_c given the data X_c and the current estimates of the parameters $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\theta}}$ is:

$$Q(\boldsymbol{\alpha}, \boldsymbol{\theta}|\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}}) = E[L_c(\boldsymbol{\alpha}, \boldsymbol{\theta}, \mathbf{X}_c, \mathbf{Z}_c)|\mathbf{X}_c, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}}] \tag{13}$$

$$= \sum_{n=1}^{N_c} \sum_{k=1}^{K} ln(\alpha_k p(\mathbf{x}_n|\boldsymbol{\theta}_k)) E[z_n^k|\mathbf{X}_c, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}}] \tag{13}$$

Since z_n^k is a binary variable, the required expectation is evaluated as follows:

$$w_n^k = E[z_n^k|\mathbf{X}_c, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}}] = P(z_n^k = 1|\mathbf{X}_c, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}}) = \frac{p(\mathbf{x}_n|\hat{\theta_k})\, P(z_n^k = 1)}{p(\mathbf{x}_n)}$$

$$= \frac{p(\mathbf{x}_n|\hat{\theta_k})\, \hat{\alpha_k}}{\sum_{r=1}^{K} p(\mathbf{x}_n|\hat{\theta_r})\, \hat{\alpha_r}} \tag{14}$$

The conditional expectation can be written as

$$Q(\boldsymbol{\alpha}, \boldsymbol{\theta}|\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}}) = \sum_{n=1}^{N_c} \sum_{k=1}^{K} w_n^k\, ln(\alpha_k p(\mathbf{x}_n|\boldsymbol{\theta}_k)) \tag{15}$$

M-step: The updated parameters in the next iteration are estimated by maximizing the above conditional expectation. The mixing coefficients are estimated as:

$$\hat{\alpha}_k = \frac{1}{N_c}\sum_{n=1}^{N_c} w_n^k,\ k = 1, \ldots, K \tag{16}$$

The AR model parameters are obtained by maximizing the Q function w.r.t. θ as follows:

$$\hat{\theta} = \underset{\theta}{arg\max} \sum_{n=1}^{N_c}\sum_{k=1}^{K} w_n^k \ln(\alpha_k p(\mathrm{x_n}|\theta_k)) \tag{17}$$

Substituting for $p(\mathbf{x}_n|\theta_k)$

$$Q(\alpha, \theta|\hat{\alpha}, \hat{\theta}) = \sum_{n=1}^{N_c}\sum_{k=1}^{K} w_n^k \left\{ ln(\alpha_k) + ln(2\pi\sigma_k^2)^{-M/2} + \frac{-1}{2\pi\sigma_k^2}\sum_{t=1}^{M}(\mathbf{a}_k^T\mathbf{x}_{nt} + x_{nt})^2 \right\} \tag{18}$$

Here, $\mathbf{x}_{nt} = [x_{n,t-1}, x_{n,t-2} \ldots x_{n,t-p}]^T$. By differentiating Q w.r.t $\mathbf{a}_k$ and σ_k^2, respectively, setting equal to zero:

$$\frac{\partial Q}{\partial \mathbf{a}_k} = \sum_{n=1}^{N_c} w_n^k \frac{\partial}{\partial \mathbf{a}_k}\left\{ \sum_{t=1}^{M} \mathbf{a}_k^T\mathbf{x}_{nt}\mathbf{x}_{nt}\mathbf{a}_k + 2\mathbf{a}_k\mathbf{x}_{nt}x_{nt} \right\} = 0 \tag{19}$$

$$\frac{\partial Q}{\partial \sigma_k^2} = \sum_{n=1}^{N_c} w_n^k \left\{ \frac{-M}{2}\frac{1}{\sigma_k^2} + \frac{1}{(\sigma_k^2)^2}\sum_{t=1}^{M}(e_n(t))^2 \right\} = 0 \tag{20}$$

the kth component AR model parameters are estimated as:

$$\hat{\mathbf{a}}_k = -\left(\sum_{n=1}^{N_c} w_n^k \mathbf{\Sigma}_n \right)^{-1} \sum_{n=1}^{N_c} w_n^k \mathbf{r}_n \tag{21}$$

$$\hat{\sigma_k^2} = \frac{\sum_{n=1}^{N_c} w_n^k \sum_{t=1}^{M}(e_n(t))^2}{M\sum_{n=1}^{N_c} w_n^k} \tag{22}$$

The E and M steps are repeated until the relative decrease in Q between two iterations is below a specified threshold.

The EM algorithm for MAR model gives the estimated AR coefficients for each mixture. These AR coefficients are converted to autocorrelation coefficients using Y–W equations (P equations with P unknowns). The autocorrelation coefficients are used in forming autocorrelation matrix and in turn in the inverse autocorrelation matrix.

2.3. Large margin mixture of AR (LMMAR) models

Let $\mathbf{\Phi}_{ck}$ denote the matrix for the kth component in the MAR model for the class c. Every time series $\mathbf{x}_n$ in the training data has a mixture label k_n (the component with highest posterior probability) together with the class label y_n. The objective in the large margin method is to ensure that each time series is closer to its target class than any other class. For each labeled time series ($\mathbf{x}_n$, y_n, k_n), the constraints are as follows:

$$\mathbf{r}_n^T(\Phi_{ck} - \Phi_{y_n k_n})\mathbf{r}_n \geq 1, \quad \forall c \neq y_n \tag{23}$$

By using the soft-max inequality, we can rewrite the above inequality as:

$$-\log\sum_{k}\exp^{-\mathbf{r}_n^T\Phi_{ck}\mathbf{r}_n} - \mathbf{r}_n^T\Phi_{y_n k_n}\mathbf{r}_n \geq 1, \quad \forall c \neq y_n \tag{24}$$

Therefore, the optimization problem for LMMAR model can be stated as follows:

$$\begin{aligned}
&\text{Minimize} \quad \sum_n\sum_c \xi_{nc} + \gamma\sum_c\sum_k trace(\Phi_{ck}) \\
&\text{subject to,} \\
&\quad 1 + \mathrm{r_n}^T\Phi_{y_n k_n}\mathrm{r_n} + \log\sum_k \exp^{-\mathbf{r}_n^T\Phi_{ck}\mathbf{r}_n} \leq \xi_{nc}, \quad \forall c \neq y_n, \quad n = 1, 2, \ldots, N \\
&\quad \xi_{nc} \geq 0, \quad n = 1, 2, \ldots, N \text{ and } c = 1, 2, \ldots, C \\
&\quad \Phi_{ck} \succ 0, \quad c = 1, 2, \ldots, C, \quad k = 1, 2, \ldots, K
\end{aligned} \tag{25}$$

The method for estimation of parameters of the LMMAR model is similar to that of the LMAR model. However, compared to the LMAR models, the LMMAR model involves a two-step procedure, first estimating the MAR component labels using the EM algorithm and then optimizing the parameters for large margin. Though we started with AR model as underlying model for time series data, the optimization function for the LMMAR model is similar to that of the LMGMM model [22]. However, the LMGMM and LMMAR models differ in the following respects:

- The LMMAR model effectively captures temporal dependency in terms of autocorrelation present in time series data. The LMGMM considers the data to be independent and identically distributed.
- Unlike LMGMM (which involves inverse of covariance matrix as parameter of optimization), the autocorrelation matrix has to be Toeplitz matrix throughout the optimization for representing a valid AR model. The Toeplitz structure of autocorrelation matrix is preserved by having the autocorrelation vector as the parameter of optimization instead of inverse autocorrelation matrix as parameter of optimization.

