Spectral Learning of Refinement HMMs

Karl Stratos[1] **Alexander M. Rush**[2] **Shay B. Cohen**[1] **Michael Collins**[1]

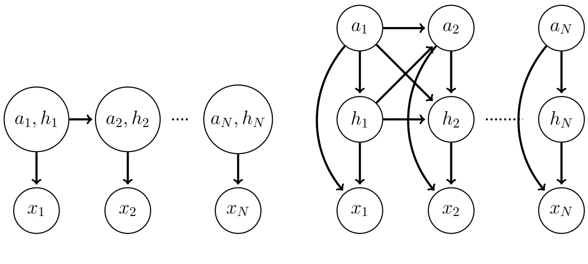
[1]Department of Computer Science, Columbia University, New-York, NY 10027, USA
[2]MIT CSAIL, Cambridge, MA, 02139, USA
{stratos,scohen,mcollins}@cs.columbia.edu, srush@csail.mit.edu

Abstract

We derive a spectral algorithm for learning the parameters of a refinement HMM. This method is simple, efficient, and can be applied to a wide range of supervised sequence labeling tasks. Like other spectral methods, it avoids the problem of local optima and provides a consistent estimate of the parameters. Our experiments on a phoneme recognition task show that when equipped with informative feature functions, it performs significantly better than a supervised HMM and competitively with EM.

1 Introduction

Consider the task of supervised sequence labeling. We are given a training set where the j'th training example consists of a sequence of observations $x_1^{(j)} ... x_N^{(j)}$ paired with a sequence of labels $a_1^{(j)} ... a_N^{(j)}$ and asked to predict the correct labels on a test set of observations. A common approach is to learn a joint distribution over sequences $p(a_1 \dots a_N, x_1 \dots x_N)$ as a hidden Markov model (HMM). The downside of HMMs is that they assume each label a_i is independent of labels before the previous label a_{i-1}. This independence assumption can be limiting, particularly when the label space is small. To relax this assumption we can refine each label a_i with a hidden state h_i, which is not observed in the training data, and model the joint distribution $p(a_1 \dots a_N, x_1 \dots x_N, h_1 \dots h_N)$. This refinement HMM (R-HMM), illustrated in figure 1, is able to propagate information forward through the hidden state as well as the label.



(a) (b)

Figure 1: (a) An R-HMM chain. (b) An equivalent representation where labels and hidden states are intertwined.

Unfortunately, estimating the parameters of an R-HMM is complicated by the unobserved hidden variables. A standard approach is to use the expectation-maximization (EM) algorithm which has no guarantee of finding the global optimum of its objective function. The problem of local optima prevents EM from yielding statistically consistent parameter estimates: even with very large amounts of data, EM is not guaranteed to estimate parameters which are close to the "correct" model parameters.

In this paper, we derive a spectral algorithm for learning the parameters of R-HMMs. Unlike EM, this technique is guaranteed to find the true parameters of the underlying model under mild conditions on the singular values of the model. The algorithm we derive is simple and efficient, relying on singular value decomposition followed by standard matrix operations.

We also describe the connection of R-HMMs to L-PCFGs. Cohen et al. (2012) present a spectral algorithm for L-PCFG estimation, but the naïve transformation of the L-PCFG model and its spectral algorithm to R-HMMs is awkward and opaque. We therefore work through the non-trivial derivation the spectral algorithm for R-HMMs.

We note that much of the prior work on spectral algorithms for discrete structures in NLP has shown limited experimental success for this family of algorithms (see, for example, Luque et al., 2012). Our experiments demonstrate empirical

Proceedings of the Seventeenth Conference on Computational Natural Language Learning, pages 56–64,
Sofia, Bulgaria, August 8-9 2013. ©2013 Association for Computational Linguistics

success for the R-HMM spectral algorithm. The spectral algorithm performs competitively with EM on a phoneme recognition task, and is more stable with respect to the number of hidden states.

Cohen et al. (2013) present experiments with a parsing algorithm and also demonstrate it is competitive with EM. Our set of experiments comes as an additional piece of evidence that spectral algorithms can function as a viable, efficient and more principled alternative to the EM algorithm.

2 Related Work

Recently, there has been a surge of interest in spectral methods for learning HMMs (Hsu et al., 2012; Foster et al., 2012; Jaeger, 2000; Siddiqi et al., 2010; Song et al., 2010). Like these previous works, our method produces consistent parameter estimates; however, we estimate parameters for a supervised learning task. Balle et al. (2011) also consider a supervised problem, but our model is quite different since we estimate a joint distribution $p(a_1 \dots a_N, x_1 \dots x_N, h_1 \dots h_N)$ as opposed to a conditional distribution and use feature functions over both the labels and observations of the training data. These feature functions also go beyond those previously employed in other spectral work (Siddiqi et al., 2010; Song et al., 2010). Experiments show that features of this type are crucial for performance.

Spectral learning has been applied to related models beyond HMMs including: head automata for dependency parsing (Luque et al., 2012), tree-structured directed Bayes nets (Parikh et al., 2011), finite-state transducers (Balle et al., 2011), and mixture models (Anandkumar et al., 2012a; Anandkumar et al., 2012b).

Of special interest is Cohen et al. (2012), who describe a derivation for a spectral algorithm for L-PCFGs. This derivation is the main driving force behind the derivation of our R-HMM spectral algorithm. For work on L-PCFGs estimated with EM, see Petrov et al. (2006), Matsuzaki et al. (2005), and Pereira and Schabes (1992). Petrov et al. (2007) proposes a split-merge EM procedure for phoneme recognition analogous to that used in latent-variable parsing.

