METHOD OF MOMENTS LEARNING FOR LEFT TO RIGHT HIDDEN MARKOV MODELS

Y. Cem Subakan[♭], Johannes Traa[♯], Paris Smaragdis[♭,♯,♮], Daniel Hsu[♯♯]

[♭]UIUC Computer Science Department, [♯♯]Columbia University Computer Science Department
[♯]UIUC Electrical and Computer Engineering Department, [♮]Adobe Systems, Inc.
{subakan2,traa2,paris}@illinois.edu, djhsu@cs.columbia.edu

ABSTRACT

In this paper, we propose a Method of Moments (MoM) algorithm for parameter learning in Left-to-Right Hidden Markov Models (LR-HMM). Compared to the conventional Expectation Maximization (EM) approach, the proposed algorithm is computationally more efficient, and hence more appropriate for large datasets. It is also asymptotically guaranteed to estimate the correct parameters. We show the validity of our approach with a synthetic data experiment and a word utterance onset detection experiment.

Index Terms— Method of Moments, Left to Right Hidden Markov Models

1. INTRODUCTION

Left-to-Right HMM (LR-HMM) is a very important HMM subclass for modeling signals with changing properties such as speech [1]. In an LR-HMM the state index increases or stays the same as time index increases. Imposing this one directional structure can be helpful in applications such as speech recognition and DNA sequence alignment [2].

Learning the parameters of HMMs from data is a non-trivial task because of the presence of hidden variables, which makes the standard maximum likelihood based objective analytically intractable, and hence there doesn't exist a closed form maximum likelihood estimator for HMMs. A very popular search heuristic for maximum likelihood learning in latent variable models such as HMMs is Expectation Maximization (EM). Despite its popularity, EM can be computationally expensive for large datasets, slow to converge and it doesn't have learning guarantees.

Recently, Method of Moments (MoM) algorithms for parameter learning in latent variable models have become popular in the machine learning community due to their computational advantages and learning guarantees [3, 4, 5, 6]. Despite being an attractive alternative for EM, the current MoM methods are unsuitable for LR-HMM because of the structural constraints of the model.

In this work, we adapt the two stage estimation procedure proposed in [7] and [8] to develop an efficient algorithm for LR-HMM. The two step estimation procedure separates the learning of the emission and transition matrix. This separation allows us to form a moment based objective function, in which we are able to enforce the structural left-to-right constraint.

Our main contribution is to provide two algorithms which estimate the left-to-right structure in a general LR-HMM, and a more constrained non-state-skipping version, which is also known as Bakis HMM [1]. The overall estimation algorithm is asymptotically guaranteed to give the true parameters. Moreover, we experimentally show on synthetic data that the proposed approach is computationally cheaper than EM, and can be used as an economical initializer for EM. We also provide a real data experiment on speech data, where we detect the onsets of a word in a recording of repeated utterances.

2. DEFINITIONS

2.1. Notation

We use the MATLAB colon notation $A(:,j), A(j,:), B(:,:,j)$, which in this case picks the j'th column/row of a matrix A, and the j'th slice of the tensor B. We use the subscript notation $x_{1:T}$ to denote $\{x_1, x_2, \ldots, x_T\}$. $\Delta^{N-1} := \{(p_1, p_2, \ldots, p_N) \in \mathbb{R}^N : p_i \geq 0 \ \forall i, \ \sum_{i=1}^N p_i = 1\}$ denotes a probability simplex in $\mathbb{R}^N$. $\Delta^{N-1\times N}$ denotes the space of column stochastic matrices in $\mathbb{R}^N$. $\mathbf{1}(arg)$ denotes an indicator function: If arg is true then the output is 1, otherwise the output is zero. For a positive integer N, let $[N] := \{1, \ldots, N\}$. Let $e_i \in \mathbb{R}^N$ denote an indicator vector, where only the i'th entry is one and the rest is zero. We use $\text{diag}(x)$ to put the vector x on the diagonal entries of a matrix. 1_N denotes an all ones vectors of length N. $A \cdot B$ denotes the element-wise multiplication of A and B matrices.