In the next section, we apply these models on simulated and real world datasets of time series.

3. Experiments and results

We perform 5-fold cross validation to find the classification performance of different models. The different models that we study are: AR model, LMAR model, MAR model, LMMAR model and SVM trained on AR coefficient based cepstral features.

In LMAR, LMMAR and SVM models, the γ hyperparameter has to be found. Also, for nonlinear SVM the parameter of the kernel function has to determined. The grid search [35] is used for finding the appropriate γ and parameters of the kernel functions.

3.1. Selection of AR model order and number of AR components

For AR and LMAR models, the important parameter that needs to be selected is the AR model order. In case of MAR and LMMAR models, we also need to select the number of components in the MAR model representing each class.

3.1.1. AR model order

The appropriate AR model order P is crucial for capturing the dependencies in the signal. A high value of P will lead to capturing of noise. A low value of P could lead to insufficient capturing of signal characteristics. The signal to noise ratio (SNR) is used for deciding the appropriate order of the AR modeling. The SNR for a signal $\mathbf{x}$ is defined to be:

$$\mathrm{SNR} = 10\log_{10}\frac{\sum_{t=1}^{M}(x(t))^2}{\sum_{t=1}^{M}(x(t) - \hat{x}(t))^2} \tag{26}$$

where $\hat{x}(t)$ is the prediction made using an AR model. The AR model order P for which the SNR is the highest is chosen.

3.1.2. Number of components in MAR model

The Bayesian information criterion (BIC) is used to select the number of components in the MAR model representing a particular class. The BIC [24] for MAR model of class c is computed as follows:

$$\text{BIC} = \sum_{n=1}^{N_c} \ln \left[\sum_{k=1}^{K} \alpha_k p(\mathbf{x}_n|\theta_k) \right] - \frac{1}{2}(K(P+1)+K)\ln N_c \quad (27)$$

The SNR and BIC based methods for choosing hyperparameters provide parsimonious values of the AR model order and number of components from optimal representation point of view. These methods do not guarantee the optimal classification performance. This necessitates validation of the models using cross validation method. We vary the AR model order and number of components around the values provided by the SNR and BIC based methods.

3.2. Study on simulated time series data

In this study we consider AR models of first order. For the first order AR model, the AR coefficient a and autocorrelation coefficient r_1 will be equal [1]. Also, two class classification problem is considered for simplicity. The AR coefficients for the data of class 1 are normally distributed in the range 0.2–0.4 with mean 0.3 and standard deviation equal to 0.05. Similarly for class 2 in the range 0.4–0.8 with mean 0.6 and standard deviation 0.1. The time series data is generated using Eq. (1). The noise variance (σ^2) is set to 0.01. The number of time series examples generated for each class is 100. Each time series example of a class is generated using the value of AR coefficient a in the corresponding range. The length of each time series is chosen as 50, after observing that autocorrelation coefficients can be estimated with sufficient accuracy. Consider a test time series example $\mathbf{x}$ generated using a as the value of AR coefficient. The corresponding autocorrelation coefficient vector is $\mathbf{r} = [1\ r_1]^T$. The distance of the time series $\mathbf{x}$ with its autocorrelation coefficient vector $\mathbf{r}$ to the model of a class with autocorrelation matrix $\mathbf{\Sigma}_c$ is $\mathbf{r}^T \mathbf{\Sigma}_c^{-1} \mathbf{r}$. The distance of time series with different values of r_1 (in the range 0.2–0.8, with step size 0.01) to the models of class 1 and class 2 is plotted in Fig. 1(a). The minimum distance occurs at 0.3 and 0.6 for classes 1 and 2, respectively. Also, it can be seen that the classification boundary is close to 0.45. It also shows the range of autocorrelation values in which the test time series example is misclassified.

However, with the LMAR model the classification boundary is optimized and the classification error is reduced as can be seen in Fig. 1(b). Also, the distances from the model have a generative interpretation. The time series data with autocorrelation coefficients in the boundary regions (around 0.4) can be referred to the domain experts to decide upon. Also, the outlier data can be found easily by setting an appropriate threshold on the distance. The SVM based method that uses AR coefficients based features will also be able to optimize the boundary. However, the generative interpretation of AR models is lost.

For our study on simulated time series classification using MAR and LMMAR models, we consider the two class classification problem in which the data of each class is represented using an MAR model with two components. Each component is represented using an AR model of order 1. For data of class 1, the AR coefficients for component 1 are in the range 0–0.1 (normally distributed with mean 0.05 and standard deviation 0.025) and for component 2 in the range 0.3–0.4 (with mean 0.35 and standard deviation 0.025)). For class 2, the AR coefficients for component 1 are in the range 0.1–0.3 (with mean 0.2 and standard deviation 0.05) and for component 2 in the range 0.4–0.6 (with mean 0.5 and standard deviation 0.05). The training data includes 50 time series examples generated for each component. First, the component label of each time series in both classes is determined by applying the EM method. Then the boundaries of the classes are optimized using the LMMAR method. The distance of a test time series example with its autocorrelation coefficient vector $\mathbf{r} = [1\ r_1]^T$ to each of the components in the MAR models for the two classes is plotted for different values of r_1 (with step size 0.01) in Fig. 2(a) for MAR models and Fig. 2(b) for LMMAR models. As can be seen in Fig. 2(a), the MAR model leads to misclassification of time series at the boundaries (0.1, 0.3 and 0.4). The LMMAR model finds the class boundaries close to 0.1, 0.3 and 0.4, which match with the actual class boundaries. Also, the rejection option and outlier detection can be performed by applying appropriate thresholds on distances. In this case, the linear SVM method cannot separate the classes with AR coefficients based features. One needs to use a kernel function to separate them, which will further make generative interpretation more difficult. Also, for this kind of multi-modal data, it is suboptimal to use a single AR model to represent each class as can be seen in Fig. 2(c). The time series data from one of the model components is misclassified from each class.