3 The R-HMM Model

We decribe in this section the notation used throughout the paper and the formal details of R-HMMs.

3.1 Notation

We distinguish row vectors from column vectors when such distinction is necessary. We use a superscript $\top$ to denote the transpose operation. We write $[n]$ to denote the set $\{1, 2, \dots, n\}$ for any integer $n \geq 1$. For any vector $v \in \mathbb{R}^m$, $\text{diag}(v) \in \mathbb{R}^{m \times m}$ is a diagonal matrix with entries $v_1 \dots v_m$. For any statement $\mathcal{S}$, we use $[[\mathcal{S}]]$ to refer to the indicator function that returns 1 if $\mathcal{S}$ is true and 0 otherwise. For a random variable X, we use $\mathbf{E}[X]$ to denote its expected value.

A tensor $C \in \mathbb{R}^{m \times m \times m}$ is a set of m^3 values $C_{i,j,k}$ for $i, j, k \in [m]$. Given a vector $v \in \mathbb{R}^m$, we define $C(v)$ to be the $m \times m$ matrix with $[C(v)]_{i,j} = \sum_{k \in [m]} C_{i,j,k} v_k$. Given vectors $x, y, z \in \mathbb{R}^m$, $C = xy^\top z^\top$ is an $m \times m \times m$ tensor with $[C]_{i,j,k} = x_i y_j z_k$.

3.2 Definition of an R-HMM

An R-HMM is a 7-tuple $\langle l, m, n, \pi, o, t, f \rangle$ for integers $l, m, n \geq 1$ and functions π, o, t, f where

- $[l]$ is a set of labels.
- $[m]$ is a set of hidden states.
- $[n]$ is a set of observations.
- $\pi(a, h)$ is the probability of generating $a \in [l]$ and $h \in [m]$ in the first position in the labeled sequence.
- $o(x|a, h)$ is the probability of generating $x \in [n]$, given $a \in [l]$ and $h \in [m]$.
- $t(b, h'|a, h)$ is the probability of generating $b \in [l]$ and $h' \in [m]$, given $a \in [l]$ and $h \in [m]$.
- $f(*|a, h)$ is the probability of generating the stop symbol $*$, given $a \in [l]$ and $h \in [m]$.

See figure 1(b) for an illustration. At any time step of a sequence, a label a is associated with a hidden state h. By convention, the end of an R-HMM sequence is signaled by the symbol $*$.

For the subsequent illustration, let N be the length of the sequence we consider. A *full sequence* consists of labels $a_1 \dots a_N$, observations $x_1 \dots x_N$, and hidden states $h_1 \dots h_N$. The model assumes

$$p(a_1 \dots a_N, x_1 \dots x_N, h_1 \dots h_N) = \pi(a_1, h_1) \times \prod_{i=1}^{N} o(x_i|a_i, h_i) \times \prod_{i=1}^{N-1} t(a_{i+1}, h_{i+1}|a_i, h_i) \times f(*|a_N, h_N)$$

```
Input: a sequence of observations $x_1 \dots x_N$; operators $\left\langle C^{b|a}, C^{*|a}, c^1_a, c^a_x \right\rangle$
Output: $\mu(a,i)$ for all $a \in [l]$ and $i \in [N]$

[Forward case]
  • $\alpha^1_a \leftarrow c^1_a$ for all $a \in [l]$.
  • For $i = 1 \dots N-1$
        $\alpha^{i+1}_b \leftarrow \sum_{a \in [l]} C^{b|a}(c^a_{x_i}) \times \alpha^i_a$ for all $b \in [l]$

[Backward case]
  • $\beta^{N+1}_a \leftarrow C^{*|a}(c^a_{x_N})$ for all $a \in [l]$
  • For $i = N \dots 1$
        $\beta^i_a \leftarrow \sum_{b \in [l]} \beta^{i+1}_b \times C^{b|a}(c^a_{x_i})$ for all $a \in [l]$

[Marginals]
  • $\mu(a,i) \leftarrow \beta^i_a \times \alpha^i_a$ for all $a \in [l], i \in [N]$
```

Figure 2: The forward-backward algorithm

A *skeletal sequence* consists of labels $a_1 \dots a_N$ and observations $x_1 \dots x_N$ without hidden states. Under the model, it has probability

$$p(a_1 \dots a_N, x_1 \dots x_N) = \sum_{h_1 \dots h_N} p(a_1 \dots a_N, x_1 \dots x_N, h_1 \dots h_N)$$

An equivalent definition of an R-HMM is given by organizing the parameters in matrix form. Specifically, an R-HMM has parameters $\left\langle \pi^a, o^a_x, T^{b|a}, f^a \right\rangle$ where $\pi^a \in \mathbb{R}^m$ is a column vector, o^a_x is a row vector, $T^{b|a} \in \mathbb{R}^{m \times m}$ is a matrix, and $f^a \in \mathbb{R}^m$ is a row vector, defined for all $a, b \in [l]$ and $x \in [n]$. Their entries are set to

- $[\pi^a]_h = \pi(a,h)$ for $h \in [m]$
- $[o^a_x]_h = o(x|a,h)$ for $h \in [m]$
- $[T^{b|a}]_{h',h} = t(b,h'|a,h)$ for $h, h' \in [m]$
- $[f^a]_h = f(*|a,h)$ for $h \in [m]$