2.2. Hidden Markov Model

Hidden Markov Model (HMM) is a statistical sequence model, where observations are generated conditioned on a latent Markov chain. The observations are denoted with x_t, which are real vectors in $\mathbb{R}^L$ in the Gaussian case, and indicator vectors $e_{1:L}$ in the discrete case. Latent states are denoted with $r_t \in [M]$. The HMM is parametrized by $\theta = (O, A, \nu)$, which are respectively the conditional mean matrix (or the emission matrix) $O = \mathbb{E}[x_t|r_t] \in \mathbb{R}^{L\times M}$, the transition matrix $A \in \Delta^{M-1\times M}$ (we use column stochasticity), and the initial state distribution $\nu \in \Delta^{M-1}$. Conditional means correspond to the mean vectors in Gaussian case, and the emission probabilities of symbols in the discrete case. Note that one may add the covariance matrices to the list of parameters for each state in the Gaussian case. Overall, the generative model of an HMM is defined as follows,

$$\begin{aligned} r_1 &\sim \text{Categorical}(\nu), \\ r_t|r_{t-1} &\sim \text{Categorical}(A(:, r_{t-1})), \ \forall t \in \{2, \ldots, T\}, \\ x_t|r_t &\sim p_{r_t}(x_t), \ \forall t \in [T], \end{aligned}$$

where $p_{r_t}(x_t)$ is the emission density that corresponds to the r_t'th state.

2.3. Left to Right HMM (LR-HMM)

As mentioned in the introduction, in an LR-HMM as time progresses state index increases or stays the same. This means that the transition matrix A of an LR-HMM is a lower triangular matrix with the entries above the main diagonal equal to zero. We denote the binary mask that corresponds to the lower diagonal structure as $\mathcal{S}_{LR}$ such that $(1 - \mathcal{S}_{LR}) \cdot A = \mathbf{0}$.

In the general LR-HMM that has just been defined, it is allowed to transition into the proceeding state by skipping intermediate states. In some applications such as speech [1], it is favorable to further constrain the transition structure such that only a pre-defined step transitions are allowed. This is also known as the Bakis model. In this paper, we consider the one-step transitions case.

The transition matrix structure in a Bakis HMM is such that only the main diagonal and the first lower diagonal are non-zero. Note that to complete the cycle (to have an ergodic Bakis HMM) one can place a non-zero number in the top-right entry.

3. LEARNING

The dominant method for learning LR-HMM is Maximum Likelihood. However, a closed form Maximum Likelihood estimator does not exist for HMMs, and a popular search heuristic is Expectation Maximization (EM) [9] algorithm. The performance of EM is highly dependent on the initialization as it is not guaranteed to converge to a global optimum. Another major drawback of EM is computational since the E-step requires forward-backward message passing for each sequence.

In this work, we adapt the two stage HMM learning algorithm in [7], which separates the learning of the emission matrix and the transition matrix. Since the estimation of the transition matrix is formulated as an individual convex optimization problem, we are able to enforce the left-to-right structure with an affine constraint in the form of $A \cdot (1 - \mathcal{S}_{LR}) = 0$. The main obstacle from directly applying a lower triangular mask $\mathcal{S}_{LR}$ is the fact that we do not know the left-to-right ordering of the states. For this reason, we propose two algorithms to learn the state index ordering for the general LR-HMM and Bakis HMM.

The resulting algorithm is computationally much cheaper, and therefore is scalable to larger datasets as shown in Section 4.1. Also unlike EM, it doesn't require initialization. We give the summary of the proposed algorithm below in Algorithm 1.

Algorithm 1 Summary of the overall learning procedure

1. Estimate $\widehat{O}$ and corresponding mixing weights $\widehat{\pi}$. (Section 3.1).
2. Estimate $\widehat{A}$ using the output of stage 1 (Section 3.2).
3. Learn the mask $\mathcal{M}$ to suppress unwanted entries for the appropriate model (Section 3.3).
4. Refine $\widehat{A}$ using the mask $\mathcal{M}$ (Section 3.4).

3.1. Estimation of the Emission Matrix

In an HMM, the emission density can be learnt independently of the transition matrix by treating the observations as i.i.d. and fitting a mixture distribution. The generative model that corresponds to this i.i.d. 'interpretation' is as follows,

$$i \sim \mathcal{U}_{[T]},\ r|i \sim \pi_i,\ x \sim p_r(x), \tag{1}$$

where, i is a random index drawn from a uniform distribution over time instances $[T]$, π_i is the state marginal at random time i, r is a state indicator and x is a data point. Marginalizing over the random index i, we obtain a mixture model with mixing weights $\pi := \sum_{i=1}^{T} p(r,i) = \frac{1}{T}\sum_{i=1}^{T} \pi_i$.

