Using Regression for Spectral Estimation of HMMs

Abstract. Hidden Markov Models (HMMs) are widely used to model discrete time series data, but the EM and Gibbs sampling methods used to estimate them are often slow or prone to get stuck in local minima. A more recent class of reduced-dimension spectral methods for estimating HMMs has attractive theoretical properties, but their finite sample size behavior has not been well characterized. We introduce a new spectral model for HMM estimation, a corresponding spectral bilinear regression model, and systematically compare them with a variety of competing simplified models, explaining when and why each method gives superior performance. Using regression to estimate HMMs has a number of advantages, allowing more powerful and flexible modeling.

1 Introduction

Hidden Markov Models (HMMs) [1] are widely used in modelling time series data from text, speech, video and genomic sequences. In applications where the dimension of the observations is much larger than the dimension of the hidden state space, spectral methods can be used project the high dimensional observations down to a much lower dimensional representation that captures the information of the hidden state in the HMM. We call this class of model "spectral HMMs" (*sHMMs*) and show in this paper that sHMMs can be estimated in a variety of ways.

Standard algorithms for HMMs estimate the unobservable transition matrix T and emission matrix O, but are prone to getting stuck in local optima (for instance the EM algorithm) or are computationally intensive (Gibbs sampling). In contrast, sHMM methods estimate a fully observable representation of T and O and are fast, do not have local minima, have nice theoretical error bound proofs, and are optimal in linear estimation sense.

[2] showed that a set of statistics using unigrams, bigrams and trigrams of observations are sufficient to estimate such models. We present a simpler estimation technique and show that it generalizes to a rich collection of regression-based methods for estimating HMMs. In regression, one can easily include more information such as a longer history, or more features about the observed data. These cannot as easily be added into a pure HMM model. Our methods are particularly useful for language modeling, where the emissions of the Hidden Markov Models are words drawn from a large vocabulary (tens or hundreds of thousands of words), and the hidden state is a much lower dimensional representation (30-100 dimensions).

HMMs of this size are widely used in modeling NLP. Many variants of and applications of HMMs have been proposed including (to present a random list

of recent work) multiple span-HMM to predict predicates in different domains [3], factorial-HMMs to resolve the pronoun anaphora [4], multi-chain HMMs to compute the meaning of terms in text [5], tree-modified HMMs to do machine translation [6], fertility-HMM to reduce word alignment errors [7] and continuous HMMs to summarize speech documents without text [8].

Our main HMM estimation method, which we call a *spectral HMM* is inspired by the observation in [2] that the 'Observable Operator' model [9] which estimates the probability of a sequence $x_1, x_2, \ldots, x_t$ as

$$Pr(x_1, x_2, \ldots, x_t) = 1^\top A(x_t)A(x_{t-1}) \cdots A(x_1)\pi \tag{1}$$

in terms of the still unobservable $A(x) = T\text{diag}(O^\top x)$ (where $x = e_i$ denotes word i in a vocabulary, and e_i denotes as usual the vector of all zeros and a one in the i^{th} position) and the unigram probabilities π, can be rewritten to be a fully observable, partially reduced model through clever projections and combinations of the moment statistics. [10] extend this to a fully reduced, fully observable model. This extension directly motivates simplified bilinear and regression estimation procedures.

We find that a wide range of spectral methods work well for estimating HMMs. HMMs have an intrinsically bi-linear model, but using a linear approximation works well in practice, especially when one still keeps the use of recursive prediction. Our regression methods are competitive with the "traditional" method of moments methods, and make it relatively easy to add in much richer sets of features than either EM or standard spectral HMM estimations.

The rest of the paper is organized as follows. In section 2 we formally describe the reduced dimension spectral HMM ($sHMM$) model and the bilinear and simplified regression models that it motivates. We also compare our $sHMM$ model against the partially reduced dimension model of [2]. Section 3 gives our experimental results, and discusses prediction accuracy of the different methods in different limits. Section 4 concludes.

2 Approximations to HMMs

Consider a discrete HMM consisting of a sequence of observations $(x_1, x_2, ..., x_t)$ at discrete times $1...t$. Each observation, x_i corresponds to one of n labels (e.g. words). There is a corresponding sequence of hidden states, $(h_1, h_2, ..., h_t)$, where h_i corresponds to one of m labels.