4 The Forward-Backward Algorithm

Given an observation sequence $x_1 \dots x_N$, we want to infer the associated sequence of labels under an R-HMM. This can be done by computing the *marginals* of $x_1 \dots x_N$

$$\mu(a,i) = \sum_{a_1 \dots a_N \colon a_i = a} p(a_1 \dots a_N, x_1 \dots x_N)$$

for all labels $a \in [l]$ and positions $i \in [N]$. Then the most likely label at each position i is given by

$$a^*_i = \arg\max_{a \in [l]} \mu(a,i)$$

The marginals can be computed using a tensor variant of the forward-backward algorithm, shown in figure 2. The algorithm takes additional quantities $\left\langle C^{b|a}, C^{*|a}, c^1_a, c^a_x \right\rangle$ called the *operators*:

- Tensors $C^{b|a} \in \mathbb{R}^{m \times m \times m}$ for $a, b \in [l]$
- Tensors $C^{*|a} \in \mathbb{R}^{1 \times m \times m}$ for $a \in [l]$
- Column vectors $c^1_a \in \mathbb{R}^m$ for $a \in [l]$
- Row vectors $c^a_x \in \mathbb{R}^m$ for $a \in [l]$ and $x \in [n]$

The following proposition states that these operators can be defined in terms of the R-HMM parameters to guarantee the correctness of the algorithm.

Proposition 4.1. *Given an R-HMM with parameters $\left\langle \pi^a, o^a_x, T^{b|a}, f^a \right\rangle$, for any vector $v \in \mathbb{R}^m$ define the operators:*

$$C^{b|a}(v) = T^{b|a} \mathit{diag}(v) \qquad c^1_a = \pi^a$$
$$C^{*|a}(v) = f^a \mathit{diag}(v) \qquad c^a_x = o^a_x$$

Then the algorithm in figure 2 correctly computes marginals $\mu(a,i)$ under the R-HMM.

The proof is an algebraic verification and deferred to the appendix. Note that the running time of the algorithm as written is $O(l^2 m^3 N)$.[1]

Proposition 4.1 can be generalized to the following theorem. This theorem implies that the operators can be linearly transformed by some invertible matrices as long as the transformation leaves the embedded R-HMM parameters intact. This observation is central to the derivation of the spectral algorithm which estimates the linearly transformed operators but not the actual R-HMM parameters.

Theorem 4.1. *Given an R-HMM with parameters $\left\langle \pi^a, o^a_x, T^{b|a}, f^a \right\rangle$, assume that for each $a \in [l]$ we have invertible $m \times m$ matrices G^a and H^a. For any vector $v \in \mathbb{R}^m$ define the operators:*

$$C^{b|a}(v) = G^b T^{b|a} \mathit{diag}(v H^a)(G^a)^{-1} \quad c^1_a = G^a \pi^a$$
$$C^{*|a}(v) = f^a \mathit{diag}(v H^a)(G^a)^{-1} \qquad c^a_x = o^a_x (H^a)^{-1}$$

Then the algorithm in figure 2 correctly computes marginals $\mu(a,i)$ under the R-HMM.

The proof is similar to that of Cohen et al. (2012).

[1] We can reduce the complexity to $O(l^2 m^2 N)$ by precomputing the matrices $C^{b|a}(c^a_x)$ for all $a, b \in [l]$ and $x \in [n]$ after parameter estimation.

5 Spectral Estimation of R-HMMs

In this section, we derive a consistent estimator for the operators $\langle C^{b|a}, C^{*|a}, c^1_a, c^a_x \rangle$ in theorem 4.1 through the use of singular-value decomposition (SVD) followed by the method of moments.

Section 5.1 describes the decomposition of the R-HMM model into random variables which are used in the final algorithm. Section 5.2 can be skimmed through on the first reading, especially if the reader is familiar with other spectral algorithms. It includes a detailed account of the derivation of the R-HMM algorithm.

For a first reading, note that an R-HMM sequence can be seen as a right-branching L-PCFG tree. Thus, in principle, one can convert a sequence into a tree and run the inside-outside algorithm of Cohen et al. (2012) to learn the parameters of an R-HMM. However, projecting this transformation into the spectral algorithm for L-PCFGs is cumbersome and unintuitive. This is analogous to the case of the Baum-Welch algorithm for HMMs (Rabiner, 1989), which is a special case of the inside-outside algorithm for PCFGs (Lari and Young, 1990).

5.1 Random Variables

We first introduce the random variables underlying the approach then describe the operators based on these random variables. From $p(a_1 \dots a_N, x_1 \dots x_N, h_1 \dots h_N)$, we draw an R-HMM sequence $(a_1 \dots a_N, x_1 \dots x_N, h_1 \dots h_N)$ and choose a time step i uniformly at random from $[N]$. The random variables are then defined as

$$\begin{aligned}
&X = x_i \\
&A_1 = a_i \text{ and } A_2 = a_{i+1} && \text{(if } i = N,\ A_2 = *\text{)} \\
&H_1 = h_i \text{ and } H_2 = h_{i+1} \\
&F_1 = (a_i \dots a_N, x_i \dots x_N) && \text{(future)} \\
&F_2 = (a_{i+1} \dots a_N, x_{i+1} \dots x_N) && \text{(skip-future)} \\
&P = (a_1 \dots a_i, x_1 \dots x_{i-1}) && \text{(past)} \\
&R = (a_i, x_i) && \text{(present)} \\
&D = (a_1 \dots a_N, x_1 \dots x_{i-1}, x_{i+1} \dots x_N) && \text{(destiny)} \\
&B = [[i = 1]]
\end{aligned}$$

Figure 3 shows the relationship between the random variables. They are defined in such a way that the future is independent of the past and the present is independent of the destiny conditioning on the current node's label and hidden state.