Following this rationale, an MoM subroutine can be utilized to obtain the estimates $\widehat{O} = O_\epsilon P^\top$ and $\widehat{\pi} = P\pi_\epsilon$ to fit a mixture distribution, where O_ϵ, π_ϵ are respectively noisy (but not permuted) versions of the true emission matrix O and the true mixing weight vector π. P is the permutation matrix that corresponds to inherent permutation of the state labels in the learning of the mixture model. A typical choice for the mixture learning subroutine is the tensor power method [5]. This procedure uses second and third order moments to learn the emission parameters of the mixture model, and requires that O has full column rank.

3.2. Initial Estimation of the Transition Matrix

The second order moment of an HMM is decomposed in terms of model parameters as follows [4, 3]:

$$\begin{aligned} Q_{2,1} &:= \mathbb{E}[x_2 x_1^\top] = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[x_{t+1} x_t^\top], \\ &= OA\text{diag}\left(\frac{1}{T}\sum_{t=1}^{T}\pi_t\right) O^\top = OA\text{diag}(\pi)O^\top, \end{aligned} \tag{2}$$

where π is the mixing weight vector defined in the previous section. Note that the sequence that corresponds to $Q_{2,1}$ is of length $T+1$. Given estimates $\widehat{O}$, $\widehat{\pi}$ from the first stage, and an empirical second order moment $\widehat{Q_{2,1}}$, we try to estimate a feasible transition matrix which minimizes the Frobenius norm of the deviation between the empirical moment and the factorization in Equation (2) using the following convex program:

$$\begin{aligned} \widehat{A} = \text{argmin}_A \ \ & \|\widehat{Q_{2,1}} - \widehat{O}A\text{diag}(\widehat{\pi})\widehat{O}^\top\|_F \\ s.t.\ \ & 1_M^\top A = 1_M^\top,\ A \geq 0, \end{aligned} \tag{3}$$

where the constraints ensure that the estimate is a column stochastic transition matrix. Note that the estimate is $\widehat{A} = PA_\epsilon P^\top$, where A_ϵ is the noisy estimate of A without permutation. Note that $\widehat{A}$ is noisy and therefore may not be in a lower triangular (or Bakis) form. In the subsequent sections, we will first recover the permutation due to the mixture learning step and then zero out the unwanted entries in $\widehat{A}$ by adding an affine constraint to the convex program that has just been described.

3.3. Learning the Refinement Mask

The goal in this step is to recover the left-to-right ordering of the states and to be able to zero out the unwanted entries in $\widehat{A}$ later on. We denote the permutation of the state labels due to the mixture learning in the first step with $\mathcal{P}$, and we denote the binary mask that will suppress the unwanted entries with $\mathcal{M}$. We provide algorithms to estimate $\mathcal{M}$ both for the general LR-HMM and the more constrained Bakis HMM.

3.3.1. Depermutation for a general LR-HMM

In the case of general LR-HMM, the transition matrix is lower diagonal in its original form. So, if we threshold the entries which are

greater than zero, and look at the row sums we will see a monotonic increase in the row sums. (First row 1, second one 2, ..., last one M). Therefore, if there is no noise, one can simply first threshold the entries and reorder the states such that row sums are increasing.

In the following Lemma, we establish the noisy version of this algorithm. We conclude that, if the minimum of the entries in the lower diagonal is larger than the maximum of the upper diagonal entries, then the same algorithm can be applied by setting the threshold to the maximum of the upper diagonal.

Lemma 1: *Let us define* $\mathcal{L} := \min_{i \geq j}(A_\epsilon)_{i.j}$, $\mathcal{U} := \max_{i<j}(A_\epsilon)_{i.j}$, *and* $\sigma_i := \sum_{j=1}^{M} \mathbf{1}((A_\epsilon)_{i,j} > \mathcal{U})$. *If* $\mathcal{L} > \mathcal{U}$, *then* $\sigma_i < \sigma_{i+1}, \forall i \in [M-1]$.

Proof: Since $\mathcal{L} > \mathcal{U}$, only the lower diagonal elements will contribute to the sum σ_i. And therefore $\sigma_i = i$. So we conclude that $\sigma_{i+1} > \sigma_i$. □

The following lemma establishes that the row sum vector of the thresholded $\widehat{A}$ is the same vector as the permuted version of the row sum vector of A_ϵ. So, the sorting we find from the row sum vector of the thresholded $\widehat{A}$ corresponds to $\mathcal{P}$.