Assume that $m << n$, as is the case, for example, when the vocabulary size n of words is much bigger than the hidden state size. Let T of size $m \times m$ denote the transition matrix; $T_{ij} = Pr(h_t = i|h_{t-1} = j)$. Let O of size $n \times m$ denote the emission matrix; $O_{ij} = Pr(x_t = e_i|h_t = j)$.

We estimate an sHMM using a matrix U which projects each observation x_t onto a low dimensional representation y_t using $y_t = U^\top x_t$, where x_t is defined as before. We work primarily in the y space, which is dimension m instead of the n-dimensional observation space. Note that unlike h, which is a discrete space, y lies in a continuous space.

U is the mapping between the original high dimension observation space and the reduced dimensional representation space. This matrix received a full treatment in [2] and therefore is not the focus of this paper. It is worth noting, however, that U is not unique, and need only satisfy a handful of properties. We call U the *eigenword* matrix, as $y = U^\top x$ forms a low dimensional representation of each word x in the vocabulary. For completeness, we note that a version of U can be easily estimated by taking the largest left singular vectors of the bigram matrix P_{21}, where

$$[P_{21}]_{i,j} = P(x_t = e_i, x_{t+1} = e_j).$$

We use this version in the empirical results presented below. This works well in theory (see details below) and adequately in practice, but better Us can be found, either by estimating U from another much bigger data set, or by using more complex estimation methods [11].

In all of our methods, we will estimate a model to predict the probability of the next item in the sequence given what has been observed so far:

$$\Pr(x_{t+1}|x_t, x_{t-1}, \ldots, x_1) = \Pr(x_{t+1}, x_t, x_{t-1}, \ldots, x_1) / \Pr(x_t, x_{t-1}, \ldots, x_1).$$

We do this in the reduced dimension space of y_i.

2.1 sHMM Model and Estimation

Our core sHMM algorithm estimates $\Pr(x_t, x_{t-1}, \ldots, x_1)$ via the method of moments, writing it in terms of $c_\infty^\top$, c_1 and $\mathcal{C}(y_t)$, and in turn writing each of these three items in terms of moments of the Ys. From [2] and [10] we have

$$Pr(x_1, x_2, \ldots, x_t) = c_\infty^\top \mathcal{C}(y_t)\mathcal{C}(y_{t-1}) \cdots \mathcal{C}(y_1)c_1 \tag{2}$$

with

$$c_1 = \mu, \quad c_\infty^\top = \mu^\top \Sigma^{-1}, \quad \mathcal{C}(y) = \mathcal{K}(y)\Sigma^{-1}$$

and parameters

$$\begin{aligned}
\mu &= \mathbb{E}(y_1) = U^\top O\, \pi \\
\Sigma &= \mathbb{E}(y_2 y_1^\top) = U^\top O\, T\, \mathrm{diag}(\pi)\, O^\top U \\
\mathcal{K}(a) &= \mathbb{E}(y_3 y_1^\top y_2^\top)a = U^\top O\, T\, \mathrm{diag}(O^\top U a)\, T\, \mathrm{diag}(\pi)\, (O^\top U)
\end{aligned}$$

This yields the following estimate of Pr():

$$\widehat{\Pr}(x_t, x_{t-1}, \ldots, x_1) = \widehat{c}_\infty^\top \widehat{\mathcal{C}}(y_t)\widehat{\mathcal{C}}(y_{t-1}) \cdots \widehat{\mathcal{C}}(y_1)\widehat{c}_1 \tag{3}$$

where

$$\widehat{c}_1 = \widehat{\mu}, \quad \widehat{c}_\infty^\top = \widehat{\mu}^\top \widehat{\Sigma}^{-1}, \quad \widehat{C}_y = \widehat{\mathcal{C}}(y) = \widehat{\mathcal{K}}(y)\widehat{\Sigma}^{-1}$$

and $\widehat{\mu}$, $\widehat{\Sigma}$ and $\widehat{\mathcal{K}}()$ are the empirical estimates of the first, second and third moments of the Y's, namely

$$\widehat{\mu} = \frac{1}{N}\sum_{i=1}^{N} Y_{i,1}, \quad \widehat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} Y_{i,2}Y_{i,1}^{\top}, \quad \widehat{\mathcal{K}}(y) = \frac{1}{N}\sum_{i=1}^{N} Y_{i,3}Y_{i,1}^{\top}Y_{i,2}^{\top}\ y$$

Here $Y_{i,t}$ indexes the N different independent observations (over i) of our data at time $t \in \{1, 2, 3\}$.

