Online Spectral Identification of Dynamical Systems

Byron Boots
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
beb@cs.cmu.edu

Geoff Gordon
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
ggordon@cs.cmu.edu

1 Introduction

Recently, a number of researchers have proposed *spectral* algorithms for learning models of nonlinear dynamical systems—for example, Hidden Markov Models (HMMs) [1, 2], Partially Observable Markov Decision Processes (POMDPs) [3], and Predictive State Representations (PSRs) [4, 3, 5]. These algorithms are attractive since they are *statistically consistent* and not subject to local optima. However, they are *batch* methods: they need to store their entire training data set in memory at once and operate on it as a large matrix, and so they cannot scale to extremely large data sets (either many examples or many features per example). In turn, this restriction limits their ability to learn accurate models of complex systems.

To remedy this drawback, we propose a fast, online spectral algorithm for PSRs. PSRs *subsume* HMMs and POMDPs [6, 4]. In fact, previous spectral learning algorithms for several types of HMMs [1, 2, 7] are more accurately described as PSR learning algorithms *applied to* HMMs. Therefore, our algorithm also improves on past algorithms for these other models. Our method leverages fast, low-rank modifications of the thin singular value decomposition [8], and uses tricks such as random projections to scale to extremely large numbers of examples and features per example. Consequently, the new method can handle orders of magnitude larger data sets than previous methods, and can therefore scale to learn systems that are too complex for previous methods.

Experiments show that our online spectral learning algorithm does a good job recovering the parameters of a nonlinear dynamical system in several partially observable domains. In our first experiment we empirically demonstrate that our online spectral learning algorithm is unbiased by recovering the parameters of a small but difficult synthetic Reduced-Rank HMM. In our second experiment we demonstrate the performance of the new method on a high-bandwidth video understanding task.

This work was presented as a technical paper at AAAI 2011 [9]. The full-length paper can be found here: http://www.cs.cmu.edu/∼beb/files/boots-gordon-online-PSRs.pdf

2 Online Spectral Updates to Dynamical System Parameters

The main contribution of our work is a novel *online* spectral learning algorithm for identifying the parameters of PSRs. A PSR is a compact description of a dynamical system that represents state as a set of predictions of observable experiments or *tests*. The key idea behind a PSR is that, if we know the expected outcomes of all possible tests, then we know everything there is to know about state. Instead of representing all possible tests, however, PSRs maintain a small number of sufficient statistics which are *linear combinations* of predictions for a (potentially very large) set of tests.

This fact encapsulates the main benefit of PSR learning algorithms: given a large set of tests, we can find low dimensional parameters using spectral methods and regression. In this respect, PSRs are closely related to the transformed representations of Kalman filters and HMMs found by *subspace identification* [10, 11, 1]. The details of PSR learning algorithms will not be discussed here (see [3] for details). Instead we will focus on the pivotal step in the PSR learning algorithm: a spectral decomposition used to discover the latent state space of the dynamical system.

We assume for simplicity that our data is a single long sequence of observations $o_{1:T}$ sampled from the PSR. Our goal will be to recover the PSR state space and dynamics M up to a similarity

transform—no more is possible, since a similarity transform doesn't affect predictions [3]. Our algorithm is based on vectors of features of histories and features of future observations. In particular, write $\chi_t = \chi(h_t) \in \mathbb{R}^\ell$ for a vector of features of history (a sequence of observations prior to time t), and write $\phi_t = \phi\left(o_{t:(t+N_F-1)}\right) \in \mathbb{R}^k$ for a vector of features of the next N_F observations, where $k, \ell \geq d$. Given the above notation, we define the following moments:

$$\Sigma = \mathbb{E}[\phi_t \chi_t^\top] \qquad \Pi = \mathbb{E}[\phi_{t+1} \chi_t^\top]$$

Here, expectations are with respect to the stationary distribution, and are therefore independent of t.

Our algorithm will estimate the above moments from data, and then manipulate the estimated moments to recover the PSR parameters. To this end, we can express the moments in terms of the PSR parameters:

$$\Sigma = RP \qquad \Sigma = RMP \tag{1}$$

Here, $R \in \mathbb{R}^{k \times d}$ and $P \in \mathbb{R}^{d \times \ell}$ are derived from PSR parameters as described in the full paper [9]. Eq. 1 implies that the ranks of Σ and Π are no greater than d, the latent dimension of the PSR (since we can write each matrix as a product of factors, at least one of which has no more than d columns).

Let U be any matrix such that $U^\mathsf{T} R$ is invertible (e.g., the d leading left singular vectors of Σ). Then, we can recover the PSR dynamics M (up to a similarity transform) from U, Σ, and Π:

$$U^\mathsf{T}\Pi(U^\mathsf{T}\Sigma)^\dagger = U^\mathsf{T}RMP(U^\mathsf{T}RP)^\dagger = U^\mathsf{T}RMPP^\dagger(U^\mathsf{T}R)^{-1} = (U^\mathsf{T}R)M(U^\mathsf{T}R)^{-1}$$

Here we have used the definitions of Σ and Π; the assumption that $U^\mathsf{T}R$ is invertible; and the assumptions that $\ell \geq d$ and that P has rank d (so that $PP^\dagger = I$).