Lemma 2: *Let* $\widehat{A} := PA_\epsilon P^\top$, *and* $\sigma_i^{\mathcal{P}} := \sum_{j=1}^{M} \mathbf{1}(\widehat{A}_{i,j} > \mathcal{U})$. *Then,* $\sigma_i^{\mathcal{P}} = \sigma_{\mathcal{P}(i)}$, *where* $\mathcal{P}$ *denotes the inherent permutation mapping of the state labels caused by the emission density estimation step.*

Proof:

$$\sigma_i^{\mathcal{P}} = \sum_{j=1}^{M} \mathbf{1}(\widehat{A}_{i,j} > \mathcal{U}) = \sum_{j=1}^{M} \mathbf{1}((A_\epsilon)_{\mathcal{P}(i),\mathcal{P}(j)} > \mathcal{U}),$$

$$\sigma_{\mathcal{P}(i)} = \sum_{j=1}^{M} \mathbf{1}((A_\epsilon)_{\mathcal{P}(i),j} > \mathcal{U}) = \sum_{j=1}^{M} \mathbf{1}((A_\epsilon)_{\mathcal{P}(i),\mathcal{P}(j)} > \mathcal{U}) = \sigma_i^{\mathcal{P}}.$$

So, we see that $\sigma_i^{\mathcal{P}} = \sigma_{\mathcal{P}(i)}$. □

We conclude that, if the noise on the upper triangular elements is less strong than the values of the lower diagonal elements, then by thresholding the matrix $\widehat{A}$ with a threshold γ and sorting $\sigma_{1:M}^{\mathcal{P}}$, we can provide an estimate for $\mathcal{P}$. In practice several γ values can be tried, and the one that maximizes $\sum_{i \geq j} \widehat{A}_{i,j}$ is chosen. The depermutation algorithm is given in algorithm 2. We index the used thresholds by e and denote the corresponding permutation mapping with $\mathcal{P}_e$. At the end of the Algorithm we choose the permutation that results in the most lower diagonal matrix.

Algorithm 2 The Depermutation Algorithm

Input: Noisy and Permuted Transition Matrix $\widehat{A}$.
Output: Binary Mask $\mathcal{M}$.

for $e = 1 : E$ **do**
  Compute $\sigma_i^{\mathcal{P}_e} := \sum_{j=1}^{M} \mathbf{1}(\widehat{A}_{i,j} > \gamma_e), \forall i \in \{1, \ldots, M\}$
  Find $\mathcal{P}_e$ by sorting $\widetilde{\sigma}_{1:M}$.
  Compute $L(e) := \sum_{i \geq j} \widehat{A}_{\mathcal{P}_e(i),\mathcal{P}_e(j)}$
end for
return $\mathcal{M} = \mathcal{P}_{\text{argmax}_{e'} L(e')}(\mathcal{S}_{LR})$;

3.3.2. *Learning the refinement mask for Bakis HMM*

In the previous section we have described an algorithm for the general LR-HMM. Unfortunately that algorithm is not applicable to the Bakis HMM case, since the lower triangular entries contain values which are zero. In this section, we describe a separate algorithm for this case.

The state transition structure of a Bakis HMM defines a Hamiltonian path (a tour along the vertices, with each vertex visited exactly once) in the state space. Finding an Hamiltonian path is known to be NP-Hard [10]. We therefore propose a greedy algorithm in Algorithm 3.

Algorithm 3 The greedy algorithm for Bakis HMM

Input: Noisy and Permuted Transition Matrix $\widehat{A}$.
Output: Binary Mask $\mathcal{M}$.

for $k = 1 : M$ **do**
  $i = 1$; $j = k$; vsts $= \{j\}$;
  while $i < M$ **do**
    $j' = \text{argmax}_{l \in \{1,\ldots M\} \setminus \text{vsts}} A_{l,j}$;
    vsts $= \{\text{vsts}, j'\}$; ▷ Add j' to the list of visited vertices.
    $j = j'$; ▷ Next state to visit is j'.
    $\text{Masks}(j', j, k) = 1$;
    $i = i + 1$;
    $A'(:,:,k) = Normalize(A. * (\text{Masks}(:,:,l) + I))$;
  end while
  $\text{Masks}(k, j', k) = 1$; ▷ Optional step to complete the cycle.
end for
$k' = \text{argmax}_{l \in \{1,\ldots M\}} p(x_{1:T}|A'(:,:,l)) - \lambda R(A(:,:,l))$;
return $\mathcal{M} = \text{Masks}(:,:,k') + I$;

This algorithm finds Hamiltonian paths starting from every state, and then picks the one yielding the highest likelihood. If the number of sequences is small, then it is possible to use a regularizer $R(.)$ for the choice of k'. For example in a musical chord segmentation experiment, one can choose the transition matrix which yields the Viterbi decoding with most uniform-length segments, since a-priori we know that chords are played for similar durations. Next, we show that if the estimated transition matrix is close to the true transition matrix, the algorithm returns the correct answer, and consequently as the number of observed sequences tends to infinity, the Algorithm is guaranteed to return the true parameters up to a permutation of the state indices.

