Spectral Learning of Predictive State Representations with Insufficient Statistics

Alex Kulesza and **Nan Jiang** and **Satinder Singh**
Computer Science & Engineering
University of Michigan
Ann Arbor, MI, USA

Abstract

Predictive state representations (PSRs) are models of dynamical systems that represent state as a vector of predictions about future observable events (tests) conditioned on past observed events (histories). If a practitioner selects finite sets of tests and histories that are known to be sufficient to completely capture the system, an exact PSR can be learned in polynomial time using spectral methods. However, most real-world systems are complex, and in practice computational constraints limit us to small sets of tests and histories which are therefore never truly sufficient. How, then, should we choose these sets? Existing theory offers little guidance here, and yet we show that the choice is highly consequential—tests and histories selected at random or by a naïve rule significantly underperform the best sets. In this paper we approach the problem both theoretically and empirically. While any fixed system can be represented by an infinite number of equivalent but distinct PSRs, we show that in the computationally unconstrained setting, where existing theory guarantees accurate predictions, the PSRs learned by spectral methods always satisfy a particular spectral bound. Adapting this idea, we propose a simple algorithmic technique to search for sets of tests and histories that approximately satisfy the bound while respecting computational limits. Empirically, our method significantly reduces prediction errors compared to standard spectral learning approaches.

Introduction

Hidden Markov models (HMMs) and their variants, which postulate state variables that are never observed, are among the most well-known models of discrete-time dynamical systems. They are usually trained with iterative expectation-maximization (EM) algorithms that alternately "guess" the latent state value and then update the model parameters assuming the guessed value is correct. This process is guaranteed to converge, however, it can often be quite slow and get stuck in local optima (Wu 1983).

Predictive state representations (PSRs), first proposed by Littman, Sutton, and Singh (2002), take a different approach. Unlike HMMs, they represent state as a vector of predictions about future events that, crucially, are observable. This means that the signal needed for learning appears directly in the data, rendering iterative algorithms unnecessary. Instead, it becomes possible to exploit recent developments in spectral learning (Hsu, Kakade, and Zhang 2012; Balle, Quattoni, and Carreras 2011; Parikh, Song, and Xing 2011; Anandkumar et al. 2012). In particular, Boots, Siddiqi, and Gordon (2010) proposed a spectral learning algorithm for PSRs that is closed-form, fast, and, under the right assumptions, statistically consistent. In addition to being potentially easier to learn, PSRs of rank k are strictly more expressive than HMMs with k states (Jaeger 2000; Siddiqi, Boots, and Gordon 2010).

The learning algorithm of Boots, Siddiqi, and Gordon (2010) takes as input a matrix of statistics indexed by sets of *tests* and *histories*; these comprise sequences of observations that might occur (tests) or might have occurred (histories) at any given point in time, and are typically determined in advance by the practitioner. If the chosen tests and histories are sufficient in the right technical sense, then the learning process is consistent. If they are not, then as far as we are aware no formal guarantees are known.

Unfortunately, sufficiency requires that the number of tests and histories is at least the linear dimension of the underlying system that generates the data. (This condition does not by itself imply sufficiency, but it is necessary.) Linear dimension is a measure of system complexity; in an HMM, for example, it is at most the number of states. While small toy problems may have modest dimension, real-world systems are typically extremely complex. At the same time, since the cost of the learning algorithm scales cubically with the dimension of the input matrices, we are usually computationally constrained to small sets of tests and histories. The sufficiency assumption, therefore, almost always fails in practice.

In this paper we propose a novel, practical method for selecting sets of tests and histories for spectral learning of PSRs in the constrained setting where sufficiency is infeasible. As shown in Figure 1, this is not trivial; randomly chosen sets of tests and histories of a fixed size exhibit a wide range of prediction error rates. A simple baseline that chooses the shortest available tests and histories (as often done in practice) has higher error than average, and in the worst case, a poor choice can produce highly uninformative predictions. The sets chosen by our method, which have the same cardinality,

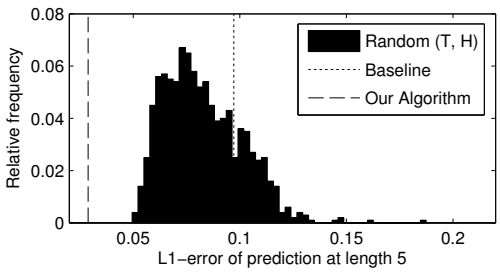

Figure 1: The distribution of L_1 variational error for 10,000 randomly chosen sets of four tests and four histories for a synthetic HMM with 100 states and 4 observations (see the Experiments section for details). The vertical lines show the error rates of our method and a baseline that uses all length-one tests and histories.

are an order of magnitude better.

