

## Mixture of Forward-Directed and Backward-Directed Autoregressive Hidden Markov Models for Time Series Modeling

Vahid Rezaei Tabar<sup>1</sup>, Hosna Fathipour<sup>2</sup>, Horacio Pérez-Sánchez<sup>3</sup>, Farzad Eskandari<sup>1</sup>, and Dariusz Plewczynski<sup>4,5</sup>

<sup>1</sup> Department of Statistics, Faculty of mathematics and Computer Sciences, Allameh Tabataba'i University, Tehran, Iran.

<sup>2</sup> Financial Mathematics Group, Faculty of Financial Sciences, Kharazmi University, Iran.

<sup>3</sup> Structural Bioinformatics and High Performance Computing Research Group (BIO-HPC) Universidad Católica de Murcia (UCAM), Spain.

<sup>4</sup> Laboratory of Functional and Structural Genomics, Center of New Technologies, University of Warsaw, Warsaw, Poland.

<sup>5</sup> Faculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw, Poland.

Received: 05/22/2017, Revision received: 08/13/2018, Published online: 03/28/2019

**Abstract.** Hidden Markov models (HMM) are a ubiquitous tool for modeling time series data. The HMM can be poor at capturing dependency between observations because of the statistical assumptions it makes. Therefore, the extension of the HMM called forward-directed Autoregressive HMM (ARHMM) is considered to handle the dependencies between observations. It is also more appropriate to use an Autoregressive Hidden Markov Model directed backward in time.

---

Corresponding Author: Vahid Rezaei Tabar (vhrezaei@gmail.com)  
Hosna Fathipour (hosnafathi@yahoo.com)  
Horacio Pérez-Sánchez (hperez@ucam.edu)  
Farzad Eskandari (ffeskandari@yahoo.com )  
Dariusz Plewczynski (dariuszplewczynski@cent.uw.edu.pl)

In this paper, we present a sequence-level mixture of these two forms of ARHMM (called MARHMM), effectively allowing the model to choose for itself whether a forward-directed or backward-directed model or a soft combination of the two models are most appropriate for a given data set. For this purpose, we use the conditional independence relations in the context of a Bayesian network which is a probabilistic graphical model. The performance of the MARHMM is discussed by applying it to the simulated and real data sets. We show that the proposed model has greater modeling power than the conventional forward-directed ARHMM.

The source code is available at <https://bitbucket.org/4dnucleome/marhmm/>.

**Keywords.** Autoregressive hidden markov model , Bayesian network, Mixture ARHMM.

**MSC:** 62-09; 62Hxx.

## 1 Introduction

Hidden Markov models (HMM) are a ubiquitous tool for modeling time series data (Baum and Petrie , 1966; Bartolucci and Pennoni, 2007; Kawamoto *et al.*, 2013; MacDonald and Zucchini, 1997; Rabiner, 1989; Zucchini *et al.*, 2016). The factorization of the joint probability of a sequence of states ( $S = \{S_1, \dots, S_T\}$ ) and observations ( $O = \{O_1, \dots, O_T\}$ ) in first-order HMM can be drawn graphically in the form shown in Figure 1. This graph, known as a Bayesian network, probabilistic graphical model or probabilistic independence network, shows the dependencies between the variable in the model (Ghahramani, 2001; Murphy and Mian, 1999; Pearl, 1988).

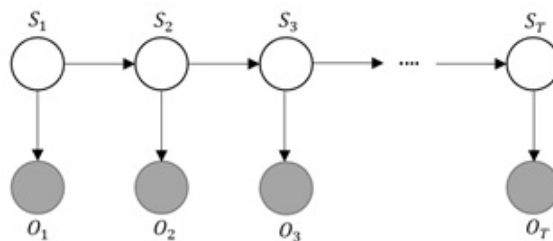


Figure 1: Graphical model for a conventional HMM.

The first-order HMM obeys the following two conditional independence relations which obtained by considering the D-separation concept in Bayesian network

$$S_t \perp \{S_1, O_1, \dots, S_{t-2}, O_{t-2}, O_{t-1}\} | S_{t-1}, \text{ and } O_t \perp \{S_1, O_1, \dots, S_{t-1}, O_{t-1}\} | S_t, \quad 2 \leq t \leq T.$$

The "D" in D-separation stands for dependence. Pearl (1988) proposed the concept of D-separation for general conditional independence in a Bayesian network as follows: Two sets of nodes X and Y are D-separated in Bayesian networks if and only if every path between X and Y is blocked.

As shown in Figure 1, the HMM can be poor at capturing dependency between observations because of the statistical assumptions it makes (Durbin et al., 1998; Dymarski, 2011; Eddy, 1998). In time series modeling, the assumption that observation has a common effect in another observation in the future, simplifies the design of the Bayesian network: directed arcs should flow forward in time (Figure 2). Therefore, the extension of the HMM called Autoregressive HMM (or forward-directed ARHMM) is considered to handle the dependencies between observations (Bartolucci et al., 2014; Shannon et al., 2013). The forward-directed ARHMM encourages correlation amongst observations by adding direct dependencies between them. In an HMM, the current observation is independent of all the other observations given the current state. Consequently, there is no explicit constraint on HMM samples to be smooth. By adding direct dependencies between observations, samples drawn from a forward-directed ARHMM are thus smoother than samples from an HMM, usually making a better generative model in time series problems (Stanculescu et al., 2014). Figure 2 shows a forward-directed ARHMM with first-order dependency (first-order Markov) in which the current observation is conditioned on the current state as well as previous observation. The forward-directed ARHMM is a Bayesian network and obeys the following

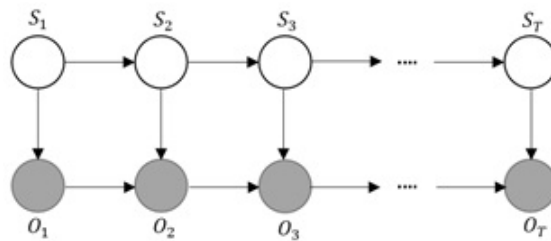


Figure 2: Graphical model for a first-order forward-directed ARHMM.

two conditional independence relations

$$S_t \perp \{S_1, O_1, \dots, S_{t-2}, O_{t-1}\} | S_{t-1}, \text{ and } O_t \perp \{S_1, O_1, \dots, S_{t-1}, O_{t-2}\} | S_t, O_{t-1}.$$

Using these conditional independence relations, the joint distribution of a sequence of states and observations can be factored in the following way

$$P(\mathbf{O}, \mathbf{S}) = P(S_1)P(O_1|S_1) \prod_{t=2}^T P(S_t|S_{t-1})P(O_t|S_t, O_{t-1}).$$

For certain stochastic processes, it may be more appropriate to use an Autoregressive Hidden Markov Model (ARHMM) directed backward in time (Reichenbach, 1991). The backward-directed ARHMM with first order dependency is shown in Figure 3. The backward-directed ARHMM is also known as a Bayesian network.

