Efficient Learning for Time Series Models by Non-Negative Moment Matrix Factorization

Abstract

In recent years, method-of-moments (MoM) based algorithms for latent variable models have been popular in the machine learning community, as a computationally cheaper alternatives to more conventional maximum likelihood based algorithms. Although there are MoM algorithms for the Hidden Markov Model (HMM), Gaussian Mixture Model (GMM), and Latent Dirichlet Allocation (LDA), it is unclear how to extend these methods to more complex models. In this paper, we develop a MoM-based approach using Non-negative Matrix Factorization (NMF) for learning several time series models. These include the Mixture of HMMs (MHMM), Switching HMM (SHMM) and Factorial HMM (FHMM). We show experimentally that NMF is considerably faster than conventional maximum-likelihood approaches and gives comparable estimation accuracies.

1. Introduction

In recent years, method-of-moments (MoM) learning algorithms with unique solution guarantees for several latent variable models have been developed in the machine learning literature. Examples include the Hidden Markov Model (HMM) (Hsu et al., 2009; Anandkumar et al., 2012c;b), Gaussian Mixture Model (GMM) (Hsu & Kakade, 2013), Latent Dirichlet Allocation (LDA) (Anandkumar et al., 2012a), and Independent Component Analysis (ICA) (Anandkumar et al., 2012b). These algorithms provide computationally cheaper alternatives to conventional Expectation-Maximization (EM) (Dempster et al., 1977) based approaches that are free of locally optimal solutions.

While there are MoM algorithms for a handful of models, a general framework that is as widely applicable as EM does not exist. Therefore, it is not clear how to derive MoM learning algorithms with unique solution guarantees for more sophisticated models such as the Mixture of Hidden Markov Models (MHMM) (Smyth, 1997), the Switching HMM (SHMM) (Murphy, 2002), or the Factorial HMM (FHMM) (Ghahramani & Jordan, 1997).

There exists Non-negative Matrix Factorization (NMF) (Lee & Seung, 1999) based MoM algorithms for learning the parameters of an HMM (Cybenko & Crespi, 2011; Lakshminarayanan & Raich, 2010; Shashanka, 2011). This approach amounts to doing alternating least squares or having an iterative multiplicative update scheme. However, it has not been applied to more sophisticated models.

In this paper, we propose a general NMF-MoM framework to learn the parameters of several time series models. We do this by converting each to an HMM with a specific transition structure and developing multiplicative update rules (Lee & Seung, 2001) that iteratively update the parameter matrices.

Although one loses the uniqueness properties of the spectral learning algorithms with NMF, the computational advantage of NMF-MoM is significant. Learning of time series models is typically done with maximum-likelihood training using EM (ML-EM). An important drawback of ML-EM is that one has to perform inference for the latent variables at each iteration. For example, for MHMM, one has to do forward-backward message-passing for each sequence at each iteration. Therefore, in abundance of data, learning the parameters of the model may be computationally expensive. However, learning the model with a NMF-MoM approach amounts to the estimation of some observable moments and the factorization of a moment matrix (or tensor) with NMF, where both of which can be executed quite fast.

We perform experiments on synthetic data to compare the accuracy and speed of the proposed learning methods with EM. We observe that our approach is substantially faster than ML-EM when data is plentiful. In addition, we achieve comparable estimation accuracies. We also conduct experiments with real datasets to show the validity of our approach.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Notation

We used square brackets $[-]$ for the indicator function. We use MATLAB notation to select a column or row of a matrix. For example, $O(:,k)$, $O(k,:)$ and $O(k,j)$ denote the k'th column and row and $(i,j)'th$ entry of the matrix O, respectively. We use boldface for matrices when doing matrix calculations. The outer product operator $\otimes$ is defined as $(a \otimes b)(i,j) = a(i)b(j)$. We use $\odot$ to denote elementwise matrix multiplication. We denote a sequence of length T as $x_{1:T} = \{x_1, x_2, \ldots, x_T\}$. We use boldface to denote a sequence so that $x_{1:T,n}$ =$\mathbf{x}_n$ and consequently we use $\mathbf{x}_{1:N} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ to denote a dataset of N sequences. Note that $x_t \in \mathbb{R}^L$ in general. The estimate for a parameter θ is shown as θ^*. Finally, we use subscript to denote parameters belonging to slices of a tensor, e.g. O_k matrix is the k'th slice of O tensor.

Some important indices, variables and parameters that are used throughout the paper are as follows. $n \in \{1,\ldots,N\}$: sequence index, $t \in \{1,\ldots,T\}$: time index, $k \in \{1,\ldots,K\}$: cluster index, $h \in \{1,\ldots,K\}$: cluster/regime indicator, $r_t \in \{1,\ldots J\}$: state indicator, $x_t \in \{1,\ldots L\}$: observation, $\mathbf{O}_k \in \mathbb{R}^{L\times J}$: cluster emission matrix, $\tilde{\mathbf{O}} = \begin{bmatrix}\mathbf{O}_1 & \mathbf{O}_2 & \cdots & \mathbf{O}_K\end{bmatrix} \in \mathbb{R}^{L\times KJ}$: stacked emission matrix, $\mathbf{B} \in \mathbb{R}^{K\times K}$: regime transition matrix, $\mathbf{A}_k \in \mathbb{R}^{J\times J}$: cluster transition matrix, $\boldsymbol{\nu}_k \in \mathbb{R}^J$: cluster initial state distibution, $\widehat{\mathbf{A}}_k = \mathbf{A}_k \text{diag}(\nu_{\mathbf{k}})$, $\tilde{\mathbf{A}} \in \mathbb{R}^{JK\times JK}$: global joint state matrix, $\boldsymbol{\pi} \in \mathbb{R}^K$: prior vector, $\mathbf{1}_{ab} \in \mathbb{R}^{a\times b}$: ones matrix.

3. Model Definitions

In this section, we give the definitions of the models used in this paper, and the corresponding variables and parameters. We also describe the modeling choices used throughout the paper.