Next, we require a set of feature functions over the random variables.

- ϕ maps F_1, F_2 to $\phi(F_1), \phi(F_2) \in \mathbb{R}^{d_1}$.

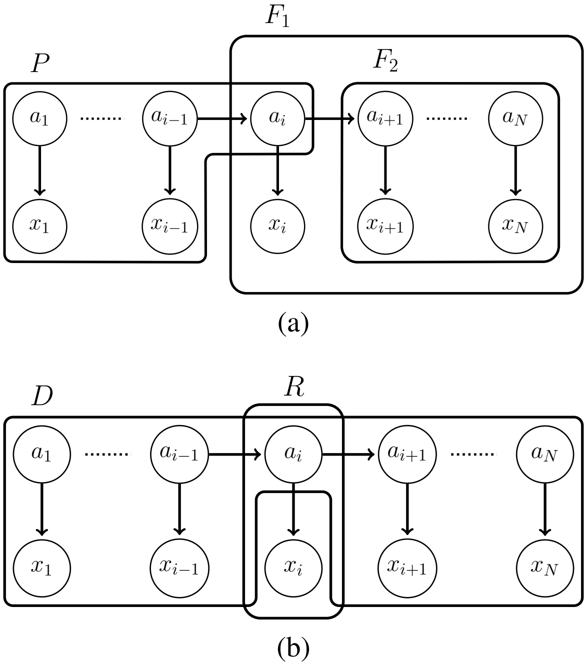


Figure 3: Given an R-HMM sequence, we define random variables over observed quantities so that conditioning on the current node, (a) the future F_1 is independent of the past P and (b) the present R is independent of the density D.

- ψ maps P to $\psi(P) \in \mathbb{R}^{d_2}$.
- ξ maps R to $\xi(R) \in \mathbb{R}^{d_3}$.
- υ maps D to $\upsilon(D) \in \mathbb{R}^{d_4}$.

We will see that the feature functions should be chosen to capture the influence of the hidden states. For instance, they might track the next label, the previous observation, or important combinations of labels and observations.

Finally, we require projection matrices

$$\Phi^a \in \mathbb{R}^{m \times d_1} \qquad \Psi^a \in \mathbb{R}^{m \times d_2}$$
$$\Xi^a \in \mathbb{R}^{m \times d_3} \qquad \Upsilon^a \in \mathbb{R}^{m \times d_4}$$

defined for all labels $a \in [l]$. These matrices will project the feature vectors of ϕ, ψ, ξ, and υ from (d_1, d_2, d_3, d_4)-dimensional spaces to an m-dimensional space. We refer to this reduced dimensional representation by the following random variables:

$$\begin{aligned}
&\underline{F}_1 = \Phi^{A_1}\phi(F_1) && \text{(projected future)} \\
&\underline{F}_2 = \Phi^{A_2}\phi(F_2) && \text{(projected skip-future: if } i = N,\ \underline{F}_2 = 1\text{)} \\
&\underline{P} = \Psi^{A_1}\psi(P) && \text{(projected past)} \\
&\underline{R} = \Xi^{A_1}\xi(R) && \text{(projected present)} \\
&\underline{D} = \Upsilon^{A_1}\upsilon(D) && \text{(projected destiny)}
\end{aligned}$$

Note that they are all vectors in $\mathbb{R}^m$.

5.2 Estimation of the Operators

Since $\underline{F}_1$, $\underline{F}_2$, $\underline{P}$, $\underline{R}$, and $\underline{D}$ do not involve hidden variables, the following quantities can be directly estimated from the training data of skeletal sequences. For this reason, they are called *observable blocks*:

$$\begin{aligned}
\Sigma^a &= \mathbf{E}[\underline{F}_1\underline{P}^\top|A_1 = a] && \forall a \in [l]\\
\Lambda^a &= \mathbf{E}[\underline{R}\,\underline{D}^\top|A_1 = a] && \forall a \in [l]\\
D^{b|a} &= \mathbf{E}[[[A_2 = b]]\underline{F}_2\underline{P}^\top\underline{R}^\top|A_1 = a] && \forall a, b \in [l]\\
d_x^a &= \mathbf{E}[[[X = x]]\underline{D}^\top|A_1 = a] && \forall a \in [l], x \in [n]
\end{aligned}$$

The main result of this paper is that under certain conditions, matrices Σ^a and Λ^a are invertible and the operators $\left\langle C^{b|a}, C^{*|a}, c_a^1, c_x^a\right\rangle$ in theorem 4.1 can be expressed in terms of these observable blocks.

$$C^{b|a}(v) = D^{b|a}(v)(\Sigma^a)^{-1} \quad (1)$$

$$C^{*|a}(v) = D^{*|a}(v)(\Sigma^a)^{-1} \quad (2)$$

$$c_x^a = d_x^a(\Lambda^a)^{-1} \quad (3)$$

$$c_a^1 = \mathbf{E}[[[A_1 = a]]\underline{F}_1|B = 1] \quad (4)$$

To derive this result, we use the following definition to help specify the conditions on the expectations of the feature functions.