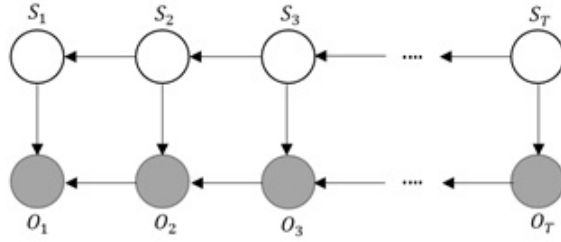


Figure 3: Graphical model for a first-order backward-directed ARHMM.

According to D-separation concept in a Bayesian network, the backward-directed ARHMM also has the following two conditional independence relations

$$S_t \perp \{S_T, O_T, \dots, S_{t+2}, O_{t+1}\} | S_{t+1} \text{ and } O_t \perp \{S_T, O_T, \dots, S_{t+1}, O_{t+2}\} | \{S_t, O_{t+1}\}.$$

The joint distribution of a sequence of states and observations for the backward-directed ARHMM is also as follows

$$P(\mathbf{O}, \mathbf{S}) = P(S_T)P(O_T|S_T) \prod_{t=T-1}^1 P(S_t|S_{t+1})P(O_t|S_t, O_{t+1}).$$

In this paper, we present a sequence-level mixture of two forms of ARHMM (called Mixture ARHMM or MARHMM), effectively allowing the model to choose for itself

whether a forward-directed or backward-directed model or a soft combination of the two is more appropriate for a given data set. The mixture model is defined as follows

$$P(O|\lambda) = \alpha_1 P_1(O|\lambda_1) + \alpha_2 P_2(O|\lambda_2). \quad (1.1)$$

where  $P_1(O|\lambda_1)$  and  $P_2(O|\lambda_2)$  are the probability of the observation sequences given the forward-directed ARHMM and backward-directed ARHMM respectively, and  $\alpha_1, \alpha_2$  are mixing weights such that;  $\alpha_1, \alpha_2 \geq 0$  and  $\alpha_1 + \alpha_2 = 1$ . A sample from the proposed model is always either a sample from a forward-directed ARHMM or from a backward-directed ARHMM. To sample from the proposed model, one first samples the mixture component and then samples the sequence given the mixture component. We show that the proposed model has greater modeling power than the conventional forward-directed ARHMM. This means that the MARHMM performs better in forecasting time series data and is a powerful method for gene expression and assigning protein sequences to protein families in the field of Biology.

Note that, we focus only on the first-order MARHMM (MAR(1)HMM). The higher-order of MARHMM (MAR(r)HMM,  $r > 1$ ) can also be considered for time series and biological data. The MAR(r)HMM makes the model very complex and computationally heavy. For MAR(r)HMM, we need to use the forward-directed and backward-directed Autoregressive higher-order HMMs and estimate many parameters (for more details see Seifert *et al.*, 2014).

This paper is organized as follows: In Section 2 and Section 3, maximum likelihood solutions for ARHMM and mixture ARHMM via EM are presented. For this purpose, we use the conditional independence relations in the context of Bayesian Network. In Section 4, for different sample sizes of simulated and real data sets, we apply both forward-directed ARHMM and MARHMM and compare the results. We finally discuss the applications of MARHMM.

## 2 Finding Maximum Likelihood ARHMM Parameters via EM

The forward-directed or backward-directed ARHMMs are constructed by 4 sets of parameters which are the initial state probability density ( $\pi$ ), transition ( $A$ ) and emission ( $\phi, E$ ) matrices. The ingredients of the forward-directed ARHMM are as follows

- Hidden state takes on  $N$  values which will be denoted by  $\{1, \dots, N\}$ .
- An observation takes  $M$  values which will be denoted by  $\{1, \dots, M\}$ .

- There are  $T$  timestamps in the model, i.e. the set of latent variables of state  $S = \{S_1, \dots, S_T\}$  and the set of observations  $O = \{O_1, \dots, O_T\}$ .
- A vector of initial state  $\pi$  with elements  $\pi(k) = P(S_1 = k)$ .
- The transition matrix  $A$ , in which the element  $a_{k'k} = P(S_t = k | S_{t-1} = k')$  is the transition probability of state  $k'$  to state  $k$  such that  $\sum_k a_{k'k} = 1$
- The initial emission matrix  $\phi$  with the element  $\phi_k(b) = P(O_1 = b | S_1 = k)$
- The emission matrix  $E$  in which the element  $e_k(b, c) = P(O_t = b | S_t = k, O_{t-1} = c)$  presents the emission probability of a current observed variable, given the previous observed variable and current hidden state.

In this work, two sets of parameters

$$\lambda_1 = (A_{L-R}, \phi_{L-R}, E_{L-R}, \pi_{L-R}), \lambda_2 = (A_{R-L}, \phi_{R-L}, E_{R-L}, \pi_{R-L}),$$

for forward-directed and backward-directed ARHMMs, together with the  $\alpha_1$  and  $\alpha_2$  are defined. Note that the index L-R is the timestamp from left to right for representing the forward-directed ARHMM and R-L is the timestamp from right to left for representing the backward-directed ARHMM.

An elegant and powerful method for finding maximum likelihood solutions for models with latent variables is called the Expectation-Maximization (EM) algorithm (Bishop, 2006; Dempster et al., 1977; Do and Batzoglou, 2008; McLachlan and Krishnan, 2007). In this section, we present EM for the forward-directed and backward-directed ARHMMs by considering  $S$  to be the latent variable. Therefore, we need to define the Forward ( $f_k(t)$ ) and Backward ( $b_k(t)$ ) algorithms which are to find out a recursive way to represent the variable sequence in both models (Borodovsky and Ekisheva, 2006; Letunic et al., 2002; Petrushin, 2000). The forward probability represents the probability of the observation sequence up to time  $t$  and the state  $k$  at time  $t$ , given model  $\lambda_1$  (or  $\lambda_2$ ) as the following formulas

$$f_{(L-R)k}(t) = P(O_1, O_2, \dots, O_t, S_t = k | \lambda_1).$$

$$f_{(R-L)k}(t) = P(O_T, O_{T-1}, \dots, O_t, S_t = k | \lambda_2).$$

Also, the backward probability represents the probability of the partial observation sequence from  $t + 1$  to the end, given state  $k$  at time  $t$  as follows

$$b_{(L-R)k}(t) = P(O_{t+1}, O_{t+2}, \dots, O_T | S_t = k, O_t, \lambda_1).$$

$$b_{(R-L)k}(t) = P(O_{t-1}, \dots, O_1 | S_t = k, O_t, \lambda_2).$$

Using the conditional independence relations in the context of Bayesian network, the recurrences for the forward algorithms for forward-directed ARHMM and backward-directed ARHMM will be as follows

$$\begin{aligned}
 f_{(L-R)k}(t) &= \sum_{k'=1}^N P(O_1, O_2, \dots, O_t, S_t = k, S_{t-1} = k') \\
 &= \sum_{k'=1}^N P(O_1, O_2, \dots, O_{t-1}, S_{t-1} = k') \cdot P(O_t, S_t = k | O_1, O_2, \dots, O_{t-1}, S_{t-1} = k') \\
 &= \sum_{k'=1}^N f_{(L-R)k'}(t-1) \cdot a_{(L-R)k'k} \cdot e_{(L-R)k}(O_t, O_{t-1}). \tag{2.1}
 \end{aligned}$$