Our approach is based on an analysis of a limiting case where the sets of tests and histories become infinitely large. Such sets are sufficient, so we know from existing theory that the PSRs learned from them (given enough data) are exact; however, we show that these PSRs have other unique properties as well. In particular, although there are an infinite number of equivalent but distinct PSRs representing any given system, the PSR learned by the spectral method from these infinite sets always satisfies a nontrivial spectral bound. We adapt this idea to the practical setting by searching for sets of finite size that approximately satisfy the bound.

We evaluate our approach on both synthetic and real-world problems. Using synthetic HMMs, we show that our method is robust to learning under a variety of transition topologies; compared to a baseline using the shortest tests and histories, our method achieves error rates up to an order of magnitude lower. We also demonstrate significantly improved prediction results on a real-world language modeling task using a large collection of text from Wikipedia.

Background

We begin by reviewing PSRs and the spectral learning algorithm proposed by Boots, Siddiqi, and Gordon (2010). At a high level, the goal is to model the output of a dynamic system producing observations from a finite set O at discrete time steps. (For simplicity we do not consider the controlled setting, in which an agent also chooses an action at each time step; however, the extension seems straightforward.)

We will assume the system has a reference condition from which we can sample observation sequences. Typically, this is either the reset condition (in applications with reset), or the long-term stationary distribution of the system, in which case samples can be drawn from a single long trajectory.

A *test* or *history* is an observation sequence in O^*. For any such sequence x, $p(x)$ denotes the probability that the system produces x in the first $|x|$ time steps after starting from the reference condition. It is not difficult to see that $p(\cdot)$ uniquely determines the system. Given a set of tests $\mathcal{T}$ and a set of histories $\mathcal{H}$, we define $P_{\mathcal{T},\mathcal{H}}$ to be the $|\mathcal{T}| \times |\mathcal{H}|$ matrix indexed by elements of $\mathcal{T}$ and $\mathcal{H}$ with $P_{t,h} = p(ht)$, where ht is the concatenation of h and t.

When $\mathcal{T} = \mathcal{H} = O^*$, $P_{\mathcal{T},\mathcal{H}}$ is a special bi-infinite matrix known as the *system-dynamics matrix*. The rank of the system-dynamics matrix is called the *linear dimension* of the system (Singh, James, and Rudary 2004). General sets $\mathcal{T}$ and $\mathcal{H}$ are called *core* if the rank of $P_{\mathcal{T},\mathcal{H}}$ is equal to the linear dimension; note that any $P_{\mathcal{T},\mathcal{H}}$ is a submatrix of the system dynamics matrix, and therefore can never have rank greater than the linear dimension. When $\mathcal{T}$ and $\mathcal{H}$ are core and have cardinality equal to the linear dimension, then they are called *minimal* core sets, since removing any element of $\mathcal{T}$ or $\mathcal{H}$ will reduce the rank of $P_{\mathcal{T},\mathcal{H}}$. Minimal core sets exist for any system with finite linear dimension.

Predictive State Representations

PSRs are usually described from the top down, showing how the desired state semantics can be realized by a particular parametric specification. However, because we are interested in PSRs that approximate (but do not exactly model) real systems, we will describe them instead from the bottom up, first defining the parameterization and prediction rules, and then discussing how various learning methods yield parameters that give accurate predictions under certain assumptions.

A PSR of rank k represents its state by a vector in $\mathbb{R}^k$ and is parameterized by a reference condition state vector $\boldsymbol{b}_* \in \mathbb{R}^k$, an update matrix $B_o \in \mathbb{R}^{k \times k}$ for each $o \in O$, and a normalization vector $\boldsymbol{b}_\infty \in \mathbb{R}^k$. Let $\boldsymbol{b}(h)$ denote the PSR state after observing history h from the reference condition (so $\boldsymbol{b}(\epsilon) = \boldsymbol{b}_*$, where ϵ is the empty history); the update rule after observing o is given by

$$\boldsymbol{b}(ho) = \frac{B_o \boldsymbol{b}(h)}{\boldsymbol{b}_\infty^\top B_o \boldsymbol{b}(h)} \,. \tag{1}$$

From state $\boldsymbol{b}(h)$, the probability of observing the sequence $o_1 o_2 \ldots o_n$ in the next n time steps is predicted by

$$\boldsymbol{b}_\infty^\top B_{o_n} \cdots B_{o_2} B_{o_1} \boldsymbol{b}(h) \,; \tag{2}$$

in particular, a PSR approximates the system function $p(\cdot)$ as

$$p(o_1 o_2 \ldots o_n) \approx \boldsymbol{b}_\infty^\top B_{o_n} \cdots B_{o_2} B_{o_1} \boldsymbol{b}_* \,. \tag{3}$$