Our HMM model is shown in Figure 1.

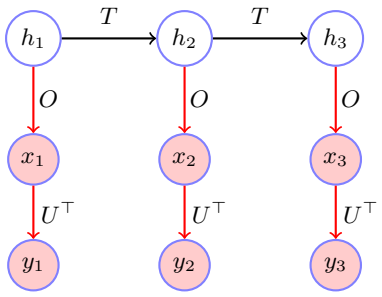

Fig. 1: The HMM with states h_1, h_2, and h_3 which emit observations x_1, x_2, and x_3. These observations are further projected onto the lower dimensional space with observations y_1, y_2, y_3 by U from which our core statistic $\mathcal{C}_y$ is computed based on $\mathcal{K} = E(y_3 y_1^{\top} y_2^{\top})$ which is a $(m \times m \times m)$ tensor.

[10] proved that the *sHMM* model is PAC learnable if the true model is an HMM and the projection matrix U has the property that range$(O) \subset$ range(U) and $|U_{ij}| \leq 1$. Given any small error ϵ and small confidence parameter δ, when the sample triples of observations are bigger than a polynomial of m, n, ϵ and δ, the probability estimated by reduced dimensional tensor $\mathcal{C}(y)$ in Eqn. 3 is smaller than ϵ with high confidence $1 - \delta$.

For any $t \in [2..\infty)$, the estimated value of y_t, denoted by $\hat{y}_t$, can be recursively estimated using the information at the previous time:

$$\hat{y}_t = \frac{\mathcal{C}(y_{t-1})\hat{y}_{t-1}}{\hat{c}_{\infty}^{\top}\mathcal{C}(y_{t-1})\hat{y}_{t-1}} \tag{4}$$

with $\hat{y}_1 = \widehat{\mu}$. Since the denominator in Eqn. 4 is a scalar constant for a particular time, we will separate the rescaling step from the recursive computation. Let $\lambda_t = \hat{c}_{\infty}^{\top}\mathcal{C}(y_{t-1})\hat{y}_{t-1}$. First we estimate $\tilde{y}_t = \mathcal{C}(y_{t-1})\hat{y}_{t-1}$ using the information from time $t-1$, then we set $\hat{y}_t = \tilde{y}_t/\lambda_t$.[1]

[1]Note the use of $\tilde{y}_t$ for the non-rescaled version of $\hat{y}_t$.

Note that once we have computed $\tilde{y}_t$, λ_t is computed deterministically; hence the key component in estimating $\hat{y}_t$ is the computation of

$$\tilde{y}_t = \mathcal{C}(y_{t-1})\,\hat{y}_{t-1}. \tag{5}$$

The observable HMM representation with $\hat{y}_1$, $\hat{c}_\infty$ and $\mathcal{C}(y)$ is sufficient to predict the probabilities of sequences of observations generated by an HMM. For joint probability of an observation sequence $(x_1, x_2, ..., x_t)$ one can use Eqn. 2. The conditional probability of the same sequence can be computed directly using $\hat{y}_t$. The conditional probability of observing i at time t is

$$Pr[x_t = e_i | x_1, x_2, ..., x_{t-1}] = [U\hat{y}_t]_i \tag{6}$$

This concludes the full presentation of the sHMM model. As mentioned in the introduction, this motivates simpler approximations which will now be discussed.