$$\begin{aligned}
 f_{(R-L)k}(t) &= \sum_{k'=1}^N P(O_T, \dots, O_{t+1}, O_t, S_t = k, S_{t+1} = k') \\
 &= \sum_{k'=1}^N P(O_T, \dots, O_{t+1}, S_{t+1} = k') \cdot P(O_t, S_t = k | O_T, O_{T-1}, \dots, O_{t+1}, S_{t+1} = k') \tag{2.2} \\
 &= \sum_{k'=1}^N f_{(R-L)k'}(t+1) \cdot a_{(R-L)k'k} \cdot e_{(R-L)k}(O_t, O_{t+1}).
 \end{aligned}$$

$$\begin{aligned}
 b_{(L-R)k}(t) &= \sum_{k'=1}^N P(O_{t+1}, O_{t+2}, \dots, O_T, S_{t+1} = k' | S_t = k, O_t) \\
 &= \sum_{k'=1}^N b_{(L-R)k'}(t+1) \cdot a_{(L-R)kk'} \cdot e_{(L-R)k'}(O_{t+1}, O_t). \tag{2.3}
 \end{aligned}$$

$$\begin{aligned}
 b_{(R-L)k}(t) &= \sum_{k'=1}^N P(O_{t-1}, \dots, O_1, S_{t-1} = k' | S_t = k, O_t) \\
 &= \sum_{k'=1}^N b_{(R-L)k'}(t-1) \cdot a_{(R-L)kk'} \cdot e_{(R-L)k'}(O_{t-1}, O_t). \tag{2.4}
 \end{aligned}$$

Suppose that we have a data set of size  $n$  i.e.,  $\{O^1, \dots, O^n\}$ . In the forward-directed and backward-directed ARHMMs, the E-step of the EM algorithm consists of calculating

the expectations  $A_{(L-R)kk'}$ ,  $A_{(R-L)kk'}$ ,  $E_{(L-R)k}(b, c)$ , and  $E_{(R-L)k}(b, c)$  by the following ways (Durbin et al., 1998)

$$\begin{aligned} A_{(L-R)kk'} &= \sum_j \sum_t P(S_t = k, S_{t+1} = k' | \mathbf{O}^j, \lambda_1) \\ &= \sum_j \frac{1}{P(\mathbf{O}^j | \lambda_1)} \sum_t f_{(L-R)k}^j(t) \cdot b_{(L-R)k'}^j(t+1) \cdot a_{(L-R)kk'} \cdot e_{(L-R)k'}(\mathbf{O}_{t+1}^j, \mathbf{O}_t^j), \end{aligned} \quad (2.5)$$

where  $f_{(L-R)k}^j(t)$  is the forward variable  $f_{(L-R)k}(t)$  defined in (2.1), calculated for sequence  $j$ , and  $b_{(L-R)k'}^j(t+1)$  is the corresponding backward variable. Similarly, we have

$$\begin{aligned} A_{(R-L)kk'} &= \sum_j \sum_t P(S_{t+1} = k, S_t = k' | \mathbf{O}^j, \lambda_2) \\ &= \sum_j \frac{1}{P(\mathbf{O}^j | \lambda_2)} \sum_t f_{(R-L)k}^j(t+1) \cdot b_{(R-L)k'}^j(t) \cdot a_{(R-L)kk'} \cdot e_{(R-L)k'}(\mathbf{O}_t^j, \mathbf{O}_{t+1}^j). \end{aligned} \quad (2.6)$$

$$\begin{aligned} E_{(L-R)k}(b, c) &= \sum_j \sum_{\{t | O_t = b, O_{t-1} = c\}} P(S_t = k | \mathbf{O}^j, \lambda_1) \\ &= \sum_j \frac{1}{P(\mathbf{O}^j | \lambda_1)} \sum_{\{t | O_t = b, O_{t-1} = c\}} f_{(L-R)k}^j(t) \cdot b_{(L-R)k}^j(t). \end{aligned} \quad (2.7)$$

$$\begin{aligned} E_{(R-L)k}(b, c) &= \sum_j \sum_{\{t | O_t = b, O_{t+1} = c\}} P(S_t = k | \mathbf{O}^j, \lambda_2) \\ &= \sum_j \frac{1}{P(\mathbf{O}^j | \lambda_2)} \sum_{\{t | O_t = b, O_{t+1} = c\}} f_{(R-L)k}^j(t) \cdot b_{(R-L)k}^j(t). \end{aligned} \quad (2.8)$$

Having calculated these expectations, the new model parameters are calculated in following way (Durbin et al., 1998)

$$a_{(L-R)kk'}^{new} = \frac{A_{(L-R)kk'}}{\sum_{k''} A_{(L-R)kk''}}. \quad (2.9)$$

$$a_{(R-L)kk'}^{new} = \frac{A_{(R-L)kk'}}{\sum_{k''} A_{(R-L)kk''}}. \quad (2.10)$$



$$e_{(L-R)k}^{new}(b, c) = \frac{E_{(L-R)k}(b, c)}{\sum_{c'} E_{(L-R)k}(b, c')}. \quad (2.11)$$

$$e_{(R-L)k}^{new}(b, c) = \frac{E_{(R-L)k}(b, c)}{\sum_{c'} E_{(R-L)k}(b, c')}. \quad (2.12)$$

In other words, the M-step just consists of plugging  $A_{(L-R)kk'}$ ,  $A_{(R-L)kk'}$ ,  $E_{(L-R)k}(b, c)$ , and  $E_{(R-L)k}(b, c)$  into the re-estimation formulas given in (2.9)-(2.12).

### 3 Finding Maximum Likelihood Mixture Parameters via EM

In this section, we present EM for the mixture ARHMM (MARHMM) by considering  $\mathbf{S}$  and  $\mathbf{y}$  (mixture component) to be the latent variables. In other words, we apply EM to the overall MARHMM model by considering  $(\mathbf{S}, \mathbf{y})$  as the latent variable. The log-likelihood expression given the mixture model parameters  $\lambda = (\alpha_1, \alpha_2, \lambda_1, \lambda_2)$  is

$$\log L(\lambda|\mathbf{O}) = \log \prod_{j=1}^n P(\mathbf{O}^j|\lambda) = \sum_{j=1}^n \log \left( \sum_{l=1}^2 \alpha_l P_l(\mathbf{O}^j|\lambda_l) \right), \quad (3.1)$$

where the functions  $P_1(\cdot)$  and  $P_2(\cdot)$  are the probability of the observation sequences given the forward-directed ARHMM and backward-directed ARHMM respectively. Based on the fact that the mixture model can be described more simply by assuming that each observed data point has a corresponding unobserved data point or latent variable, we assume there are unobserved data items  $y_i \in \{1, 2\}$ , which means that the  $i^{th}$  sample was generated by the forward-directed ARHMM or backward-directed ARHMM. If we know the values of  $\mathbf{Y}$ , the likelihood becomes

$$\log L(\lambda|\mathbf{O}, \mathbf{Y}) = \log P(\mathbf{O}, \mathbf{Y}|\lambda) = \sum_{j=1}^n \log \left( \alpha_{y_j} P_{y_j}(\mathbf{O}^j|\lambda_{y_j}) \right), \quad y_j \in \{1, 2\}.$$

