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 compressed models as one does in practice.

2. BACKGROUND AND RELATED WORK

Predictive state representations 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 a 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 less 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 the work of [5] who used non-spectral methods. Spectral algorithms emerged later and became of 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 timing applications where choosing additional operators is easiest to do. We then progress to the general case of systems with multiple observations and develop a heuristic for choosing effective operators from data.

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. Alternatively, one could use cross-validation.

4.1 Double Loops

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

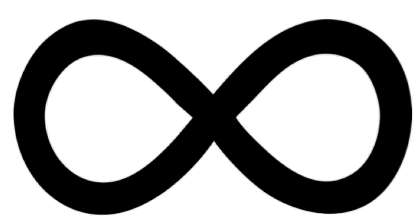

Figure 1: Double Loop Environment

In both cases, the PSR with the Base System has 100 % less error than without. 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 weightings from the learned PSRs to predict distances between the robot and objects in the environment.

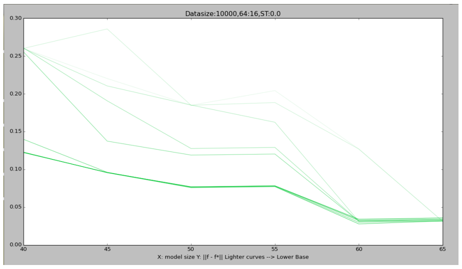

Figure 2: No noise

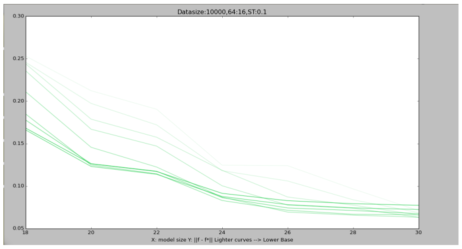

Figure 3: Corridor noise

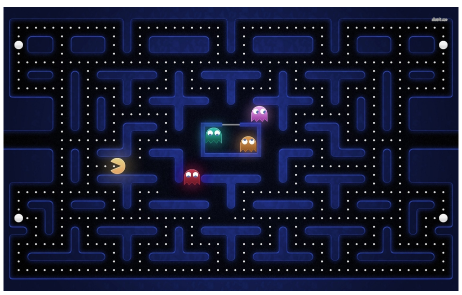

Figure 4: Pacman Environment

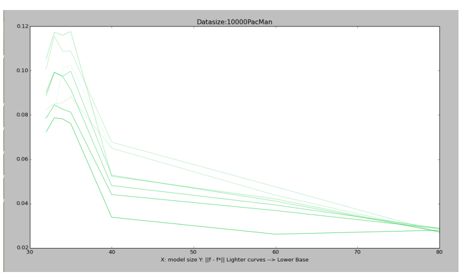

Figure 5: Timing predictions in Pacman

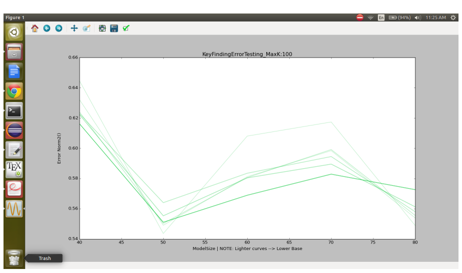

Figure 6: Distance predictons from key

Here, the Base System outperforms the naive by 100% for timing and 45% for distances.

4.3 Multiple Observations

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

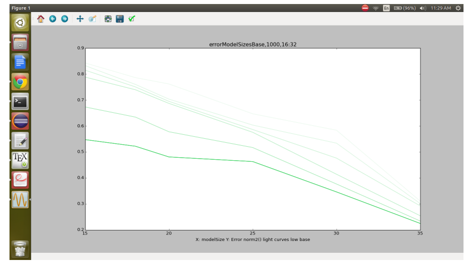

Figure 7: Predicting wall colors

Here, the Base System outperforms that naive approach by 55%. For this environment, we construct the Base System separately for each symbol. In general, one might want to use a custom heuristic to optimize the construction.

5. RELEVANCE AND FUTURE WORK

The spectral framework for learning in partially observable environments has better theoretical guarantees [**?**] than non-spectral methods. In this work, we showed a way to significantly improve results when one wants a smaller model of the environment as often occurs in practice. In future work, we hope to see a theoretical analysis of the apparent improvement of the Base System and further optimization of it's 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.

6.1 References

Generated by bibtex from your .bib file. Run latex, then bibtex, then latex twice (to resolve references) to create the .bbl file. Insert that .bbl file into the .tex source file and comment out the command `\thebibliography`.