Definition: *Let* $\epsilon := \|\widehat{A} - PAP^\top\|_1$, *where* $\|.\|_1$ *computes the sum of absolute values of the argument.*

Lemma 3: *If* $\epsilon \leq \min_j \max_{i \neq j} A_{i,j}$, *then the output of Algorithm 3 satisfies* $(1 - \mathcal{M}) \cdot \widehat{A} = \mathbf{0}$.

Proof: The condition requires that the deviation ϵ should be smaller than the smallest of second largest column entries in A. If this is satisfied, then then the algorithm will find the true Hamiltonian path since the true path will remain unaltered in $\widehat{A}$. □

Theorem: *As the number of observed sequences* $N \to \infty$, *Algorithm 3 is guaranteed to find the true mask* $\mathcal{M}$.

Proof Sketch: As $N \to \infty$, the estimates of the tensor power method converges to the true emission matrix O and the mixing

weights π. Furthermore, due to law of large numbers $\widehat{Q_{2,1}} \to Q_{2,1}$. When this is the case one can show that a pseudo-inverse estimator $\widehat{O}^{\dagger}\widehat{Q_{2,1}}(\widehat{O}^{\top})^{\dagger}\text{diag}(\widehat{\pi})^{-1}$ converges to A. Since $\text{argmin}_{A'}\|Q_{2,1} - OA'\text{diag}(\pi)O^{\top}\|_F$ is in the feasible region, the solution of the optimization problem in Section 3.2 is equal to this pseudo inverse estimator, and therefore $\epsilon \to 0$. This results in the condition in Lemma 3 being satisfied, and therefore Algorithm 3 is guaranteed to return the true mask $\mathcal{M}$. □

3.4. Refinement of the estimated transition matrix

As discussed earlier, once the mask $\mathcal{M}$ is learned, we re-run the convex optimization procedure described in Section 3.2 with an additional constraint defined by $\mathcal{M}$ to suppress the unwanted non-zero elements in the estimated transition matrix.

$$\widehat{A}_{ref} = \text{argmin}_A \ \|\widehat{Q_{2,1}} - \widehat{O}A\text{diag}(\widehat{\pi})\widehat{O}^{\top}\|_F \quad (4)$$
$$s.t. \ \ 1_M^{\top}A = 1_M^{\top}, \ A \geq 0, \ A \cdot (1 - \mathcal{M}) = \mathbf{0}.$$

Once the refined transition matrix is obtained with this second round of convex optimization, the learning procedure is concluded. Note that besides the computational advantage of the method which is illustrated in Section 4.1, the overall method is very easy to implement. Once the emission matrix is obtained, one can use an off-the shelf convex programming language such as CVX [11] for the estimation of the transition matrix.

4. EXPERIMENTS

4.1. Synthetic Data Experiment

In this Section, we experimentally studied the time-accuracy tradeoff for the proposed algorithm (MoM), expectation maximization (EM), and EM initialized by MoM for various number of EM iterations. For sequence lengths 400, 4000, and 40000 we generated 10 sequences for two classes from Bakis HMM, with 4 hidden states and Gaussian emission model. The means of the Gaussians were drawn from a zero mean unit variance Gaussian. The observation model was a Gaussian with variance 8. We learned Bakis HMMs from these sequences. We then did Viterbi decoding on 10 test sequences generated from the same Bakis HMMs used in training. For EM learning we used a code which has the E-step implemented in MEX. For EM, we did 5 random initializations, and accepted the new set of parameters when we observed an increase in the log-likelihood. We repeated the experiment for 5 times. Error bars show the standard deviation of accuracy over the random repeats. We observed that the variance of the repeats vanished for longer sequences. The time-accuracy tradeoff curves averaged over 5 repeats are given in Figure 1. We see that MoM is faster for longer sequences and thus more scalable than EM.