2.2 Bilinear Regression Model

Our *sHHM* model (5) that outputs the current $\tilde{y}_t$ is bilinear in y_{t-1} and $\hat{y}_{t-1}$. In other words, let $y_{j,t}$ be the j^{th} element of y_t, and $[\mathcal{C}]_{ijk} = c_{ijk}$. Then we can write

$$\tilde{y}_{j,t} = \sum_{i,k} c_{ijk} y_{i,t-1} \hat{y}_{k,t-1} \tag{7}$$

This leads naturally to our first simplified estimation technique–using linear regression by regressing $\tilde{y}_t$ on the outer product of y_{t-1} and $\hat{y}_{t-1}$ as shown in eqn. 7. We call this estimation method *Bilin-RRegr*. "Bilin"since it is Bilinear, "R" for recursive, since it is recursively estimated and predicted using the previous value of $\hat{y}_{k,t-1}$, and "Regr", since it is estimated using regression.

A note on training this model: in order to learn the parameters c_{ijk} we first estimate $\tilde{y}$'s and $\hat{y}$'s using linear regression on our empirically collected trigram data using the actual $y = U^\top x$'s as the "responses" to be predicted. We then estimate the parameters in 7 using a second regression in which these intial estimates of y form the responses. One could iterate this to fixed point, but the above process is in practice sufficient.

Also, although *sHMM* uses the method of moments to estimate the parameters while *Bilin-RRegr* uses linear regression, when used to make predictions the two methods are used identically.

2.3 Other Regression Models:

As mentioned in the introduction, many methods can be used to estimate the $sHMM$ model. We focus on two main simplifications: one can linearize the bilinear model, and one can drop the recursive estimation. Recursion shows up in two places: when doing estimation, one can regress either on y_t and $\hat{y}_t$ or on y_t and y_{t-1}, and when using the model to predict, one can do a "rolling"

prediction, in which y_{t+1} is predicted using the observed y_t and the predicted $\hat{y}_t$. These choices are made independently. For example the base spectral HMM method uses trigrams (no recursion) to estimate, but uses recursion to predict.

The bilinear equation in Eqn 7 can be linearized to give a simpler model to estimate $\tilde{y}_t$ using regression on y_{t-1} and $\hat{y}_{t-1}$. In the experimental results below, we call the resulting recursive linear model *Lin-RReg*:

$$\tilde{y}_t = \alpha\, y_{t-1} + \beta\, \hat{y}_{t-1} \tag{8}$$

We can also further simplify either the recursive bilinear model in Eqn 7 or the recursive linear model of Eqn 8 by noting that a simple linear estimate of $\hat{y}_{t-1}$ is $\hat{y}_{t-1} = Ay_{t-2}$. Since the matrix A is arbitrary, it can be folded into the model, giving a simple linear regression, *Lin-Regr*, model

$$\begin{aligned}\hat{y}_t &= \alpha\, y_{t-1} + \beta_1\, \hat{y}_{t-1} \\ &= \alpha\, y_{t-1} + \beta_2 A\, y_{t-2} \\ &= \alpha\, y_{t-1} + \beta\, y_{t-2}\end{aligned}$$

Note that here we estimate $\hat{y}$ directly instead of first estimating the unscaled $\tilde{y}$ and then rescaling to get our $\hat{y}$. Similarly, one can build a non-recursive bilinear model *Bilin-Regr*.

All of the above estimators work completely in the reduced dimension space Y. They are summarized in Table 1, along the single-lag version of *Lin-Regr*, *Lin-Regr-1*, and a couple of partially reduced dimension models which are described in the following section.

2.4 Partially Reduced Dimension Models

Instead of our fully dimension-reduced model $sHMM$, one can, following [2] estimate a tensor $\mathcal{B}(x)$, which is only projected into the reduced dimension space in two of its three components. $\mathcal{B}(x)$ thus takes an observation x, and produces an $m \times m$ matrix, unlike $\mathcal{C}(y)$ which takes a reduced dimension y and produces an $m \times m$ matrix.[2]

Given $\mathcal{B}(x)$, which is estimated from bigram an trigram occurrence counts, similarly to $\mathcal{C}(y)$, the probability of the next item in a sequence is predicted using

[2]Those familiar with the original paper will note that we have slightly re-interpreted B_x, which Hsu et al. call a matrix, and that what we call x here, they call δ_x.