3.3. Study on abnormal ECG pattern classification

The electrocardiogram (ECG) describes the electrical activity of the heart. The Hotler ECG device is used for recording ECG. In routine physical checkups, the ECG signal is recorded and studied by physicians. However, for cardiac patients, ECG is recorded throughout the day. Physicians then analyze and interpret the ECG time series. As it is time consuming and tedious, there is a need for automatic ECG pattern recognition system [6,14,26].

The MIT-BIH ECG database contains data 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. The categories of ECG signals in these databases are: normal sinus rhythm (NSR), atrial premature contraction (APC), premature ventricular contraction (PVC), supra ventricular contraction (SVT), ventricular tachycardia (VT), and ventricular fibrillation (VF).

3.3.1. Preprocessing

The sampling frequencies for time series in the databases are different and are as follows – arrhythmia database: 360 Hz, ventricular database: 250 Hz and supraventricular database: 128 Hz. Therefore, the examples in arrhythmia and supraventricular databases were re-sampled to 250 Hz. It is important to detect the R peaks in the ECG signals. The R peaks of ECG were detected using the Tompkin's algorithm [23]. The length of time series data 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 the normal ECG of an adult, the heart beat rate ranges between 60 and 100 beats per minute. In APC, the RR interval is shorter and in VF it is even shorter compared to NSR. Therefore, in the current study 300 sample length time series is used in AR modeling. Differencing (third order) of the ECG signal is carried out to account for the non-stationary nature of the ECG signals. The effective length of the time series after differencing is 297. The number cardiac cycles used in this study as follows: NSR (10,000), SVT (6121), APC (2505), PVC (6183), VT (470), VF (484).

3.3.2. Selection of model hyperparameters

The average SNR levels for training data of different classes and for different AR orders are shown in Fig. 3. It is seen that the improvement in SNR is relatively less beyond the fourth order of AR model, except in the case of SVT. Therefore, we set the model order of four as initial model order in our experiments.

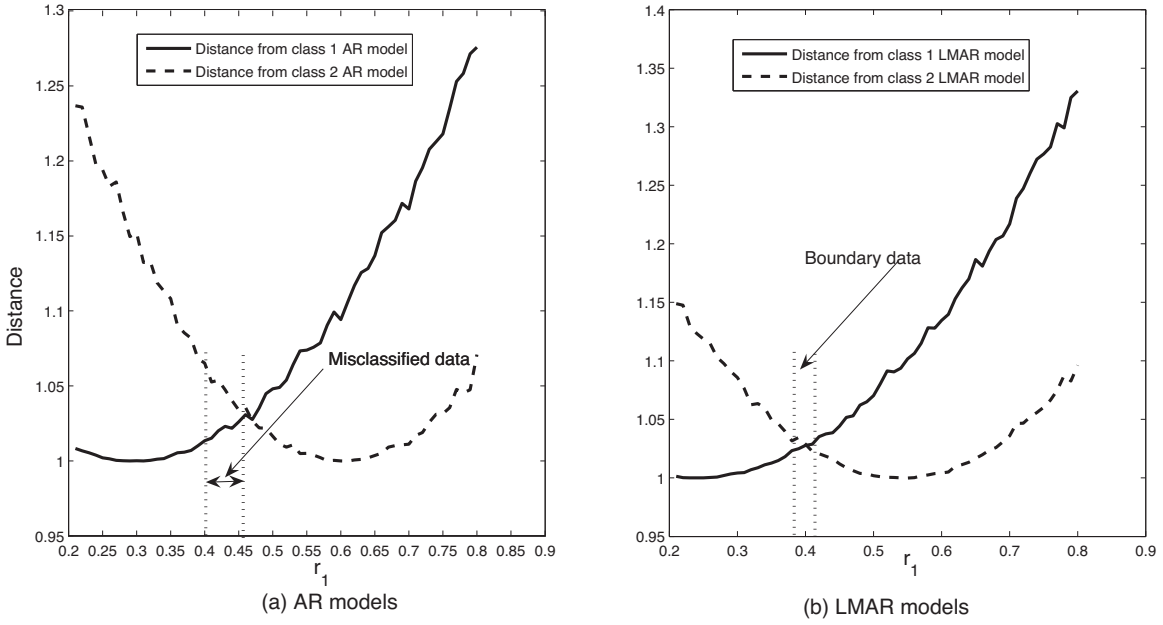

Fig. 1. The distance of a time series to the (a) AR models and (b) LMAR models of two classes, for different values of r_1 used in generating the test time series.

The BIC values for different classes with increasing number of components in the MAR model are shown in Fig. 4. It can be seen that the increase in the BIC values is insignificant for the number of components beyond three. The initial value of number of AR components is set to 3 for NSR, APC and SVT classes, and 2 for VT, VF and PVC classes.

3.3.3. Study on ECG classification using AR models

Experiments were performed with models developed in the previous sections. We studied the classification performance for different values of model order used in the AR and LMAR models. Fig. 5 shows the average fivefold classification performance for different values of model order. It can be seen that for the model order of 5, the best classification performance is achieved. The 5-fold average classification performance of the LMAR models is given in Table 1. The corresponding classification performance for the AR models is given in parentheses.

Next, we perform similar experiments with MAR models and the corresponding LMMAR models. The number of components and the AR model order used to represent data of each class are varied. The 5-fold classification performance is shown in Fig. 6. The model MAR_1 uses 2 components for NSR, APC, and SVT and 1 component for PVC, VT and VF. For MAR_2 and MAR_3 models, the number of components for each class was increased by one and two, respectively.