We now turn to setting the parameters $\boldsymbol{b}_*$, B_o, and $\boldsymbol{b}_\infty$. Let $\mathcal{T}$ and $\mathcal{H}$ be minimal core sets, and define $P_{o\mathcal{T},\mathcal{H}}$ to be the $|\mathcal{T}| \times |\mathcal{H}|$ matrix with $[P_{o\mathcal{T},\mathcal{H}}]_{t,h} = p(hot)$. James and Singh (2004) showed that if the PSR parameters are chosen to be

$$\begin{aligned} \boldsymbol{b}_* &= P_{\mathcal{T},\{\epsilon\}} \\ B_o &= P_{o\mathcal{T},\mathcal{H}} P^+_{\mathcal{T},H} \qquad\qquad \forall o \in O \\ \boldsymbol{b}_\infty^\top &= P_{\{\epsilon\},\mathcal{H}} P^+_{\mathcal{T},\mathcal{H}} \,, \end{aligned} \tag{4}$$

where P^+ is the pseudoinverse of P, then Equation (3) holds with equality. That is, a system of linear dimension d, which has minimal core sets of cardinality d, can be modeled exactly by a rank d PSR. Moreover, in this case we can interpret the state vector $\boldsymbol{b}(h)$ as containing the probabilities of the tests in $\mathcal{T}$ given that h has been observed from the reference condition. This interpretation gives the PSR its name.

Equation (4) can be viewed as a consistent learning algorithm: if the P-statistics are estimated from data, then the

derived parameters converge to an exact PSR as the amount of data goes to infinity. In fact, consistency holds even when $\mathcal{T}$ and $\mathcal{H}$ are not minimal (as long as they are core). However, since the rank of the PSR grows with the cardinality of $\mathcal{T}$, computationally it is desirable to keep these sets small. Identifying small core sets of tests and histories is not trivial; if they are not known in advance, then the problem of finding them is called the *discovery problem* (Singh et al. 2003).

Boots, Siddiqi, and Gordon (2010) proposed an alternative learning algorithm that uses spectral techniques to control the rank of the learned PSR even when core sets $\mathcal{T}$ and $\mathcal{H}$ are large. In some ways this approach ameliorates the discovery problem, since finding large core sets is easier than finding small ones. The spectral method involves first obtaining the left singular vectors of the matrix $P_{\mathcal{T},\mathcal{H}}$ to form $U \in \mathbb{R}^{|\mathcal{T}| \times d}$ (recall that since $\mathcal{T}$ and $\mathcal{H}$ are core, $P_{\mathcal{T},\mathcal{H}}$ has rank d); then the parameters are set as follows:

$$\begin{aligned} \boldsymbol{b}_* &= U^\top P_{\mathcal{T},\{\epsilon\}} \\ B_o &= U^\top P_{o\mathcal{T},\mathcal{H}} \left(U^\top P_{\mathcal{T},H}\right)^+ \qquad \forall o \in O \\ \boldsymbol{b}_\infty^\top &= P_{\{\epsilon\},\mathcal{H}} \left(U^\top P_{\mathcal{T},H}\right)^+ . \end{aligned} \tag{5}$$

Note that the resulting PSR is of rank d regardless of the size of core sets $\mathcal{T}$ and $\mathcal{H}$. Moreover, it remains exact when the input statistics are exact, and consistent when the statistics are estimated from data.

Insufficient Statistics

While the spectral algorithm in Equation (5) makes it possible to use larger core sets of tests and histories without unnecessarily increasing the rank of the learned PSR, it does not address a potentially more serious issue: the rank necessary to learn most real-world systems exactly is impossibly large.

The runtime of spectral PSR learning is usually dominated by the singular value decomposition of $P_{\mathcal{T},\mathcal{H}}$, which requires $O(d^3)$ time if $|\mathcal{T}| = |\mathcal{H}| = d$. Though this is polynomial, in practice it typically means that we are limited to perhaps a few thousand tests and histories given modern computational constraints. (If the number of observations $|O|$ is very large, then the multiplications needed to compute all of the B_o matrices may require $\mathcal{T}$ and $\mathcal{H}$ to be even smaller.)

On the other hand, the linear dimension of any real-world system is likely to be effectively unbounded due to intrinsic complexity as well as external influences and sensor noise (which from the perspective of learning are indistinguishable from the underlying system). This makes it doubtful that test and history sets small enough to be computationally tractable can ever be core.

In this paper, therefore, we are interested in developing techniques for learning PSRs in the *insufficient* setting, where recovering an exact model is infeasible, but we still want to achieve good performance. To our knowledge, this setting is not addressed by any existing analysis. (A related *low-rank* setting is discussed by Kulesza, Nadakuditi, and Singh (2014).)