Therefore, the function Q for MARHMM is given by (Bilmes, 1998):

$$Q(\lambda, \lambda^{old}) = \sum_{l=1}^2 \sum_{j=1}^n \log \alpha_l \cdot P(l|\mathbf{O}^j, \lambda^{old}) + \sum_{l=1}^2 \sum_{j=1}^n \log \left( P_l(\mathbf{O}^j|\lambda_l) \right) \cdot P(l|\mathbf{O}^j, \lambda^{old}), \quad (3.2)$$

in which

$$P(l|\mathbf{O}^j, \boldsymbol{\lambda}^{old}) = \frac{\alpha_l^{old} P_l(\mathbf{O}^j | \boldsymbol{\lambda}_l^{old})}{\alpha_1^{old} P_1(\mathbf{O}^j | \boldsymbol{\lambda}_1^{old}) + \alpha_2^{old} P_2(\mathbf{O}^j | \boldsymbol{\lambda}_2^{old})}, \quad l = 1, 2.$$

$$P_1(\mathbf{O}^j | \boldsymbol{\lambda}_1^{old}) = \sum_{k=1}^N P(O_1^j, O_2^j, \dots, O_T^j, S_T^j = k | \boldsymbol{\lambda}_1^{old}) = \sum_{k=1}^N f_{(L-R)k}^j(T).$$

$$P_2(\mathbf{O}^j | \boldsymbol{\lambda}_2^{old}) = \sum_{k=1}^N P(O_T^j, O_{T-1}^j, \dots, O_1^j, S_1^j = k | \boldsymbol{\lambda}_2^{old}) = \sum_{k=1}^N f_{(R-L)k}^j(1).$$

To maximize (3.2), we can maximize the term containing  $\alpha_l$  and the term containing  $\lambda_l$  independently since they are not related. To find the expression for  $\alpha_l$ , the Lagrange multiplier  $\beta$  with the constraint that  $\alpha_1 + \alpha_2 = 1$ , is solved by the following equation

$$\frac{\partial}{\partial \alpha_l} \left[ \sum_{l=1}^2 \sum_{j=1}^n \log \alpha_l \cdot P(l|\mathbf{O}^j, \boldsymbol{\lambda}^{old}) + \beta(\alpha_1 + \alpha_2 - 1) \right] = 0.$$

Thus we have

$$\alpha_l = \frac{1}{n} \sum_{j=1}^n P(l|\mathbf{O}^j, \boldsymbol{\lambda}^{old}), \quad l = 1, 2.$$

To find the expression for  $\lambda_l$ , we maximize the second term of (3.2). For this purpose, we maximize the auxiliary function Q rather than to directly maximize  $\log P_l(\mathbf{O}^j | \lambda_l)$ . Therefore, we have

$$\arg \max_{\lambda_l} \sum_{l=1}^2 \sum_{j=1}^n \log \left( P_l(\mathbf{O}^j | \lambda_l) \right) \cdot P(l|\mathbf{O}^j, \boldsymbol{\lambda}^{old}) = \arg \max_{\lambda_l} \sum_{l=1}^2 \sum_{j=1}^n Q(\lambda_l, \lambda_l^{old}) \cdot P(l|\mathbf{O}^j, \boldsymbol{\lambda}^{old}),$$

in which

$$Q(\lambda_l, \lambda_l^{old}) = \sum_{s^j} \log \left( P_l(\mathbf{O}^j, S^j | \lambda_l) \right) \cdot P_l(S^j | \mathbf{O}^j, \lambda_l^{old}).$$

Taken together, we have

$$\begin{aligned}
& \sum_{l=1}^2 \sum_{j=1}^n Q(\lambda_l, \lambda_l^{old}).P(l|\mathbf{O}^j, \lambda^{old}) \\
&= \sum_{l=1}^2 \sum_{j=1}^n \left( \sum_{S^j} \log \left( P_l(\mathbf{O}^j, S^j | \lambda_l) \right) \cdot P_l(S^j | \mathbf{O}^j, \lambda_l^{old}) \right) \cdot P(l|\mathbf{O}^j, \lambda^{old}) \\
&= \sum_{j=1}^n \left( \sum_{S^j} \log \left( P_1(\mathbf{O}^j, S^j | \lambda_1) \right) \cdot P_1(S^j | \mathbf{O}^j, \lambda_1^{old}) \right) \cdot P(1|\mathbf{O}^j, \lambda^{old}) \\
&+ \sum_{j=1}^n \left( \sum_{S^j} \log \left( P_2(\mathbf{O}^j, S^j | \lambda_2) \right) \cdot P_2(S^j | \mathbf{O}^j, \lambda_1^{old}) \right) \cdot P(2|\mathbf{O}^j, \lambda^{old}). \tag{3.3}
\end{aligned}$$

Since the parameters  $\lambda_1$  and  $\lambda_2$  we wish to optimize are now independently split into the two terms in the sum, we can optimize each term individually. We only maximize the first term in (3.3), and the second term maximizes in a similar way. According to Figure 2, we have

$$P_1(\mathbf{O}^j, S^j | \lambda_1) = P_1(S_1^j) P_1(\mathbf{O}_1^j | S_1^j) \prod_{t=2}^T P_1(S_t^j | S_{t-1}^j) P_1(\mathbf{O}_t^j | S_t^j, \mathbf{O}_{t-1}^j).$$

Therefore, the first term of (3.3) will be as follows (Movellan , 2003)

$$\begin{aligned}
& \sum_{j=1}^n \left( \sum_{S^j} \log \left( P_1(\mathbf{O}^j, S^j | \lambda_1) \right) \cdot P_1(S^j | \mathbf{O}^j, \lambda_1^{old}) \right) \cdot P(1|\mathbf{O}^j, \lambda^{old}) \\
&= \sum_{j=1}^n \left( \sum_{k=1}^N \log \left( \pi_{(L-R)}(k) \right) \cdot P_1(S_1^j = k | \mathbf{O}^j, \lambda_1^{old}) \right) \cdot P(1|\mathbf{O}^j, \lambda^{old}) \\
&+ \sum_{j=1}^n \left( \sum_{k=1}^N \sum_{b=1}^M \log \left( \phi_{(L-R)k}(b) \right) \cdot P_1(S_1^j = k | \mathbf{O}^j, \lambda_1^{old}) \right) \cdot P(1|\mathbf{O}^j, \lambda^{old}) \\
&+ \sum_{j=1}^n \left( \sum_{k=1}^N \sum_{k'=1}^N \sum_{t=2}^{T^j} \log \left( a_{(L-R)kk'} \right) \cdot P_1(S_t^j = k, S_{t+1}^j = k' | \mathbf{O}^j, \lambda^{old}) \right) \cdot P(1|\mathbf{O}^j, \lambda^{old}) \\
&+ \sum_{j=1}^n \left( \sum_{k=1}^N \sum_{b=1}^M \sum_{c=1}^M \sum_{t=2}^{T^j} \log \left( e_{(L-R)k}(b, c) \right) \cdot P_1(S_t^j = k | \mathbf{O}^j, \lambda^{old}) \right) \cdot P(1|\mathbf{O}^j, \lambda^{old}). \tag{3.4}
\end{aligned}$$

