Low-Rank Spectral Learning with Weighted Loss Functions

Alex Kulesza
University of Michigan

Nan Jiang
University of Michigan

Satinder Singh
University of Michigan

Abstract

Kulesza et al. [2014] recently observed that low-rank spectral learning algorithms, which discard the smallest singular values of a moment matrix during training, can behave in unexpected ways, producing large errors even when the discarded singular values are arbitrarily small. In this paper we prove that when learning predictive state representations those problematic cases disappear if we introduce a particular weighted loss function and learn using sufficiently large sets of statistics; our main result is a bound on the loss of the learned low-rank model in terms of the singular values that are discarded. Practically speaking, this suggests that regardless of the model rank we should use the largest possible sets of statistics, and we show empirically that this is true on both synthetic and real-world domains.

1 INTRODUCTION

Predictive state representations (PSRs) are compact models of dynamical systems that represent state as a vector of predictions about future observable events. More general than hidden Markov models (HMMs), PSRs are appealing because they can be learned directly from data without inferring hidden variables or applying iterative methods like expectation-maximization (EM). In particular, Boots et al. [2010] proposed a spectral learning algorithm for PSRs, based closely on an HMM learning algorithm by Hsu et al. [2012] and related to a variety of spectral learning methods that have been proposed in other settings [Balle et al., 2013, Anandkumar et al., 2012, Cohen et al., 2014, Parikh et al., 2011], that is closed-form, fast, and, under appropriate conditions, consistent.

Despite these advantages, Kulesza et al. [2014] observed recently that the *full-rank* assumption required for consistency in such spectral algorithms can be quite unrealistic, and when it is violated the results can be unpredictable. For the full-rank assumption to be met, the rank parameter used to control PSR complexity must exactly match the underlying system to be learned; when this is true, the singular value decomposition used during learning does not discard any information. When the assumption holds approximately, that is, when the PSR rank is less than the true rank but the singular values discarded are small—Kulesza et al. [2014] call this *low-rank* spectral learning—we might expect intuitively that learning should be approximately correct. It turns out that this is not true. In fact, Kulesza et al. [2014] found that in some cases discarding an arbitrarily small singular value can lead to maximally large errors.

Unfortunately, statistical considerations generally make it difficult to estimate the correct rank. More importantly, even if we knew it, the rank of any realistic system would likely be so large that we could not collect enough data or afford a powerful enough computer to learn a full-rank PSR. While Kulesza et al. [2014] identified certain assumptions under which low-rank spectral learning provably succeeds, those assumptions are themselves strong and difficult-to-verify constraints on the systems that can be learned. Currently, then, we have some worrying examples but no general-purpose guarantees on the performance of low-rank spectral learning, even though in practice this is the setting in which we almost always find ourselves.

In this paper we aim to address this situation by proving the first error bound for low-rank spectral learning that does not depend on assumptions about the underlying system. We rely instead on two main methodological considerations. First, we assume that the practitioner provides a weighting function over observation sequences that is used to define a loss function; we argue that this is conceptually fundamental, since an approximate learning algorithm must be able to quantify tradeoffs between different types of prediction errors. Second, we assume that the sets of *test* and *history* se-

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

quences used by the learning algorithm are sufficiently large; in general, we assume they are infinite.

This second assumption is what allows us to avoid the problematic examples of Kulesza et al. [2014], but we cannot work with infinite sets in practice. Instead, the method we analyze can be viewed as a limiting case of increasingly large but finite sets; thus, for any target PSR rank, our theory supports using the largest manageable sets of tests and histories. We evaluate this advice empirically on a variety of synthetic domains as well as a real-world text prediction problem, finding that the error of a rank-k PSR generally drops monotonically as the sets used to learn it grow.

In the next section we provide some necessary background on spectral learning for PSRs. We then discuss low-rank learning and the need for a weighting function in Section 3 before proceeding to our main result in Section 4 and empirical evaluations in Section 5.

2 BACKGROUND

We begin by reviewing PSRs and the spectral learning algorithm proposed by Boots et al. [2010]. At a high level, the goal is to model the output of a dynamical system producing observations from a finite set $\mathcal{O}$ 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, it is straightforward to extend our analysis.)

We will assume the system has a reference condition from which we can sample observation sequences. Typically, this will be 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 $\mathcal{O}^*$. For any such sequence x, $\Pr(x)$ denotes the probability that the system produces x in the first $|x|$ time steps after starting from the reference condition. Note that the function $\Pr(\cdot)$ completely specifies the system. Given a set of tests $\mathcal{T}$ and a set of histories $\mathcal{H}$, $P_{\mathcal{T},\mathcal{H}}$ is the $|\mathcal{T}| \times |\mathcal{H}|$ matrix indexed by elements $\mathcal{T}$ and $\mathcal{H}$ with $[P_{\mathcal{T},\mathcal{H}}]_{t,h} = \Pr(ht)$, where ht is the concatenation of h and t.

When $\mathcal{T} = \mathcal{H} = \mathcal{O}^*$, $P_{\mathcal{T},\mathcal{H}}$ is a special bi-infinite matrix known as the *system-dynamics matrix*, which we will denote by M. The rank of the system-dynamics matrix is called the *linear dimension* of the system [Singh et al., 2004], and sets of tests $\mathcal{T}$ and histories $\mathcal{H}$ 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 M, and therefore can never have rank greater than the linear dimension.)

2.1 Predictive State Representations

A PSR of rank k represents state using vectors in $\mathbb{R}^k$; it is parameterized by a triple $\mathcal{B} = (\boldsymbol{b}_*, \{B_o\}, \boldsymbol{b}_\infty)$, where $\boldsymbol{b}_* \in \mathbb{R}^k$ is a reference condition state vector, $B_o \in \mathbb{R}^{k\times k}$ is an update matrix for each $o \in \mathcal{O}$, and $\boldsymbol{b}_\infty \in \mathbb{R}^k$ is a normalization vector. 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 string); 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 $o_1 o_2 \dots o_n$ in the next n time steps is predicted by

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

and, in particular, the PSR approximates the system function $\Pr(\cdot)$ as

$$\Pr_{\mathcal{B}}(o_1 o_2 \cdots o_n) = \boldsymbol{b}_\infty^\top B_{o_n} \cdots B_{o_2} B_{o_1} \boldsymbol{b}_* \ . \tag{3}$$

The goal of learning is to choose parameters $\mathcal{B}$ so that $\Pr_B \approx \Pr$.

