Learning PSRs with The Base System

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, where hidden markov models (HMMs) are often the model of choice. We take particular interest in the class of partially observable systems which are compressible, namely systems in which one can achieve good performance with a smaller model. There exists a well known spectral algorithm for learning a Predictive State Representation of the environment from empirical data [NAME]. Traditionally, truncation of this learned PSR has been performed naively. We provide an extension to [NAME]'s algorithm for improved learning of smaller models. In empirical tests, our approach strongly outperforms it's ancestor in both in predictive and computational performance.

2. BACKGROUND AND RELATED WORK

Predictive state representations are used as a model for predicting the probability of observations in a dynamical system. [NAME] gives an algorithm which makes use of Hankel matrices and a singular value decomposition to obtain a PSR from data. One can control the number of states in the PSR by including states with highest singular values. The reason for using less states is twofold. First, noise in empirical data artificial creates states with low singular values. Secondly, reducing the number of states is necessary in practice for computational performance.

Previous work with PSR learning has included planning [pierre luc] and natural language processing [other source]. Our work focuses on improving the learning of PSRs for general applications.

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. This extra machinery allows us to richly express state transitions with truncated models and provides better experimental performance. We first apply this system to timing applications where the construction of the Base System is easiest to standardize. We then progress to the general case of systems with multiple observations and develop a heuristic for constructing the Base System effectively from data.

4. RESULTS AND CONTRIBUTIONS

4.1 Preliminaries

In the experiments that follow, we produce observations by simulating robot motion in labyrinth environments. We compare PSRs which use the base system to different degrees. To measure performance, the predictions of each PSR are compared to the actual probability distribution of the observations.

4.2 Double Loops

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

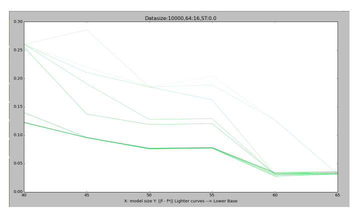

Figure 1: No noise

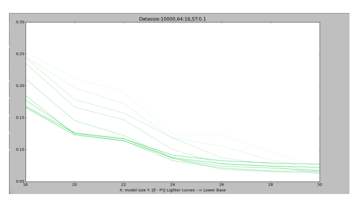

Figure 2: Corridor noise

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.

4.3 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.

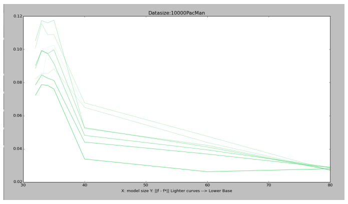

Figure 3: Timing predictions in Pacman

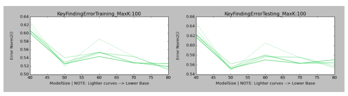

Figure 4: Distance predictons from key

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

4.4 Multiple Observations

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

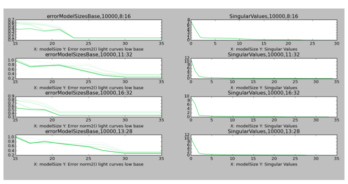

Figure 5: Predicting wall colors

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 heuristic to optimize the construction of the base.

4.5 Relevance

The spectral framework for learning in partially observable environments has better theoretical guarantees [REFERENCE] than non-spectral methods. In this work, we showed a way to significantly improve results in practical settings, that is when one wants a smaller model. In future work, we hope to see a theoretical analysis of the apparent improvement and optimization in how one constructs the Base System for the multiple observation case.

APPENDIX

A. HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. In the **appendix** environment, the command **section** is used to indicate the start of each Appendix, with alphabetic order designation (i.e. the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure *within* an Appendix, start with **subsection** as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

A.1 Introduction

A.2 The Body of the Paper

A.2.1 Type Changes and Special Characters

A.2.2 Math Equations

Inline (In-text) Equations

Display Equations

A.2.3 Citations

A.2.4 Tables

A.2.5 Figures

A.2.6 Theorem-like Constructs

A Caveat for the TEX Expert

A.3 Conclusions

A.4 Acknowledgments

A.5 Additional Authors

This section is inserted by LATEX; you do not insert it. You just add the names and information in the `\additionalauthors` command at the start of the document.

A.6 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`.

B. MORE HELP FOR THE HARDY

The acm_proc_article-sp document class file itself is chock-full of succinct and helpful comments. If you consider yourself a moderately experienced to expert user of LATEX, you may find reading it useful but please remember not to change it.