4.2. Real Data Experiment

In this section, we work on detecting the speech onsets on a long sequence consisted of 48 concatenated utterances of digit 7 by the same person. We trained an ergodic Bakis HMM on the sequence, and used the Viterbi state sequence decoding to detect the utterance onsets. We defined an onset as a transition from the last state to the first state of the HMM, which are respectively determined by setting the first state as the very first and last elements of the Viterbi sequence. We used 29-dimensional MFCC features. To measure the performance of the onset detection we used the precision, recall and F-measure criterions defined in [12]. Similar to the previous section

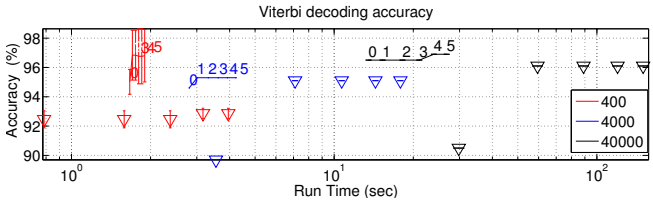

Figure 1: Time-Accuracy tradeoff curve for Synthetic Experiment. Different colors correspond to different sequence lengths. Triangles show the performance of randomly initialized EM. Different points with the same color correspond to an random EM initialization. (The further to the right, the larger the number of initializations) The solid curves show the performance of EM initialized with MoM. The numbers correspond to the number of EM iterations after initialization (0 means MoM only). Time axis is logarithmic.

we compared the proposed algorithm (MoM), randomly initialized EM (we used 5 random restarts and report only the restarts until the best F-measure), and EM initialized by MoM. Note that, the forward-backward part of the EM code is implemented with MEX, and the proposed method is implemented in MATLAB, with the optimization part implemented with CVX [11]. The F-measure - time tradeoff curves for 4 different sequence lengths are given in Figure 2. We are able to use sequences longer than 48 utterances by replicating the sequence. The numbers given in the legend correspond to the number of replicates.

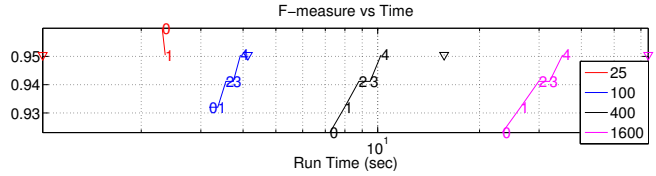

Figure 2: F-measure/time tradeoff curves. Different colors correspond to different number of replicates. Triangles correspond to randomly initialized EM. The solid curves correspond to EM initialized with MoM. The numbers show the number of EM iterations after initialization (0 means MoM only). Time axis is logarithmic.

As can be seen from the figure, the proposed Algorithm provides a more scalable alternative to EM. Even though the forward backward part of the EM code is implemented with MEX, the proposed algorithm is faster in longer sequence lengths. We also see that it's a fast way for initializing EM.

5. CONCLUSIONS

We have proposed a novel algorithm based on Method of Moments for learning Left to Right HMMs. As we see in synthetic and real data experiments, the proposed algorithm is a scalable alternative for EM and also can be used to initialize EM. Unlike EM, the proposed algorithm also has asymptotic convergence guarantee.

6. ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant No. 1319708.

7. REFERENCES

[1] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition (1989)," *Proceedings of the IEEE*, pp. 257–286, 1989.

[2] O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models (Springer Series in Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[3] D. Hsu, S. M. Kakade, and T. Zhang, "A spectral algorithm for learning hidden markov models," *Journal of Computer and System Sciences*, no. 1460-1480, 2009.

[4] A. Anandkumar, D. Hsu, and S. Kakade, "A method of moments for mixture models and hidden markov models," in *COLT*, 2012.

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

[6] D. Hsu and S. Kakade, "Learning mixtures of spherical gaussians: moment methods and spectral decompositions," in *Fourth Innovations in Theoretical Computer Science*, 2013.

[7] A. Kontorovich, B. Nadler, and R. Weiss, "On learning parametric-output hmms," in *International Conference of Machine Learning (ICML)*, 2013.

[8] A. T. Chaganty and P. Liang, "Estimating latent-variable graphical models using moments and likelihoods," in *International Conference of Machine Learning (ICML)*, 2014.

[9] A. Dempster, N. Laird, and D.B.Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, 1977.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.

[11] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[12] P. Brossier and P. Leveau, "2013:audio onset detection," http://www.music-ir.org/mirex/wiki/2013:Audio_Onset_Detection, June 2013.