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

We propose a method-of-moments algorithm for parameter learning in Left-to-Right Hidden Markov Models. Compared to the conventional Expectation Maximization 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 Hidden Markov Models (LR-HMMs) comprise an important subclass of Hidden Markov Models (HMMs) for modeling time series data [1]. In an LR-HMM, the hidden state space is linearly ordered; as time progresses, the state index either increases or stays the same. Imposing this one directional structure is extremely 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 alternatives to (or initializers for) EM in the machine learning community due to their computational advantages and theoretical guarantees [3, 4, 5, 6]. However, current MoM algorithms 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-stage 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 algorithm is a statistically consistent estimator of the true model parameters. 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, respectively, the j'th column and row of a matrix A, and the j'th slice of the third-order tensor B. We use the subscript notation $x_{1:T}$ to denote the set $\{x_1, x_2, \ldots, x_T\}$. The probability simplex in $\mathbb{R}^N$ is denoted by $\Delta^{N-1} := \{(p_1, p_2, \ldots, p_N) \in \mathbb{R}^N : p_i \geq 0 \ \forall i, \ \sum_{i=1}^N p_i = 1\}$, and $\Delta^{N-1\times N}$ is the space of column-stochastic matrices in $\mathbb{R}^N$. The indicator function is denoted by $\mathbf{1}(arg)$: 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. All ones vectors of length N is denoted by 1_N. Element-wise multiplication of A and B matrices is denoted by $A \odot B$.

2.2. Hidden Markov Models

HMMs are statistical sequence models where observations $x_{1:T}$ are generated independently conditioned on a latent Markov chain $r_1 \to r_2 \to \cdots$. The observation at time t is denoted by a random vector x_t in $\mathbb{R}^L$ (where discrete observations are encoded using indicator vectors), and the corresponding latent state is denoted by $r_t \in [M]$. HMM distributions with discrete observations in the set $e_{1:L}$ are parametrized by $\theta = (O, A, \nu)$: an emission matrix $O \in \Delta^{L-1\times M}$, a transition matrix $A \in \Delta^{M-1\times M}$, and an initial state distribution $\nu \in \Delta^{M-1}$. (We use column-stochastic matrices.) For HMMs with continuous observations, the emission matrix O is replaced by (parameters of) M emission distributions over $\mathbb{R}^L$; in this case, we regard O as the conditional mean matrix where $O(:,j) = \mathbb{E}[x_t|r_t{=}j]$ (which is consistent with the discrete observations case). Overall, the generative model of an HMM is defined as follows:

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

where p_{r_t} is the emission density corresponding to the r_t'th state.

2.3. Left-to-Right and Bakis HMMs

As mentioned in the introduction, in an LR-HMM, the state index increases or stays the same as time progresses. 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, and there exists at least one non-zero entry below the diagonal per column.

For a family of transition matrices $\mathcal{A} \subseteq \Delta^{M-1\times M}$, we say $\mathcal{S} \in \{0,1\}^{M\times M}$ is the binary mask for $\mathcal{A}$ if $\mathcal{S}_{i,j} = \mathbf{1}(\exists A \in \mathcal{A}$ s.t. $A_{i,j}>0)$, so $A = A \odot \mathcal{S}$ for all $A \in \mathcal{A}$. Therefore, the binary mask $\mathcal{S}_{LR}$ for LR-HMMs is given by $(\mathcal{S}_{LR})_{i,j} = \mathbf{1}(i \geq j)$.

As can be inferred from the above paragraph, general LR-HMMs allow transitions between non-adjacent states (e.g., a transition from state 2 to 5). In some applications such as speech recognition [1], it is desirable to further constrain the transition structure so that only transitions among neighboring states are allowed. Such models are called Bakis HMMs. In this work, we consider Bakis HMMs with one-step transitions (i.e., from state i to $i+1$). The transition matrix structure in a Bakis HMM is such that only the main diagonal and the first lower diagonal are non-zero. Thus, the binary mask $\mathcal{S}_B$ for Bakis HMMs is given by $(\mathcal{S}_B)_{i,j} = \mathbf{1}(i{=}j$ or $i{=}j{+}1)$. We also consider Bakis HMMs where transitions from the final state to the initial state are permitted, so the corresponding mask $\mathcal{S}_{B'}$ in this case also has $(\mathcal{S}_{B'})_{1,M} = 1$.

