Modelling Sparse Dynamical Systems with Compressed Predictive State Representations

William L. Hamilton WHAMIL3@CS.MCGILL.CA
Reasoning and Learning Laboratory, School of Computer Science, McGill University, Montreal, QC, Canada

Mahdi Milani Fard MMILAN1@CS.MCGILL.CA
Reasoning and Learning Laboratory, School of Computer Science, McGill University, Montreal, QC, Canada

Joelle Pineau JPINEAU@CS.MCGILL.CA
Reasoning and Learning Laboratory, School of Computer Science, McGill University, Montreal, QC, Canada

Abstract

Efficiently learning accurate models of dynamical systems is of central importance for developing rational agents that can succeed in a wide range of challenging domains. The difficulty of this learning problem is particularly acute in settings with large observation spaces and partial observability. We present a new algorithm, called Compressed Predictive State Representation (CPSR), for learning models of high-dimensional partially observable uncontrolled dynamical systems from small sample sets. The algorithm exploits a particular sparse structure present in many domains. This sparse structure is used to compress information during learning, allowing for an increase in both the efficiency and predictive power. The compression technique also relieves the burden of domain specific feature selection. We present empirical results showing that the algorithm is able to build accurate models more efficiently than its uncompressed counterparts, and we provide theoretical results on the accuracy of the learned compressed model.

1. Introduction

Learning accurate models of complex domains is a fundamental problem for developing rational autonomous agents. In the case of dynamical systems, the learned models can be used for a variety of tasks, such as tracking, prediction, resource allocation, planning, and much more. The problem of learning models of dynamical systems is particularly difficult in settings with large observation spaces and partial observability.

Predictive state representations (PSRs) offer one model for discrete time finite action and observation systems, which represent system states as predictions about future events. Unlike other popular frameworks for modelling dynamic systems, such as *Hidden Markov Models* (HMMs) for uncontrolled systems (Rabiner, 1990) and *Partially Observable Markov Decision Processes* (POMDPs) (Kaelbling et al., 1998) for controlled systems, PSRs do not rely on hidden or latent states. Instead, PSR models are rooted directly in observable quantities. This allows PSRs to be constructed without expectation maximization (EM) style algorithms, and thus allows for the efficient construction of globally optimal models. Furthermore, PSRs, which are closely related to certain spectral learning algorithms for HMM's (Hsu et al., 2008), are a more general model of stochastic dynamic systems that contain the latent state model as a special case (Singh et al., 2004).

Despite the fact that there are many theoretical results demonstrating the rich representational capacity of PSRs (Singh et al., 2004; Wiewiora, 2007), there is a lack of efficient algorithms for learning these representations. A number of promising algorithms based on spectral learning have been proposed recently (Boots et al., 2009; Boots & Gordon, 2011). These have been shown to perform well on some challenging tasks. However, they are still somewhat limited in large observation spaces, since they typically consider a combinatorial number of observation sequences. This can be avoided by assuming a functional representation of the

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

observation space, but this requires additional knowledge from a human designer.

The goal of this paper is to provide a sound agnostic general purpose algorithm for learning PSRs from large-dimensional batch data. The algorithm exploits a particular sparse structure that is present in many domains. This sparse structure is used to compress information using random projections, thus achieving efficient learning without requiring explicit structural information about the domain. The primary benefits of our approach are that it is algorithmically simple, it is applicable to a large variety of domains, and it provides guarantees on the fidelity of the learned compressed model.

The current paper focuses exclusively on uncontrolled systems (i.e. observation-only systems, no actions), as our theoretical results pertain directly to the quality of the learned model in that case. However, our algorithm is easily extended to include actions using techniques already developed for other PSR representations (e.g. TPSRs). We present empirical results showing the good performance of our approach on two contrasting domains.

2. Technical Background

This section outlines a number of useful technical concepts related to our approach.

2.1. Predictive State Representations

Predictive State Representations (PSRs) (Singh et al., 2004) are designed to represent dynamical systems directly with observable events. The dynamics are captured through probability distributions over *tests* (sequences of possible future observations, denoted τ) conditioned on *histories* (sequences of past observations, denoted h). Formally, a PSR in an uncontrolled partially observable system can be defined by four elements $\{\mathcal{O}, Q, \mathbf{m}_0, F\}$, where $\mathcal{O}$ is a set of observations, Q is a set of *sufficient (core) tests*, $\mathbf{m}_0$ is a vector of prior probabilities (i.e. prediction vector conditioned on an empty history), and F is a function relating tests in Q to all possible tests.

