Spectral learning for structured partially observable environments

Lucas Langer
lucas.langer@mail.mcgill.ca

1. PROBLEM AND MOTIVATION

We consider the problem of learning models of time series data in partially observable environments. Typical applications arise in robotics and reinforcement learning with HMMs and POMDPs being the models of choice. We take interest in environments with structured observations. Standard learning algorithms are not designed to exploit patterns which arise in many practical applications. As a result, we focus on extending a current learning algorithm to exploit such structure. Our approach yields both better predictive accuracy and computational performance when learning smaller models as one does in practice.

2. BACKGROUND AND RELATED WORK

Predictive state representations (PSR) are used as a model for computing a probability distribution over observations in a dynamical system [4]. There exists a well known spectral algorithm which learns a PSR from empirical data [3]. The algorithm makes use of Hankel matrices and singular value decomposition. One can control the number of states in the PSR by only including states with high singular values. The reason for using fewer states is twofold. First, noise in empirical data artificially creates extra states with low singular values. Secondly, reducing the number of states is necessary in practice for computational performance.

Learning of PSRs began with work on non-spectral methods [5]. Spectral algorithms emerged later and became of particular interest because they delivered theoretical guarantees far better than other methods. [3]. On the applied side, spectral learning of PSRs has shown promise in planning with timing information [1] and in natural language processing for dependency parsing [2].

3. APPROACH AND UNIQUENESS

In our work, we extend the standard PSR learning algorithm by developing a new machinery for performing queries which we call the Base System. The main idea in the Base System is to include transition operators for sequences of observations in addition to those for single observations. We first apply the Base System to predict the time spent by an agent in a stochastic environment. We then progress to systems with multiple observations. Finally, we develop two heuristics: one for choosing operators from data, and another for applying operators in an effective order. The former uses an iterative greedy algorithm, while the latter uses a dynamic programming algorithm.

4. RESULTS AND CONTRIBUTIONS

In the experiments that follow, we produce observations by simulating robot motion in stochastic labyrinth environments. The robot explores the labyrinths until it leaves through one of the doors. We compare PSRs learned with the generic algorithm to PSRs learned with different degrees of the Base System. To measure the performance of a PSR, we compare predictions to the actual probability distribution over observations.

4.1 Double Loops

In the first experiment we look at the time spent in double loop labyrinths.

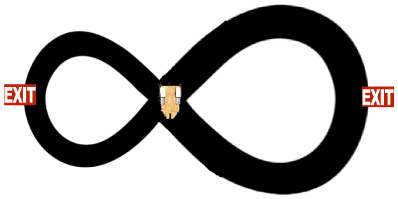

Figure 1: Double Loop Environment

The PSR with the Base System significantly lower errors across all model sizes (Figures 2 and 3). In particular, we note that noise in the durations of loops doesn't harm the performance of the Base System.

4.2 PacMan Labyrinth

In the second experiment, we look at timing for a PacMan-Type labyrinth. In addition, we use state weights from the learned PSRs to predict distances between the robot and objects in the environment.