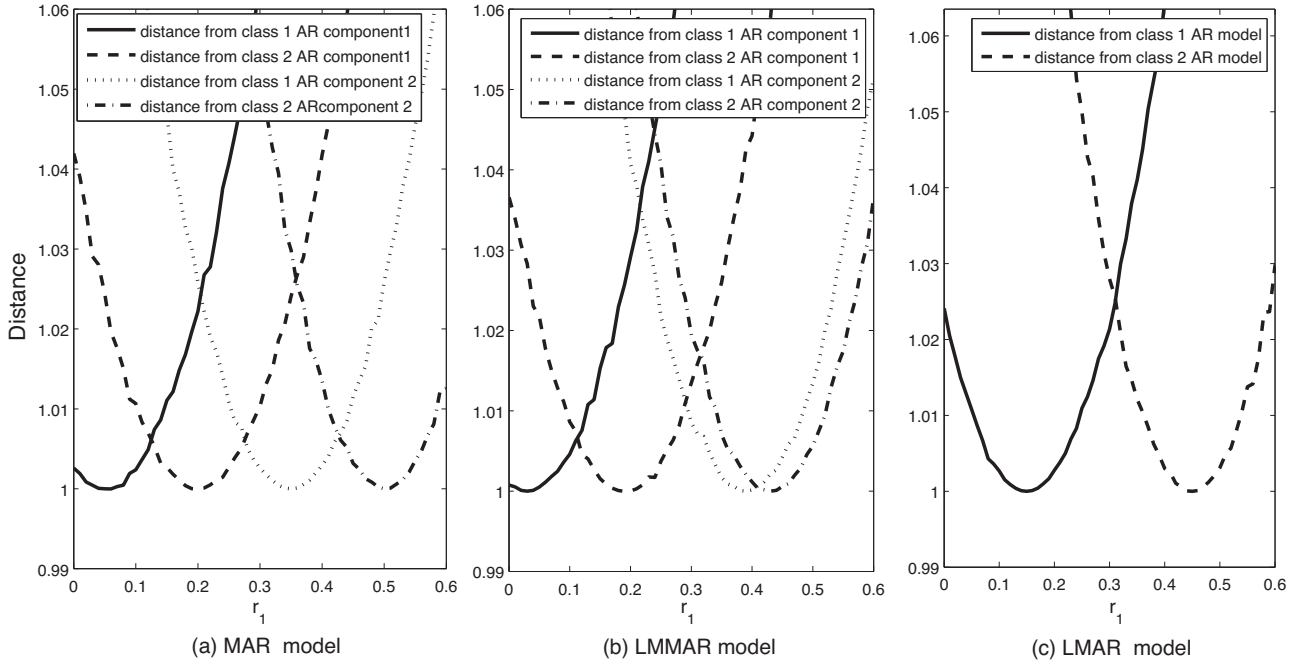

Fig. 2. The distance of a time series to the (a) MAR model components, (b) LMMAR model components and (c) LMAR model of two classes, for different values of r_1 used in generating the test time series.

Table 1
Average 5-fold classification accuracies and misclassification accuracies (in %) for LMAR model and in parentheses for AR model on ECG data.

Actual class	Hypothesized class					
	NSR	*APC*	*PVC*	*SVT*	*VF*	*VT*
NSR	98.13 (94.15)	0.83 (3.15)	0.32 (1.46)	0.44 (0.53)	0.19 (0.53)	0.09 (0.18)
APC	0.16 (2.04)	97.56 (93.12)	0.60 (1.70)	1.55 (2.31)	0.07 (0.29)	0.06 (0.54)
PVC	1.28 (0.29)	1.14 (2.29)	97.04 (93.39)	0.29 (2.27)	0.16 (1.43)	0.09 (0.33)
SVT	0.29 (2.23)	1.06 (3.04)	1.18 (1.54)	97.12 (92.06)	0.29 (0.76)	0.06 (0.37)
VF	0.19 (1.68)	0.13 (1.44)	1.03 (2.69)	0.13 (2.22)	97.26 (85.32)	1.26 (6.65)
VT	0.13 (1.21)	0.14 (1.56)	0.07 (2.33)	0.15 (3.10)	2.17 (7.70)	97.34 (84.10)

Table 2
5-Fold average classification accuracies and misclassification accuracies (in %) for LMMAR model and in parentheses for MAR model on ECG data.

Actual class	Hypothesized class					
	NSR	*APC*	*PVC*	*SVT*	*VF*	*VT*
NSR	98.99 (93.56)	0.61 (3.44)	0.13 (1.17)	0.15 (1.1)	0.06 (0.39)	0.06 (0.34)
APC	0.26 (1.19)	98.60 (94.41)	0.61 (1.34)	0.42 (1.84)	0.06 (0.63)	0.05 (0.59)
PVC	0.1 (1.67)	1.01 (2.31)	98.76 (91.92)	0.05 (1.18)	0.05 (1.24)	0.03 (1.68)
SVT	0.22 (1.24)	0.03 (1.34)	0.8 (2.48)	98.77 (92.63)	0.13 (1.25)	0.05 (1.06)
VF	0.18 (2.32)	0 (1.64)	0.01 (2.11)	0.02 (3.4)	98.51 (85.23)	1.28 (5.3)
VT	0.15 (1.4)	0.01 (2.27)	0.03 (2.14)	0.18 (3.86)	2.22 (5.33)	97.41 (85.0)

It can be seen from Fig. 6 that the best performance is obtained for the AR model order 5 and the MAR_1 model. The 5-fold classification performance for $LMMAR_1$ model is given in Table 2. The corresponding classification performance of MAR_1 model is given in parentheses.

It is seen that the large margin AR models give a significantly better performance than the AR models. The overall average classification accuracy is 90.84% for the AR models and 97.96% for the large margin AR models. Similarly, the overall classification accuracy is 91.77% for the MAR models and 98.59% for the large margin MAR models.

Rejection option: In the current application, the classification errors are not desired. Mostly, the data which are closer to the boundary are misclassified. It is appropriate to refer such uncertain cases to human experts (cardiologists). Therefore, we applied a threshold on the distance computed in Eq. (5), for the data to be classified. The threshold is set to the maximum distance corresponding to 95% of the correctly classified training data of a particular class. The overall classification accuracy is 99.73% for LMMAR, 92.16% for MAR model, 99.64% for LMAR model and 90.39% for AR model. However, 5.71% of data for LMMAR, 8.91% for MAR, 5.72% for LMAR and 8.66% for AR model were rejected as unclassified and referred further for investigation by cardiologists. This threshold can be further fine tuned for different classes depending on the risk involved in wrong classification. For example, it can be

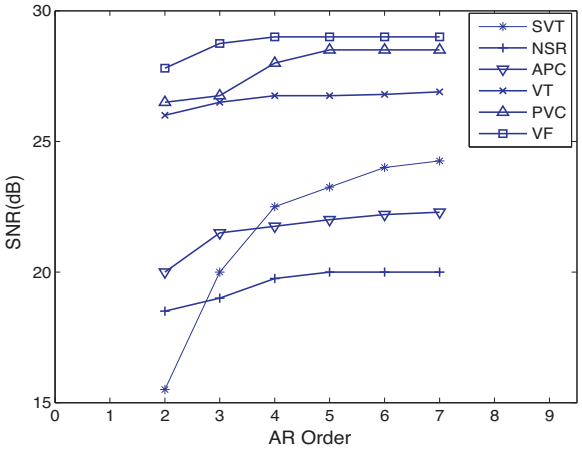

Fig. 3. Average signal to noise ratio of different ECG classes for different model orders.

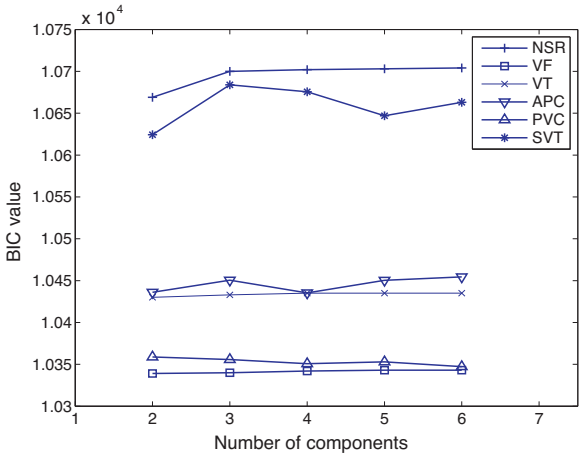

Fig. 4. BIC values of different ECG classes for different number of components used in MAR models.