Also the resulting $m \times m$ matrices are identical, specifically

$$\begin{aligned}C(y) &= \mathcal{K}(y)\Sigma^{-1} \\ &= (U^\top O)T\ \mathrm{diag}(O^\top U y)(U^\top O)^{-1} \\ &= (U^\top O)T\ \mathrm{diag}(O^\top x)(U^\top O)^{-1} \\ &= \mathcal{B}(x)\end{aligned}$$

the same recursive (rolling) method described above. The fundamental equation is similar in form:

$$Pr(x_1, x_2, \ldots, x_t) = b_\infty^\top \mathcal{B}(x_t)\mathcal{B}(x_{t-1}) \cdots \mathcal{B}(x_1) b_1 \tag{9}$$

See [2] for details. We call this method HKZ after its authors.

Our fully reduced dimension $sHMM$ offers several advantages over the original HKZ method. Working entirely in the reduced dimension space reduces number of parameters to be estimated from $m^2 n$ to m^3. This comes at a cost in that the theorems for $sHMM$ require U to contain full range of O instead of only just being full dimension.

The other big change in this paper over [2] is the use of linear regression to estimate the model. Computing a regression, unlike using the method of moments, requires computing the inverse of the covariance of the features (the outer product of y_t and $\hat{y}_t$). At the cost of doing the matrix inversion, we get more accurate estimates, particularly for the rarer emissions.

Using a regression model also gives a tremendous increase in flexibility; The regression can easily include more terms of history, giving more accurate estimates, particularly for more slowly changing or non-Markovian processes. This comes at a cost of estimating more parameters, but if the history is included in linear, instead of a bilinear model, this is relatively cheap.

3 Experiments

In this section, we present experimental results on synthetic and real data for a variety of algorithms for estimating spectral HMMs.

Table (1) lists the methods we used in our experiments. The number of parameters being estimated in each case (not including the U projection matrix) are listed on the right side. We expect models with more parameters to better on larger training sets and worse on smaller ones.

Method	Equation	Num. Params.
sHMM	$\tilde{y}_t = \mathcal{C}(y_{t-1})\hat{y}_{t-1}$	m^3
Bilin-RRegr	$\tilde{y}_t = \mathcal{C}(y_{t-1})\hat{y}_{t-1}$	m^3
Bilin-Regr	$\hat{y}_t = \Gamma(y_{t-1})y_{t-2}$	m^3
Lin-RRegr	$\tilde{y}_t = \alpha\, y_{t-1} + \beta\, \hat{y}_{t-1}$	$2m^2$
Lin-Regr	$\hat{y}_t = \alpha\, y_{t-1} + \beta\, y_{t-2}$	$2m^2$
Lin-Regr-1	$\hat{y}_t = \alpha\, y_{t-1}$	m^2
Lin-Regr-X	$\hat{x}_t = \alpha\, x_{t-1} + \beta\, x_{t-2}$	$2n^2$
HKZ	$\tilde{y}_t = \mathcal{B}(x_{t-1})\hat{y}_{t-1}$	$m^2 n$
EM(BaumWelch)	*MLE*	m^2

Table 1: **Methods compared in our experiments.** "Num Params" is the number of parameters, not including the $m \times n$ parameters for U. $\hat{y}$ denotes the estimate of y scaled by λ_t as in Eqn. 4, and $\tilde{y}$ denotes the unscaled estimate.

3.1 Synthetic data test

The synthetic data is generated by constructing HMMs as follows: A potential transition matrix T is generated with normally distributed elements. It is accepted if its second eigenvalue is in the range 0.9 ± 0.1. Similarly, emission matrices O are generated with normally distributed elements and accepted if the second eigenvalue is in 0.8 ± 0.1. This allows us to generate a selection of HMMs, but to control the length of memory of the HMM and the difficulty of estimating it.

We run the experiments as follows. For each of 10 runs, we generate a random HMM model (T, O) as described above and use it to generate a longer observation sequence as training data and 100 short (length 10) sequences as test data.

We then estimate the various models using the training data. First we build the unigram P_1, bigram P_{21} and trigram P_{3x1} of the observations and use them to estimate the projection matrix U and model parameters such as α, β, Γ and $\mathcal{C}$ and $\mathcal{B}$. U consists of the first m singular vectors corresponding to the m largest singular values of P_{21}. For the EM algorithm we use the R package [12]. Finally, we apply every method on each of test sequences and predict the last observation of each test sequence given the proceeding observations.