We formulate the problem as a variant of the PSR discovery problem for spectral learning, where rather than searching for small core sets of tests and histories, we are looking for sets that will perform well despite not being core. While we could in principle treat this as a standard model selection problem, the number of possible $\mathcal{T}$ and $\mathcal{H}$ is exponentially large, so huge amounts of data would be needed to choose sets based on empirical estimates of their performance without overfitting. Instead, we seek a measure for characterizing the likely performance of $\mathcal{T}$ and $\mathcal{H}$ that does not rely on validation data. We next describe a limiting-case analysis that motivates the measure we will eventually propose.

Limiting-Case Analysis

In order to get insight into the behavior of spectral PSR learning, we begin by considering the theoretical case where $\mathcal{T} = \mathcal{H} = O^*$; that is, where we have not only sufficient statistics but complete statistics. Moreover, we will assume that we have access to the exact system-dynamics matrix $P_{\mathcal{T},\mathcal{H}}$, so finite-sample effects do not come into play. These are highly unrealistic assumptions, but they represent what should be the best-case scenario for PSR learning. By understanding how the spectral method behaves in this ideal setting, where the resulting PSR is guaranteed to be exact, we can hopefully develop useful heuristics to improve performance in practice.

We will make use of the fact that $P_{o\mathcal{T},\mathcal{H}}$ is now actually a submatrix of $P_{\mathcal{T},\mathcal{H}}$, which is possible since both matrices are bi-infinite. In particular, for all $o \in O$ we define the bi-infinite operator R_o with rows and columns indexed by $\mathcal{T}$, where $[R_o]_{t,t'} = \mathbb{I}(t' = ot)$ ($\mathbb{I}$ is the indicator function). Then,

$$\begin{aligned} \left[R_o P_{\mathcal{T},\mathcal{H}}\right]_{t,h} &= \sum_{t'} [R_o]_{t,t'} P_{t',h} && (6) \\ &= P_{ot,h} = p(hot) = \left[P_{o\mathcal{T},\mathcal{H}}\right]_{t,h} , && (7) \end{aligned}$$

and thus $P_{o\mathcal{T},\mathcal{H}} = R_o P_{\mathcal{T},\mathcal{H}}$.

In order for the spectral algorithm to apply, $P_{\mathcal{T},\mathcal{H}}$ must have a singular value decomposition; while this is always true for finite matrices, in the infinite setting certain technical conditions are required. (For instance, if the system becomes fully deterministic then the singular values of the system-dynamics matrix can tend to infinity.) Since we are interested in the general behavior of the learning algorithm, we will not attempt a detailed characterization of such systems and instead simply assume that a singular value decomposition $P_{\mathcal{T},\mathcal{H}} = U \Sigma V^\top$ exists.

We can now express B_o from Equation (5) as

$$\begin{aligned} B_o &= U^\top R_o P_{\mathcal{T},\mathcal{H}} \left(U^\top P_{\mathcal{T},\mathcal{H}}\right)^+ && (8) \\ &= U^\top R_o P_{\mathcal{T},\mathcal{H}} V \Sigma^+ && (9) \\ &= U^\top R_o U . && (10) \end{aligned}$$

Let $\sigma_1(A)$ denote the first (largest) singular value of a matrix A. U is an orthogonal matrix because it contains the left singular vectors of $P_{\mathcal{T},\mathcal{H}}$, therefore $\sigma_1(U) = 1$; similarly, R_o is a binary matrix with at most a single 1 per row or column, so $\sigma_1(R_o) = 1$. Since for matrices A and B we have $\sigma_1(AB) \leq \sigma_1(A)\sigma_1(B)$ (Horn and Johnson 2012), we conclude that

$$\sigma_1(B_o) \leq 1 . \tag{11}$$

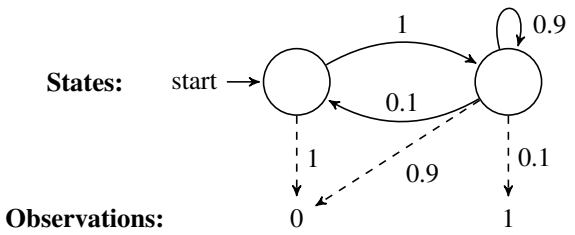

Figure 2: A simple HMM with two states and two observations. Solid edges indicate state transitions, and dotted edges show observation probabilities.

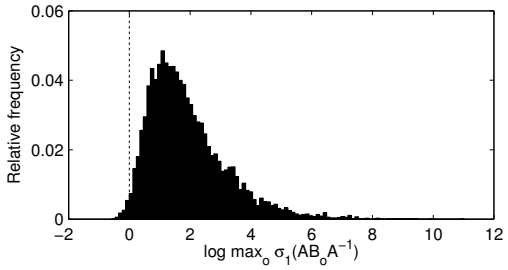

Figure 3: The distribution of $\log \max_o \sigma_1(AB_oA^{-1})$ for 10,000 random HMMs and transformations A.