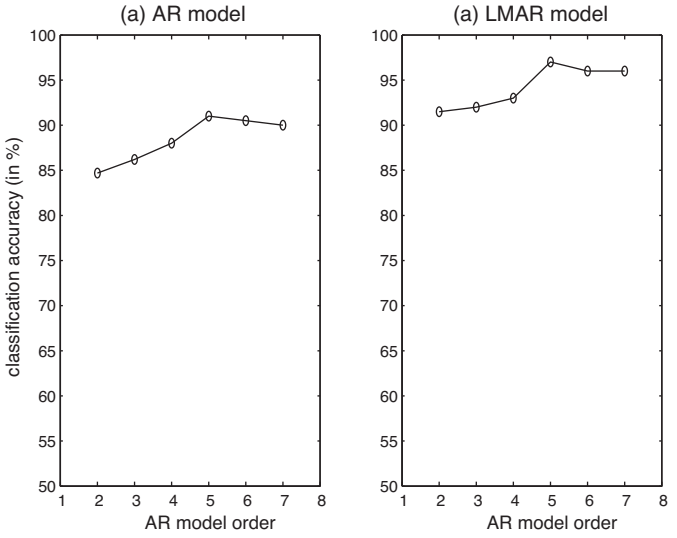

Fig. 5. Average 5-fold classification performance of (a) AR model and (b) LMAR model based classifiers for different values of model order P on ECG data.

tolerated when the NSR class data is misclassified into arrhythmias, but not vice versa.

Outlier detection: In the current application, the data of all the classes may not be available or scarce while training. In such cases, when the model is posed with the unseen class data, it is desired that the data is not classified into one of the existing classes. It should be termed as novel data. Also, it is more informative if we can say that the novel data is more similar to one of the existing classes than the others. To verify the capability of LMMAR model towards this, we performed training using the data of NSR, APC and VF classes and found the classification performance for the data of all classes. The NSR and APC classes are chosen because they are common ECG classes and the data is abundantly available for these classes. The VF class is included to check the similarity of VT and VF classes. We apply a threshold on the distance corresponding to the 94.95% of correctly classified training data of a particular classs, to determine the outliers. It was found that the average accuracies for the *NSR*, *APC* and *VF* classes are 99.14%, 98.22% and 97.93%, respectively. However, 5.98% data from these classes is rejected as outliers. Most of the data of *SVT* and *PVC* classes was classified as outliers (with 0.98% acceptance). From *VT* class, 12.31% of examples are classified as *VF* class and the rest are rejected as outliers. From this result, it is seen that the *VF* class is more similar to the *VT* class than to others. Similar outlier detection performance was obtained using LMAR model with 0.98% acceptance for *SVT* and *PVC*. The outlier detection using AR and MAR models is slightly better with 0.93% acceptance for *SVT* and *PVC*.

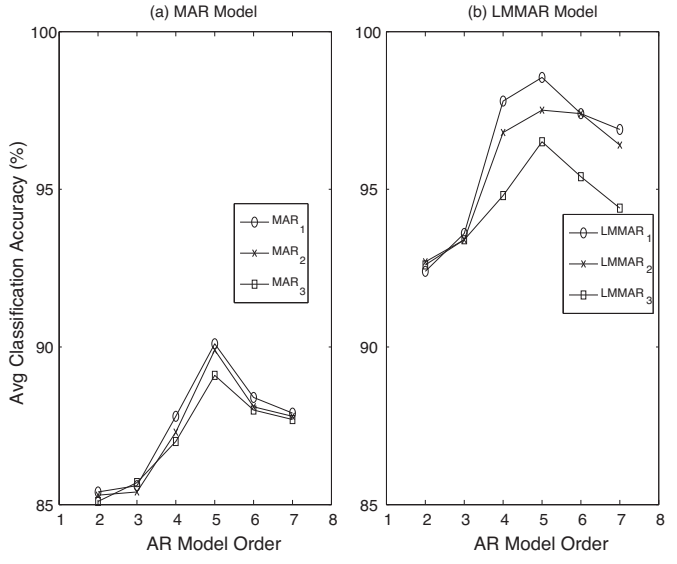

Fig. 6. Average 5-fold classification performance of (a) MAR model and (b) LMMAR model based classifiers for different values of model order P and different number of components K on ECG data.

Table 3
5-Fold average classification accuracies and misclassification accuracies (in %) for SVM using AR features on ECG data.

Actual class	Hypothesized class					
	NSR	*APC*	*PVC*	*SVT*	*VF*	*VT*
NSR	98.24	0.79	0.68	0.14	0.05	0.10
APC	0.66	98.21	0.78	0.26	0.05	0.04
PVC	0.15	1.08	98.58	0.09	0.06	0.04
SVT	0.18	0.33	0.54	98.75	0.16	0.04
VF	0.25	0.06	0.05	0.09	96.10	3.45
VT	0.35	0.09	0.09	0.06	3.21	96.20

3.3.4. Study on ECG classification using SVMs

Further, we performed classification experiments using SVMs with fifth order AR coefficients based cepstral features [16]. We used the Gaussian kernel and one-versus-one strategy for multi-class classification. The 5-fold average classification performance of SVMs is given in Table 3 for AR model order 5. It can be seen that the SVMs give a classification performance of 97.74%. This classification accuracy is less than that of the LMAR and LMMAR models. The SVMs do not carry the generative model interpretation of AR models and the large margin AR models.

3.4. E-set speech time series classification

We evaluated the proposed models on multi-speaker, utterance recognition for E-set data [31,33,34]. This data set is considered to be difficult data to classify because the classes differ only in the consonant part. In this subsection, we present the results of our study on this time series classification task.

The E-set recognition task involves recognition of spoken utterances of nine letters in English alphabet: {b, c, d, e, g, p, t, v, z}. The Oregon Graduate Institute (OGI) spoken letter database is used in this study [31]. The data set consists of 300 utterances from 150 speakers (75 male and 75 female) for each letter. From each of the speech time series, the consonant part is extracted by retaining the first 40% of the time series and discarding the rest, as in [32]. The initial AR model order is chosen to be 12 in accordance with [18].

First, we compare the results obtained by applying the AR and LMAR models. For different values of AR model order, the 5-fold classification performance of AR model and LMAR model is shown in Fig. 7. It can be seen that for the model order of 15 and 16, respectively, the AR model and LMAR models give the best classification performance. The 5-fold average classification performance for the AR and LMAR models is found to be 57.4% and 83.1%, respectively.