Definition. *For each $a \in [l]$, define matrices $I^a \in \mathbb{R}^{d_1 \times m}$, $J^a \in \mathbb{R}^{d_2 \times m}$, $K^a \in \mathbb{R}^{d_3 \times m}$, $W^a \in \mathbb{R}^{d_4 \times m}$ by*

$$\begin{aligned}
[I^a]_{k,h} &= \boldsymbol{E}[[\phi(F_1)]_k|A_1 = a, H_1 = h]\\
[J^a]_{k,h} &= \boldsymbol{E}[[\psi(P)]_k|A_1 = a, H_1 = h]\\
[K^a]_{k,h} &= \boldsymbol{E}[[\xi(R)]_k|A_1 = a, H_1 = h]\\
[W^a]_{k,h} &= \boldsymbol{E}[[\upsilon(D)]_k|A_1 = a, H_1 = h]
\end{aligned}$$

In addition, let $\Gamma^a \in \mathbb{R}^{m \times m}$ be a diagonal matrix with $[\Gamma^a]_{h,h} = P(H_1 = h|A_1 = a)$.

We now state the conditions for the correctness of Eq. (1-4). For each label $a \in [l]$, we require that

Condition 6.1 I^a, J^a, K^a, W^a have rank m.

Condition 6.2 $[\Gamma^a]_{h,h} > 0$ for all $h \in [m]$.

The conditions lead to the following proposition.

Proposition 5.1. *Assume Condition 6.1 and 6.2 hold. For all $a \in [l]$, define matrices*

$$\begin{aligned}
\Omega_1^a &= \boldsymbol{E}[\phi(F_1)\psi(P)^\top|A_1 = a] \in \mathbb{R}^{d_1 \times d_2}\\
\Omega_2^a &= \boldsymbol{E}[\xi(R)\upsilon(D)^\top|A_1 = a] \in \mathbb{R}^{d_3 \times d_4}
\end{aligned}$$

Let $u_1^a \dots u_m^a \in \mathbb{R}^{d_1}$ and $v_1^a \dots v_m^a \in \mathbb{R}^{d_2}$ be the top m left and right singular vectors of Ω^a. Similarly, let $l_1^a \dots l_m^a \in \mathbb{R}^{d_3}$ and $r_1^a \dots r_m^a \in \mathbb{R}^{d_4}$ be the top m left and right singular vectors of Ψ^a. Define projection matrices

$$\begin{aligned}
\Phi^a &= [u_1^a \dots u_m^a]^\top & \Psi^a &= [v_1^a \dots v_m^a]^\top\\
\Xi^a &= [l_1^a \dots l_m^a]^\top & \Upsilon^a &= [r_1^a \dots r_m^a]^\top
\end{aligned}$$

Then the following $m \times m$ matrices

$$\begin{aligned}
G^a &= \Phi^a I^a & \mathcal{G}^a &= \Psi^a J^a\\
H^a &= \Xi^a K^a & \mathcal{H}^a &= \Upsilon^a W^a
\end{aligned}$$

are invertible.

The proof resembles that of lemma 2 of Hsu et al. (2012). Finally, we state the main result that shows $\left\langle C^{b|a}, C^{*|a}, c_a^1, c_x^a\right\rangle$ in Eq. (1-4) using the projections from proposition 5.1 satisfy theorem 4.1. A sketch of the proof is deferred to the appendix.

Theorem 5.1. *Assume conditions 6.1 and 6.2 hold. Let $\langle \Phi^a, \Psi^a, \Xi^a, \Upsilon^a\rangle$ be the projection matrices from proposition 5.1. Then the operators in Eq. (1-4) satisfy theorem 4.1.*

In summary, these results show that with the proper selection of feature functions, we can construct projection matrices $\langle \Phi^a, \Psi^a, \Xi^a, \Upsilon^a\rangle$ to obtain operators $\left\langle C^{b|a}, C^{*|a}, c_a^1, c_x^a\right\rangle$ which satisfy the conditions of theorem 4.1.

6 The Spectral Estimation Algorithm

In this section, we give an algorithm to estimate the operators $\left\langle C^{b|a}, C^{*|a}, c_a^1, c_x^a\right\rangle$ from samples of skeletal sequences. Suppose the training set consists of M skeletal sequences $(a^{(j)}, x^{(j)})$ for $j \in [M]$. Then M samples of the random variables can be derived from this training set as follows

- At each $j \in [M]$, choose a position i_j uniformly at random from the positions in $(a^{(j)}, x^{(j)})$. Sample the random variables $(X, A_1, A_2, F_1, F_2, P, R, D, B)$ using the procedure defined in section 5.1.

This process yields M samples

$(x^{(j)}, a_1^{(j)}, a_2^{(j)}, f_1^{(j)}, f_2^{(j)}, p^{(j)}, r^{(j)}, d^{(j)}, b^{(j)})$ for $j \in [M]$

Assuming $(a^{(j)}, x^{(j)})$ are i.i.d. draws from the PMF $p(a_1 \dots a_N, x_1 \dots x_N)$ over skeletal sequences under an R-HMM, the tuples obtained through this process are i.i.d. draws from the joint PMF over $(X, A_1, A_2, F_1, F_2, P, R, D, B)$.