Suppose that $\mathcal{T}$ and $\mathcal{H}$ are core sets of tests and histories, so the rank of $P_{\mathcal{T},\mathcal{H}}$ is equal to d, the linear dimension of the system. Let $o\mathcal{T}$ denote the set $\{ot \mid t \in \mathcal{T}\}$, and let $U \in \mathbb{R}^{|\mathcal{T}|\times d}$ be a matrix containing the left singular vectors of the matrix $P_{\mathcal{T},\mathcal{H}}$. Boots et al. [2010] showed that if the PSR parameters are chosen to be

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

where A^+ is the pseudoinverse of A, then $\Pr_{\mathcal{B}} = \Pr$. That is, a system of linear dimension d can be modeled exactly by a rank d PSR, and one such PSR is recovered by the so-called spectral learning algorithm in Equation (4). Note that this algorithm is statistically consistent: 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.

3 LOW-RANK SPECTRAL LEARNING

While the spectral algorithm of Boots et al. [2010], along with the closely related work of Hsu et al. [2012], has many nice properties, it requires knowing and using the linear dimension d to compute U. In principle we can obtain d from the rank of $P_{\mathcal{T},\mathcal{H}}$, but in practice the statistics are only estimates, and as pointed out by Kulesza et al. [2014] accurately estimating rank in this setting is quite difficult [Benaych-Georges and Nadakuditi, 2012]. Moreover, even if known, the rank is likely to be prohibitively large from a computational and statistical standpoint for any interesting real-world system. Here, our theoretical interest is in understanding the consequences of violating the full-rank assumption, so we follow Kulesza et al. [2014] and assume for now that we have access to exact P-statistics but that d is unknown or very large. (In Section 5, we will study the empirical behavior of learning with finite data sets.)

For settings where we cannot use the true d, Kulesza et al. [2014] proposed the idea of *low-rank* spectral learning, where $U \in \mathbb{R}^{|\mathcal{T}| \times k}$ contains only the k principal left singular vectors of $P_{\mathcal{T},\mathcal{H}}$ for some hyperparameter $k < d$; the PSR parameters can then be computed using Equation (4) as before. While this approach is intuitive and was in fact proposed informally by Boots et al. [2010], Kulesza et al. [2014] showed that it leads to surprising problems. In particular, even omitting from U a single singular vector with an arbitrarily small (but nonzero) corresponding singular value can produce an uninformative model with maximum error.

Our aim is to show that these problems can be ameliorated by (a) introducing a convergent weighting function on observation sequences and (b) choosing sufficiently large sets $\mathcal{T}$ and $\mathcal{H}$. We will prove that in the limit, when $\mathcal{T} = \mathcal{H} = \mathcal{O}^*$ (but k remains constant), low-rank spectral learning behaves in a predictable way, with weighted loss bounded by the omitted singular values. We first discuss the need for a weighting function before proceeding to our main theorem in Section 4.

3.1 Weighted Loss

Existing theoretical guarantees for spectral learning generally apply uniformly across all sequences [Hsu et al., 2012]; that is, they show that $\Pr_{\mathcal{B}}(x)$ approaches $\Pr(x)$ regardless of x. In the low-rank setting, where we cannot hope to recover the underlying system exactly, this uniform performance is not generally possible. We argue that a low-rank model must trade off (for instance) short sequence prediction against longer sequence prediction, and cannot be optimal for both.

To see why, consider the system that (from the reset condition) yields the observation "a" for the first ten time steps and the observation "b" at all remaining time steps. That is, the observation sequence from reset is "aaaaaaaaaabbbbbbb…". The linear dimension of this process is 11, so we could learn it exactly with a rank 11 PSR, but suppose that due to computational constraints we wish to learn a model $\mathcal{B}$ of rank one. We will examine, for various choices of $\mathcal{B}$, the types of errors that result when we use $\Pr_{\mathcal{B}}$ to predict Pr on observation sequences of different lengths.

First, let $\boldsymbol{b}_* = \boldsymbol{b}_\infty = 1$ and set $B_a = 1, B_b = 0$. It is clear from Equation (3) that $\Pr_{\mathcal{B}}$ assigns a probability of one to any sequence consisting solely of "a"s, and therefore matches Pr for the first 10 time steps. Thus, a simple rank-one model achieves zero error for predictions up to length 10. However, for predictions of length 11 and beyond, this model is maximally inaccurate—it assigns a probability of one to a sequence ("aaa…") that is never observed, and predicts zero probability for the sequence that is, in fact, always observed.

On the other hand, consider the rank-one PSR given by $\boldsymbol{b}_* = \boldsymbol{b}_\infty = 1$ and $B_a = B_b = 0.5$. Now $\Pr_{\mathcal{B}}$ predicts that all sequences of observations are equally likely. It is a poor predictor of Pr in general; the real system is completely deterministic, while this PSR is the maximum-entropy rank-one model. However, it is still better than our first model for lengths greater than 10, since it at least assigns *some* nonzero probability to the observed sequence.

So we have described two models. The first achieves zero error for predictions of length at most 10, but maximal error (under any reasonable metric) for longer predictions. The second achieves error somewhere between zero and the maximum at all lengths. Which of these two is preferable? We argue that the answer fundamentally depends on how the practitioner values the accuracy of different kinds of predictions. There is no globally dominating model—it is not possible to achieve uniformly zero error with a rank one model—therefore we need some additional information to tell us which choice is better.