Equivalent PSRs

The bound in Equation (11) may not seem surprising at first; after all, products of the PSR update matrices are used in Equation (3) to predict probabilities that must always be in $[0, 1]$, so we know they cannot blow up. And yet, for any system of finite linear dimension there are an infinite number of exact PSRs, and as we will see they do not all satisfy Equation (11)[1]. Perhaps more importantly, the PSRs we learn in practice generally do not satisfy the bound; later, we will use this idea to improve the empirical performance of the spectral learning algorithm. First, though, we discuss some of the interesting implications of Equation (11).

Consider the simple system shown as an HMM in Figure 2. Using the direct HMM-to-PSR construction described by Jaeger (2000), we can exactly model this system with the following rank 2 PSR:

$$\boldsymbol{b}_* = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad B_0 = \begin{bmatrix} 0 & 0.09 \\ 1 & 0.81 \end{bmatrix} \tag{12}$$

$$\boldsymbol{b}_\infty^\top = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 0 & 0.01 \\ 0 & 0.09 \end{bmatrix} \tag{13}$$

Since $\sigma_1(B_0) \approx 1.228$, this is an instance of a PSR that does not satisfy the bound in Equation (11). Yet if we apply the spectral learning algorithm in Equation (5), we obtain an equivalent PSR where $\max_o \sigma_1(B_o) \approx 0.909$.

More generally, for any rank k PSR with parameters $(\boldsymbol{b}_\infty, \{B_o\}, \boldsymbol{b}_*)$, an invertible $k \times k$ matrix A generates an equivalent PSR with parameters $(\boldsymbol{b}_\infty A^{-1}, \{AB_oA^{-1}\}, A\boldsymbol{b}_*)$—it is easy to see that the As cancel out in expressions like Equation (3). If, for instance, we let $A = \mathrm{diag}(a, 1)$ for some constant $a > 1$, then

$$B_o = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow AB_oA^{-1} = \begin{bmatrix} 0 & a \\ 1/a & 0 \end{bmatrix}, \tag{14}$$

and thus $\sigma_1(AB_oA^{-1}) = a$. Obviously, by choosing an appropriate a we can make this quantity as large as we like. Note that scaling B_o does not affect the claim.

We have shown that the direct construction of Jaeger (2000) does not necessarily satisfy Equation (11), and further that we can construct examples where Equation (11) is arbitrarily violated. For yet another perspective, we show in Figure 3 the distribution of $\max_o \sigma_1(AB_oA^{-1})$ when the B_o matrices are learned exactly using the spectral algorithm from randomly generated HMMs with 10 states and 10 observations, and A is a random matrix with independent normally-distributed entries. We see that the transformation by A nearly always brings σ_1 above 1 ($\log \sigma_1 > 0$), typically to about 5, but sometimes up to 100,000 or more. Thus the guarantee in Equation (11) does seems to say something "special" about the particular PSRs found by spectral learning, in the sense that transformed variants rarely satisfy the bound.

This "specialness" is what we hope to exploit. Though we will never be working in the infinite setting analyzed above, Equation (11) guarantees that, for a system of linear dimension d, not only does there exist an exact PSR of rank d, but there exists an exact rank d PSR where the singular values of the B_o matrices are also bounded by 1. Additionally, it tells us that the spectral algorithm will learn one of these bounded models in the limit of infinitely large sets $\mathcal{T}$ and $\mathcal{H}$. Both of these facts motivate using Equation (11) as an objective with which to choose among finite sets of tests and histories. This objective will always steer us toward at least one exact model, and moreover encourages the learning algorithm to behave as it would in the idealized setting.

Although this analysis does not provide any formal guarantees for our approach (as far as we are aware no guarantees of any kind are known in the insufficient setting), we will show later that it has significant advantages in practice.

Our Algorithm

We formulate the algorithmic problem as follows: given a maximum size k, which is determined by the practitioner based on computational constraints, find sets $\mathcal{T}$ and $\mathcal{H}$ with cardinality k to minimize the largest singular value of the update matrices $\{B_o\}$:

$$\underset{\substack{\mathcal{T},\mathcal{H} \\ |\mathcal{T}|=|\mathcal{H}|=k}}{\arg\min} \max_o \sigma_1(B_o)\,, \tag{15}$$

where B_o depends on $\mathcal{T}$ and $\mathcal{H}$ via the spectral procedure in Equation (5).