The primary goal of a PSR is to maintain probabilities of the form: $P(\tau_i|h_j) = P(o^1_{t+1}o^2_{t+2}...o^3_{t+3}|o^4_1 o^5_2...o^6_t)$ for all i, j (superscripts label different observations and subscripts denote time). If we knew $P(\tau_i|h_j)\,\forall i \forall j$ (i.e. all possible tests conditioned on all possible histories) then trivially we would have all the information necessary to characterize a system. Of course, obtaining estimates for all possible events (i.e. tests/histories of arbitrary length) is not feasible. This is where the set Q of *sufficient tests* comes into play.

We say a set of tests Q is sufficient if and only if we can form a *prediction vector* of these tests $\mathbf{p}(Q|h) = [P(\tau_1|h), P(\tau_2|h), ...P(\tau_{|Q|}, |h)]^T$ such that for any history and for all tests there is a projection function f_τ such that $P(\tau|h) = f_\tau(\mathbf{p}(Q|h))$. That is, the set Q is sufficient if and only if all other tests can be represented by some function of the tests in Q. Throughout this paper (as with much of the previous work on PSRs), we restrict our attention to *linear* projection functions. Thus the function f_τ can be expressed as a vector $\mathbf{m}_\tau$, and the expression above can be rewritten as:

$$P(\tau|h) = \mathbf{m}_\tau^T \mathbf{p}(Q|h). \tag{1}$$

Once we have these projection functions, we need only maintain $\mathbf{p}(Q|h)$ for one particular history (i.e. the current history) at each time step. Thus, using time-dependent notation, we need to maintain only a single prediction vector $\mathbf{m}_t = \mathbf{p}(Q|h)$.

We then define $\mathbf{M}_{o^l}$ to be a matrix with $\mathbf{m}_{o^l\tau_i}\ \forall \tau \in Q$ as rows, where $\mathbf{m}_{o^l\tau_i}$ is defined by $P(\tau o^l|h) = \mathbf{m}_{o^l\tau_i}\mathbf{m}_t$. In other words, $\mathbf{M}_{o^l}$ contains the projection functions for all tests that consist of the observation o^l appended to a core test $\tau_i \in Q$. Using this matrix, the prediction vector can be recursively updated after seeing observation o^l by:

$$\mathbf{m}_{t+1} = \frac{\mathbf{M}_{o^l}\mathbf{m}_t}{\mathbf{m}_\infty \mathbf{M}_{o^l}\mathbf{m}_t}, \tag{2}$$

where the normalizing factor $\mathbf{m}_\infty$ satisfies $\mathbf{m}_\infty \mathbf{m}_t = 1\,\forall t$.

The above update equation also reveals the full set of parameters which belong to F in our initial definition of PSR's. These parameters consist of the following: $|O|$ matrices $\mathbf{M}_{o^l}$ of size $|Q| \times |Q|$ (one for each observation) and the $|Q| \times 1$ normalizing factor $\mathbf{m}_\infty$.

These learned parameters, along with the prediction vector $\mathbf{m}_t$, provide a complete and sufficient model of a system. For any test $\tau_i \notin Q$, where τ_i consists of some observation sequence, for example $o^1_{t+1}, o^2_{t+2}...o^n_T$, we can compute $\mathbf{m}_{\tau_i}$ with:

$$\mathbf{m}_{\tau_i} = \mathbf{m}_\infty \mathbf{M}_{o^n_T}\mathbf{M}_{o^{n-1}_{T-1}} \cdots \mathbf{M}_{o^1_{t+1}}. \tag{3}$$

The PSR model can also be used to produce T-step predictions (i.e. the probability $P(o^l_{t+T}|h_t)$ of seeing an observation o^l T-steps in the future) by:

$$P(o^j_{t+T}|h_t) = \mathbf{c}_\infty \mathbf{M}_{o^l}(\mathbf{M}_\star)^{T-1}\mathbf{m}_t, \tag{4}$$

where $\mathbf{M}_\star = \sum_{o^i \in \mathcal{O}} \mathbf{M}_{o^l}$ is a matrix that can be computed once and stored as a parameter for quick computation (Wiewiora, 2007).