In this paper, we assume that the information comes in the form of a weighting function $w : \mathcal{O}^* \to \mathbb{R}$, which we use to define a loss function for measuring the performance of a PSR:

$$\mathcal{L}(\mathcal{B}) = \sum_{x \in \mathcal{O}^*} w^2(x) \left[\Pr(x) - \Pr_{\mathcal{B}}(x)\right]^2 \ . \tag{5}$$

We require that w satisfies the technical condition

$$\sum_{n=0}^{\infty} (n+1) \max_{|x|=n} w^2(x) < \infty \ . \tag{6}$$

In general, a bi-infinite system-dynamics matrix may not have a valid singular value decomposition, but this condition ensures that the weighted system-dynamics matrix we define in Section 4 does; it also ensures that the loss is finite.

As suggested by the form of Equation (6), it can be convenient to choose a weighting function that depends only on the length of x. For example, we could choose

$$w(x) = \frac{1}{2^{|x|}} \ ; \tag{7}$$

then the infinite sum in Equation (6) is equal to 16/9. Alternatively, we could choose $w(x) = \mathbb{I}(|x| \leq n)$ for any finite length n, or $w(x) = 1/(|x|+1)^p$ for $p > 1$.

Our goal will be to find a PSR $\mathcal{B}$ of rank k to minimize $\mathcal{L}(\mathcal{B})$. Since the weighting function defines the loss, it is a fundamental component of this learning problem; as we will see later, the choice of weight function can affect the difficulty of learning a given system, as well as the behavior of the spectral algorithm.

4 ANALYSIS

In this section we will establish an upper bound (Theorem 1) on the loss of a modified low-rank spectral learning algorithm that uses the full system-dynamics matrix M. Of course, we are never truly in this situation since M is infinite (and our computers are not), but later we will discuss how the result informs the use of spectral methods in realistic settings.

We begin by defining the weighted system dynamics matrix

$$\hat{M}_{t,h} = w(ht)M_{t,h} \ . \tag{8}$$

Under this definition, Equation (6) ensures that the squared sum of the entries of $\hat{M}$ is bounded:

$$\begin{aligned}
\sum_{t,h\in\mathcal{O}^*} |\hat{M}_{th}|^2 &= \sum_{t,h\in\mathcal{O}^*} w^2(ht)M_{th}^2 && (9)\\
&= \sum_{x\in\mathcal{O}^*} (|x|+1)w^2(x)\mathrm{Pr}^2(x) && (10)\\
&= \sum_{n=0}^{\infty}(n+1)\sum_{|x|=n} w^2(x)\mathrm{Pr}^2(x) && (11)\\
&\leq \sum_{n=0}^{\infty}(n+1)\max_{|x|=n} w^2(x) && (12)\\
&< \infty \ , && (13)
\end{aligned}$$

where we use the fact that each sequence x can be split into a history h and a test t in exactly $|x|+1$ ways, and $\sum_{|x|=n}\mathrm{Pr}(x) = 1$. (Note that for some systems the unweighted M may already have this property, but by introducing a weighting function we guarantee it in every case.) The bi-infinite matrix $\hat{M}$ therefore describes a Hilbert-Schmidt operator and has a singular value decomposition (SVD) given by

$$\hat{M} = U\Sigma V^\top \ , \tag{14}$$

where U and V are infinite orthogonal matrices and Σ is a bi-infinite diagonal matrix whose diagonal entries are the singular values $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots$ [Kennedy and Sadeghi, 2013, Smithies and Varga, 2006]. Below, we denote by U_k the $\infty \times k$ matrix containing the first k columns of U (similarly for V_k) and by Σ_k the $k \times k$ upper-left submatrix of Σ.

In order to learn a PSR we will need several quantities, analogous to those in Equation (4), that are derived from $\hat{M}$. Let P_* denote the column of $\hat{M}$ corresponding to ϵ; that is, $P_* = \hat{M}\mathbf{1}_\epsilon$, where $\mathbf{1}_\epsilon$ is an infinite binary vector with a single one in the position indexed by ϵ. Likewise, let $P_\infty = \mathbf{1}_\epsilon^\top \hat{M}$ denote the row of $\hat{M}$ corresponding to ϵ. Finally, let P_o for any $o \in \mathcal{O}$ denote the bi-infinite matrix whose t, h entry is given by $\hat{M}_{t,ho}$. P_o contains only the columns of $\hat{M}$ that are indexed by a history ending in o, and we can write it as $P_o = \hat{M}R_o$, where R_o is the bi-infinite binary matrix with $[R_o]_{h_1h_2} = 1$ if and only if $h_1 = h_2o$.

Finally, since we will be learning from $\hat{M}$ instead of M, our prediction function must "undo" the weighting function; for any sequence $x = o_1o_2\cdots o_n$ we redefine

$$\mathrm{Pr}_{\mathcal{B}}(x) = \frac{1}{w(x)}\boldsymbol{b}_\infty^\top B_{o_n}\cdots B_{o_1}\boldsymbol{b}_* \ . \tag{15}$$

With these definitions we will prove the following theorem.

Theorem 1. *Assume that the weighted system-dynamics matrix $\hat{M}$ has rank k or greater. Let $\hat{M} = U\Sigma V^\top$ be a singular value decomposition with singular values $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots$, and let $\mathcal{B}$ be the rank k weighted PSR given by*

$$\begin{aligned}
\boldsymbol{b}_* &= U_k^\top P_*\\
B_o &= U_k^\top P_o\left(U_k^\top\hat{M}\right)^+\\
\boldsymbol{b}_\infty^\top &= P_\infty^\top\left(U_k^\top\hat{M}\right)^+ \ .
\end{aligned} \tag{16}$$

Then $\mathcal{L}(\mathcal{B}) \leq \sum_{i=k+1}^{\infty}\sigma_i^2$.

Kulesza et al. [2014] showed that serious problems can arise when applying low-rank spectral learning to an unweighted finite submatrix of M. In contrast, Theorem 1 guarantees that, given access to the infinite weighted system-dynamics matrix $\hat{M}$, the low-rank spectral learning algorithm in Equation (16) behaves in a predictable way, with loss bounded directly by the omitted singular values.