Input: samples of $(X, A_1, A_2, F_1, F_2, P, R, D, B)$; feature functions ϕ, ψ, ξ, and υ; number of hidden states m

Output: estimates $\left\langle \hat{C}^{b|a}, \hat{C}^{*|a}, \hat{c}^1_a, \hat{c}^a_x \right\rangle$ of the operators used in algorithm 2

[Singular Value Decomposition]

- For each label $a \in [l]$, compute empirical estimates of

$$\Omega^a_1 = \mathbf{E}[\phi(F_1)\psi(P)^\top | A_1 = a]$$
$$\Omega^a_2 = \mathbf{E}[\xi(R)\upsilon(D)^\top | A_1 = a]$$

and obtain their singular vectors via an SVD. Use the top m singular vectors to construct projections $\left\langle \hat{\Phi}^a, \hat{\Psi}^a, \hat{\Xi}^a, \hat{\Upsilon}^a \right\rangle$.

[Sample Projection]

- Project (d_1, d_2, d_3, d_4)-dimensional samples of

$$(\phi(F_1), \phi(F_2), \psi(P), \xi(R), \upsilon(D))$$

with matrices $\left\langle \hat{\Phi}^a, \hat{\Psi}^a, \hat{\Xi}^a, \hat{\Upsilon}^a \right\rangle$ to obtain m-dimensional samples of

$$(\underline{F}_1, \underline{F}_2, \underline{P}, \underline{R}, \underline{D})$$

[Method of Moments]

- For each $a, b \in [l]$ and $x \in [n]$, compute empirical estimates $\left\langle \hat{\Sigma}^a, \hat{\Lambda}^a, \hat{D}^{b|a}, \hat{d}^a_x \right\rangle$ of the observable blocks

$$\begin{aligned} \Sigma^a &= \mathbf{E}[\underline{F}_1 \underline{P}^\top | A_1 = a] \\ \Lambda^a &= \mathbf{E}[\underline{R}\ \underline{D}^\top | A_1 = a] \\ D^{b|a} &= \mathbf{E}[[[A_2 = b]]\underline{F}_2 \underline{P}^\top \underline{R}^\top | A_1 = a] \\ d^a_x &= \mathbf{E}[[[X = x]]\underline{D}^\top | A_1 = a] \end{aligned}$$

and also $\hat{c}^1_a = \mathbf{E}[[[A_1 = a]]\underline{F}_1 | B = 1]$. Finally, set

$$\begin{aligned} \hat{C}^{b|a}(v) &\leftarrow \hat{D}^{b|a}(v)(\hat{\Sigma}^a)^{-1} \\ \hat{C}^{*|a}(v) &\leftarrow \hat{D}^{*|a}(v)(\hat{\Sigma}^a)^{-1} \\ \hat{c}^a_x &\leftarrow \hat{d}^a_x(\hat{\Lambda}^a)^{-1} \end{aligned}$$

Figure 4: The spectral estimation algorithm

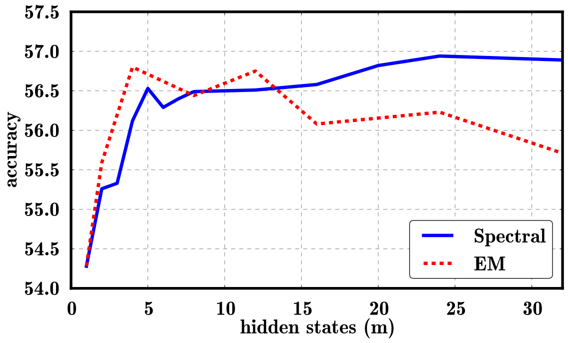


Figure 5: Accuracy of the spectral algorithm and EM on TIMIT development data for varying numbers of hidden states m. For EM, the highest scoring iteration is shown.

The algorithm in figure 4 shows how to derive estimates of the observable representations from these samples. It first computes the projection matrices $\langle \Phi^a, \Psi^a, \Xi^a, \Upsilon^a \rangle$ for each label $a \in [l]$ by computing empirical estimates of Ω^a_1 and Ω^a_2 in proposition 5.1, calculating their singular vectors via an SVD, and setting the projections in terms of these singular vectors. These projection matrices are then used to project (d_1, d_2, d_3, d_4)-dimensional feature vectors

$$\left(\phi(f^{(j)}_1), \phi(f^{(j)}_2), \psi(p^{(j)}), \xi(r^{(j)}), \upsilon(d^{(j)})\right)$$

down to m-dimensional vectors

$$\left(\underline{f}^{(j)}_1, \underline{f}^{(j)}_2, \underline{p}^{(j)}, \underline{r}^{(j)}, \underline{d}^{(j)}\right)$$

for all $j \in [M]$. It then computes correlation between these vectors in this lower dimensional space to estimate the observable blocks which are used to obtain the operators as in Eq. (1-4). These operators can be used in algorithm 2 to compute marginals.

As in other spectral methods, this estimation algorithm is consistent, i.e., the marginals $\hat{\mu}(a, i)$ computed with the estimated operators approach the true marginal values given more data. For details, see Cohen et al. (2012) and Foster et al. (2012).

7 Experiments

We apply the spectral algorithm for learning R-HMMs to the task of phoneme recognition. The goal is to predict the correct sequence of phonemes $a_1 \dots a_N$ for a given a set of speech frames $x_1 \dots x_N$. Phoneme recognition is often modeled with a fixed-structure HMM trained with EM, which makes it a natural application for spectral training.