3. LEARNING

The dominant method for learning LR-HMMs is the Expectation Maximization (EM) algorithm [9]. EM is a heuristic for maximum likelihood estimation that locally optimizes the likelihood objective. Due to the non-convexity of the likelihood objective, EM may return a local maximizer, and hence the performance of EM depends strongly on the initial parameter setting. It is common to run EM many times with different (random) initializations, but this can be computationally inefficient. Another drawback of EM is that each iteration requires a computationally expensive 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 $A \odot (1_M 1_M^\top - \mathcal{S}_{LR}) = 0$. The main obstacle from directly applying the true mask $\mathcal{S}_{LR}$ is due to the fact that we do not know the left-to-right ordering of the states after the parameter estimation step. For this reason, we first need to learn the permuted binary mask $\mathcal{M} := P\mathcal{S}_{LR}P^\top$, where P is the permutation matrix that corresponds to the permutation mapping introduced by the parameter estimation procedure. We propose two algorithms to learn the state index ordering for the general LR-HMM and Bakis HMM.

The resulting general learning algorithm (outlined in Algorithm 1) is computationally cheaper than EM and is scalable to large datasets, as we demonstrate in Section 4.1.

Algorithm 1 Outline of the overall learning procedure

1. Estimate $\widehat{O}$ and 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,

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

where i is a random index drawn from a uniform distribution $\mathcal{U}_{[T]}$ over time indices $[T]$, π_i is the state marginal distribution for (random) time i, r is a hidden state variable, and x is an observation. Marginalizing out the random index i, we see that x has a mixture distribution with mixing weights $\pi := \frac{1}{T}\sum_{i=1}^{T} \pi_i$.

Following this rationale, an MoM subroutine can be used to estimate O and π, up to some unknown permutation P of the state labels (given as a permutation matrix). Let $\widehat{O}$ and $\widehat{\pi}$ denote these estimates, and let $O_\epsilon := \widehat{O}P^{-\top}$ and $\pi_\epsilon := P^{-1}\widehat{\pi}$ be the corresponding estimates after un-permuting the state labels. One 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]:

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

where π is the mixing weight vector defined in the previous subsection. (For simplicity, we assume the sequence that corresponds to $Q_{2,1}$ has length $T+1$.) Given estimates $\widehat{O}, \widehat{\pi}$ from the first stage, and an empirical second-order moment $\widehat{Q_{2,1}}$, we compute a transition matrix that minimizes the Frobenius norm of the deviation between the empirical moment and the factorization from Equation (2) by solving the following convex program:

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

As before, let $A_\epsilon := P^{-1}\widehat{A}P^{-\top}$ denote the estimate after un-permuting the state labels. Note that the estimate $\widehat{A}$ is (yet) not constrained to be in a lower triangular (or Bakis) form; we address this issue in the next subsections.

3.3. Learning the Refinement Mask

We now discuss how to recover the left-to-right ordering of the states which is needed to enforce the lower triangular (or Bakis) structure in $\widehat{A}$.

Let $\mathcal{P}\colon [M] \to [M]$ denote the permutation corresponding to the permutation matrix P from the first step of the algorithm. Once again note that $\mathcal{M} = P\mathcal{S}P^\top$ denotes the permuted binary mask that corresponds to transition matrix structure (e.g., $\mathcal{M} = P\mathcal{S}_{LR}P^\top$ in the general LR-HMM case). We give an algorithm to estimate $\mathcal{M}$ for both 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 triangular in its original form. This means that for each $i \in [M]$, the i'th row in A has the fewest non-zero entries among all rows $i' \geq i$ after excluding the diagonal entries of A and columns $j < i$. Thus, a simple greedy algorithm recovers the permutation $\mathcal{P}$ from a sufficiently accurate estimate $\widehat{A}$ of the transition matrix A (Algorithm 2).