3.1. Hidden Markov Model

In Hidden Markov Model (HMM), an observed sequence $\mathbf{x} = x_{1:T}$ is generated conditioned on a latent Markov chain $r_{1:T}$, with $r_t \in \{1,\ldots J\}$. Given the model parameters $\theta = (O, A, \nu)$, the likelihood $p(x_{1:T}|\theta)$ of an observation sequence $x_{1:T}$ of length T is defined as follows:

$$\begin{aligned} p(x_{1:T}|\theta) &= \sum_{r_{1:T}} p(x_{1:T}, r_{1:T}|\theta) \\ &= \sum_{r_{1:T}} \prod_{t=1}^{T} p(x_t|r_t)p(r_t|r_{t-1}) \\ &= \sum_{r_{1:T}} \prod_{t=1}^{T} O(x_t, r_t)A(r_t, r_{t-1}) \end{aligned} \tag{1}$$

The model parameters are defined as follows:

- $\nu(u) = p(r_1 = u|r_0) = p(r_1 = u)$, the initial latent state distribution.
- $A(u,v) = p(r_t = u|r_{t-1} = v)$, $t \geq 2$, latent state transition matrix.
- $O(:,u) = \mathbb{E}[x_t|r_t = u]$, emission matrix.

where, $\nu \in \mathbb{R}^J$, $A \in \mathbb{R}^{J\times J}$ and $O \in \mathbb{R}^{L\times J}$. A column $O(:,u)$ of the observation matrix O is defined as the expectation of x_t, conditioned on the latent state r_t. The choice of $p(x_t|r_t)$, determines what the columns of O correspond to. Some frequently used choices are:

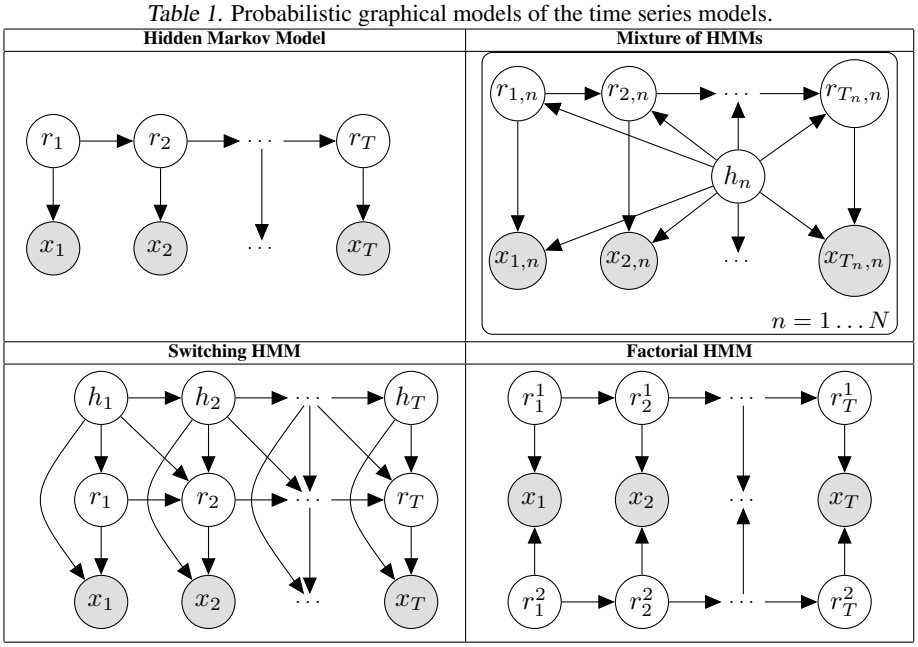
- Gaussian: $p(x_t|r_t = u) = \mathcal{N}(x_t; \mu_u, \sigma^2)$ then, $O(:,u) = \mathbb{E}[x_t|r_t = u] = \mu_u$.
- Poisson: $p(x_t|r_t = u) = \mathcal{PO}(x_t; \lambda_u)$ then, $O(:,u) = \mathbb{E}[x_t|r_t = u] = \lambda_u$.
- Multinomial: $p(x_t|r_t = u) = \text{Multinomial}(p_u, S)$ then, $O(:,u) = \mathbb{E}[x_t|r_t = u] = p_u$.

The first choice is multivariate, isotropic Gaussian with mean $\mu_u \in \mathbb{R}^L$. The second distribution is Poisson with intensity parameter $\lambda_u \in \mathbb{R}^L$. This choice is particularly useful for counts data. The last density is a multinomial distribution, with parameter $p_u \in \mathbb{R}^L$, and number of draws S. In this work, we use the Multinomial distribution in all of the models. In audio modeling applications, for example, S is taken to be a large number so that the observations describe the discrete Fourier transform (DFT) magnitude of the signal over time (Mysore et al., 2010).

3.2. Mixture of HMMs

Mixture of HMMs is a useful model for clustering sequences where each one is modeled with one of the K HMMs. Given model parameters $\theta = (O_{1:K}, A_{1:K}, \nu_{1:K}, \pi)$, the likelihood $p(\mathbf{x}_n|\theta)$ of an observation sequence $\mathbf{x}_n = \{x_{1,n}, x_{2,n}, \ldots, x_{T_n,n}\}$ of length T_n is computed as convex combination of the likelihood of K hidden Markov models:

Table 1. Probabilistic graphical models of the time series models.



$$p(\mathbf{x}_n|\theta) = \sum_{k=1}^{K} p(h_n = k)p(\mathbf{x}_n|h_n = k) \tag{2}$$

$$= \sum_{k=1}^{K} \pi_k \sum_{\mathbf{r}_n} p(\mathbf{x}_n, \mathbf{r}_n|h_n = k)$$

$$= \sum_{k=1}^{K} \pi_k \sum_{r_{1:T_n,n}} \prod_{t=1}^{T_n} p(x_{t,n}|r_{t,n}, h_n = k)p(r_{t,n}|r_{t-1,n}, h_n = k)$$

$$= \sum_{k=1}^{K} \pi_k \sum_{r_{1:T_n,n}} \prod_{t=1}^{T_n} O_k(x_{t,n}, r_{t,n})A_k(r_{t,n}, r_{t-1,n})$$

where, $h_n \in \{1, 2, \dots, K\}$ is the latent cluster indicator, and $\mathbf{r}_n = \{r_{1,n}, r_{2,n}, \dots, r_{T_n,n}\}$ is the latent state sequence of the observed sequence $\mathbf{x}_n$. $\pi \in \mathbb{R}^K$ is defined as the cluster prior probability vector. Note that, if a sequence is assigned to k'th cluster (meaning $h_n = k$), then the transition matrix A_k and the observation matrix O_k are used to generate that particular sequence. That is to say, we have a mixture of HMMs where each mixture component is specified by the model parameters θ_k.