Of course, working with an infinite system-dynamics matrix is impossible in practice, but the result is still informative in several ways. First, if we choose a weighting function that is nonzero on only a finite set of sequences X, then $\hat{M}$ has only a finite number of nonzero entries. In particular, if $\mathcal{T}$ and $\mathcal{H}$ contain (respectively) all the suffixes and prefixes of sequences in X, then low-rank spectral learning on a weighted version of $P_{\mathcal{T},\mathcal{H}}$ is equivalent to Equation (16). (Alternatively, we can think of $\mathcal{T}$ and $\mathcal{H}$ as defining a particular harsh weighting function that only cares about predictions on sequences that appear in those sets.) So in some cases we actually can implement the algorithm in Theorem 1.

Second, even in the general case, the convergence properties of the weighting function guarantee that $\hat{M}$ can be described as a limit of finite-rank matrices [Riesz and Sz.-Nagy, 1990]. In particular, given a data set, the maximum likelihood estimate of $\hat{M}$ will contain only a finite number of nonzeros, and therefore we can effectively implement Equation (16) by choosing $\mathcal{T}$ and $\mathcal{H}$ to contain just the sequences observed in the data. If the data set grows such that every sequence with nonzero probability is eventually observed, then the maximum likelihood estimate approaches $\hat{M}$ (and the estimated singular values approach $\{\sigma_i\}$). Thus we can implement an algorithm that converges, in the limit of data, to the behavior in the theorem.

Still, a data set may contain a very large number of sequences, making this technique expensive. Instead, the most practical approach is usually to simply choose sets $\mathcal{T}$ and $\mathcal{H}$ that are as large as is manageable. Indeed, as $\mathcal{T}$ and $\mathcal{H}$ grow to include longer and longer sequences, the zero-padded bi-infinite extension of $P_{\mathcal{T},\mathcal{H}}$ converges to $\hat{M}$ as well. While in practice it may seem statistically problematic to accurately estimate a very large $P_{\mathcal{T},\mathcal{H}}$ matrix, Denis et al. [2014] showed that the concentration of the empirical $P_{\mathcal{T},\mathcal{H}}$ around its mean is essentially independent of dimension, and argued that statistical considerations should therefore not prevent us from using large $\mathcal{T}$ and $\mathcal{H}$. Thus, we can hope that as the sets of tests and histories that we use get larger, the empirical performance of low-rank spectral learning will transition from the problematic regime of Kulesza et al. [2014] to the well-behaved regime of Theorem 1. We show experimentally that this is true in Section 5.

An interesting consequence of Theorem 1 is that the weight function, which is chosen by the practitioner and used to derive $\hat{M}$, has the potential to make learning easier (in the sense of reducing singular values of $\hat{M}$, for instance, by having many zeros) or harder (in the sense of increasing the bound in Theorem 1; we will show an example in Section 4.1). We argue that this is of fundamental importance: a system is not by itself easy or hard to learn; the difficulty of the learning problem depends on the loss function we aim to optimize.

To prove Theorem 1 we will use the following lemma, whose proof is straightforward but omitted for space.

Lemma 1.

$$\mathbf{1}_\epsilon \mathbf{1}_\epsilon^\top = I - \sum_{o\in\mathcal{O}} R_o R_o^\top. \tag{17}$$

Proof of Theorem 1. Since $\hat{M} = U\Sigma V^\top$ is a singular value decomposition, we have $U_k^\top \hat{M} = \Sigma_k V_k^\top$ and $\hat{M} V_k \Sigma_k^{-1} = U_k$. Using these facts, we can rewrite Equation (16):

$$\boldsymbol{b}_* = U_k^\top \hat{M} \mathbf{1}_\epsilon \qquad = \Sigma_k V_k^\top \mathbf{1}_\epsilon \tag{18}$$

$$B_o = U_k^\top \hat{M} R_o V_k \Sigma_k^{-1} \qquad = \Sigma_k V_k^\top R_o V_k \Sigma_k^{-1} \tag{19}$$

$$\boldsymbol{b}_\infty^\top = \mathbf{1}_\epsilon^\top \hat{M} V_k \Sigma_k^{-1} \qquad = \mathbf{1}_\epsilon^\top U_k \ , \tag{20}$$

where $(U_k^\top \hat{M})^+ = (\Sigma_k V_k^\top)^+ = V_k \Sigma_k^{-1}$ because $V_k^\top$ has orthonormal rows.

Now, for any sequence $x = o_1 o_2 \dots o_n$ we have

$$\begin{aligned} & w(x)\mathrm{Pr}_{\mathcal{B}}(x) \\ &\quad = \boldsymbol{b}_\infty^\top B_{o_n} \cdots B_{o_1} \boldsymbol{b}_* \qquad (21) \\ &\quad = \left(\mathbf{1}_\epsilon^\top U_k\right) \left(\Sigma_k V_k^\top R_{o_n} V_k \Sigma_k^{-1}\right) \\ &\qquad\quad \cdots \left(\Sigma_k V_k^\top R_{o_1} V_k \Sigma_k^{-1}\right) \left(\Sigma_k V_k^\top \mathbf{1}_\epsilon\right) \qquad (22) \\ &\quad = \mathbf{1}_\epsilon^\top \hat{M} V_k V_k^\top R_{o_n} V_k V_k^\top \cdots R_{o_1} V_k V_k^\top \mathbf{1}_\epsilon \ . \qquad (23) \end{aligned}$$

Thus, prediction is effectively a series of alternating projection $(V_k V_k^\top)$ and shift (R_o) operations on the row $P_\infty = \mathbf{1}_\epsilon^\top \hat{M}$, until finally the element corresponding to the empty string is extracted. Note that, if we omit the low-rank projections, we obtain exact predictions:

$$\mathbf{1}_\epsilon^\top \hat{M} R_{o_n} \cdots R_{o_1} \mathbf{1}_\epsilon = \hat{M}_{x,\epsilon} = w(x)\mathrm{Pr}(x) \ . \tag{24}$$

Let $W \equiv V_k V_k^T$ denote the orthonormal projection matrix, which is symmetric and satisfies $W^2 = W$ and $(I - W)^2 = I - W$. Let $\textsc{rw}_x \equiv R_{o_n} W \cdots R_{o_1} W$, so that Equation (23) can be abbreviated as

$$\mathrm{Pr}_{\mathcal{B}}(x) = \frac{1}{w(x)} \mathbf{1}_\epsilon^\top \hat{M} W \textsc{rw}_x \mathbf{1}_\epsilon \ , \tag{25}$$

and similarly let $R_x \equiv R_{o_n} \cdots R_{o_1}$.