2.2. The TPSR Learning Algorithm

Early results showed that a minimal PSR model of an uncontrolled partially observable system will be at least as concise as a corresponding HMM (Singh et al., 2004). However, they did not provide an efficient method for learning such a representation. Currently, the most successful learning method for PSRs, termed transformed PSRs (TPSRs), is based on spectral learning algorithms (Rosencrantz et al., 2004; Boots et al., 2009). In this approach, a large set of tests is chosen (large enough so that it almost certainly contains a sufficient set as a subset), and empirical estimates are obtained for these tests. These estimates are then grouped into matrices, and the size of these matrices are reduced using spectral projection methods. The TPSR parameters can then be learned using regression and these reduced estimates.

Formally, one defines two *observable matrices* $\mathcal{P}_{\mathcal{T},\mathcal{H}}$, $\mathcal{P}_{\mathcal{H}}$, and $|\mathcal{O}|$ *observable matrices* $\mathcal{P}_{\mathcal{T},o^l,\mathcal{H}}$ (one for each observation). $\mathcal{P}_{\mathcal{T},\mathcal{H}}$ is a $|\mathcal{T}|\times|\mathcal{H}|$ matrix which contains the joint probabilities of all specified tests and possible histories. $\mathcal{P}_{\mathcal{H}}$ is a $|\mathcal{H}| \times 1$ vector containing the marginal probabilities of each possible history. The $\mathcal{P}_{\mathcal{T},o^l,\mathcal{H}}$ matrices are also of size $|\mathcal{T}| \times |\mathcal{H}|$ and contain the joint probabilities of observing each history, followed by a particular observation (corresponding to that matrix) and a test. The transpose of the thin SVD of $\mathcal{P}_{\mathcal{T},\mathcal{H}}$, denoted U^T, is then used to reduce the dimension of the observable matrices by left multiplying each observable matrix, excluding $\mathcal{P}_{\mathcal{H}}$, by U^T with some of the least significant singular vectors removed. The use of SVD also allows for the tuning of the dimension of the representation by removing least significant vectors from U.

The TPSR approach can also be extended to work with features of tests and histories (Boots et al., 2009; Boots & Gordon, 2011). This is useful in cases where the observation space is too complex for standard tests to be used. When features of tests and histories are used, however, they are specified in a domain-specific manner, such as through kernel methods in continuous domains (Boots et al., 2009). Some authors have also used randomized Fourier methods to efficiently approximate kernel-based feature selection (Boots & Gordon, 2011). These methods are quite successful in continuous domains. However, they still require domain-specific specification, and in some cases they also require an extremely large number of features in order to obtain high prediction quality (Boots & Gordon, 2011). The benefit of the algorithm presented here is that it implicitly performs general purpose feature selection of tests using random compression.

2.3. Compressed Estimation

Despite the fact that spectral algorithms can specify a small dimension for a transformed space, there are still a number of computational limitations. They require that the $|\mathcal{T}| \times |\mathcal{H}|$ matrix $\mathcal{P}_{\mathcal{T},\mathcal{H}}$ be estimated in its entirety, and that the $\mathcal{P}_{\mathcal{T},o^l,\mathcal{H}}$ matrices be partially estimated as well. In many domains, such as the *PocMan* (Silver & Veness, 2010) domain described below, these observable matrices can become far too large and cannot be manipulated directly. More importantly, these algorithms also require that spectral decomposition be performed on this large $\mathcal{P}_{\mathcal{T},\mathcal{H}}$ matrix, which can be prohibitively expensive. In order to circumvent these computational constraints, the CPSR algorithm we propose (in the next section) performs *compressed estimation.*

This method is borrowed from the field of compressed sensing and works by projecting matrices down to spaces determined via randomly generated bases. More formally, a $m \times n$ matrix $\mathbf{Y}$ is compressed to a $d \times n$ matrix $\mathbf{X}$ (where $d < m$) by:

$$\mathbf{X} = \Phi\mathbf{Y}, \tag{5}$$

where Φ is a $d \times m$ projection matrix composed of entries drawn from the gaussian distribution $\mathcal{N}(0, 1/d)$ (Baraniuk & Wakin, 2009).