Each method in table (1) was tested varying various properties: training sequence lengths (figure 2a), the dimension of observations (figure 2b), and the state transition probabilities (figure 3). In the last table, the second eigenvalue (2nd EV) of the transition matrix is varied. When this is close to 1, the process mixes slowly. In other words, it behaves close to a deterministic process. When this 2nd eigenvalue is close to zero, the process mixes rapidly. Basically it behaves like a sequence of IID hidden states. Hence more naive estimators will do well.

We report the prediction accuracy averaged over the 10 runs. We count a prediction as correct if the true observation has the highest estimated probability.

3.2 NLP data test

We also evaluated our sHMM and rHMM on real NLP data sets. As with the synthetic data experiment, we predict the last word of a test sequence using the proceeding words.

We use the New York Times Newswire Service (*nyt-eng*) portion of English Gigaword Fourth Edition corpus (LDC2009T13) in Penn Treebank [13]. We used a vocabulary of ten thousand words, including tokens for punctuation, sentence boundaries, and a single word token for all out-of-vocabulary words. The corpus consisted of approximately 1.3 billion words from 1.8 million document. Our training and test data set are drawn randomly without replacement from the *nyt-eng* corpus. The training data consists of long sequences of observations with lengths varying from 1K to 1000K. The test data consists of 10,000 sequences of observations of length 100.

Following the language modeling literature, we use perplexity to measure how well our estimated language models fit the test data [14, 15]. Suppose a predicted distribution of a word x is p and the true distribution is q, the perplexity $PP(x)$

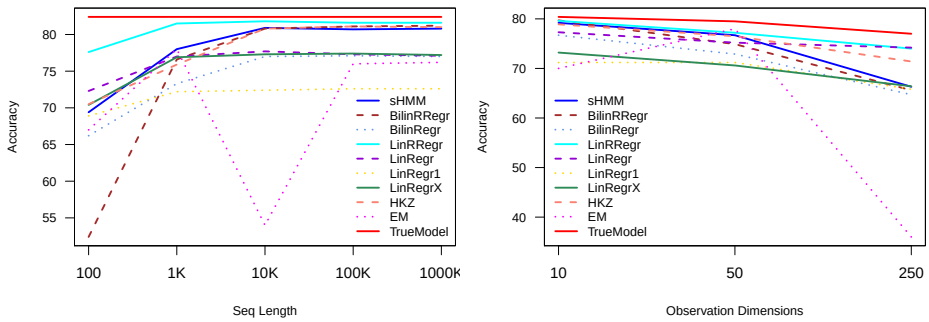

Fig. 2: **Prediction accuracy on synthetic data.** Number of correct predictions of the 10th observation given the proceeding 9 observations on 100 HMM sequences generated with dimension of states $m = 4$, second eigenvalue of transition matrix $T = 0.9$, second eigenvalue of emission matrix $O = 0.8$. Results are the average of 10 runs. The standard errors of 10 runs ranged from .06 to 3.1. **Left: Accuracy as a result of training sequence length. Observation dimension n = 10. Right: Accuracy as a result of observation dimension. Training length 10K**

is defined as $PP(x) = 2^{H(p,q)}$, where $H(p,q)$ is the cross-entropy of p and q. i.e. $H(p) = -\sum_x q(x) \log_2 \frac{qx}{p(x)}$. Because our true distribution q is a unit vector with only one element 1 at the x-th dimension, the actual computing of perplexity of word x is simplified as $PP(x) = \frac{1}{p_x}$. A lower perplexity $PP(x)$ indicates a better prediction on x.

We use the same test procedure and methods as for the synthetic data set. The perplexities of language models on *nyt-eng* corpus are shown in figures (4a) and (4b) with vocabularies of 1,000 and 10,000 words.

The results show several main trends, which are illustrated by two-way comparisons

- Fully reduced $sHMM$ vs. Partially reduced method HKZ
  - For small training sequences, $sHMM$ is better than HKZ, as one would expect, since $sHMM$ has far fewer parameters to estimate; $\mathcal{C}(y)$ is m/n times smaller than $\mathcal{B}(x)$. As theory predicts, in the limit of infinite training data, the two models are equivalent.
- Fully reduced $sHMM$ vs. Bilinear recursive regression *Bilin-RReg.*
  - On synthetic data generated from an HMM, for smaller training sets $sHMM$ performs better.
- Bilinear regression *Bilin-RReg.* vs. Linear regression *Lin-RReg.*
  - As expected, the simpler model linear model works better with short training sequences (We are not regularizing our regression, and so overfitting is possible). *Lin-RReg* unlike *Bilin-RReg*, is not a correct model of an HMM, and so will not perform as well in the limit of infinite training data.