Therefore, adding the Lagrange multiplier, using the constraint (i.e.  $\sum_k \pi_{(L-R)}(k) = 1$ ) and setting the derivative equal to zero, the estimates of the new parameters in terms of the old parameters will be as follows (Bilmes, 1998)

$$\pi_{(L-R)}^{new}(k) = \frac{\sum_{j=1}^n P_1(S_1^j = k | \mathbf{O}^j, \lambda_1^{old}) \cdot P(1 | \mathbf{O}^j, \lambda^{old})}{\sum_{j=1}^n P(1 | \mathbf{O}^j, \lambda^{old})}, \quad (3.5)$$

$$\phi_{(L-R)k}^{new}(b) = \frac{\sum_{j=1}^n \delta(O_1^j = b) P_1(S_1^j = k | \mathbf{O}^j, \lambda_1^{old}) \cdot P(1 | \mathbf{O}^j, \lambda^{old})}{\sum_{j=1}^n P_1(S_1^j = k | \mathbf{O}^j, \lambda_1^{old}) \cdot P(1 | \mathbf{O}^j, \lambda^{old})}, \quad (3.6)$$

$$a_{(L-R)kk'}^{new} = \frac{\sum_{j=1}^n \sum_{t=2}^{T^j} P_1(S_t^j = k, S_{t+1}^j = k' | \mathbf{O}^j, \lambda_1^{old}) \cdot P(1 | \mathbf{O}^j, \lambda^{old})}{\sum_{j=1}^n \sum_{t=2}^{T^j} P_1(S_t^j = k | \mathbf{O}^j, \lambda_1^{old}) P(1 | \mathbf{O}^j, \lambda^{old})}, \quad (3.7)$$

$$e_{(L-R)k}^{new}(b, c) = \frac{\sum_{j=1}^n \sum_{\{t | O_t^j = b, O_{t-1}^j = c\}}^{T^j} P_1(S_t^j = k | \mathbf{O}^j, \lambda_1^{old}) \cdot P(1 | \mathbf{O}^j, \lambda^{old})}{\sum_{j=1}^n \sum_{\{t | O_{t-1}^j = c\}}^{T^j} P_1(S_t^j = k | \mathbf{O}^j, \lambda_1^{old}) \cdot P(1 | \mathbf{O}^j, \lambda^{old})}, \quad (3.8)$$

where  $\delta(O_1^j = b)$  tells us to include only those cases in which  $O_1^j = b$ .

## 4 Experiment

### 4.1 Simulation Study

We generated 4 different data sets  $25 \times 150$  (i.e. 25 subjects over 150 days),  $30 \times 150$ ,  $60 \times 150$  and  $100 \times 150$  of randomly sampled discrete data with 2 hidden states and 3 observation values regarding the following matrices

$$P(S_t | S_{t-1}) = \begin{bmatrix} 0.7 & 0.3 \\ 0.9 & 0.1 \end{bmatrix}, \quad P(O_t | S_t) = \begin{bmatrix} 0.5 & 0.3 & 0.2 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}.$$

We then estimate the parameters of forward-directed ARHMM and MARHMM for these data sets regarding the following initial values

$$A = \begin{bmatrix} 0.6 & 0.4 \\ 0.8 & 0.2 \end{bmatrix}, \quad \pi = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}.$$

$$P_{L-R}(O_t|s_t, O_{t-1} = 1) = P_{R-L}(O_t|s_t, O_{t+1} = 1) = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}.$$

$$P_{L-R}(O_t|s_t, O_{t-1} = 2) = P_{R-L}(O_t|s_t, O_{t+1} = 2) = \begin{bmatrix} 0.1 & 0.6 & 0.3 \\ 0.8 & 0.1 & 0.1 \end{bmatrix}.$$

$$P_{L-R}(O_t|s_t, O_{t-1} = 3) = P_{R-L}(O_t|s_t, O_{t+1} = 3) = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.7 & 0.2 & 0.1 \end{bmatrix}.$$

The initial values are obtained by testing different values on complete data sets (fourth simulated data set). This means that we have acted upon different initial values and chosen the best ones (Figure 4). We also choose  $\alpha_1 = \alpha_2 = 0.5$ .

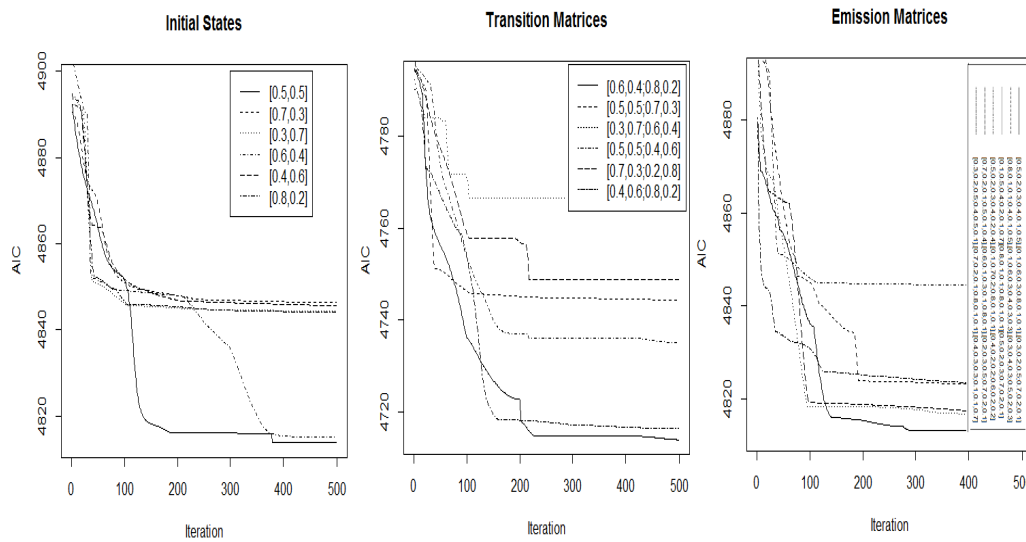


Figure 4: Comparing different initial values.

For evaluation of performance of the ARHMM and MARHMM, we compute the Akaike information criterion (AIC) values of forward-directed ARHMM and MARHMM at each iteration of the EM algorithm. The model which gives the minimum AIC is

selected as the best. Convergence is generally detected by computing the value of the AIC after each iteration and halting when it appears not to be changing in a significant manner from one iteration to the next. AIC is to select the model that minimizes the negative log-likelihood penalized by the number of parameters as specified in the (4.1)

$$\text{AIC} = 2k - 2 \log(L), \quad (4.1)$$

where  $L$  is the maximum value of the likelihood function for the model and  $k$  is the number of estimated parameters. Note that the log-likelihood for forward-directed ARHMM and MARHMM are defined respectively as follows

$$\log L(\lambda_1 | \mathcal{O}) = \sum_{j=1}^n \log \left( P_1(\mathcal{O}^j | \lambda_1) \right) = \sum_{j=1}^n \log \left( \sum_{k=1}^N f_{(L-R)k}^j(T) \right). \quad (4.2)$$

$$\begin{aligned} \log L(\lambda | \mathcal{O}) &= \sum_{j=1}^n \log \left( \sum_{l=1}^2 \alpha_l P_l(\mathcal{O}^j | \lambda_l) \right) \\ &= \sum_{j=1}^n \log \left( \alpha_1 P_1(\mathcal{O}^j | \lambda_1) + \alpha_2 P_2(\mathcal{O}^j | \lambda_2) \right) \\ &= \sum_{j=1}^n \log \left( \alpha_1 \sum_{k=1}^N f_{(L-R)k}^j(T) + \alpha_2 \sum_{k=1}^N f_{(R-L)k}^j(1) \right). \end{aligned} \quad (4.3)$$

We plotted AIC values vs. iterations for different data sets using forward-directed ARHMM and MARHMM (Figure 5).