3.3. Switching HMM

Switching state space HMM is a generalization of Mixture of HMMs, where the cluster indicator variable h is allowed to take differing values within a single sequence, as illustrated in the graphical model in Table 1. Allowing h to change within a sequence enables us to use this model for segmentation. The emission and transition matrices used to generate the observations vary as h changes throughout the sequence.

Given the model parameters $\theta = (O_{1:K}, A_{1:K}, \nu_{1:K}, B, \pi)$, the likelihood $p(x_{1:T}|\theta)$ of a given sequence $x_{1:T}$ is defined as follows:

$$p(x_{1:T}|\theta) = \sum_{h_{1:T}, r_{1:T}} p(x_{1:T}, h_{1:T}, r_{1:T}|\theta_{1:K}) \tag{3}$$

$$= \sum_{h_{1:T}, r_{1:T}} \prod_{t=1}^{T} p(x_t|r_t, h_t)p(r_t|r_{t-1}, h_t, h_{t-1})p(h_t|h_{t-1})$$

$$= \sum_{h_{1:T}, r_{1:T}} \prod_{t=1}^{T} O_{r_t}(x_t, r_t)\Big\{[h_t = h_{t-1}]A_{h_t}(r_t, r_{t-1}) + [h_t \neq h_{t-1}]A_{h_t,h_{t-1}}(r_t, r_{t-1})\Big\}B(h_t, h_{t-1})$$

where $h_t \in \{1, \dots K\}$ is the latent regime indicator, and $r_t \in \{1, \dots J\}$ is the latent state indicator. $A_{h_t,h_{t-1}}$ denotes the transition matrix to be used whenever there is a regime change ($h_t \neq h_{t-1}$). If there is no regime change ($h_t \neq h_{t-1}$), then $A_{h_t,h_{t-1}} = A_{h_t}$. Note that $A_{h_t,h_{t-1}}$ matrices are fixed in learning. The regime transition matrix $B \in \mathbb{R}^{K\times K}$ is defined as $B(u, v) = p(h_t = u|h_{t-1} = v)$, and regime initial distribution $\pi \in \mathbb{R}^K$ is defined as $\pi(u) = p(h_1 = u)$.

3.4. Factorial HMM

The factorial HMM is useful for modeling additive mixtures. An observed sequence is modeled as the weighted sum of the outputs of K HMMs. For $K = 2$, given the

model parameters $\theta = (O_{1:2}, A_{1:2}, \nu_{1:2})$, the likelihood $p(x_{1:T}|\theta)$ of a given sequence $x_{1:T}$ is defined as follows:

$$\begin{aligned} p(x_{1:T}|\theta) &= \sum_{r^1_{1:T}, r^2_{1:T}} p(x_{1:T}, r^1_{1:T}, r^2_{1:T}|\theta) \qquad (4) \\ &= \sum_{r^1_{1:T}, r^2_{1:T}} \prod_{t=1}^{T} p(x_t|r^1_t, r^2_t) p(r^1_t|r^1_{t-1}) p(r^2_t|r^2_{t-1}) \\ &= \sum_{r^1_{1:T}, r^2_{1:T}} \prod_{t=1}^{T} \frac{1}{2}\Big\{O_1(x_t, r^1_t) + O_2(x_t, r^2_t)\Big\} \\ &\qquad \times A_1(r^1_t, r^1_{t-1}) A_2(r^2_t, r^2_{t-1}) \end{aligned}$$

where $r^1_{1:T}$ is the first, and $r^2_{1:T}$ is the second latent chain. Conditioned on them, the observation sequence $x_{1:T}$ is generated using the observation model $\frac{1}{2}\Big\{O_1(x_t, r_t) + O_2(x_t, r_t)\Big\}$. Note that the mixing weights are set to $\frac{1}{2}$ in this paper. In general, mixing weights are also parameters to be estimated.

4. MoM-NMF algorithms for learning

A common approach to learn latent variable models like the ones that are defined in Section 3 is to use a maximum likelihood criterion. A parameter set θ^* is chosen such that the log-likelihood of the observed data is maximized:

$$\theta^* = \operatorname*{argmax}_{\theta} \log p(x|\theta) = \operatorname*{argmax}_{\theta} \log \sum_{h} p(x, h|\theta)\,, \quad (5)$$

where x is the observed data, h is a set of hidden variables, and θ is the parameter set to be optimized. Because of the presence of a summation over the latent variables h, which may be hard to compute in general, it is easier to iteratively maximize a lower bound on the log likelihood. The estimate in iteration τ is:

$$(\theta^\tau)^* = \operatorname*{argmax}_{\theta^\tau} \mathbb{E}_{p(\theta^{\tau-1}|x,h)}[\log p(x, h|\theta^\tau)]\,. \quad (6)$$

This EM lower bound involves computing the posterior $p(\theta^{\tau-1}|x, h)$ over the model parameters at each iteration. In the temporal models that we consider, this requires a message passing algorithm. For example, for a mixture of HMMs, we need to compute the posterior for each sequence $\mathbf{x}_n$ for each cluster k at each iteration τ. This can be computationally prohibitive for large datasets. The number of calculations required per iteration is $\mathcal{O}(J^2KTN + JKLTN)$, where N is the number of sequences and T is the average sequence length. Since the dependence is linear in N, EM is infeasible for large datasets.