While one could imagine a variety of ways to turn this objective into a concrete learning algorithm, we propose a simple local search method that is simple to implement and works well in practice. Our method is described in Algorithm 1, where SPECTLEARN$(D, \mathcal{T}, \mathcal{H})$ denotes an imple-

[1] In fact, the bound in Equation (11) holds even when learning in the weighted finite automaton setting, where the values of the $p(\cdot)$ function are unconstrained (Balle et al. 2013).

Algorithm 1 Search for sets of k tests and histories that approximately minimize $\max_o \sigma_1(B_o)$.

Input: dataset D, initial $\mathcal{T}$ and $\mathcal{H}$ of size k, distributions $p_\mathcal{T}$/$p_\mathcal{H}$ over candidate tests/histories, number of rounds r

$\{B_o\} := \textsc{SpectLearn}(D, \mathcal{T}, \mathcal{H})$
$\sigma_{\text{opt}} := \max_{o \in O} \sigma_1(B_o)$
for $i = 1, \ldots, r$ **do**
  Sample $h \notin \mathcal{H} \sim p_\mathcal{H}$
  for $h' \in \mathcal{H}$ **do**
    $\{B_o\} := \textsc{SpectLearn}(D, \mathcal{T}, \mathcal{H} \setminus h' \cup \{h\})$
    $\sigma(h') := \max_{o \in O} \sigma_1(B_o)$
  $h^* = \arg\min_{h'} \sigma(h')$
  if $\sigma(h^*) < \sigma_{\text{opt}}$ **then**
    $\sigma_{\text{opt}} := \sigma(h^*)$
    $\mathcal{H} := \mathcal{H} \setminus h^* \cup \{h\}$
  [Repeat the same procedure for $\mathcal{T}$]

Output: $\mathcal{T}$,$\mathcal{H}$

mentation of Equation (5) using P-statistics estimated from dataset D with tests $\mathcal{T}$ and histories $\mathcal{H}$. Starting with a default $\mathcal{T}$ and $\mathcal{H}$ of the desired size, we iteratively sample a single new test (history) and consider using it to replace each element of $\mathcal{T}$ ($\mathcal{H}$). If the best replacement is an improvement in terms of $\max_o \sigma_1(B_o)$, then we keep it. After a fixed number of rounds, we stop and return the current $\mathcal{T}$ and $\mathcal{H}$.

Experiments

We demonstrate Algorithm 1 in both synthetic and real-world domains.

Synthetic Domains

We learn PSRs to model randomly generated HMMs with 100 states and 4 observations. The observation probabilities in a given state are chosen uniformly at random from $[0, 1]$ and then normalized. The initial state distribution is generated in the same way. Transition probabilities are chosen to reflect three different state topologies:

- **Random**: Each state has 5 possible successor states, selected uniformly at random.
- **Ring**: The states form a ring, and each state can only transition to itself or one of its 2 neighbors.
- **Grid**: The states form a 10×10 toric grid, and each state can only transition to itself or one of its 4 neighbors.

In each case, the non-zero entries of the transition matrix are chosen uniformly at random from $[0, 1]$ and normalized.

We measure the performance of a PSR by comparing its predicted distributions over observation sequences of length 1–10 to the true distributions given by the underlying HMM using L_1 variational distance. Since at longer lengths there are too many sequences to quickly compute the exact L_1 distance, we estimate it using 100 uniformly sampled sequences, which is sufficient to achieve low variance. Because an inexact PSR may predict negative probabilities, we clamp the predictions to $[0, \infty)$ and approximately normalize them by uniformly sampling sequences to estimate the normalization constant.

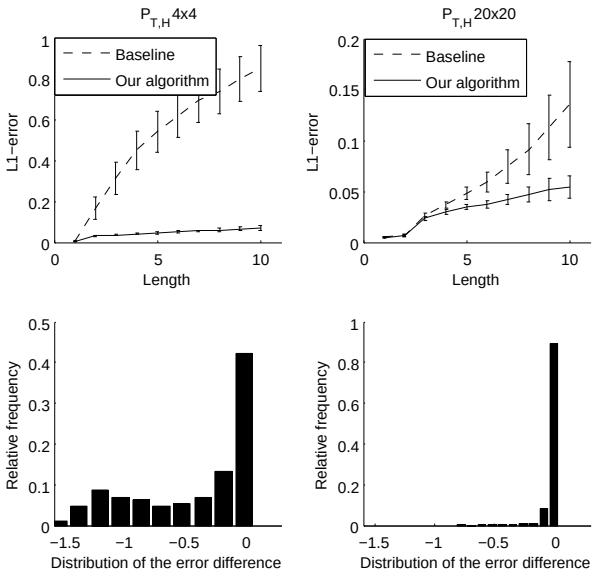

Figure 4: Results with random-topology synthetic HMMs. First row: L_1 error vs. prediction length. Second row: distribution of the difference in error between Algorithm 1 and the baseline.