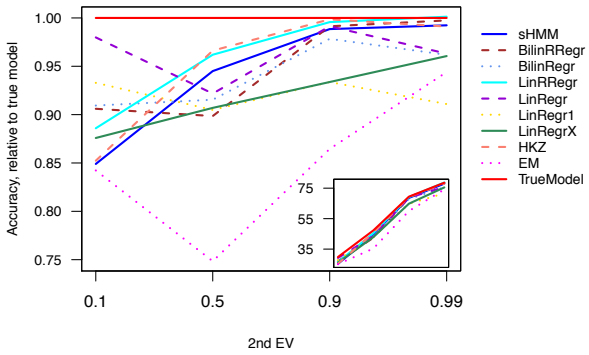

Fig. 3: **Prediction accuracy relative to that of True Model on synthetic data in terms of the second eigenvalue of the transition matrix. Inset: actual prediction accuracy.** Number of correct predictions of the 10th observation given the proceeding 9 observations on 100 HMM sequences. Results are the average of 10 runs. The standard errors of 10 runs ranged from 1.3 to 3.7. The model parameters are the number of states $m = 4$, the number of observations $n = 10$, the training length = 10K, and the second eigenvalue of the emission matrix $O = 0.8$.

- Recursive Methods (*Bilin-RReg*, *Lin-RReg*) vs. non-recursive ones (*Bilin-Reg*, *Lin-Reg*)
  - Recursive prediction always helps for linear models. For the more complicated bilinear model, recursion helps if there is sufficient training data. Keeping more lags in the model helps (e.g. *Lin-Regr* vs. *Lin-Regr1*).
- EM method *EM*
  - The *EM* method is prone to get stuck in local minima and often gives poor results. One could use a more sophisticated EM method, such as random restarts or annealing methods, but a major advantage of all of the spectral methods presented here is that they are fast (see, for example [16]) and guaranteed to converge to a good solution.

4 Discussion

HMM's are intrinsically nonlinear, but it is often advantageous to use a linear estimator, even when the data are generated by a model that is not linear. Linear estimators are fast, are guaranteed to converge to a single global optimum, and one can prove strong theorems about them. None of these properties are true of iterative algorithms such as EM for estimating nonlinear models.

We compared two major classes of techniques for estimating HMMs, method of moments ($sHMM$ and HKZ) and regression methods. All the methods presented here inherit the advantage of [2]'s method in that they use the projection matrix U containing the singular vectors of the bigram co-occurrence matrix to

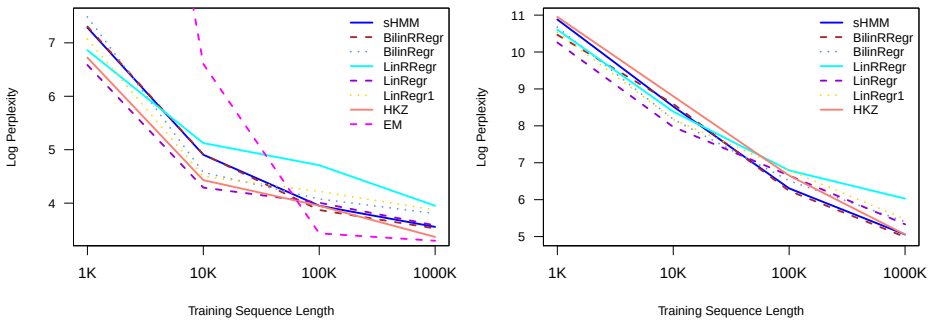

Fig. 4: **Log of perplexities of language models on *nyt-eng* corpus. Left: corpus vocabulary size 1000 words. Right: corpus vocabulary size 10,000 words** Note: EM has been excluded on the right in order to preserve a sensible y-axis scale- the performance was poor across all sequence lengths.

reduce the observations from a high dimension observation space X to a low dimension space Y. The Y space captures the information inherent in the hidden states and has same dimension as the hidden states. In this sense, the Y space can be seen as a linear transformation of the hidden state space. One could, of course, do regression in the original observation space, but that leads to models with vastly more parameters, making bilinear models prohibitively expensive. Models in the reduced dimension Y space have far fewer parameters and hence lower computational and sample complexity.