A computationally cheaper learning scheme is as follows. We can compute moment estimates from the data and estimate the model parameters via a factorization of the moment matrix/tensor. We note that the 2nd order moment matrices for these models all factorize in the form:

$$\mathbf{P} := \mathbb{E}[x_t \otimes x_{t-1}] = \tilde{\mathbf{O}}\tilde{\mathbf{A}}\tilde{\mathbf{O}}^\top\,, \quad (7)$$

where $\mathbf{P} \in \mathbb{R}^{L\times L}$ is defined as the true moment matrix, and the block matrix $\tilde{\mathbf{O}} \in \mathbb{R}_+^{L\times JK}$ contains the HMMs' emission matrices, stacked side-by-side. $\tilde{\mathbf{A}} \in \mathbb{R}_+^{JK\times JK}$ is the joint state distribution matrix that contains the transition structure of the model embedded in a single, larger HMM. The details of $\tilde{\mathbf{A}}$ for each model are given in the subsequent sections.

Having made this observation, the learning problem reduces to the following non-negative factorization problem:

$$\theta^* = (\tilde{\mathbf{O}}^*, \tilde{\mathbf{A}}^*) = \operatorname*{argmin}_{\tilde{\mathbf{O}}\geq 0, \tilde{\mathbf{A}}\geq 0} d(\mathbf{V} \| \tilde{\mathbf{O}}\tilde{\mathbf{A}}\tilde{\mathbf{O}}^\top)\,, \quad (8)$$

where $\mathbf{V}$ is an empirical estimate for the moment matrix $\mathbf{P} = \mathbb{E}[x_t \otimes x_{t-1}]$, and $d(.\|.)$ is an appropriate divergence measure. We use KL divergence as it is commonly used in NMF community. We continue by providing 2nd order moment matrix decomposition descriptions for each model. Higher order decompositions for each model are given in the supplemental material.

4.1. Hidden Markov Model

The 2nd-order moment matrix of hidden Markov model is decomposed as follows:

$$\begin{aligned} \mathbf{P} =& \mathbb{E}\left[x_{t+1} \otimes x_t\right] \qquad (9) \\ =& \sum_{r_t} \sum_{r_{t+1}} \mathbb{E}[x_{t+1}|r_{t+1}] p\left(r_{t+1}|r_t\right) p(r_t) \mathbb{E}[x_t|r_t] \\ =& \sum_{r_t} \sum_{r_{t+1}} O(x_{t+1}, r_{t+1}) A(r_{t+1}|r_t) \nu(r_t) O(x_t, r_t) \\ =& \mathbf{O}\mathbf{A}\mathrm{diag}(\boldsymbol{\nu})\mathbf{O}^\top = \mathbf{O}\widehat{\mathbf{A}}\mathbf{O}^\top \end{aligned}$$

where $\mathbf{O}$ and $\mathbf{A}$ are respectively the emission and transition matrices of the HMM. Note that $\widehat{\mathbf{A}} = p\left(r_{t+1}, r_t\right)$ is the joint consecutive state distribution. Thus, the true moment $\mathbf{P}$ is:

$$\mathbf{P} = \mathbf{O}\widehat{\mathbf{A}}\mathbf{O}^\top = \tilde{\mathbf{O}}\tilde{\mathbf{A}}\tilde{\mathbf{O}}^\top\,. \quad (10)$$

The multiplicative update rules to factorize the empirical moment matrix $\mathbf{V}$ are:

Table 2. Global joint state distributions $\tilde{\mathbf{A}}$ for the time series models.

Hidden Markov Model	Mixture of HMMs
$\widehat{\mathbf{A}}$	$\begin{bmatrix} \pi_1\widehat{\mathbf{A}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \pi_2\widehat{\mathbf{A}}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \pi_K\widehat{\mathbf{A}}_K \end{bmatrix}$
Switching HMM	**Factorial HMM**
$\begin{bmatrix} \mathbf{A}_1\, B_{11} & \mathbf{A}_{12}\, B_{12} & \cdots & \mathbf{A}_{1K}\, B_{1K} \\ \mathbf{A}_{21}\, B_{21} & \mathbf{A}_2\, B_{22} & \cdots & \mathbf{A}_{2K}\, B_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{K1}\, B_{K1} & \mathbf{A}_{K2}\, B_{K2} & \cdots & \mathbf{A}_K\, B_{KK} \end{bmatrix} diag\,(\tilde{\boldsymbol{\pi}})$	$\frac{1}{K^2}\begin{bmatrix} \widehat{\mathbf{A}}_1 & \widehat{\mathbf{A}}_1\mathbf{1}_{JJ}\widehat{\mathbf{A}}_2 & \cdots & \widehat{\mathbf{A}}_1\mathbf{1}_{JJ}\widehat{\mathbf{A}}_K \\ \widehat{\mathbf{A}}_2\mathbf{1}_{JJ}\widehat{\mathbf{A}}_1 & \widehat{\mathbf{A}}_2 & \cdots & \widehat{\mathbf{A}}_2\mathbf{1}_{JJ}\widehat{\mathbf{A}}_K \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{\mathbf{A}}_K\mathbf{1}_{JJ}\widehat{\mathbf{A}}_1 & \widehat{\mathbf{A}}_K\mathbf{1}_{JJ}\widehat{\mathbf{A}}_2 & \cdots & \widehat{\mathbf{A}}_K \end{bmatrix}$

$$\tilde{\mathbf{O}} \leftarrow \tilde{\mathbf{O}} \odot \frac{\left(\frac{\mathbf{V}_{\cdot}}{\widehat{\mathbf{V}}}\right)\tilde{\mathbf{O}}\tilde{\mathbf{A}}^{\top} + \left(\frac{\mathbf{V}_{\cdot}}{\widehat{\mathbf{V}}}\right)^{\top}\tilde{\mathbf{O}}\tilde{\mathbf{A}}}{\mathbf{1}_{LL}\tilde{\mathbf{O}}\left(\tilde{\mathbf{A}}^{\top} + \tilde{\mathbf{A}}\right)}, \tag{11}$$

