Improving Predictive State Representations via Gradient Descent

Nan Jiang, Alex Kulesza, Satinder Singh
Computer Science & Engineering, University of Michigan

Abstract Predictive state representations (PSRs) are models of dynamical systems using predictions about future observations as representation of state. They can be learned via a spectral algorithm that is computationally efficient and statistically consistent when the model complexity matches that of the true system. In practice, however, model mismatch is inevitable, and spectral learning may fail to find optimal models. To address this problem, we propose using gradient methods for improving spectrally-learned PSRs. We show that only a small amount of additional gradient optimization can lead to significant performance gains, and moreover that initializing gradient methods with the spectral learning solution yields better models in significantly less time than starting from scratch.

Background: Spectral Learning for PSRs vs Expectation Maximization for HMMs

Example sequence modelling task: daily weather

	Example state vector on day *t*	Model parameters (*m* observations)	Prediction rule	Learning algorithm
Hidden Markov Models (HMMs) *State is the posterior of the latent variable*	$\Pr(z_t = \text{sunny}) = 0.2$ $\Pr(z_t = \text{cloudy}) = 0.3$ $\Pr(z_t = \text{rainy}) = 0.5$	***n*: #states** Initial state vector: $[\pi(\text{sunny})]_{n\times 1}$ Transition: $[T(\text{sunny}\mid\text{cloudy})]_{n\times n}$ Emission: $[O(\text{sunny}\mid\text{sunny})]_{m\times n}$	Pr(sunny, cloudy, sunny, rainy) = $\mathbf{1}^\top T O_{\text{rainy}} T O_{\text{sunny}} T O_{\text{cloudy}} T O_{\text{sunny}} \pi$ $\text{diag}\{O(\text{rainy}\mid\cdot)\}$	Expectation Maximization • Iterative and slow • Local optimum (no consistency guarantee) • Optimizing data likelihood
	can always convert to			
Predictive State Representations (PSRs) *State is predictions of future events*	$\Pr(\text{day } t{+}1: \text{sunny}) = 0.4$ $\Pr(\text{day } t{+}1: \text{cloudy}) = 0.2$ $\Pr(\text{day } t{+}1: \text{rainy}, \text{day } t{+}2: \text{rainy}) = 0.1$ or its linear transformation	***n*: model rank** Initial state vector: $[b_*]_{n\times 1}$ Update operators: $[B_{\text{sunny}}]_{n\times n}$ (one for each observation) Normalization vector: $[b_\infty]_{n\times 1}$	Pr(sunny, cloudy, sunny, rainy) = $b_\infty^\top B_{\text{rainy}} B_{\text{sunny}} B_{\text{cloudy}} B_{\text{sunny}} b_*$	Spectral Learning • Closed-form and fast • Consistent when "*full-rank*", i.e., model rank = system rank • No optimization

Main Idea and the Algorithm

Practical Limitations of Spectral Learning for PSRs

Real systems are complex ⇨ models are always *low-rank*

- can fail badly [1]
- bounded error guarantee requires infinite computation [2]

Our Solution: Refine spectrally-learned PSRs by optimizing data likelihood.

Naive Attempt Given a dataset D, minimize over model $\mathcal{B}$

$$loss(\mathcal{B}) = -\sum_{x\in D} \log Pr_{\mathcal{B}}(x), \quad \text{where } Pr_{\mathcal{B}}(x) = b_\infty^\top B_{x_{|x|}} \cdots B_{x_1} b_*$$

Issue minimizer: $\mathcal{B} \to \infty$
Reason: no model validity constraints

Our Solution: *Rectify and normalize the predictions.*

New Objective $$loss(\mathcal{B}) = -\sum_{x\in D} \log \frac{|Pr_{\mathcal{B}}(x))|}{\sum_{|y|=|x|} |Pr_{\mathcal{B}}(y))|}$$

Issue normalizing constant is hard to deal with.

Our Solution: *Contrastive Divergence.*[3]

Stochastic gradient is

$$-\nabla \log \left|Pr_{\mathcal{B}}(x)\right| + \nabla \log \left|Pr_{\mathcal{B}}(y)\right| \quad \text{where } x \sim D,\ y \sim \mathcal{B}\,.$$

Issue sampling y still needs the normalizing constant!