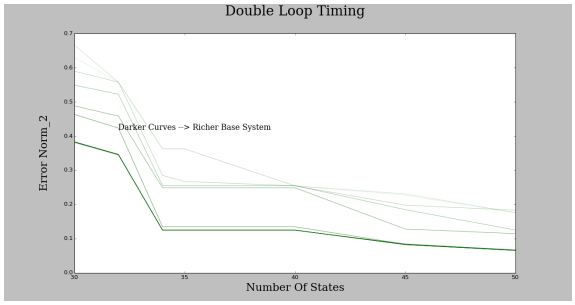

Figure 2: No Noise in Loops

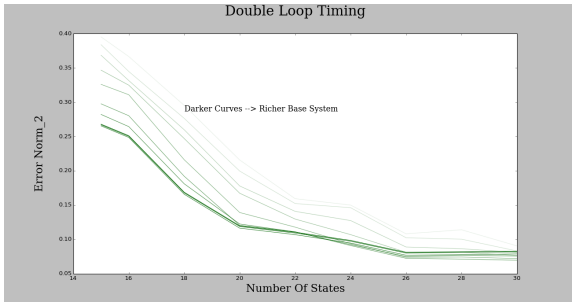

Figure 3: Noise in Loops

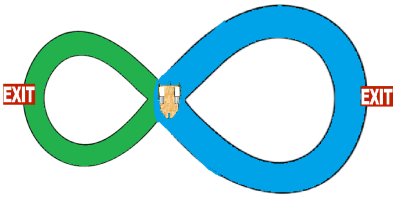

Figure 4: Multiple Observation Environment

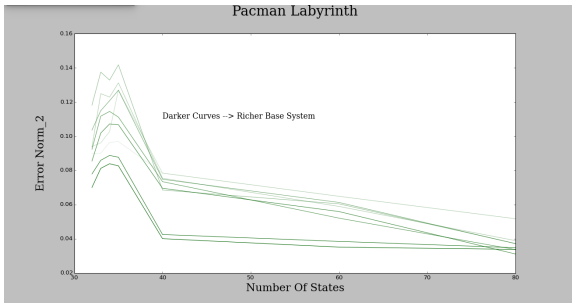

Figure 5: Timing Predictions in Pacman

The Base System outperforms the naive PSR for the Pacman environment (Figure 5). It also does better for predicting distances (Figure 6).

4.3 Multiple Observations

Next, we change our set of observations to wall colors of the labyrinth.

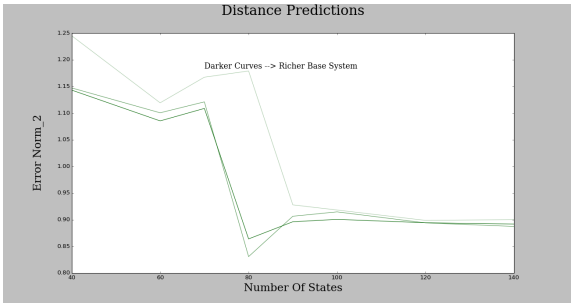

Figure 6: Distance Predictons

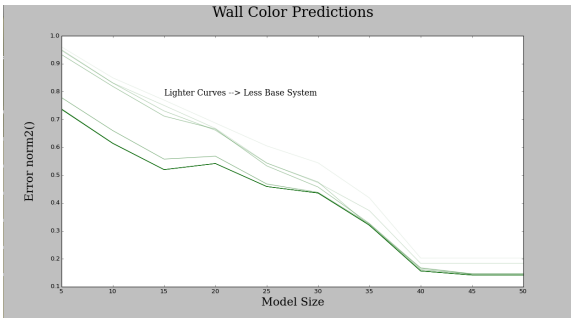

Figure 7: Predicting Wall Colors

The Base System also does better for loops with multiple observations (Figure 7).

5. CONCLUSION AND FUTURE WORK

In this work, we showed a way to significantly improve performance of truncated models in applied environments. For future work, we leave a theoretical analysis of the Base System and further optimization of its construction .

6. REFERENCES

[1] P.-L. Bacon, B. Balle, and D. Precup. Learning and planning with timing information in markov decision processes. *2nd Multidisciplinary Conference on Reinforcement Learning and Decision Making*, 2015.

[2] B. Balle, X. Carreras, F. M. Luque, and A. Quattoni. Spectral learning of weighted automata. *Machine Learning*, pages 1–31, 2013.

[3] B. Boots, S. M. Siddiqi, and G. J. Gordon. Closing the learning-planning loop with predictive state representations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 1369–1370. International Foundation for Autonomous Agents and, 2010.

[4] M. L. Littman, R. S. Sutton, and S. P. Singh. Predictive representations of state. *NIPS*, 14:1555–1561, 2001.

[5] E. Wiewiora. Learning predictive representations from a history. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, pages 964–971, 2005.