$$\tilde{\mathbf{A}} \leftarrow \tilde{\mathbf{A}} \odot \frac{\tilde{\mathbf{O}}^{\top}\left(\frac{\mathbf{V}_{\cdot}}{\widehat{\mathbf{V}}}\right)\tilde{\mathbf{O}}}{\tilde{\mathbf{O}}^{\top}\mathbf{1}_{LL}\tilde{\mathbf{O}}}. \tag{12}$$

where $\widehat{\mathbf{V}}$ is the approximation of $\mathbf{V}$ constructed with $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{O}}$. Matrix division is performed element-wise. Iterating (11) and (12) is guaranteed to converge to a local minimum of (8) (Lee & Seung, 2001).

4.2. Mixture of Hidden Markov Models

The 2nd-order moment matrix of the mixture of HMMs is decomposed as follows:

$$\begin{aligned}\mathbf{P} =& \mathbb{E}[x_{t+1} \otimes x_t] = \sum_{h=1}^{K} p(h) \cdots \\ & \sum_{r_t}\sum_{r_{t+1}} \mathbb{E}[x_{t+1}|r_{t+1}, h]p(r_t, r_{t+1}|h)\mathbb{E}[x_t|r_t, h] \\ =& \sum_{h=1}^{K} \pi(h) \sum_{r_t}\sum_{r_{t+1}} O_h(x_{t+1}, r_{t+1})A_h(r_{t+1}|r_t) \\ & \qquad \ldots \nu_k(r_t)O_h(x_t, r_t) \\ =& \sum_{k=1}^{K} \pi_k \mathbf{O}_k \widehat{\mathbf{A}}_k \mathbf{O}_k^{\top} = \tilde{\mathbf{O}}\tilde{\mathbf{A}}\tilde{\mathbf{O}}^{\top}\end{aligned} \tag{13}$$

We have dropped the sequence index n to avoid clutter and defined a block-diagonal $\tilde{\mathbf{A}}$ for the entire MHMM (see Table 2). Since the factorization is identical in form to that of the single HMM case, the NMF updates are identical to (11) and (12). A similar approach was taken to derive update rules based on the 3rd and 4th moments (see supplemental materials). Note that if $\tilde{\mathbf{A}}$ is initialized as block-diagonal, it will remain this way throughout the multiplicative updates.

4.3. Switching Hidden Markov Model

The 2nd-order moment matrix for the switching HMM is decomposed as follows:

$$\begin{aligned}\mathbf{P} =& \mathbb{E}[x_{t+1} \otimes x_t] = \sum_{h_t}\sum_{h_{t+1}} p\,(h_t, h_{t+1}) \sum_{r_t}\sum_{r_{t+1}} \cdots \\ & \mathbb{E}[x_{t+1}|r_{t+1}, h_{t+1}]p(r_{t+1}|r_t, h_t, h_{t+1})p(r_t|h_t)\mathbb{E}[x_t|r_t, h_t] \\ =& \sum_{h_t}\sum_{h_{t+1}} B(h_t, h_{t+1}) \sum_{r_t}\sum_{r_{t+1}} O_{h_{t+1}}(x_{t+1}, r_{t+1}) \cdots \\ & \qquad A_{h_{t+1},h_t}(r_{t+1}, r_t)\nu_{h_t}(r_t)O_{h_t}(x_t, r_t) \\ =& \sum_{k=1}^{K}\sum_{k'=1}^{K} \mathbf{O}_k\, \tilde{\mathbf{A}}_{(kk')}\mathbf{O}_{k'}^{\top} = \tilde{\mathbf{O}}\tilde{\mathbf{A}}\tilde{\mathbf{O}}^{\top}\end{aligned} \tag{14}$$

where $\tilde{\mathbf{A}}_{(kk')}$ denotes the $(k, k')^{\text{th}}$ block of the $\tilde{\mathbf{A}}$ matrix defined for the entire SHMM (see Table 2) and $\mathbf{B}$ is the regime transition matrix. The blocks of $\tilde{\mathbf{A}}$ encode the state transition information given the regime indicators h_t and h_{t+1}. The multiplicative update for $\tilde{\mathbf{O}}$ is identical to (11). The other updates are:

$$\mathbf{A}_{kk'} \leftarrow \mathbf{A}_{kk'} \odot \frac{\mathbf{O}_k^{\top}\left(\frac{\mathbf{V}_{\cdot}}{\widehat{\mathbf{V}}}\right)\mathbf{O}_{k'}}{\mathbf{O}_k^{\top}\mathbf{1}_{LL}\mathbf{O}_{k'}}, \tag{15}$$

$$B_{kk'} \leftarrow B_{kk'} \odot \frac{\mathbf{1}_{1L}\left[\left(\frac{\mathbf{V}_{\cdot}}{\widehat{\mathbf{V}}}\right) \odot \mathbf{O}_k\, \tilde{\mathbf{A}}_{(kk')}\, \mathbf{O}_{k'}^{\top}\right]\mathbf{1}_{L1}}{\mathbf{1}_{1L}\left[\mathbf{O}_k\, \tilde{\mathbf{A}}_{(kk')}\, \mathbf{O}_{k'}^{\top}\right]\mathbf{1}_{L1}}, \tag{16}$$

$$\tilde{\boldsymbol{\pi}} \leftarrow \tilde{\boldsymbol{\pi}} \odot \frac{\left[\tilde{\mathbf{A}}^{\top}\tilde{\mathbf{O}}^{\top}\left(\frac{\mathbf{V}_{\cdot}}{\widehat{\mathbf{V}}}\right) \odot \tilde{\mathbf{O}}^{\top}\right]\mathbf{1}_{L1}}{\tilde{\mathbf{A}}^{\top}\tilde{\mathbf{O}}^{\top}\mathbf{1}_{L1} \odot \tilde{\mathbf{O}}^{\top}\mathbf{1}_{L1}}, \tag{17}$$

where:

$$\tilde{\boldsymbol{\pi}} = \begin{bmatrix} \pi_1 \boldsymbol{\nu}_1^\top & \pi_2 \boldsymbol{\nu}_2^\top & \cdots & \pi_K \boldsymbol{\nu}_K^\top \end{bmatrix}^\top \ . \tag{18}$$