Results Figure 4 shows the results averaged over 100 HMMs with random topologies, comparing our method to a baseline that uses the shortest available tests and histories. We include results for $k = 4$, where the baseline includes all tests and histories of length one, and $k = 20$, where the baseline includes all tests and histories of length one and two. Both algorithms receive exact P-statistics and do not need to estimate them from data. Our algorithm is initialized at the baseline $\mathcal{T}$ and $\mathcal{H}$, and we sample new tests and histories whose length is one observation longer than the longest sequences in the baseline sets; the sampling probability of a sequence x is proportional to $p^2(x)$. We run our algorithm for 10 rounds. Except as noted, all experiments use this setup.

Our algorithm significantly improves on the baseline at all prediction lengths, and dramatically so for $k = 4$. In the bottom half of the figure, we show the distribution of the error difference between our algorithm and the baseline across HMMs. Though in many cases the two are nearly equal, our algorithm almost never underperforms the baseline.

Figure 5 extends these results to the ring and grid topologies. We see qualitatively similar results, although our algorithm does not significantly improve on the baseline for ring topologies at $k = 20$. This may be because $k = 20$ is a relatively generous limit for this simpler toplogy, so less is gained by a careful choice of $\mathcal{T}$ and $\mathcal{H}$.

The dependence of our algorithm on r, the number of rounds, is illustrated in Figure 6. It is clear that more rounds lead to improved performance, suggesting that the objective derived from Equation (11) acts as a useful proxy; it is also clear that the earliest rounds are the most beneficial. Notice that the error bars are large for the baseline, but shrink for our algorithm as the number of rounds increases; this suggests that our search procedure not only reduces error but also

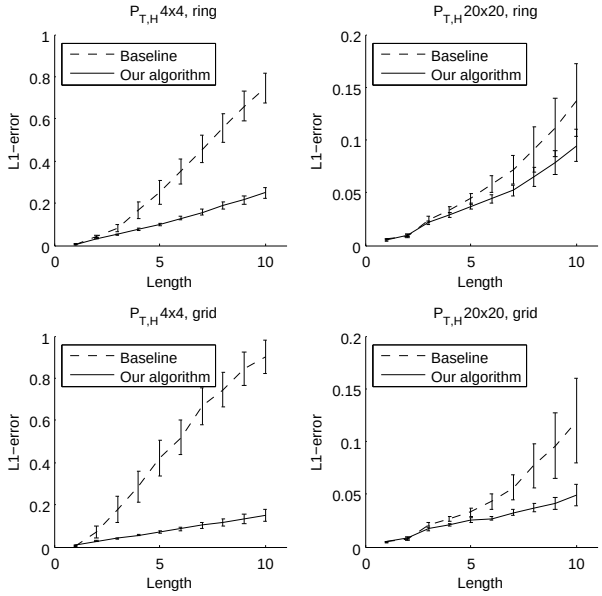

Figure 5: Results with ring and grid topology HMMs.

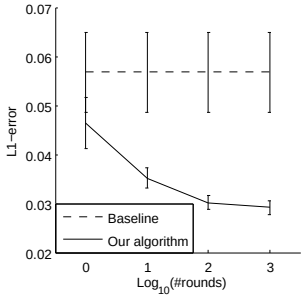

Figure 6: L_1 error at length 5 vs. number of rounds, averaged over 100 random-topology HMMs, $k = 20$.

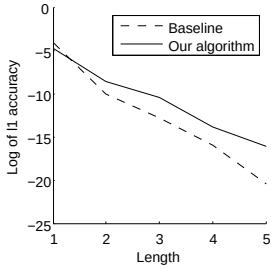

Figure 7: Results for modeling Wikipedia text: log of L_1 accuracy (higher is better) vs. prediction length.

reduces variance, which may be independently valuable.

In reality we do not get perfect P-statistics, so in Figure 8 we show how performance changes when the statistics are estimated from a dataset containing sampled observation sequences. We sample observation sequences of length 7 from a random-topology HMM and estimate $p(ht)$ by dividing the number of sequences with prefix ht by the total number of sequences. In this setting, the distributions used to sample new tests and histories in our algorithm are also estimated from the data. Our algorithm continues to outperform the baseline for all dataset sizes.