We train and test on the TIMIT corpus of spoken language utterances (Garofolo and others, 1988). The label set consists of $l = 39$ English phonemes following a standard phoneme set (Lee and Hon, 1989). For training, we use the `sx` and `si` utterances of the TIMIT training section made up of

$$\begin{array}{ll}
\phi(F_1) & a_{i+1} \times x_i,\ a_{i+1},\ x_i,\ \texttt{np}(a_i \dots a_N) \\
\psi(P) & (a_{i-1}, x_{i-1}),\ a_{i-1},\ x_{i-1},\ \texttt{pp}(a_1 \dots a_i) \\
\xi(R) & x_i \\
\upsilon(D) & a_{i-1} \times x_{i-1},\ a_{i-1},\ x_{i-1},\ \texttt{pp}(a_1 \dots a_i), \\
 & \texttt{pos}(a_1 \dots a_N)
\end{array}$$

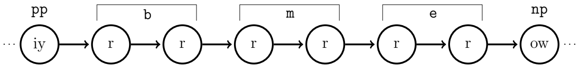


Figure 6: The feature templates for phoneme recognition. The simplest features look only at the current label and observation. Other features indicate the previous phoneme type used before a_i (pp), the next phoneme type used after a_i (np), and the relative position (beginning, middle, or end) of a_i within the current phoneme (pos). The figure gives a typical segment of the phoneme sequence $a_1 \dots a_N$

$M = 3696$ utterances. The parameter estimate is smoothed using the method of Cohen et al. (2013).

Each utterance consists of a speech signal aligned with phoneme labels. As preprocessing, we divide the signal into a sequence of N overlapping frames, 25ms in length with a 10ms step size. Each frame is converted to a feature representation using MFCC with its first and second derivatives for a total of 39 continuous features. To discretize the problem, we apply vector quantization using euclidean k-means to map each frame into $n = 10000$ observation classes. After preprocessing, we have 3696 skeletal sequence with $a_1 \dots a_N$ as the frame-aligned phoneme labels and $x_1 \dots x_N$ as the observation classes.

For testing, we use the core test portion of TIMIT, consisting of 192 utterances, and for development we use 200 additional utterances. Accuracy is measured by the percentage of frames labeled with the correct phoneme. During inference, we calculate marginals μ for each label at each position i and choose the one with the highest marginal probability, $a_i^* = \arg\max_{a \in [l]} \mu(a, i)$.

The spectral method requires defining feature functions ϕ, ψ, ξ, and υ. We use binary-valued feature vectors which we specify through features templates, for instance the template $a_i \times x_i$ corresponds to binary values for each possible label and output pair (ln binary dimensions).

Figure 6 gives the full set of templates. These feature functions are specially for the phoneme labeling task. We note that the HTK baseline explicitly models the position within the current phoneme as part of the HMM structure. The spectral method is able to encode similar information naturally through the feature functions.

Method	Accuracy
EM(4)	56.80
EM(24)	56.23
SPECTRAL(24), no np, pp, pos	55.45
SPECTRAL(24), no pos	56.56
SPECTRAL(24)	56.94

Figure 7: Feature ablation experiments on TIMIT development data for the best spectral model ($m = 24$) with comparisons to the best EM model ($m = 4$) and EM with $m = 24$.

Method	Accuracy
UNIGRAM	48.04
HMM	54.08
EM(4)	55.49
SPECTRAL(24)	55.82
HTK	55.70

Figure 8: Performance of baselines and spectral R-HMM on TIMIT test data. Number of hidden states m optimized on development data (see figure 5). The improvement of the spectral method over the EM baseline is significant at the $p \leq 0.05$ level (and very close to significant at $p \leq 0.01$, with a precise value of $p \leq 0.0104$).

We implement several baseline for phoneme recognition: UNIGRAM chooses the most likely label, $\arg\max_{a \in [l]} p(a|x_i)$, at each position; HMM is a standard HMM trained with maximum-likelihood estimation; EM(m) is an R-HMM with m hidden states estimated using EM; and SPECTRAL(m) is an R-HMM with m hidden states estimated with the spectral method described in this paper. We also compare to HTK, a fixed-structure HMM with three segments per phoneme estimated using EM with the HTK speech toolkit. See Young et al. (2006) for more details on this method.

An important consideration for both EM and the spectral method is the number of hidden states m in the R-HMM. More states allow for greater label refinement, with the downside of possible overfitting and, in the case of EM, more local optima. To determine the best number of hidden states, we optimize both methods on the development set for a range of m values between 1 to 32. For EM, we run 200 training iterations on each value of m and choose the iteration that scores best on the development set. As the spectral algorithm is non-iterative, we only need to evaluate the development set once per m value. Figure 5 shows the development accuracy of the two method as we adjust the value of m. EM accuracy peaks at 4 hidden states and then starts degrading, whereas the spectral method continues to improve until 24 hidden states.

Another important consideration for the spectral method is the feature functions. The analysis suggests that the best feature functions are highly informative of the underlying hidden states. To test this empirically we run spectral estimation with a reduced set of features by ablating the templates indicating adjacent phonemes and relative position. Figure 7 shows that removing these features does have a significant effect on development accuracy. Without either type of feature, development accuracy drops by 1.5%.

We can interpret the effect of the features in a more principled manner. Informative features yield greater singular values for the matrices Ω_1^a and Ω_2^a, and these singular values directly affect the sample complexity of the algorithm; see Cohen et al. (2012) for details. In sum, good feature functions lead to well-conditioned Ω_1^a and Ω_2^a, which in turn require fewer samples for convergence.

Figure 8 gives the final performance for the baselines and the spectral method on the TIMIT test set. For EM and the spectral method, we use best performing model from the development data, 4 hidden states for EM and 24 for the spectral method. The experiments show that R-HMM models score significantly better than a standard HMM and comparatively to the fixed-structure HMM. In training the R-HMM models, the spectral method performs competitively with EM while avoiding the problems of local optima.