We can fix the off-diagonal blocks in $\tilde{\mathbf{A}}$ for simplicity. For example, we may know that when a regime change occurs (i.e. $h_t \neq h_{t+1}$), all states are equally likely at time $t+1$. In this case, $\mathbf{A}_{kk'} = \frac{1}{J}\mathbf{1}_{JJ} \quad , \quad k \neq k'$. Alternatively, we may impose that $r_{t+1} = 1$ (e.g. for left-to-right models used for speech or gesture datasets), in which case we have that $\mathbf{A}_{kk'} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^\top \mathbf{1}_{1J} \ , \ k \neq k'$.

4.4. Factorial Hidden Markov Model

In order for our approach to be tractable, an emission model $O(x_t, r_t^{1:K})$ is assumed:

$$O(x_t, r_t^{1:K}) = \frac{1}{K}\sum_{k=1}^{K} O_k(x_t, r_t^k) \ . \tag{19}$$

where $r_t^{1:K} = \left\{r_t^1, \ldots, r_t^K\right\}$ is the set of HMM states at time t. Since we average over the entire sequence to compute moments, this is equivalent to assuming that the mixing proportions $p(r_t^k)$ are uniform *on average*. This is reasonable in many cases.

The 2[nd]-order moment matrix for the factorial HMM is decomposed as follows:

$$\begin{aligned} \mathbf{P} =& \mathbb{E}\left[x_{t+1} \otimes x_t\right] \\ =& \sum_{r_t^{1:K}} \sum_{r_{t+1}^{1:K}} \mathbb{E}[x_{t+1}|r_{t+1}^{1:K}] p\left(r_{t+1}^{1:K}, r_t^{1:K}\right) \mathbb{E}[x_t|r_t^{1:K}] \\ =& \frac{1}{K^2}\sum_{k=1}^{K}\left[\mathbf{O}_k \widehat{\mathbf{A}}_k \mathbf{O}_k^\top + \sum_{k' \neq k} \mathbf{O}_k \widehat{\mathbf{A}}_k \mathbf{1}_{JJ} \widehat{\mathbf{A}}_{k'} \mathbf{O}_{k'}^\top\right] \\ =& \tilde{\mathbf{O}}\tilde{\mathbf{A}}\tilde{\mathbf{O}}^\top \ . \end{aligned} \tag{20}$$

where we have defined an $\tilde{\mathbf{A}}$ for the entire FHMM (see Table 2). This formulation can easily be generalized to the case where the HMMs have differing numbers of states J.

The NMF update for $\tilde{\mathbf{O}}$ is identical to (11). The updates for the transition matrices can be written as:

$$\mathbf{A}_k \leftarrow \mathbf{A}_k \odot \frac{\mathbf{E}_k \mathbf{R}_1 \left(\tilde{\mathbf{O}}^\top \left(\frac{\mathbf{V}_\cdot}{\widehat{\mathbf{V}}}\right) \tilde{\mathbf{O}} \odot \mathbf{Q}_k\right) \mathbf{R}_2 \mathbf{E}_k^\top}{\mathbf{E}_k \mathbf{R}_1 \left(\tilde{\mathbf{O}}^\top \mathbf{1}_{LL} \tilde{\mathbf{O}} \odot \mathbf{Q}_k\right) \mathbf{R}_2 \mathbf{E}_k^\top} \ . \tag{21}$$

where, for example, when $K = 2$:

$$\mathbf{Q}_1 = \begin{bmatrix} \mathbf{1}_{JJ} & \mathbf{1}_{JJ} \\ \mathbf{1}_{JJ} & \mathbf{0} \end{bmatrix}, \ \mathbf{Q}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{1}_{JJ} \\ \mathbf{1}_{JJ} & \mathbf{1}_{JJ} \end{bmatrix}, \tag{22}$$

$$\mathbf{R}_1 = \begin{bmatrix} \mathbf{I} & \mathbf{1}_{JJ} A_2^\top \\ \mathbf{1}_{JJ} A_1^\top & \mathbf{I} \end{bmatrix}, \ \mathbf{R}_2 = \begin{bmatrix} \mathbf{I} & A_1^\top \mathbf{1}_{JJ} \\ A_2^\top \mathbf{1}_{JJ} & \mathbf{I} \end{bmatrix},$$

$$\mathbf{E}_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \ , \ \mathbf{E}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} .$$

We also derived updates based on the factorization of the 3[rd] moment tensor (see supplemental materials).

4.5. Computational complexity

The computation complexities of the ML-EM and NMF-MoM algorithms are given in Table 3. The length T of the data sequences (and the number of sequences N, in the case of the MHMM) appears as a multiplicative factor for the ML-EM algorithms and as an additive term for the NMF algorithms. Although ML-EM typically converges in fewer iterations, message-passing in the E step is very costly. This shows the computational advantage of the MoM approach. We also note that for the factorial HMM, the EM algorithm runs in time that is exponential in the number of latent state chains K. However, this dependence is at worst cubic in the MoM methods. Thus, the computational advantage of the proposed approach may be substantial for complicated models.

5. Experiments

In all the synthetic experiments described in this section, parameter matrices were generated in the same way. The columns of the $\mathbf{O}$ matrices were generated from a Dirichlet distribution with parameters $\alpha_l = 0.15$, $l = 1, \ldots, L$. This ensures that they are relatively sparse. The transition matrices were generated as $\mathbf{A} = \beta\mathbf{I} + (1-\beta)\mathbf{I}_s$, where $0.6 < \beta < 1$ and $\mathbf{I}_s$ is the identity matrix with a downward circular shift applied to the rows. Thus, the $\mathbf{A}$ matrices are ergodic but relatively sparse. Prior vectors were generated from a uniform distribution over the probability simplex. For the NMF-MoM algorithms, parameters are initialized with random non-negative values.