Using this notation, we can put the weighted PSR loss (Equation (5)) into a quadratic form in $\mathbf{1}_\epsilon^\top \hat{M}$:

$$\begin{aligned} \mathcal{L}(\mathcal{B}) &= \sum_{x\in\mathcal{O}^*} w^2(x) \left[M_{x,\epsilon} - \frac{1}{w(x)} \mathbf{1}_\epsilon^\top \hat{M} W \textsc{rw}_x \mathbf{1}_\epsilon \right]^2 \\ &= \sum_{x\in\mathcal{O}^*} \left[\hat{M}_{x,\epsilon} - \mathbf{1}_\epsilon^\top \hat{M} W \textsc{rw}_x \mathbf{1}_\epsilon \right]^2 \qquad (26) \\ &= \sum_{x\in\mathcal{O}^*} \left[\mathbf{1}_\epsilon^\top \hat{M} R_x \mathbf{1}_\epsilon - \mathbf{1}_\epsilon^\top \hat{M} W \textsc{rw}_x \mathbf{1}_\epsilon \right]^2 \qquad (27) \\ &= \sum_{x\in\mathcal{O}^*} (\mathbf{1}_\epsilon^\top \hat{M}) \left(W \textsc{rw}_x - R_x \right) \mathbf{1}_\epsilon \\ &\qquad \cdot \mathbf{1}_\epsilon^\top \left(W \textsc{rw}_x - R_x \right)^\top (\mathbf{1}_\epsilon^\top \hat{M})^\top \ . \qquad (28) \end{aligned}$$

We can also rewrite the bound $\sum_{i=k+1}^\infty \sigma_i^2$ as a quadratic form in $\mathbf{1}_\epsilon^\top \hat{M}$. Since W is a projection onto the top k singular vectors of $\hat{M}$, we have $\sum_{i=k+1}^\infty \sigma_i^2 = \|\hat{M} - \hat{M}W\|_F^2$, where $\|\cdot\|_F$ denotes the Frobenius norm. On the other hand, we can write the squared Frobenius norm as the sum of the squared L_2-norms of the rows; if $A_{x,\cdot}$ denotes the x row of matrix A, we have

$$\sum_{i=k+1}^{\infty} \sigma_i^2 = \|\hat{M} - \hat{M}W\|_{\mathcal{F}}^2 \tag{29}$$

$$
\begin{aligned}
&= \sum_{x\in\mathcal{O}^*} \|[\hat{M}(I-W)]_{x,\cdot}\|_2^2 && (30)\\
&= \sum_{x\in\mathcal{O}^*} \|\hat{M}_{x,\cdot}(I-W)\|_2^2 && (31)\\
&= \sum_{x\in\mathcal{O}^*} \|\mathbf{1}_\epsilon^\top \hat{M} R_x (I-W)\|_2^2 && (32)\\
&= \sum_{x\in\mathcal{O}^*} (\mathbf{1}_\epsilon^\top \hat{M}) R_x (I-W) R_x^\top (\mathbf{1}_\epsilon^\top \hat{M})^\top \ .
\end{aligned}
$$

With $\sum_{i=k+1}^{\infty}\sigma_i^2$ and $\mathcal{L}(\mathcal{B})$ (and thus their difference) in the same quadratic form, it suffices to show that

$$
\begin{aligned}
\sum_{x\in\mathcal{O}^*} \big[& R_x(I-W)R_x^\top \\
& -(W\textsc{rw}_x - R_x)\mathbf{1}_\epsilon \mathbf{1}_\epsilon^\top (W\textsc{rw}_x - R_x)^\top \big] \quad (33)
\end{aligned}
$$

is a positive semidefinite matrix. To simplify notation, define the following operator for any matrix A:

$$
A^{2\top} \equiv AA^\top \ . \quad (34)
$$

Then, applying Lemma 1, Equation 33 is equal to

$$
\begin{aligned}
&\sum_{x\in\mathcal{O}^*} \Big[(R_x(I-W))^{2\top} && (35)\\
&\quad - (W\textsc{rw}_x - R_x)\left(I - \sum_{o\in\mathcal{O}} R_o^{2\top}\right)(W\textsc{rw}_x - R_x)^\top \Big] \\
&= \sum_{x\in\mathcal{O}^*} \Big[(R_x(I-W))^{2\top} - (W\textsc{rw}_x - R_x)^{2\top} \Big] \\
&\quad + \sum_{x\in\mathcal{O}^*}\sum_{o\in\mathcal{O}} (W\textsc{rw}_x R_o - R_x R_o)^{2\top} \ . && (36)
\end{aligned}
$$

Since $\textsc{rw}_\epsilon = R_\epsilon = I$, the first term of the first sum disappears:

$$
\begin{aligned}
&(R_\epsilon(I-W))^{2\top} - (W\textsc{rw}_\epsilon - R_\epsilon)^{2\top} \\
&\qquad = (I-W)^2 - (W-I)^2 = 0 \ . \quad (37)
\end{aligned}
$$