Similarly, for different values of AR model order and different number of components, the 5-fold classification performance of MAR model and LMMAR model is shown in Fig. 8. The initial number of components in MAR model for each class is set to be 2. It can be seen that for the model order of 14 and 15, respectively, the MAR model and LMMAR model give the best classification performance. The number of components is 3 for both models. The average classification performance for the MAR and LMMAR models is found to be 58.7% and 87.6%, respectively. For MAR model and LMMAR models, the BIC method of finding the number of components is found

Table 4
5-Fold average classification accuracies and misclassification accuracies (in %) for LMMAR and in parentheses for LMAR model on E-set data.

Actual class	Hypothesized class								
	b	*c*	*d*	*e*	*g*	*p*	*t*	*v*	*z*
b	87.1 (82.1)	0 (0.3)	5.2 (6.8)	0.9 (3.1)	0 (0.1)	1.2 (2.1)	0 (0)	4.8 (5.5)	0.8 (0)
c	0 (0.1)	90.6 (87.1)	0.3 (0.8)	0.2 (0.1)	0 (1.1)	5.7 (4.9)	0 (0)	3.2 (4.8)	0 (1.1)
d	0.2 (1.2)	2.7 (1.8)	84.1 (81.9)	0 (0.3)	0 (0)	8.2 (8.7)	1.8 (2.1)	1.6 (3.9)	1.4 (0.1)
e	0 (1.1)	0 (0.9)	1.4 (1.6)	93.6 (93.4)	0 (1.7)	4.2 (0)	0 (1.3)	0.8 (0)	0 (0)
g	2.0 (4.1)	0.3 (0.2)	1.2 (0)	0 (2.2)	92.5 (89.1)	3.1 (3.7)	0.9 (0)	0 (0.7)	0 (0)
p	1 (3.1)	1.1 (1.1)	9.2 (10.8)	0 (0.1)	2.1 (2.3)	81.4 (76.6)	5.6 (4.9)	0.3 (1.1)	0.3 (0)
t	2.2 (0.7)	3.8 (0.7)	0 (0)	0 (0)	0 (0)	8.4 (14.2)	85.2 (84.1)	0.2 (0.3)	0 (0)
v	5.7 (6.4)	3.3 (4.6)	0 (2.4)	0.2 (0)	0.8 (0)	7.1 (7.6)	0 (0)	81.8 (77.2)	1.1 (1.8)
z	0 (0)	2.1 (4.1)	0.3 (0)	0 (0)	0 (0.8)	4.6 (5.2)	0 (0)	6.2 (10.9)	86.8 (79.0)

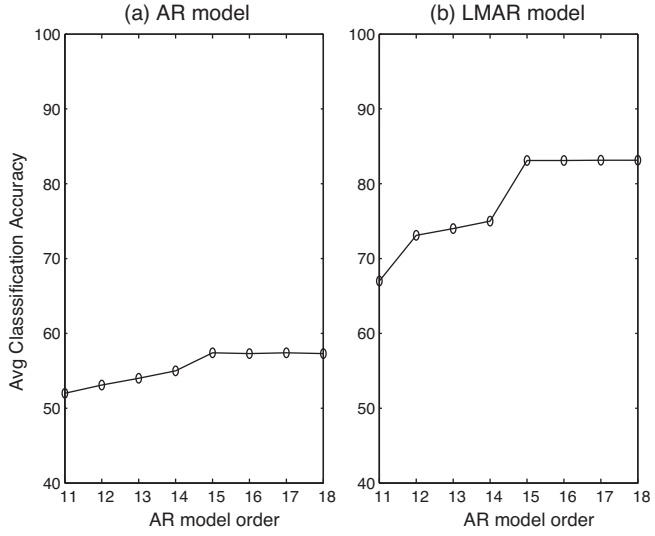

Fig. 7. 5-Fold average classification performance of (a) AR model and (b) LMAR model based classifiers for different values of model order *P* on E-set data.

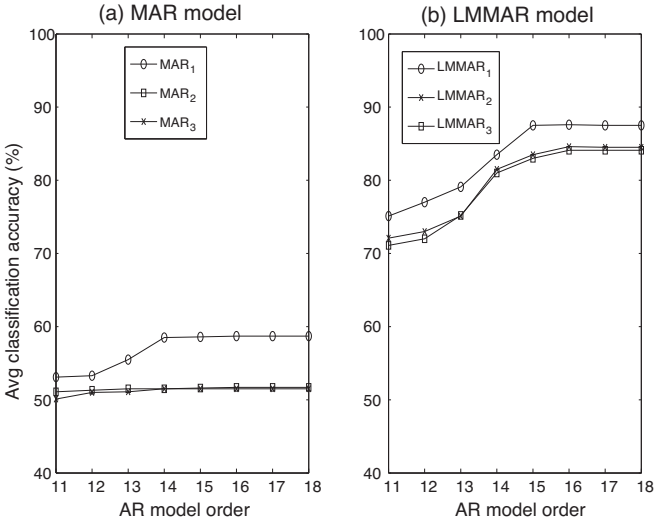

Fig. 8. 5-Fold average classification performance of (a) MAR model and (b) LMMAR model based classifiers for different values of model order *P* and different number of components *K* on E-set data.

to be not useful, as it suggested to choose approximately 5 number of components for each class. The classification performance of LMAR and LMMAR models is shown in Table 4.

With AR coefficients (order 15) based cepstral features, the average classification performance for SVM is found to be 82.1% and is shown in Table 5. From these results, we can infer that the classes *c*, *e*, *g* are easy to discriminate from the other classes. The class *p* is confused with all the other classes. From experiments, we also observe that the model size (AR model order and number of components in the mixture) has to be sufficiently large enough for good discrimination between different classes.

Among all classes, *e* class is distinct in the sense that all other classes have consonants. This class is detected as outlying class with 100% accuracy using AR model by applying a threshold corresponding to the distance for 95% correctly classified training data from all other classes. We studied the outlier detection performance of LMAR and LMMAR models by using the data of all classes other than the *e* class for training. We found that the data of *e* class was detected as the outlying class with 100% accuracy, both with the LMAR and LMMAR models. This suggests that the generative interpretation of AR models is retained by the LMAR models and LMMAR models.

The average performance of LMMAR is found to be better than that of the SVM that uses AR coefficients based cepstral features. However, the performance of LMMAR model on E-set data is not better compared to the large margin hidden Markov models [33], that make use of mel-frequency cepstral coefficients (MFCC) as features and HMMs as the underlying model. Autoregressive hidden Markov models, trained with large margin method may give a better performance for E-set recognition.