5.1. Mixture of HMMs

5.1.1. Synthetic Data

The goal of the experiments with synthetic data is to understand the impact of dataset size on sequence clustering accuracy and computation time. We generated 100 sets of MHMM data consisting of 3 clusters, each of which contains 10 sequences and $J = 3$, $L = 9$. We report the clustering accuracies and estimation time for sequences of

Table 3. Computational complexities of ML-EM and NMF-MoM algorithms per iteration.

Model	EM	2^{nd} order	3^{rd} order	4^{th} order
MHMM	$\mathcal{O}(J^2KTN + JKLTN)$	$\mathcal{O}(J^2KL + JKL^2)$	$\mathcal{O}(J^2KL^2 + JKL^3)$	$\mathcal{O}(J^2KL^3 + JKL^4)$
SHMM	$\mathcal{O}(J^2K^2T + JKLT)$	$\mathcal{O}(J^2K^2L + JKL^2)$	$\mathcal{O}(J^2K^2L^2 + JKL^3)$	
FHMM	$\mathcal{O}(J^{2K}T + J^KLT)$	$\mathcal{O}(JKL^2 + J^2K^2L + J^3K^3)$	$\mathcal{O}(J^2K^3L^2 + JK^3L^3 + J^3K^3L)$	

length $L \in [10, 10,000]$ (see supplemental materials for pseudocode).

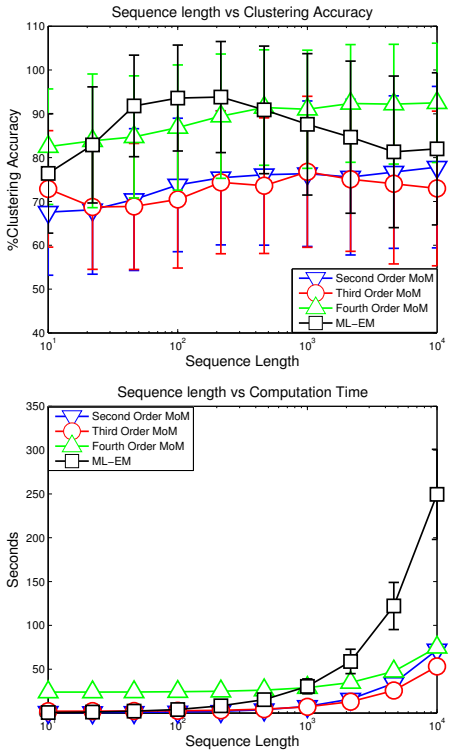


Figure 1. **Top:** Sequence Length vs Clustering accuracy Experiment, **Bottom:** Sequence Length vs Computation Time Experiment

We see in Figure 1 that the 4^{th} order MoM algorithm performs nearly as well as the maximum-likelihood based EM algorithm with short sequences. With long sequences, the MoM algorithm performs slightly better. The savings in computation with NMF-MoM is significant when the sequences are long. The increase in computation time is due to the assignment step for each sequence after learning (a single forward pass for each sequence). However, the situation is much worse for ML-EM since we need to do forward-backward passes for each sequence in every iteration. The EM iterations were stopped automatically as soon as the clustering became stable.

5.1.2. Real Data

We used the 3^{rd} order decomposition algorithm for the mixture of HMMs on the "Character Trajectories" dataset from

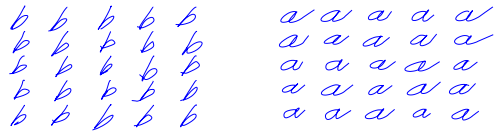


Figure 2. Clustering "a" and "b" character trajectories.

the UCI machine learning repository (Bache & Lichman, 2013). The data consists of handwritten character trajectories written on a PDA screen. The data in its original format is a 3-dimensional, continuous time series. The first two dimensions are the derivative of the spatial coordinates and the third is the pen pressure as a function of time. We discretized the data using K-means clustering with 14 clusters. The original dataset contains 2,858 sequences in 20 clusters. When we use the first 160 data items containing 97 "a" letters and 63 "b" letters, we obtain a 98% clustering accuracy. The clustering (with moment estimation) takes 7.5 seconds. Examples of clustered characters are given in Figure 2. NMF-MoM is able to cluster the trajectories correctly despite within-cluster variations.

5.2. Switching HMM

We ran 100 trials with synthetic data in which parameter matrices for a 2-regime SHMM were generated and a training sequence was sampled from the resulting model. The regime transition matrix was strongly diagonal. Each regime had 3 states. We also generated 10 test sequences of length 1,000 and dimensionality 10. Observation vectors were sampled from the appropriate multinomial distribution with 10 draws. The off-diagonal transition matrix blocks in $\tilde{\mathbf{A}}$ were set to be uniform. We learned the parameters with 2^{nd} and 3^{rd} order NMF-MoM algorithms as well as ML-EM. The NMF-MoM algorithms were run for 10,000 iterations each and the ML-EM algorithm was terminated when the log likelihood improved by less than 0.01%. The Viterbi algorithm was run on the test sequences and the decoding accuracy was measured via regime labeling. Thus, the specific state reported within a regime is irrelevant.