Therefore, letting $\textsc{wr}_x \equiv W R_{o_n} W \cdots R_{o_1}$, the above is equal to

$$
\begin{aligned}
&\sum_{|x|\geq 1} \Big[(R_x(I-W))^{2\top} - (W\textsc{rw}_x - R_x)^{2\top} \Big] \\
&\quad + \sum_{x\in\mathcal{O}^*}\sum_{o\in\mathcal{O}} (\textsc{wr}_{ox} - R_{ox})^{2\top} && (38)\\
&= \sum_{|x|\geq 1} \Big[(R_x(I-W))^{2\top} - (\textsc{wr}_x W - R_x)^{2\top} \\
&\qquad + (\textsc{wr}_x - R_x)^{2\top} \Big] \ . && (39)
\end{aligned}
$$

Note that we replaced the double sum $\sum_{x\in\mathcal{O}^*}\sum_{o\in\mathcal{O}}$ with $\sum_{|x|\geq 1}$ since the set generated by prepending every observation to every sequence is just the set of all sequences with length at least one. The final step is to show that each term in the remaining sum is positive semidefinite. Manipulating the second inner term,

$$
\begin{aligned}
&(\textsc{wr}_x W - R_x)^{2\top} \\
&= (\textsc{wr}_x - R_x - \textsc{wr}_x(I-W))^{2\top} && (40)\\
&= (\textsc{wr}_x - R_x)^{2\top} + (\textsc{wr}_x(I-W))^{2\top} \\
&\quad - \textsc{wr}_x(I-W)(\textsc{wr}_x - R_x)^\top \\
&\quad - (\textsc{wr}_x - R_x)(I-W)(\textsc{wr}_x)^\top && (41)\\
&= (\textsc{wr}_x - R_x)^{2\top} + (\textsc{wr}_x(I-W))^{2\top} \\
&\quad - (\textsc{wr}_x(I-W))^{2\top} + \textsc{wr}_x(I-W)R_x^\top \\
&\quad - (\textsc{wr}_x(I-W))^{2\top} + R_x(I-W)(\textsc{wr}_x)^\top && (42)\\
&= (\textsc{wr}_x - R_x)^{2\top} - (\textsc{wr}_x(I-W))^{2\top} \\
&\quad + \textsc{wr}_x(I-W)R_x^\top + R_x(I-W)(\textsc{wr}_x)^\top \ . && (43)
\end{aligned}
$$

Combining with the remaining two inner terms, each full term of the summation in Equation 39 is equal to

$$
\begin{aligned}
&(R_x(I-W))^{2\top} + (\textsc{wr}_x(I-W))^{2\top} \\
&\qquad - \textsc{wr}_x(I-W)R_x^\top - R_x(I-W)(\textsc{wr}_x)^\top \\
&\quad = (R_x(I-W) - \textsc{wr}_x(I-W))^{2\top} \ , && (44)
\end{aligned}
$$

which is positive semidefinite. Therefore $\mathcal{L}(\mathcal{B}) \leq \sum_{i=k+1}^{\infty}\sigma_i^2$ for any $\hat{M}$. □

4.1 Non-monotonicity

We previously argued that Theorem 1 supports choosing the largest manageable sets of tests and histories, and in the next section we will empirically validate that claim. However, we first pause to note that it is not strictly guaranteed that larger sets are always better; it is possible to add tests and histories but reduce prediction accuracy. Intuitively, this can happen when some "unexpected" property of the problem, which cannot be easily modeled with low rank, appears in the added sequences. We will create such a situation by manipulating the weighting function, emphasizing the important role it can play. (Of course, the system itself can also cause non-monotonic behavior, even when the weight function is "normal.")

Consider a system with a single observation, so that the system-dynamics matrix M consists of all ones. Let the weighting function be $w(x) = r^{|x|}$ for some $|r| < 1$. When the rows and columns are ordered by sequence length, the weighted system-dynamics matrix $\hat{M}$ is a Hankel matrix: it has constant skew-diagonals equal to 1, r, r^2, etc. (See Figure 1a.)

Since any row of $\hat{M}$ is a geometric sequence with ratio r, $\hat{M}$ has rank one, and thus any nonempty sets of tests and histories will produce a perfect rank-one spectral model. Now suppose that we modify the weighting

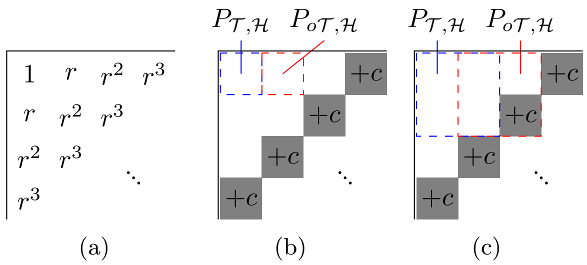

Figure 1: (a) $\hat{M}$ has constant skew-diagonals. (b) Only the shaded entries change under the modified weights, but when $\mathcal{T} = \mathcal{H} = \{\epsilon\}$, the P-statistics do not depend on them. (c) When $\mathcal{T}$ and $\mathcal{H}$ are expanded, the modified weights affect learning.

function slightly, setting $w(x) = r^{|x|} + c$ when $|x| = 3$, and leaving it otherwise unchanged. The new $\hat{M}$ is depicted in Figure 1b, and (in general) has rank five.

If we now perform rank-one spectral learning with $\mathcal{T} = \mathcal{H} = \{\epsilon\}$, $P_{\mathcal{T},\mathcal{H}}$ and $P_{o\mathcal{T},\mathcal{H}}$ do not depend on c since they involve no sequences of length three (see Figure 1b). The learned model, therefore, will make the same predictions as before. These predictions are correct for all sequences except the one with the modified weight—the sequence of length three—and therefore have a finite loss of c^2.