3.5. EEG time series classification

EEG time series have been widely used in the area of human–computer interaction for the study of underlying human brain process. We study the EEG dataset from the UCI KDD archive for our experiments [http://kdd.ics.uci.edu]. This EEG dataset arose from a large study to examine EEG correlates of genetic predisposition to alcoholism. There are two kinds of subjects in the data: control subjects and alcoholic subjects. Multichannel EEG time series were recorded for these two kinds of subjects. Each subject is exposed to stimuli that are pictures of objects chosen from the 1980 Snodgrass and Vanderwart picture set. It contains measurements, sampled at 256 Hz for 1 s, from 64 electrodes placed on

Table 5
5-Fold average classification accuracies and misclassification accuracies (in %) for SVM model using AR coefficients based features on E-set data.

Actual class	Hypothesized class								
	b	*c*	*d*	*e*	*g*	*p*	*t*	*v*	*z*
b	80.2	0	4.8	3.3	2.7	4.0	1.2	2.8	1.0
c	0	87.4	1.6	1.9	0.1	5.8	0.2	2.7	0.3
d	0	5.1	81.9	0	0	8.3	1.7	1.6	1.4
e	2.3	1.7	3.1	85.9	0	6.7	0	0.2	0.1
g	6.1	0.2	1.7	0	82.1	7.9	2	0	0
p	1.4	1.6	10.7	0	2.1	77.1	6.9	0	0
t	2.0	3.7	2.3	0	1.6	7.4	81.8	0	1.2
v	7.8	1.2	0	0	2.1	6.0	1.9	80	1
z	2.1	3.7	1.2	0	1.4	5.6	0	5.2	80.8

the scalp. The whole dataset includes 122 subjects, with each subject performed 120 trials where different stimuli were shown. The dataset contains data for each subject for each of three matching paradigms, single picture, two matching picture, two non-matching pictures, respectively. We include in our datasets time series from two channel F4 for each trial of the subjects. We perform classification on these datasets with the goal of separating time series of control and alcoholic subjects. Following previous work by Keirn and Aunon [36], we start with AR model with order 6 in our experiments to represent the EEG time series. A differencing step is first applied to the time series as the preprocessing step to remove the nonstationary trend.

First, we compare the results obtained by applying the AR and LMAR models. For different values of AR model order, the 5-fold classification performance of AR model and LMAR model on EEG dataset is shown in Fig. 9. It can be seen that for the model order of 7 and 8, respectively, the AR model and LMAR models give the best classification performance. The 5-fold average classification performance for the AR and LMAR models is found to be 59.3% and 74.2%, respectively.

Similarly, for different values of AR model order and different number of components, the 5-fold classification performance of MAR model and LMMAR model on EEG dataset is shown in Fig. 10. The initial number of components in MAR model for each class is set to be 2. It can be seen that for the model order of 6 and 7, respectively, the MAR model and LMMAR model give the best classification performance. The number of components is 2 for both models. The 5-fold average classification performance for the MAR and LMMAR models is found to be 79.9% and 95.9%, respectively. For MAR model and LMMAR models, the BIC method of finding the

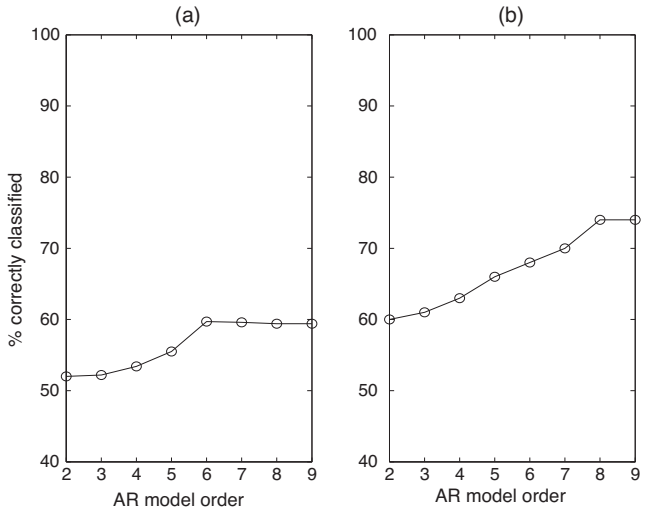

Fig. 9. 5-Fold average classification performance of (a) AR model (b) LMAR model based classifiers for different values of model order *P* on EEG data.

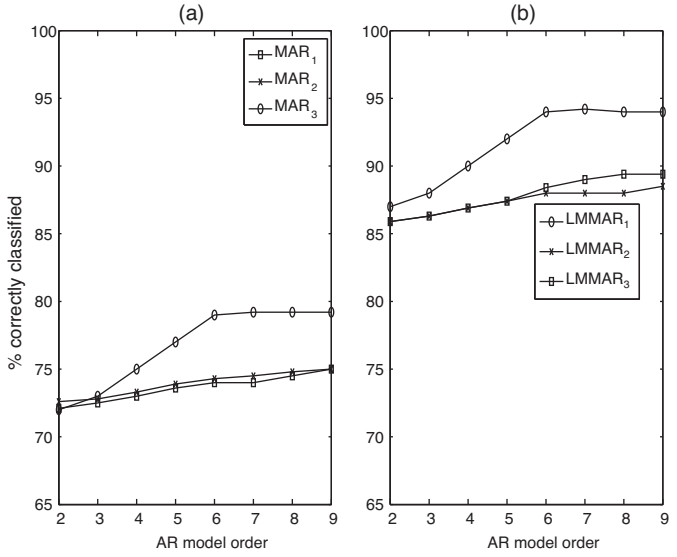

Fig. 10. 5-Fold average classification performance of (a) MAR model and (b) LMMAR model based classifiers for different values of model order *P* and different number of components *K* on EEG data.

Table 6
5-Fold average classification accuracies and misclassification accuracies (in %) for LMMAR and in parentheses for LMAR model on EEG data.

Actual class	Hypothesized class	
	Control	Alcoholic
Control	93.2 (72.9)	6.8 (27.1)
Alcoholic	2.9 (25.3)	97.1 (74.7)

number of components is found to be not useful, as it suggested to choose 3 number of components for each class. The 5-fold classification performance of LMAR and LMMAR models is shown in Table 6.

With AR coefficients (order 6) based cepstral features, the 5-fold average classification performance for SVM is found to be 83.5% and is shown in Table 7. The overall performance of LMMAR is found to be better than that of the SVM that uses cepstral based features.

Table 7
5-Fold average classification accuracies and misclassification accuracies (in %) for SVM model using AR coefficients based features on EEG data.

Actual class	Hypothesized class	
	Control	Alcoholic
Control	81.2	18.8
Alcoholic	12.9	87.1

Table 8
Classification accuracy (%) with 95% confidence intervals for different methods for the ECG, E-set and EEG benchmark datasets.