Figure 5 shows that the MARHMM yields lower AIC values than the forward-directed ARHMM, especially in large data sets.

Because we know the true parameters in transition and emission matrices, we use the similarity distance criterion between two matrices to evaluate how good the estimating parameters are. For this purpose, the Frobenius distance is considered as follows (Golub and Van Loan, 2012)

$$\text{Distance} = F_{A,B} = \sqrt{\text{trace}(A - B) * (A - B)'},$$

where  $A$  and  $B$  are the true and estimating matrices respectively. The lower values of distance represent the proper estimations. The results of the distance criterion for the

ARHMM and MARHMM are shown in Table 1. The results show that the MARHMM give the higher accuracy estimations.

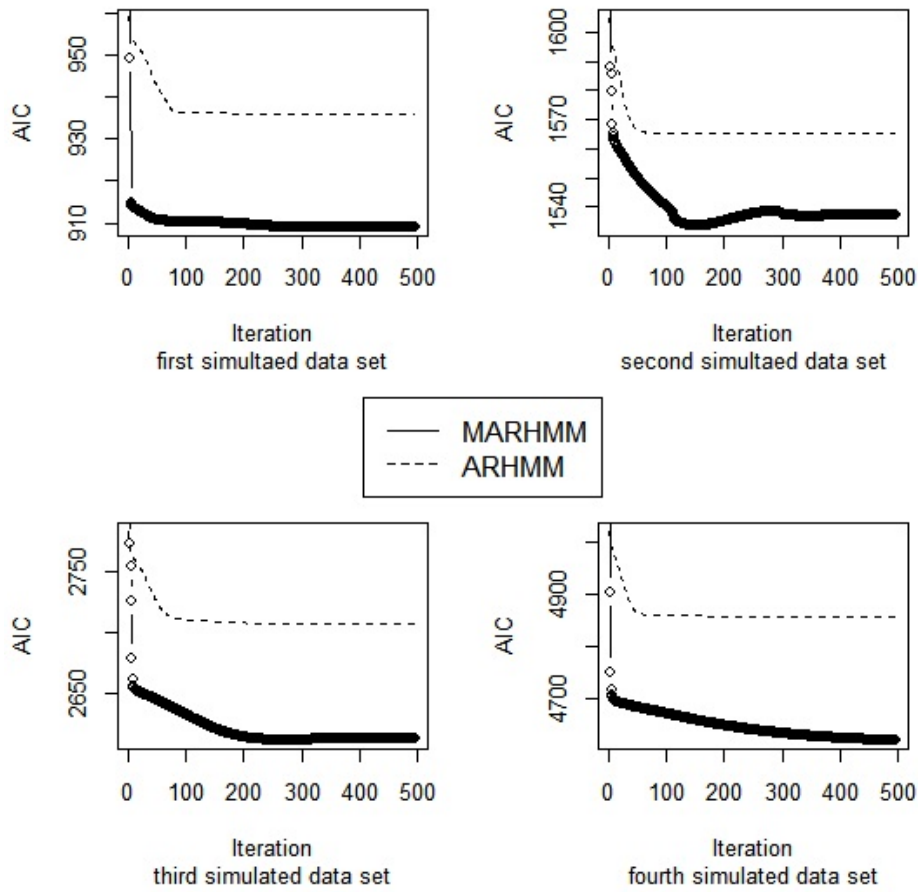


Figure 5: The AIC values vs. iterations for different simulated data sets. For the first simulated data set:  $\hat{\alpha}_1 = 0.1000, \hat{\alpha}_2 = 0.9000$ , for the second simulated data set:  $\hat{\alpha}_1 = 0.3350, \hat{\alpha}_2 = 0.6650$ , for the third simulated data set:  $\hat{\alpha}_1 = 0.5175, \hat{\alpha}_2 = 0.4825$ , for the fourth simulated data set:  $\hat{\alpha}_1 = 0.4567, \hat{\alpha}_2 = 0.5433$ .

Table 1: The distance criterion for transition and emission matrices.

Data sets	Transition Matrix		Emission Matrix	
	ARHMM	MARHMM	ARHMM	MARHMM
First Simulated data set	0.40	0.2	0.55	0.24
Second simulated data set	0.30	0.14	0.40	0.14
Third simulated data set	0.20	0.07	0.33	0.11
Fourth simulated data set	0.08	0.05	0.15	0.09

It seems that the MARHMM is a more complex model than the ARHMM. For both algorithms, we compute the time complexity. Using time complexity makes it easy to estimate the running time of an algorithm. The time complexity for both the MARHMM and ARHMM is reported in Table 2. Although the consumed times in the process of the MARHMM are more than ARHMM, the MARHMM has higher accuracy (Table 1). It should be noted that the algorithms are implemented in Matlab R2017a.

Table 2: Comparing time complexity (in seconds).

Size of Data sets	ARHMM	MARHMM
25 × 150	6.8s	8.9s
30 × 150	9.1s	14.4s
60 × 150	22.3s	34.9s
100 × 150	25.8s	40.1s

The MARHMM can be used for gene expression and assigning protein sequences to protein families in the field of biology and also for forecasting time series data. We first discuss the assignment of protein sequences in the next subsection and make the prediction for the next observations in Subsection 4.3.

## 4.2 Assigning Protein Sequences to Protein Families using MARHMM

In this section, we use the Pfam database which is a well-known data set of protein families (Finn *et al.*, 2016). It is widely used to align new protein sequences to the known proteins of a given family. There are two components in Pfam: Pfam-A and Pfam-B. The entries of Pfam-A have high quality. As shown in Table 3, we use the top twenty protein families of Pfam-A for assigning the protein sequences to protein families using forward-directed ARHMM and MARHMM. Here, we have three hidden states named



Table 3: Top twenty protein families in Pfam database.

profile	Number of sequence	
	Seed	Full
WD40	1465	378719
ABC_tran	55	369723
zf-C2H2	159	340711
pkinase	38	236455
MFS_1	192	214283
Response_reg	52	176760
Ank_2	203	172686
BPD_transp_1	81	148125
HATPase	658	133923
LRR_8	63	133230
RRM_1	72	131391
Helicase	422	119885
PPR_2	226	98670
Mito_carr	161	89340
fn_3	98	88510
AMP-binding	145	87704
I-set	48	87027
adh_short	44	86592
PPR	459	85615
HisKA	265	85578

Match (M), Delete (D), and Insert (I), and 20 amino acids as observations  $\{O_1, O_2, \dots, O_{20}\}$  (Eddy, 1998). To assess the performance of the MARHMM, ten sequences from each of the top twenty families are randomly removed. Totally, we have 200 removed sequences which are used as test sequences, while the others form the training set. We repeat this procedure 20 times. We estimate the emission and transition matrices for training sets of each protein family using forward-directed ARHMM and MARHMM. We then use the AIC to assign each test sequence to families. The means of the numbers of correctly assigned proteins to the top twenty protein families are shown in Table 4. Based on the results, the assignment of sequences to the protein families using the MARHMM is considerably improved. For all protein families, more than 80 percent of removed sequences are assigned correctly using the MARHMM.