The above equations yield a simple spectral learning algorithm: compute $\widehat{\Sigma} = \frac{1}{T}\sum_{t=1}^{T} \phi_t \chi_t^\mathsf{T}$ and $\widehat{\Pi} = \frac{1}{T}\sum_{t=1}^{T} \phi_t \chi_t^\mathsf{T}$. Compute the first d left singular vectors of $\widehat{\Sigma}$ and collect them into a matrix $\widehat{U}$. Finally, compute $\widehat{M} = \widehat{U}^\mathsf{T}\widehat{\Pi}(\widehat{U}^\mathsf{T}\widehat{\Sigma})^\dagger$.

The naïve algorithm requires storing the matrices in Eq. 1, updating these matrices given new information, and recomputing the PSR parameters.[1] This works well when the number of features of tests and histories is relatively small, and in cases where data is collected in batch. Unfortunately, these restrictions can be limiting for many real-world data sets. In practice, the number of features may need to be quite large in order to accurately estimate the parameters of the PSR. Additionally, we are often interested in estimating PSRs from massive datasets, updating PSR parameters given a new batch of data, or learning PSRs online from a data stream. In this work we develop several computationally efficient extensions to overcome these practical obstacles to learning in real-world situations

PSR parameters are generally much lower dimensional than the moments used to estimate them (e.g. a $d \times d$ matrix rather than a $k \times l$ matrix). Therefore, the key idea of the current work is to update the lower-dimensional parameters directly, instead of the naïve updates suggested above, by taking advantage of numerical algorithms for updating singular value decompositions efficiently [8]. The crux of the algorithm involves sequential rank-1 updating schemes for computing a thin SVD of the matrix $\widehat{\Sigma}$. In this setting we assume that at each time step we are given a new set of vectors representing a single sample of features of the future ϕ_t, ϕ_{t+1} and features of histories χ_t. The main computational savings come from using incremental SVD to update $\widehat{U}, \widehat{S}, \widehat{V}$ directly, which is much more efficient than the naïve additive update when the number of new data points is much smaller than the number of features in ϕ_t and χ_t. See the long version of this paper for details and the additional steps required for learning a full set of PSR parameters [9].

3 Random Projections for High Dimensional Feature Spaces

Despite their simplicity and wide applicability, HMMs, POMDPs, and PSRs are limited in that they are usually restricted to discrete observations, and the state is usually restricted to have only moderate cardinality. Recently, Song et al. proposed a spectral learning algorithm for HMMs with continuous observations by representing distributions over these observations and continuous latent states as embeddings in an infinite dimensional Hilbert space [7]. These Hilbert Space Embeddings

[1] In fact, the situation is significantly worse for the full PSR algorithm than the sketch provided above. See [9] for details.

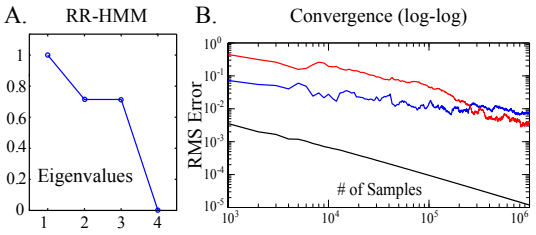

Figure 1: A synthetic RR-HMM. (A.) The eigenvalues of the true transition matrix. (B.) RMS error in the nonzero eigenvalues of the estimated transition matrix vs. number of training samples, averaged over 10 trials. The error steadily decreases, indicating that the PSR model is becoming more accurate, as we incorporate more training data.

of HMMs (HSE-HMMs) use essentially the same framework as other spectral learning algorithms for HMMs and PSRs, but avoid working in the infinite-dimensional Hilbert space by the well-known "kernel trick." HSE-HMMs have been shown to perform well on several real-world datasets, often beating the next best method by a substantial margin. However, they scale poorly due to the need to work with the kernel matrix, whose size is quadratic in the number of training points.

We can overcome this scaling problem and learn PSRs that approximate HSE-HMMs using *random features* for kernel machines [12]: we construct a large but finite set of random features which let us *approximate* a desired kernel using ordinary dot products. The benefit of random features is that we can use fast linear methods that do not depend on the number of data points to approximate the original kernel machine. If we combine random features with the above online learning algorithm, we can approximate an HSE-HMM very closely by using an extremely large number of random features. Such a large set of features would overwhelm batch spectral learning algorithms, but our online method allows us to approximate an HSE-HMM accurately, and scale HSE-HMMs to orders of magnitude larger training sets or even to *streaming* datasets with an inexhaustible supply of training data.

4 Experimental Results

We designed two sets of experiments to evaluate the statistical properties and practical potential of our online spectral learning algorithm. In the first experiment we show the convergence behavior of the algorithm. In the second experiment we demonstrate how the combination of online spectral updates and random features allows us to model a high-bandwidth, high-dimensional video, where the amount of training data would overwhelm a kernel-based method like HSE-HMMs and the number of features would overwhelm a PSR batch learning algorithm.