Solution Monte-Carlo Markov Chain.
Our design 1-round Gibbs-sampling starting from x:

a random position → $x_1 \ldots x_{i-1} x_i x_{i+1} \ldots x_{|x|}$ ← replace with $o \sim p$

$$p(o) = \frac{\left|Pr_{\mathcal{B}}(x_1 \ldots x_{i-1}\, o\, x_{i+1} \ldots x_{|x|})\right|}{\sum_{o'} \left|Pr_{\mathcal{B}}(x_1 \ldots x_{i-1}\, o'\, x_{i+1} \ldots x_{|x|})\right|}$$

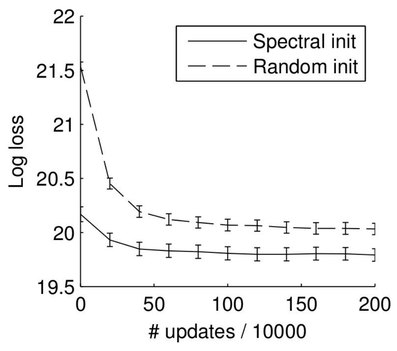
Algorithm Standard stochastic gradient descent with gradients approximated via Contrastive Divergence, plus momentum acceleration.

Experiments

Synthetic HMMs

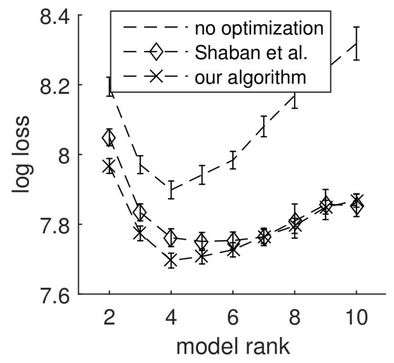
State transition follows ring topology.

Random parameters.



Comparing spectral vs random initialization for our algorithm

- 100 states, 10 observations.
- Asymptotic gain of spectral over random initialization.



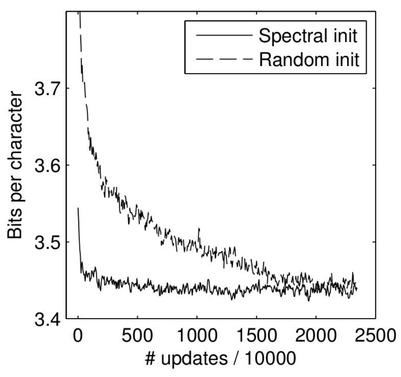
Comparing to a recent approach to refining spectrally-learned HMMs[4]

- 10 states, 20 observations.
- Our algorithm has gain in the low rank region.

Wikipedia Data

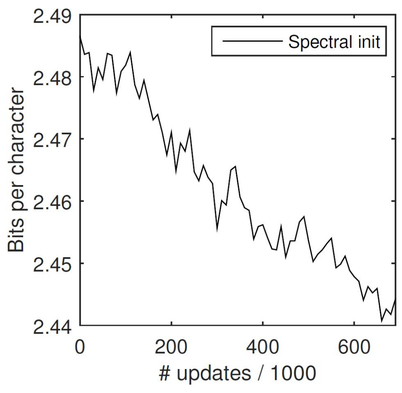
~1G Wikipedia text (articles randomly concatenated)

Character-level modelling (86 chars).



Small model (rank 20) experiment

- Spectral initialization converges much earlier than random initialization.



Large model (rank 1000) experiment

- Random initialization starts with bpc>6 and flattens around bpc=4.
- Spectral init keeps improving.

Reference

[1] Kulesza, A.; Rao, N. R.; and Singh, S. Low-Rank Spectral Learning. AISTATS 2014.
[2] Kulesza, A.; Jiang, N.; and Singh, S. Low-Rank Spectral Learning with Weighted Loss Functions. AISTATS 2015.
[3] Hinton, G. Training Products of Experts by Minimizing Contrastive Divergence. Neural computation 2002.
[4] Shaban, A.; Farajtabar, M.; Xie, B.; Song, L.; and Boots, B. Learning Latent Variable Models by Improving Spectral Solutions with Exterior Point Methods. UAI 2015.