Table 4: The mean of thre numbers of correctly assigned sequences.

profile	Mean	
	ARHMM	MARHMM
WD40	8.2	9.1
ABC_tran	9.1	9.8
zf-C2H2	8.7	9.9
pkinase	8.5	9.4
MFS 1	8.8	9.3
Response_reg	8.7	9.2
Ank 2	9.1	9.9
BPD transp 1	8.2	8.9
HATPase	9.2	9.9
LRR_8	8.5	9.2
RRM_1	7.7	8.8
Helicase	8.4	9.4
PPR_2	8.3	9.1
Mito carr	7.8	8.6
fn_3	8.4	8.9
AMP-binding	8.3	9.1
I-set	7.9	8.7
adh short	8.4	9.6
PPR	7.3	8.4
HisKA	8.4	8.9

### 4.3 Alcoholism Treatment Trial Data

We also fit forward-directed ARHMM and MARHMM to the real data set which is from a clinical trial of the drug Naltrexone that was conducted at the Centre for Studies of Addictions (CSA) at the University of Pennsylvania. This data set contains daily observations from 240 subjects over 168 days. The subjects were volunteers who had been diagnosed with an AUD. The hidden states in this data set are labeled as Abstinence (A), Moderate Drinking (M), and Heavy Drinking (H) and the observations are determined corresponding to no drinking (O=1), light drinking (O=2) and heavy drinking (O=3). A complete description of the trial is available in Shirley *et al.* (2010). We randomly choose different subjects: 25, 50, 100 and 240 over 168 days. In order to reduce the risk of being trapped in a poor local maximum, a large number of initial values were tested. We then consider the following initial values for estimating the

parameters:

$$A = \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0.7 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.3 \end{bmatrix}, \quad \pi = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix}.$$

$$P_{L-R}(O_t|s_t, O_{t-1} = 1) = P_{R-L}(O_t|s_t, O_{t+1} = 1) = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.4 & 0.1 & 0.5 \\ 0.2 & 0.7 & 0.1 \end{bmatrix}.$$

$$P_{L-R}(O_t|s_t, O_{t-1} = 2) = P_{R-L}(O_t|s_t, O_{t+1} = 2) = \begin{bmatrix} 0.1 & 0.6 & 0.3 \\ 0.8 & 0.1 & 0.1 \\ 0.3 & 0.4 & 0.5 \end{bmatrix}.$$

$$P_{L-R}(O_t|s_t, O_{t-1} = 3) = P_{R-L}(O_t|s_t, O_{t+1} = 3) = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.7 & 0.2 & 0.1 \\ 0.1 & 0.4 & 0.5 \end{bmatrix}.$$

We set  $\alpha_1 = \alpha_2 = 0.5$  as the initial values of mixing weights. We then estimate the parameters of forward-directed ARHMM and MARHMM. We plotted the AIC values vs. iterations for different subjects using forward-directed ARHMM and MARHMM (Figure 6). Figure 6 shows that the MARHMM yields lower AIC values than forward-directed ARHMM. In other words, the mixture of forward-directed and backward-directed ARHMMs has greater modeling power than the conventional forward-directed ARHMM. Thus, it can be used for forecasting in time series data. For forecasting, we only focus on the complete data set (240 over 168 days). To assess the performance of the MARHMM, last eight observations from each sequence are removed. This means that our training data set has 240 over 160 observations and the test data set contains 240 over 8 observations. Our aim is predicting the removed observations from each sequence.

We first estimate the emission and transition matrices for the training data set using forward-directed ARHMM and MARHMM. We then use these estimations for predicting the next observations. For this purpose, we focus on one of the estimated matrix  $P(O_t|s_t, O_{t-1} = c)$ ,  $c = 1, 2, 3$ , due to the value of the previous observation (here the value of previous observation is determined by 160<sup>th</sup> observation). Then the element of the estimated matrix  $P(O_t|s_t, O_{t-1} = c)$  which gets the maximum value can be considered as the prediction of the next observation. The number of correctly predicted observation using the forward directed ARHMM and MARHMM are 0.6 and 0.75 respectively. Thus, using the MARHMM makes it possible to predict the next observations with

higher probability. Note that we can also predict the most probable path for hidden states.

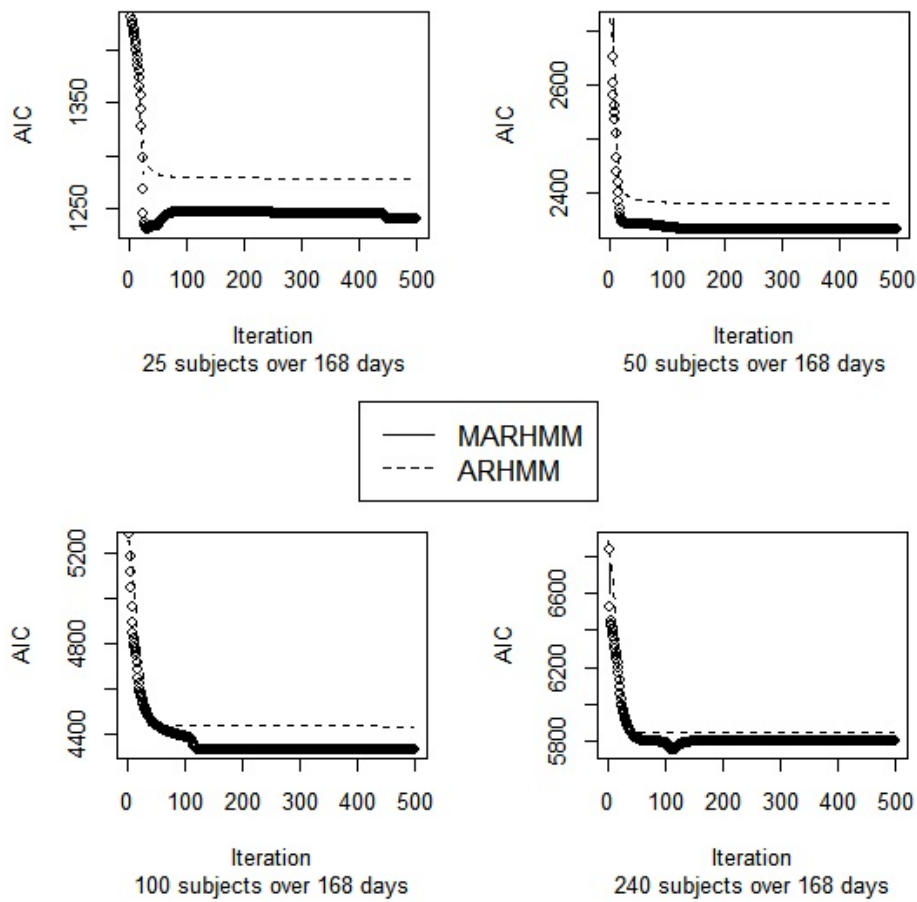


Figure 6: AIC values vs. iterations for 25 ,50, 100 and 240 subjects over 168 days. For 25 subjects over 168 days:  $\hat{\alpha}_1 = 0.4926$ ,  $\hat{\alpha}_2 = 0.5074$ , for 50 subjects over 168 days:  $\hat{\alpha}_1 = 0.1997$ ,  $\hat{\alpha}_2 = 0.8003$ , for 100 subjects over 168 days:  $\hat{\alpha}_1 = 0.4416$ ,  $\hat{\alpha}_2 = 0.5584$ , for 240 subjects over 168 days:  $\hat{\alpha}_1 = 0.5313$ ,  $\hat{\alpha}_2 = 0.4687$ .