4.1 A Synthetic Example

First we demonstrate the convergence behavior of our algorithm on a difficult synthetic HMM from Siddiqi et al. [2]. This HMM is 2-step observable, with 4 states, 2 observations, and a rank-3 transition matrix. (So, the HMM is reduced rank (an "RR-HMM") and features of multiple observations are required to disambiguate state.)

We sample observations from the true model and then estimate the model using the our online spectral learning algorithm. Since we only expect to recover the transition matrix up to a similarity transform, we compare the eigenvalues of the estimated transition matrix in the learned model to the eigenvalues of the transition matrix T of the true model. Fig. 1 shows that the learned eigenvalues converge to the true ones as the amount of data increases.

4.2 Modeling Video

Next we look at the problem of mapping from video: we collected a sequence of 11,000 160×120 grayscale frames at 24 fps in an indoor environment (a camera circling a conference room, occasionally switching directions; each full circuit took about 400 frames). This data was collected by hand, so the camera's trajectory is quite noisy. The high frame rate and complexity of the video mean that learning an accurate model requires a very large dataset. Unfortunately, a dataset of this magnitude makes learning an HSE-HMM difficult or impossible: e.g., the similar but less complex example of Song et al. [7] used only 1500 frames.

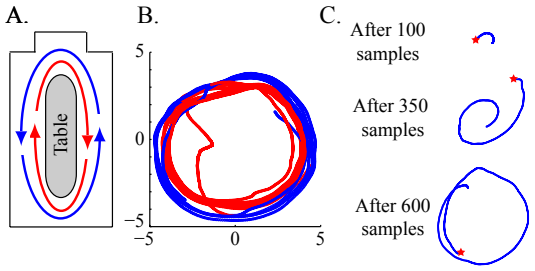

Figure 2: Modeling video. (A.) Schematic of the camera's environment. (B.) The second and third dimension of the learned belief space (the first dimension contains normalization information). Points are colored red when the camera is traveling clockwise and blue when traveling counter-clockwise. The learned state space separates into two manifolds, one for each direction, connected at points where the camera changes direction. (The manifolds appear on top of one another, but are separated in the fourth latent dimension.) (C.) Loop closing: estimated historical camera positions after 100, 350, and 600 steps. Red star indicates current camera position. The camera loops around the table, and the learned map "snaps" to the correct topology when the camera passes its initial position.

Instead, we used random Fourier features and an online PSR to approximate a HSE-HMM with Gaussian RBF kernels. We used tests and histories based on 400 sequential frames from the video, generated 100,000 random features, and learned a 50-dimensional PSR. To duplicate this setup, the batch PSR algorithm would have to find the SVD of a 100,000$\times$100,000 matrix; by contrast, we can efficiently update our parameters by incorporating 100,000-element feature vectors one at a time and maintaining 50×50 and $50\times$100,000 matrices.

Figure 2 shows our results. The final learned model does a surprisingly good job at capturing the major features of this environment, including both the continuous location of the camera and the discrete direction of motion (either clockwise or counterclockwise). Furthermore, the fact that a general-purpose online algorithm learns these manifolds is a powerful result: we are essentially performing simultaneous localization and mapping in a difficult loop closing scenario, *without* any prior knowledge (even, say, that the environment is three-dimensional, or whether the sensor is a camera, a laser rangefinder, or something else).

Acknowledgements

Byron Boots and Geoffrey J. Gordon were supported by ONR MURI grant number N00014-09-1-1052. Byron Boots was supported by the NSF under grant number EEEC-0540865.

References

[1] Daniel Hsu, Sham Kakade, and Tong Zhang. A spectral algorithm for learning hidden Markov models. In *COLT*, 2009.

[2] Sajid Siddiqi, Byron Boots, and Geoffrey J. Gordon. Reduced-rank hidden Markov models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS-2010)*, 2010.

[3] Byron Boots, Sajid M. Siddiqi, and Geoffrey J. Gordon. Closing the learning-planning loop with predictive state representations. In *Proceedings of Robotics: Science and Systems VI*, 2010.

[4] Matthew Rosencrantz, Geoffrey J. Gordon, and Sebastian Thrun. Learning low dimensional predictive representations. In *Proc. ICML*, 2004.

[5] Byron Boots and Geoff Gordon. Predictive state temporal difference learning. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 271–279. 2010.

[6] Satinder Singh, Michael James, and Matthew Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Proc. UAI*, 2004.

[7] L. Song, B. Boots, S. M. Siddiqi, G. J. Gordon, and A. J. Smola. Hilbert space embeddings of hidden Markov models. In *Proc. 27th Intl. Conf. on Machine Learning (ICML)*, 2010.

[8] Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications*, 415(1):20–30, 2006.

[9] Byron Boots, Sajid Siddiqi, and Geoffrey Gordon. An online spectral learning algorithm for partially observable nonlinear dynamical systems. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI-2011)*, 2011.

[10] P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer, 1996.

[11] Tohru Katayama. *Subspace Methods for System Identification*. Springer-Verlag, 2005.

[12] Ali Rahimi and Ben Recht. Random features for large-scale kernel machines. In *Neural Infomration Processing Systems*, 2007.