Algorithm 2 The Depermutation Algorithm for General LR-HMM

Input: Transition matrix $\widehat{A}$, threshold $\gamma \geq 0$ (default: $\gamma = 0$).
Output: Binary mask $\widehat{\mathcal{M}}$.
Initialize $\widetilde{\mathcal{M}}_{i,j} := \mathbf{1}(\widehat{A}_{i,j} > \gamma)$ for all $(i,j) \in [M]^2$; $v := [M]$.
for $i = 1, 2, \ldots, M$ **do**
  $\mathcal{P}(i) := \arg\min_{i' \in v} \sum_{j \in v \setminus \{i'\}} \widetilde{\mathcal{M}}_{i',j}$.
  $v := v \setminus \{\mathcal{P}(i)\}$.
end for
return $\widehat{\mathcal{M}} = P\mathcal{S}_{LR}P^\top$, where P is the permutation matrix corresponding to $\mathcal{P}$.

Algorithm 2 takes a threshold parameter γ as input, which in practice can be tuned using cross-validation (e.g., try several values for γ and choose the result yielding the model with highest held-out likelihood). In the next section, we describe an algorithm specifically tailored for Bakis HMMs that avoids this extra parameter.

3.3.2. Learning the refinement mask for Bakis HMM

The state transition structure of a Bakis HMM defines an 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. This algorithm finds Hamiltonian paths starting from every state, and then picks the one yielding the highest likelihood.

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 entries of the argument.*

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

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 the empirical moment converges to the true moment: $\widehat{Q_{2,1}} \to Q_{2,1}$. When this is the case, one can show that a pseudo-inverse estimator $\widehat{O}^\dagger \widehat{Q_{2,1}} (\widehat{O}^\top)^\dagger \mathrm{diag}(\widehat{\pi})^{-1}$ converges to A. Since $\arg\min_{A'} \|Q_{2,1} - OA'\mathrm{diag}(\pi)O^\top\|_F$ 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 $\mathcal{M}$. □

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**
  Initialize $\mathrm{Masks}(:,:,k) = \mathbf{0}$;
  $i = 1$; $j = k$; $\mathrm{vsts} = \{j\}$;
  while $i < M$ **do**
    $j' = \arg\max_{l \in \{1,\ldots,M\} \setminus \mathrm{vsts}} \widehat{A}_{l,j}$;
    $\mathrm{vsts} = \{\mathrm{vsts}, j'\}$; ▷ Add j' to the list of visited vertices.
    $\mathrm{Masks}(j', j, k) = 1$;
    $j = j'$; ▷ Next state to visit is j'.
    $i = i + 1$;
  end while
  $\mathrm{Masks}(k, j', k) = 1$; ▷ Optional step to complete the cycle.
  $\widehat{A}'(:,:,k) = \mathrm{Normalize}(\widehat{A} \odot (\mathrm{Masks}(:,:,k) + I))$; ▷ Normalize $\widehat{A}$ according to the estimated mask
end for
$k' = \arg\max_{l \in \{1,\ldots,M\}} p(x_{1:T} | \widehat{A}'(:,:,l))$;
▷ Pick the mask with the largest likelihood.
return $\mathcal{M} = \mathrm{Masks}(:,:,k') + I$;

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.

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

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 trade-off 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 of EM 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

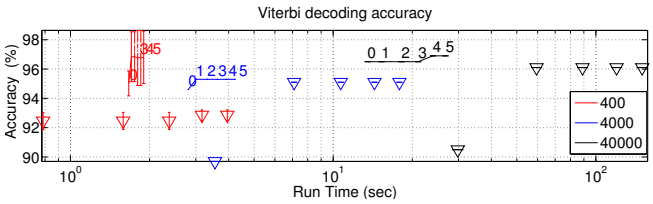

Figure 1: Time-Accuracy tradeoff curve for Synthetic Experiment. Different colors correspond to different sequence lengths. Triangles show the performance of randomly initialized EM. Different points with the same color correspond to an random EM initialization. (The further to the right, the larger 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.

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 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.

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.

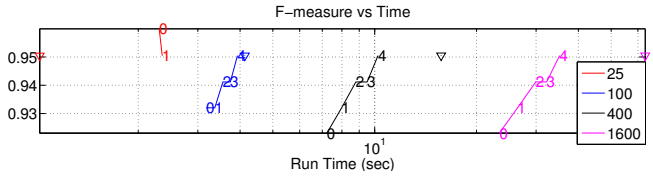

Figure 2: F-measure/time tradeoff curves. Different colors correspond to different number of replicates. Triangles correspond to randomly initialized EM. The solid curves correspond to EM initialized with MoM. The numbers show the number of EM iterations after initialization (0 means MoM only). Time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

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," *Journal of Machine Learning Research*, 2014.

[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.