If we now expand $\mathcal{T}$ and $\mathcal{H}$ to include the sequence of length one (see Figure 1c), but continue to perform rank-one learning, we have

$$P_{\mathcal{T},\mathcal{H}} = \begin{bmatrix} 1 & r \\ r & r^2 \end{bmatrix} \quad P_{o\mathcal{T},\mathcal{H}} = \begin{bmatrix} r & r^2 \\ r^2 & r^3 + c \end{bmatrix} . \quad (45)$$

The first left singular vector of P is proportional to $[1 \; r]^\top$, and so (it is straightforward to verify) the learned parameter is $B = r + \frac{cr^2}{r^4+2r^2+1}$. If $c = 0$, we recover the unaltered weight function and B is the true ratio. However, if c is sufficiently large, then B will be greater than one, the model's predictions will diverge, and its loss will be infinite. This leads to the following claim.

Claim 1. *Let $\mathcal{B}(\mathcal{T}, \mathcal{H}, k)$ denote the PSR obtained from rank-k spectral learning using tests $\mathcal{T}$ and histories $\mathcal{H}$. There exist systems M, ranks k, and sets of observation sequences $\mathcal{T}, \mathcal{T}', \mathcal{H}, \mathcal{H}'$ such that $\mathcal{T} \subseteq \mathcal{T}'$ and $\mathcal{H} \subseteq \mathcal{H}'$, but $\mathcal{L}(B(\mathcal{T}, \mathcal{H}, k)) < \mathcal{L}(\mathcal{B}(\mathcal{T}', \mathcal{H}', k))$.*

Nonetheless, we will show in Section 5 that using large $\mathcal{T}$ and $\mathcal{H}$ is usually beneficial in practice.

5 EXPERIMENTS

We aim to show (a) that the loss of a low (fixed) rank PSR model tends to decrease monotonically as the $\mathcal{T}$ and $\mathcal{H}$ used to learn it grow, (b) that this phenomenon persists when the statistics in $P_{\mathcal{T},\mathcal{H}}$ are estimated from data (and hence are inexact), and finally (c) that this phenomenon holds in a real-world data set.

5.1 Learning Synthetic HMMs

Domains We generate HMMs with 100 states and 4 observations as follows. The observation probabilities in a given state are chosen uniformly at random from $[0, 1]$ and then normalized. Transition probabilities are chosen to reflect three different hidden-state topologies:

- **Random**: Each state has 5 possible next 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.

For each topology, the non-zero entries of the transition matrix are chosen uniformly at random from $[0, 1]$ and then normalized; the initial state distribution is built in the same way. The system-dynamics matrices for these HMMs generally have rank 100 [Singh et al., 2004].

PSR Learning We use a weighting function that is constant up to length 10 and zero thereafter: $w(x) = \mathbb{I}(|x| \leq 10)$. Histories and tests indexing $\hat{M}$ are sorted by length, and within length lexicographically. We then let $P_{\mathcal{T},\mathcal{H}}$ be the $|\mathcal{T}| \times |\mathcal{H}|$ top-left corner of $\hat{M}$. (Kulesza et al. [2015] showed that, given target sizes $|\mathcal{T}|$ and $|\mathcal{H}|$, it is usually possible to improve performance by choosing $\mathcal{T}$ and $\mathcal{H}$ in a more sophisticated way; here, we use the top-left corner of $\hat{M}$ in order to isolate the effect of size.) For our experiments we fix $|\mathcal{T}| = |\mathcal{H}|$. Given $\hat{M}$, $|\mathcal{H}|$, and model rank k, we learn a PSR using Equation (16) with the finite $P_{\mathcal{T},\mathcal{H}}$ in place of $\hat{M}$.

Evaluation Since our weighting function is 0 for sequences of length > 10, the loss (Equation (5)) can be computed without performing an infinite sum. Still, there are too many sequences to tractably compute the exact loss. Instead, we estimate Equation (5) using 100 uniformly sampled sequences of each length, which is sufficient to achieve low variance. Because an inexact PSR may predict negative probabilities, we clamp the predicted probabilities to $[0, \infty)$ and normalize them.

Results In Figure 2 we vary $|\mathcal{H}|$ from 10 to 100, plotting the loss of the learned PSR vs. $|\mathcal{H}|$ for fixed model ranks of 10, 30, and 50. In all three domain topologies, for each rank the loss monotonically decreases as $|\mathcal{H}|$ increases. This satisfies objective (a): providing larger $\mathcal{T}$ and $\mathcal{H}$ for a (fixed) low-rank yields better PSRs. In

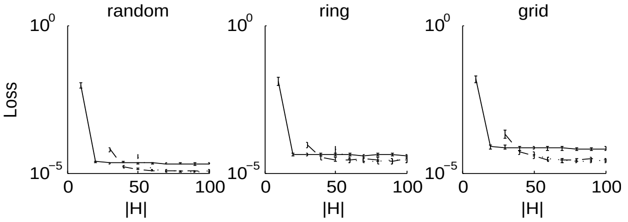

Figure 2: Average loss of low-rank PSRs on synthetic HMMs of three different topologies, using exact statistics. The solid curve is rank 10, the dashed curve is rank 30, and the dotted curve is rank 50.

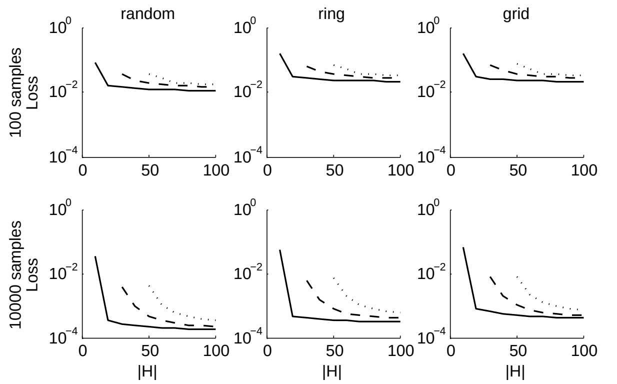

Figure 3: Average loss of low-rank PSRs on synthetic HMMs using statistics estimated from data. The curves correspond to rank as in Figure 2.

Figure 3, we show that this phenomenon persists when $P_{\mathcal{T},\mathcal{H}}$ is estimated from data (objective (b)). We use 100 or 10,000 sample trajectories (sufficiently long to populate $P_{\mathcal{T},\mathcal{H}}$) to estimate the statistics, and then repeat the experiment in Figure 2. Figure 3 shows that loss continues to decrease with increasing $|\mathcal{H}|$.