Accuracy and training time results for these experiments are shown in Figure 3. The NMF-MoM algorithms perform better than ML-EM in terms of regime labeling. They also don't suffer from a rapid increase in time complexity with increasing sequence length that is incurred due to the

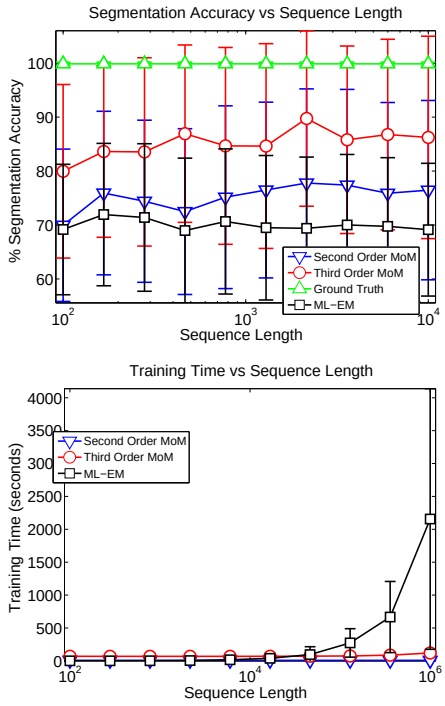


Figure 3. Results of synthetic trials for the switching HMM. **Top**: Segmentation accuracy. **Bottom**: Training time.

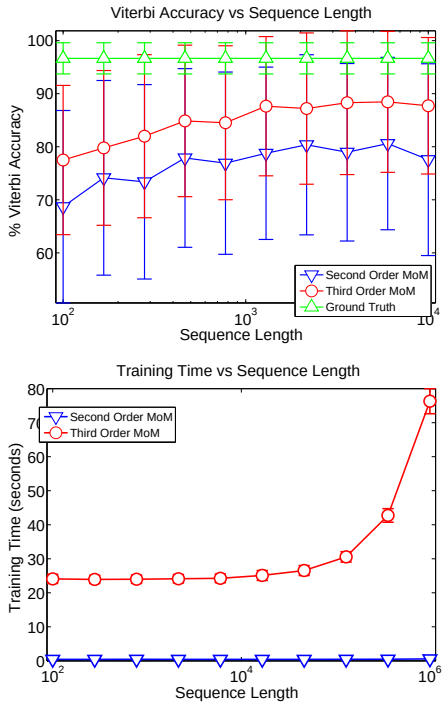


Figure 4. Results of synthetic trials for the factorial HMM. **Top**: Viterbi decoding accuracy. **Bottom**: Training time.

forward and backward passes of EM. The 3rd order NMF-MoM algorithm performs the best since it is able to take advantage of higher-order moment information.

5.3. Factorial HMM

The same synthetic trials were run for the factorial HMM as were run for the switching HMM. The 2nd and 3rd order NMF-MoM algorithms were run on the training sequence for 1,000 iterations each. The Viterbi algorithm was then applied to find the most likely state sequences for the test data.

The decoding accuracy and training time of the algorithms are shown in Figure 4. The 3rd order algorithm performs better than the 2nd order one, but it also takes significantly longer to run. This is due both to the increased amount of moment information and to the increased complexity of the updates.

We note that the ML-EM algorithm corresponding to the multinomial FHMM is non-trivial to implement since a closed-form expression for the mean updates cannot be found. These updates can be replaced with a gradient descent procedure. However, evaluating the objective function in this optimization step (i.e. data log likelihood) requires a forward pass along the observation sequence. In addition, we must consider J^K possible state pairs in both the E and M steps. These factors make the ML-EM algorithm extremely computationally prohibitive and so we have omitted a numerical comparison with it.

6. Conclusions

In this paper, we introduced a learning framework for time series models such as the mixture of HMMs, switching HMM, and factorial HMM. This involves computing observable moments and factorizing a non-negative moment matrix/tensor in terms of the model parameters. This is a general approach since we only require that the factorization exists. The main advantage of the proposed NMF-MoM algorithms is computational. Once moments are computed, the learning procedure is independent of the amount of data. We showed that for large datasets, this advantage is substantial. This suggests that this approach may be beneficial for on-line learning. At the same time, we observed that NMF-MoM often performs just as well as ML, if not better. As future work, it may also be beneficial to consider other divergence measures besides the KL criterion used in this work. Finally, we note that NMF-MoM may reach local optima. It is important to explore clever initialization strategies and methods for finding solutions with uniqueness guarantees, as has been done in the spectral learning community.

References

Anandkumar, A., Foster, D. P., Hsu, D., Kakade, S.M., and Liu, Y.K. Two svds suffice: Spectral decompositions for probabilistic topic modeling and latent dirichlet allocation. In *Neural Information Processing Systems*, 2012a.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S.M., and Telgarsky, M. Tensor decompositions for learning latent variable models. *arXiv:1210.7559v2*, 2012b.

Anandkumar, A., Hsu, D., and Kakade, S.M. A method of moments for mixture models and hidden markov models. In *COLT*, 2012c.

Bache, K. and Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Cybenko, G. and Crespi, V. Learning Hidden Markov Models using nonnegative matrix factorization. *IEEE Transactions on Information Theory*, 57(6):3963–3970, 2011.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1): 1–38, 1977.

Ghahramani, Z. and Jordan, M. I. Factorial hidden Markov models. *Journal of Machine Learning*, 29(2-3):245–273, 1997.

Hsu, Daniel and Kakade, Sham M. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Fourth Innovations in Theoretical Computer Science*, 2013.

Hsu, Daniel, Kakade, Sham M., and Zhang, Tong. A spectral algorithm for learning hidden markov models a spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, (1460-1480), 2009.

Lakshminarayanan, Balaji and Raich, Raviv. Non-negative matrix factorization for parameter estimation in hidden markov models. *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, pp. 89–94, 2010.

Lee, D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

Lee, D and Seung, H S. Algorithms for non-negative matrix factorization. *Conference on Advances in Neural Information Processing Systems*, pp. 556–562, 2001.

Murphy, K. P. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, Computer Science Division, 2002.

Mysore, Gautham J., Smaragdis, Paris, and Raj, Bhiksha. Non-negative Hidden Markov Modeling of Audio with Application to Source Separation. *In Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA / ICA)*, 2010.

Shashanka, Madhusudana. A fast algorithm for discrete hmm training using observed transitions. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

Smyth, P. Clustering sequences with hidden markov models. In *Advances in neural information processing systems*, 1997.