The method of moments models are simple to estimate, requiring only unigram bigram and trigram counts, and *not* requiring any recursive estimation (only recursive prediction). However, using regression models to estimate HMMs allows us far more flexibility than the method of moments models. Simple linear models can be used when training data are limited. Bilinear models that are identical to the $sHMM$ model can be used when more data are available. Longer histories can be used to estimate slowly changing HMMs (e.g. when the second eigenvalue of the transition matrix is close to 1) or when one does not believe that the HMM model is correct. Richer feature sets such as part of speech tags can also be added to the regression models when they are available.

Much work has been done generalizing the (partially reduced) HKZ method [17, 18] and extending it and our fully reduced $sHMM$ to probabilistic parsers [19, 20, 16]. We believe that extensions of the regression-based estimators presented in this paper should prove valuable in these settings as well.

References

1. Baum, L.E., Eagon, J.A.: An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. Bull. Amer. Math. Soc. (1967)

2. Hsu, D., Kakade, S.M., T.Zhang: A spectral algorithm for learning hidden markov models. COLT (2009)
3. Huang, F., Yates, A.: Open-domain semantic role labeling by modeling word spans. In: Association of Computational Linguistics (ACL). (2010)
4. Li, D., Miller, T., Schuler, W.: A pronoun anaphora resolution system based on factorial hidden markov models. In: Association of Computational Linguistics (ACL). (2011)
5. Turdakov, D., Lizorkin, D.: Hmm expanded to multiple interleaved chains as a model for word sense disambiguation. In: PACLIC. (2009)
6. Zabokrtsky, Z., Popel, M.: Hidden markov tree model in dependency-based machine translation. ACL-IJCNLP (2009)
7. Zhao, S., Gildea, D.: A fast fertility hidden markov model forword alignment using mcmc. EMNLP (2010)
8. Maskey, S., Hirschberg, J.: Summarizing speech without text using hidden markov models. In: Association of Computational Linguistics (ACL). (2006)
9. Jaeger, H.: Observable operator models for discrete stochastic time series. Neural Computation **12(6)** (2000)
10. Foster, D., Rodu, J., Ungar, L.: Spectral dimensionality reduction for HMMs. ArXiV (2012)
11. Dhillon, P., Foster, D., Ungar, L.: Multi-view learning of word embeddings via cca. In: NIPS. (2011)
12. Himmelmann, S.S.D.L.: HMM: Hidden Markov Models. (2010)
13. Robert Parker, e.a.: English gigaword fourth edition. Linguistic Data Consortium, Philadelphia (2009)
14. Brown, P.F., deSouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Computational Linguistics (1992)
15. Rosenfeld, R.: A maximum entropy approach to adaptive statistical language modeling. Computer Speech and Language, 10:187228 (1996)
16. Cohen, S.B., Stratos, K., Collins, M., Foster, D.P., Ungar, L.: Experiments with spectral learning of latent-variable pcfgs. NAACL (2013)
17. Siddiqi, S., Boots, B., Gordon, G.: Reduced-rank hidden markov models. Proc. 13th Intl. Conf. on Artificial Intelligence and Statistics (AISTATS) (2010)
18. Song, L., Boots, B., Siddiqi, S., Gordon, G., Smola, A.: Hilbert space embeddings of hidden markov models. Proc. 27th Intl. Conf. on Machine Learning (ICML) (2010)
19. Luque, F., Quattoni, A., Balle, B., Carreras, X.: Spectral learning for non-deterministic dependency parsing. In: EACL. (2012)
20. Cohen, S., Stratos, K., Collins, M., Foster, D., Ungar, L.: Spectral learning of latent-variable pcfgs. In: Association of Computational Linguistics (ACL). Volume 50. (2012)