5.2 Wikipedia Text Prediction

To address objective (c), we turn to a large corpus of real-world Wikipedia text treated as a time series where each character is an observation. Following Sutskever et al. [2011], we take 1GB of text as training data (treated as a single long sequence), and leave the rest for testing. The entries of the system-dynamics matrix are estimated as follows: given a training character sequence, for any history h and any test t, the estimated joint probability is the probability of observing ht at a randomly selected position in the sequence.

We evaluate a PSR on the test set by making predictions incrementally: at any position in the test sequence, the model predicts the immediate next observation based on the current state vector, sees the true next observation, makes a state update, moves to the next position, and repeats this procedure. Following Sutskever et al. [2011], we reset the state vector to $\boldsymbol{b}_*$ after every 250 observations, and in computing the model's quality we ignore the predictions made on the first 50 observations out of each 250 (though these observations are still used for state updates). Finally, the average likelihood of the predictions is calculated as the performance of the model.

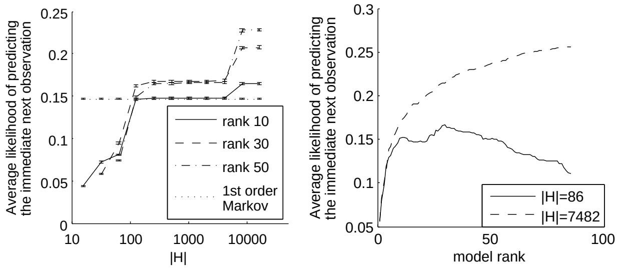

Figure 4: Performance of low-rank PSRs on Wikipedia data. **Left**: Test likelihood vs. $|\mathcal{H}|$ for three different model ranks and a baseline. **Right**: Test likelihood vs. model rank for two different values of $|\mathcal{H}|$.

Results The left plot of Figure 4 shows low-rank PSR performance curves for three different choices of rank (10, 30 and 50) over a range of $|\mathcal{H}|$. The key empirical phenomenon, that for each fixed rank the performance improves monotonically with $|\mathcal{H}|$, is also observed here. As a baseline we include the performance of a first-order Markov model of the training data (which has an implicit rank of 86 [Singh et al., 2004]). In the right plot, the x-axis is model rank and the two curves correspond to $\mathcal{T}$ and $\mathcal{H}$ containing all strings up to length 1 ($|\mathcal{H}| = 86$; solid line) and 2 ($|\mathcal{H}| = 7482$; dashed line). Again, for any fixed model rank a larger $|\mathcal{H}|$ helps (i.e., the dashed curve dominates the solid curve). Additionally, for $|\mathcal{H}| = 86$, performance increases with rank up to a point and then decreases; this is consistent with overfitting to the noise in $P_{\mathcal{T},\mathcal{H}}$.

6 CONCLUSION

In contrast to the undesirable behavior found by Kulesza et al. [2014], we proved that the introduction of a weighting function and the use of sufficient numbers of tests and histories is enough to guarantee that low-rank spectral learning for PSRs is well-behaved. Empirical evaluations support the use of the largest manageable sets $\mathcal{T}$ and $\mathcal{H}$ for any fixed model rank.

Acknowledgements

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

Animashree Anandkumar, Daniel Hsu, and Sham M. Kakade. A method of moments for mixture models and hidden Markov models. In *Proceedings of the Twenty-Fifth Annual Conference on Learning Theory*, 2012.

Borja Balle, Xavier Carreras, Franco M. Luque, and Ariadna Quattoni. Spectral learning of weighted automata. *Machine Learning*, pages 1–31, 2013.

Florent Benaych-Georges and Raj Rao Nadakuditi. The singular values and vectors of low rank perturbations of large rectangular random matrices. *Journal of Multivariate Analysis*, 111:120–135, 2012.

Byron Boots, Sajid M. Siddiqi, and Geoffrey J. Gordon. Closing the learning-planning loop with predictive state representations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 1369–1370, 2010.

Shay B. Cohen, Karl Stratos, Michael Collins, Dean P. Foster, and Lyle Ungar. Spectral learning of latent-variable pcfgs: Algorithms and sample complexity. *Journal of Machine Learning Research*, 15:2399–2449, 2014. URL `http://jmlr.org/papers/v15/cohen14a.html`.

François Denis, Mattias Gybels, and Amaury Habrard. Dimension-free concentration bounds on hankel matrices for spectral learning. In *Proceedings of The 31st International Conference on Machine Learning*, pages 449–457, 2014.

Daniel Hsu, Sham M. Kakade, and Tong Zhang. A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences*, 78 (5):1460–1480, 2012.

Rodney A. Kennedy and Parastoo Sadeghi. *Hilbert Space Methods in Signal Processing*. Cambridge University Press, 2013. ISBN 9781107010031.

Alex Kulesza, Raj Rao Nadakuditi, and Satinder Singh. Low-rank spectral learning. In *Proceedings of the 17th Conference on Artificial Intelligence and Statistics*, 2014.

Alex Kulesza, Nan Jiang, and Satinder Singh. Spectral learning of predictive state representations with insufficient statistics. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015.

Ankur P. Parikh, Le Song, and Eric P. Xing. A spectral algorithm for latent tree graphical models. In *Proceedings of The 28th International Conference on Machine Learning*, 2011.

Frigyes Riesz and Bela Sz.-Nagy. *Functional Analysis*. Dover Books on Mathematics Series. Dover Publications, 1990. ISBN 9780486662893.

Satinder Singh, Michael R. James, and Matthew R. Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 512–519. AUAI Press, 2004.

Laura Smithies and Richard S. Varga. Singular value decomposition Geršgorin sets. *Linear algebra and its applications*, 417(2):370–380, 2006.

Ilya Sutskever, James Martens, and Geoffrey E. Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1017–1024, 2011.