## 5 Discussion

In this paper, a novel generalization of the forward-directed ARHMM is introduced and its parameters are estimated via the EM algorithm. For this purpose,

- We first present EM for the forward-directed ARHMM considering  $S$  to be the latent variable.
- We then present EM for the MARHMM by considering  $S$  and  $y$  (mixture component) to be the latent variables.

In other words, we apply EM to the overall MARHMM model by considering  $(S, y)$  as the latent variable. The results of this work show that considering the mixture of ARHMMs has greater modeling power than the conventional forward-directed ARHMM. This mixture model can also be used for sequence labeling problems such as part-of-speech (POS) tagging, text chunking in natural language processing, and decomposition structures of handwriting recognition (Takiguchi *et al.*, 2014). As an example, consider the situation where we are going to annotate a three-word sentence with part-of-speech tags. Therefore, the two possible ways of decomposition one from left-to-right and the other one from right-to-left consider finding the best highest probability sequence of the three-word sentence.

## Acknowledgments

Some parts of this work were carried out when Vahid Rezaei Tabar was visiting Center of New Technology of University of Warsaw, Poland. Vahid Rezaei Tabar and Farzad Eskandari are grateful to the Department of Statistics at Allameh Tabataba'i University (No. 040, July 15, 2017).

This work has been supported by the Polish National Science Centre (2014/15/B/ST6/05082), Foundation for Polish Science (TEAM to DP) and by a grant from the Department of Science and Technology, India under Indo-Polish/Polish-Indo project No.: DST/INT/POL/P-36/2016. The work was co-supported by grant 1U54DK107967-01 Nucleome Positioning System for Spatiotemporal Genome Organization and Regulation within 4DNucleome NIH program.

## References

- Bartolucci, F. and Pennoni, F. (2007), A class of latent Markov models for capture recapture data allowing for time, heterogeneity, and behavior effects. *Biometrics*, **63**(2), 568–578.
- Bartolucci, F., Bacci, S. and Pennoni, F. (2014), Longitudinal analysis of self-reported health status by mixture latent autoregressive models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **63**(2), 267–288.
- Baum, L. E. and Petrie, T. (1966), Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, **37**(6), 1554–1563.
- Bilmes, Jeff A. (1998), A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International Computer Science Institute*, **4**(510), 126.
- Bishop, C. M. (2006), *Mixture models and the em algorithm*. Microsoft Research, Cambridge.
- Borodovsky, M. and Ekisheva, S. (2006), *Problems and solutions in biological sequence analysis*. Cambridge University Press.
- Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977), Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society, Series B (methodological)*, 1–38.
- Do, C. B. and Batzoglou, S. (2008), What is the expectation maximization algorithm?. *Nature Biotechnology*, **26**(8), 897.
- Durbin, R., Eddy, S. R., Krogh, A. and Mitchison, G. (1998), *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press.
- Dymarski, P. (2011), *Hidden Markov models, theory and applications*. In Tech Open Access Publishers.
- Eddy, S. R. (1998), Profile hidden Markov models. *Bioinformatics (Oxford, England)*, **14**(9), 755–763.
- Finn, R. D., Coghill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L. and Salazar, G. A. (2016), The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, **44**(D1), D279–D285.

- Ghahramani, Z. (2001), An introduction to hidden Markov models and Bayesian networks. *International journal of pattern recognition and artificial intelligence*, **15**(01), 9–42.
- Golub, G. H. and Van Loan, C. F. (2012), *Matrix computations* (Vol. 3). JHU Press.
- Kawamoto, R., Nazir, A., Kameyama, A., Ichinomiya, T., Yamamoto, K., Tamura, S. and Kinoshita, Y. (2013), Hidden Markov model for analyzing time-series health checkup data. *MedInfo*, **192**, 491–495.
- Letunic, I., Goodstadt, L., Dickens, N. J., Doerks, T., Schultz, J., Mott, R. and Bork, P. (2002), Recent improvements to the SMART domain-based sequence annotation resource. *Nucleic acids research*, **30**(1), 242–244.
- MacDonald, I. L. and Zucchini, W. (1997), *Hidden Markov and other models for discrete-valued time series* (Vol. 110). CRC Press.
- McLachlan, G., Krishnan, T. (2007), *The EM algorithm and extensions* (Vol. 382). John Wiley & Sons.
- Movellan, J. R. (2003), Tutorial on hidden Markov models, *Machine perception laboratory online tutorials*.
- Murphy, K. and Mian, S. (1999), *Modelling gene expression data using dynamic Bayesian networks* (Vol. 104). Technical report, Computer Science Division, University of California, Berkeley, CA.
- Pearl, J. (1988), *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.
- Petrushin, V. A. (2000), Hidden markov models: Fundamentals and applications. In *Online Symposium for Electronics Engineer*.
- Rabiner, L. R. (1989), A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**(2), 257–286.
- Reichenbach, H. (1991), *The direction of time* (Vol. 65). University of California Press.
- Shannon, M., Zen, H. and Byrne, W. (2013), Autoregressive models for statistical parametric speech synthesis. *IEEE transactions on audio, speech, and language processing*, **21**(3), 587–597.

- Seifert, M., Abou-El-Ardat, K., Friedrich, B., Klink, B. and Deutsch, A. (2014), Autoregressive higher-order hidden Markov models: exploiting local chromosomal dependencies in the analysis of tumor expression profiles. *PLoS one*, **9**(6), e100295.
- Shirley, K. E., Small, D. S., Lynch, K. G., Maisto, S. A. and Oslin, D. W. (2010), Hidden Markov models for alcoholism treatment trial data. *The Annals of Applied Statistics*, 366–395.
- Stanculescu, I., Williams, C. K. and Freer, Y. (2014), Autoregressive hidden Markov models for the early detection of neonatal sepsis. *IEEE journal of biomedical and health informatics*, **18**(5), 1560–1570.
- Takiguchi, T., Nakamura, S., Hou, Q. and Shikano, K. (1997), Model adaptation based on HMM decomposition for reverberant speech recognition. In Acoustics, Speech, and Signal Processing, 1997. ICASSP-97. *IEEE International Conference* (Vol. 2), 827–830.
- Zucchini, W., MacDonald, I. L. and Langrock, R. (2016), *Hidden Markov models for time series: an introduction using R*. Chapman and Hall/CRC.