Wikipedia Text Prediction

Finally, we apply our algorithm to model a real-world text dataset of over 6.5 million sentences from Wikipedia articles (Sutskever, Martens, and Hinton 2011). The text contains 85 unique characters that consitute our observation set O, and each "time step" consists of a single character. We use the

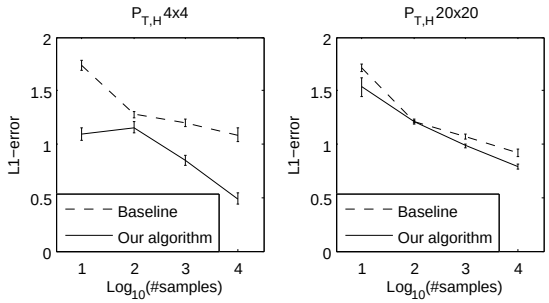

Figure 8: Results with P-statistics estimated from sampled data: L_1 error at length 5 vs. dataset size.

beginning of a sentence as our reference condition, so $p(x)$ is estimated from the number of times x appears as the prefix of a sentence.

We set $k = 85$, therefore our baseline consists of all tests and histories of length one. As before, we clamp negative predictions to zero, initialize our algorithm using the baseline sets, and sample new tests and histories of length two with probability $\propto p^2(\cdot)$. We run our algorithm for 100 rounds.

The majority of the data is used for training, but we reserve 100,000 sentences for evaluation. For each evaluation sentence, we predict the first 1–5 characters using the learned PSR. For lengths up to 3, we normalize our predictions exactly; for longer lengths, we use 500,000 uniformly sampled strings to estimate the normalization constant.

We cannot compute L_1 distance in this setting, since the true distribution over strings is unknown. Instead, we compute the mean probability assigned by the model to the observed strings; we refer to this metric as L_1 accuracy since it is a linear transformation of the L_1 distance to the δ distribution that assigns probability 1 to the observed string.

Figure 7 plots the L_1 accuracy obtained by the baseline and by our algorithm. Our algorithm produces meaningfully improved accuracy for all lengths greater than one.

Conclusion

We proposed a simple algorithm for choosing sets of tests and histories for spectral learning of PSRs, inspired by a limiting-case bound on the singular values of the learned parameters. By attempting to minimize the bound in practice, we regularize our model towards a known good solution. Experiments show that our approach significantly outperforms a standard shortest-tests/histories baseline on both synthetic and real-world domains. Future work includes developing more effective techniques to optimize Equation (15).

Acknowledgments

This work was supported by NSF grant 1319365. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the views of the sponsors.

References

[Anandkumar et al. 2012] Anandkumar, A.; Foster, D.; Hsu, D.; Kakade, S.; and Liu, Y.-K. 2012. A spectral algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems 25*, 926–934.

[Balle et al. 2013] Balle, B.; Carreras, X.; Luque, F. M.; and Quattoni, A. 2013. Spectral learning of weighted automata: a forward-backward perspective. *Machine Learning* 1–31.

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

[Boots, Siddiqi, and Gordon 2010] Boots, B.; Siddiqi, S. M.; and Gordon, G. J. 2010. Closing the learning-planning loop with predictive state representations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, 1369–1370.

[Horn and Johnson 2012] Horn, R. A., and Johnson, C. R. 2012. *Matrix analysis*. Cambridge university press.

[Hsu, Kakade, and Zhang 2012] Hsu, D.; Kakade, S. M.; and Zhang, T. 2012. A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences* 78(5):1460–1480.

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

[James and Singh 2004] James, M. R., and Singh, S. 2004. Learning and discovery of predictive state representations in dynamical systems with reset. In *Proceedings of the 21st International Conference on Machine Learning*, 53. ACM.

[Kulesza, Nadakuditi, and Singh 2014] Kulesza, A.; Nadakuditi, R. R.; and Singh, S. 2014. Low-rank spectral learning. In *Proceedings of the 17th Conference on Artificial Intelligence and Statistics*.

[Littman, Sutton, and Singh 2002] Littman, M. L.; Sutton, R. S.; and Singh, S. 2002. Predictive representations of state. In *Advances in Neural Information Processing Systems 14*, 1555–1561.

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

[Siddiqi, Boots, and Gordon 2010] Siddiqi, S. M.; Boots, B.; and Gordon, G. J. 2010. Reduced-rank hidden Markov models. In *International Conference on Artificial Intelligence and Statistics*, 741–748.

[Singh et al. 2003] Singh, S.; Littman, M. L.; Jong, N. K.; Pardoe, D.; and Stone, P. 2003. Learning predictive state representations. In *Proceedings of the 20th International Conference on Machine Learning*, 712–719.

[Singh, James, and Rudary 2004] Singh, S.; James, M. R.; and Rudary, M. R. 2004. Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 512–519. AUAI Press.

[Sutskever, Martens, and Hinton 2011] Sutskever, I.; Martens, J.; and Hinton, G. E. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning*, 1017–1024.

[Wu 1983] Wu, C. 1983. On the convergence properties of the EM algorithm. *The Annals of Statistics* 11(1):95–103.