8 Conclusion

This paper derives a spectral algorithm for the task of supervised sequence labeling using an R-HMM. Unlike EM, the spectral method is guaranteed to provide a consistent estimate of the parameters of the model. In addition, the algorithm is simple to implement, requiring only an SVD of the observed counts and other standard matrix operations. We show empirically that when equipped with informative feature functions, the spectral method performs competitively with EM on the task of phoneme recognition.

Appendix

Proof of proposition 4.1. At any time step $i \in [N]$ in the algorithm in figure 2, for all label $a \in [l]$ we have a column vector $\alpha_a^i \in \mathbb{R}^m$ and a row vector $\beta_a^i \in \mathbb{R}^m$. The value of these vectors at each index $h \in [m]$ can be verified as

$$[\alpha_a^i]_h = \sum_{\substack{a_1 \dots a_i, h_1 \dots h_i: \\ a_i = a, h_i = h}} p(a_1 \dots a_i, x_1 \dots x_{i-1}, h_1 \dots h_i)$$

$$[\beta_a^i]_h = \sum_{\substack{a_i \dots a_N, h_i \dots h_N: \\ a_i = a, h_i = h}} p(a_{i+1} \dots a_N, x_i \dots x_N, h_{i+1} \dots h_N | a_i, h_i)$$

Thus $\beta_a^i \alpha_a^i$ is a scalar equal to

$$\sum_{\substack{a_1 \dots a_N, h_1 \dots h_N: \\ a_i = a}} p(a_1 \dots a_N, x_1 \dots x_N, h_1 \dots h_N)$$

which is the value of the marginal $\mu(a, i)$. □

Proof of theorem 5.1. It can be verified that $c_a^1 = G^a \pi^a$. For the others, under the conditional independence illustrated in figure 3 we can decompose the observable blocks in terms of the R-HMM parameters and invertible matrices

$$\Sigma^a = G^a \Gamma^a (\mathcal{G}^a)^\top \qquad \Lambda^a = H^a \Gamma^a (\mathcal{H}^a)^\top$$
$$D^{b|a}(v) = G^b T^{b|a} \mathrm{diag}(v H^a) \Gamma^a (\mathcal{G}^a)^\top$$
$$D^{*|a}(v) = f^a \mathrm{diag}(v H^a) \Gamma^a (\mathcal{G}^a)^\top \qquad d_x^a = o_x^a \Gamma^a (\mathcal{H}^a)^\top$$

using techniques similar to those sketched in Cohen et al. (2012). By proposition 5.1, Σ^a and Λ^a are invertible, and these observable blocks yield the operators that satisfy theorem 4.1 when placed in Eq. (1-3). □

References

A. Anandkumar, D. P. Foster, D. Hsu, S.M. Kakade, and Y.K. Liu. 2012a. Two svds suffice: Spectral decompositions for probabilistic topic modeling and latent dirichlet allocation. *Arxiv preprint arXiv:1204.6703*.

A. Anandkumar, D. Hsu, and S.M. Kakade. 2012b. A method of moments for mixture models and hidden markov models. *Arxiv preprint arXiv:1203.0683*.

B. Balle, A. Quattoni, and X. Carreras. 2011. A spectral learning algorithm for finite state transducers. *Machine Learning and Knowledge Discovery in Databases*, pages 156–171.

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar. 2012. Spectral learning of latent-variable PCFGs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar. 2013. Experiments with spectral learning of latent-variable pcfgs. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

D. P. Foster, J. Rodu, and L.H. Ungar. 2012. Spectral dimensionality reduction for hmms. *Arxiv preprint arXiv:1203.6130*.

J. S. Garofolo et al. 1988. Getting started with the darpa timit cd-rom: An acoustic phonetic continuous speech database. *National Institute of Standards and Technology (NIST), Gaithersburgh, MD*, 107.

D. Hsu, S.M. Kakade, and T. Zhang. 2012. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*.

H. Jaeger. 2000. Observable operator models for discrete stochastic time series. *Neural Computation*, 12(6):1371–1398.

K. Lari and S. J. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 4(1):35–56.

K.F. Lee and H.W. Hon. 1989. Speaker-independent phone recognition using hidden markov models. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(11):1641–1648.

F. M. Luque, A. Quattoni, B. Balle, and X. Carreras. 2012. Spectral learning for non-deterministic dependency parsing. In *EACL*, pages 409–419.

T. Matsuzaki, Y. Miyao, and J. Tsujii. 2005. Probabilistic cfg with latent annotations. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 75–82. Association for Computational Linguistics.

A. Parikh, L. Song, and E.P. Xing. 2011. A spectral algorithm for latent tree graphical models. In *Proceedings of the 28th International Conference on Machine Learning*.

F. Pereira and Y. Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics*, pages 128–135. Association for Computational Linguistics.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440. Association for Computational Linguistics.

Slav Petrov, Adam Pauls, and Dan Klein. 2007. Learning structured models for phone recognition. In *Proc. of EMNLP-CoNLL*.

L. R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

S. Siddiqi, B. Boots, and G. J. Gordon. 2010. Reduced-rank hidden Markov models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS-2010)*.

L. Song, B. Boots, S. Siddiqi, G. Gordon, and A. Smola. 2010. Hilbert space embeddings of hidden markov models. In *Proceedings of the 27th International Conference on Machine Learning*. Citeseer.

S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, XA Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al. 2006. The htk book (for htk version 3.4).