Dataset	AR	LMAR	MAR	LMMAR	SVM
ECG	90.84 ± 1.67	97.96 ± 1.78	91.77 ± 1.79	98.59 ± 1.56	97.74 ± 1.85
E-set	57.41 ± 2.67	83.12 ± 2.22	58.73 ± 2.67	87.60 ± 2.15	82.13 ± 2.68
EEG	59.31 ± 3.07	74.22 ± 2.97	79.91 ± 2.62	95.92 ± 2.66	83.54 ± 2.59

Table 8 gives the average 5-fold classification accuracies with 95 % confidence intervals obtained by different methods for 3 different datasets.

4. Conclusion

In this work we proposed a new method for time series classification. The proposed method inspired by discriminative training of generative models, has the richness of generative models and performance of discriminative methods. A mixture AR model used to represent each class is trained discriminatively using the large margin methods. The proposed method is applied on ECG and E-set data and found to perform better than the state-of-the-art method based on SVM, that makes use of AR modeling for feature extraction. The time series mixture modeling being an important problem, the estimates of mixture model parameters can be improved by using Bayesian estimation, instead of maximum likelihood estimates. Also, the autoregressive hidden Markov model [18] can be used as the underlying generative model, for speech recognition tasks. Our method can be extended for classifying multivariate time series.

References

[1] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, Time Series Analysis, Forecasting and Control, 3rd ed., Prentice Hall, New Jersey, 1994.
[2] V.N. Vapnik, Statistical Learning Theory, John Wiley & Sons, New York, 1998.
[3] T.W. Liao, F.M. Tang, J. Qu, P.J. Blau, Grinding wheel condition monitoring with boosted minimum distance classifiers, Mechanical Systems and Signal Processing 22 (2008) 217–232.
[4] T.W. Liao, Feature extraction and selection from acoustic emission signals with an application in grinding wheel condition monitoring, Engineering Applications of Artificial Intelligence 23 (2010) 74–84.
[5] H.A. Guvenir, B. Acar, G. Demiroz, A. Cekin, A supervised learning algorithm for arrhythmia analysis, Computers in Cardiology, Sweden 24 (1997) 433–436.
[6] M.H. Song, J. Lee, P.C. Sung, J.L. Kyoung, K.Y. Sun, Support vector machine based arrhythmia classification using reduced features, International Journal of Control, Automation and Systems 2 (2005) 571–579.
[7] M.J.F. Gales, Discriminative models for speech recognition, in: Proceedings of the Information Theory and Applications Workshop, 2007, pp. 170–176.
[8] O. Yakhnenko, A. Silvescu, V. Honavar, Discriminatively trained Markov model for sequence classification, in: Proceedings of the International Conference on Data Mining, 2005, pp. 498–505.
[9] B. Taskar, C. Guestrin, D. Koller, Max-margin Markov networks, in: Proceedings of the Neural Information Processing Systems, 2003, pp. 25–32.
[10] F. Sha, L.K. Saul, Large margin hidden Markov models for automatic speech recognition, Proceedings of the Neural Information Processing Systems (2006) 1249–1256.
[11] X. He, L. Deng, W. Chou, Discriminative learning in sequential pattern recognition, IEEE Signal Processing Magazine 25 (2008) 14–36.
[12] A.B. Chan, N. Vasconcelos, Probabilistic kernels for the classification of autoregressive visual processes, in: IEEE Conference on Computer Vision and Pattern Recognition, San Diego, 2005, pp. 846–851.
[13] T. Jebara, R. Kondor, A. Howard, K. Bennett, N. Cesa-bianchi, Probability product kernels, Journal of Machine Learning Research 5 (2004) 819–844.
[14] D. Ge, N. Srinivasan, S.M. Krishnan, Cardiac arrhythmia classification using autoregressive modeling, Biomedical Engineering, OnLine: http://www.biomedical-engineering-online.com/content/1/1/5, 2002.
[15] N. Huan, R. Palaniappan, Neural network classification of autoregressive features from electroencephalogram signals for brain computer interface design, Journal of Neural Engineering (3) (2004) 142–150.
[16] K. Kalpakis, D. Gada, V. Puttagunta, Distance measures for effective clustering of ARIMA time series, in: Proceedings of the IEEE International Conference on Data Mining, 2001, pp. 273–280.
[17] B.V. Kini, C.C. Sekhar, Large margin AR model for time series classification, in: International Conference on Pattern Recognition, Tampa, 2008, pp. 1–4.
[18] L. Rabiner, B. Juang, Fundamentals of Speech Recognition, Prentice Hall Signal Processing Series, New Jersey, 1993.
[19] J. Nocedal, S.J. Wright, Numerical Optimization, 2nd ed., Springer-Verlag, New York, 2006.
[20] F. Sha, Large Margin Training of Acoustic Models for Speech Recognition, Ph.D. Theis, University of Pennsylvania, 2007.
[21] L. Haupt, S.E. Haupt, Practical Genetic Algorithms, John Wiley & Sons, 1998.
[22] F. Sha, L.K. Saul, Large margin Gaussian mixture modeling for phonetic classification and recognition, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2006, pp. 265–268.
[23] J. Pan, W.J. Tompkins, A real-time QRS detection algorithm, IEEE Transactions on Biomedical Engineering 32 (1985) 230–236.
[24] Y. Xiong, D. Yeung, Time series clustering with ARMA mixtures, Journal of Pattern Recognition 37 (2004) 1675–1689.
[25] C.S. Wong, W.K. Li, On a mixture autoregressive model, Journal of the Royal Statistical Society 62 (2000) 95–115.
[26] D.A. Coast, R.M. Stren, C.G. Cano, S.A. Briller, An approach to cardiac arrhythmia analysis using hidden Markov models, IEEE Transactions on Biomedical Engineering 37 (1990) 826–836.
[31] ISOLET Corpus, Release 1.1, Center for Spoken Language Understanding, Oregon Graduate Institute, Hillsboro, 2000.
[32] S. Katagiri, C.H. Lee, A new HMM/LVQ hybrid algorithm for speech recognition, in: Proceedings of the Global Telecommunications IEEE Conference, 1990, pp. 1032–1036.
[33] H. Jiang, X. Li, C. Liu, Large margin hidden Markov models for speech recognition, IEEE Transactions on Audio, Speech, and Language Processing 14 (2006) 1584–1595.
[34] L. Gu, K. Rose, Substate tying with combined parameter training and reduction in tied-mixture HMM design, IEEE Transactions on Audio and Speech Processing 10 (2002) 137–145.
[35] C. Hsu, C. Chang, C. Lin, A Practical Guide to Support Vector Classification, Technical Report, Dept. of Computer Science and Information Engineering, National Taiwan University, Taipei, 2003.
[36] Z.A. Kerin, J.I. Aunon, A new model for communication between man and his surroundings, IEEE Transactions on Biomedical Engineering 37 (12) (1990) 1209–1214.