The fidelity of this technique depends only on what is called the *sparsity* of the matrix $\mathbf{Y}$. Sparsity in this context refers to the maximum number of non-zero entries which occur in any column of $\mathbf{Y}$. Formally, if we denote a column vector of $\mathbf{Y}$ by $\mathbf{y}_i$, we say that a matrix is k-sparse if:

$$k \geq ||\mathbf{y}_i||_0 \forall \mathbf{y}_i \in \mathbf{Y}, \tag{6}$$

where $||\cdot||_0$ denotes Donoho's zero "norm."

This method of compression has a number of useful properties. It is computationally efficient, as it requires only the construction of the matrix $\mathbf{X}$. The matrix multiplication above can, in fact, be avoided, and one can work in the compressed space directly. Furthermore, as we discuss later in the paper, the fidelity of the compression and the size of the compressed representation have only a logarithmic dependency on the original dimension (m), allowing for massive compression in some cases.

This compression technique is very well suited for application to PSRs. Informally, the sparsity condition is the requirement that for every history h_i, only a subset of all tests have non-zero probabilities (a more formal definition appears in the theory section below). This seems realistic in many domains. For example, in the PocMan domain described below, we empirically found the average column sparsity of the matrices to be roughly 0.018% (i.e. approximately 0.018% of entries in a column were non-zero).

3. The Compressed PSR Algorithm

Given the technical background outlined above, the compressed PSR (CPSR) algorithm is relatively straight-forward. It first constructs the random matrix Φ. It then directly computes empirical estimates of the compressed matrices $\Phi\mathcal{P}_{\mathcal{T},\mathcal{H}}$ and $\Phi\mathcal{P}_{\mathcal{T},o^l,\mathcal{H}}$ from the batch of data, as well as the marginal probability of histories, $\mathcal{P}_{\mathcal{H}}$.

The $\Phi\hat{\mathcal{P}}_{\mathcal{T},\mathcal{H}}$ and $\Phi\hat{\mathcal{P}}_{\mathcal{T},o^l,\mathcal{H}}$ estimates are then used to construct $\mathbf{C}_{o^l}$ matrices for all $o^l \in \mathcal{O}$, using linear regression. Intuitively, these $\mathbf{C}_{o^l}$ matrices are linear operators that encode, at any particular history, the joint probability of seeing that history and then the observation o^l. A c_∞ vector is constructed in a similar manner but with $\hat{\mathcal{P}}_{\mathcal{H}}$ replacing the $\Phi\hat{\mathcal{P}}_{\mathcal{T},o^l,\mathcal{H}}$ estimates. This vector functions as a normalizer and is used to convert the joint probabilities computed with the $\mathbf{C}_{o^l}$ matrices to conditional ones. That is, $\mathbf{c}_\infty$ and a $\mathbf{C}_{o^l}$ matrix are used together to compute the conditional probability of o^l given a history. Lastly, a $\mathbf{c_0}$ is constructed from the first column of $\Phi\hat{\mathcal{P}}_{\mathcal{T},\mathcal{H}}$ and simply defines an initial probability distribution over tests. It is used as the initial prediction vector in the CPSR model of the system. The $\mathbf{C}_{o^l}$ matrices, the $\mathbf{c}_\infty$ normalizer, and the initial prediction vector $\mathbf{c_0}$ form a sufficient model and can be used with equations (1)-(4) to make predictions and track through a system.

```
CPSR Algorithm
  Inputs: $\mathcal{T}$: set of tests , $\mathcal{H}$: set of indicative events, $\mathcal{W}$: a diverse sample of events, $d$: projection size.
  Returns: CPSR model parameterized by $\{\mathbf{c}_o, \mathbf{c}_\infty, \mathbf{C}_{o^l}\ \forall o^l \in \mathcal{O}\}$

1: Sample $\Phi^{d\times|\mathcal{T}|}$ with i.i.d. entries from $\mathcal{N}(0, 1/d)$.
2: Obtain empirical estimates for $\mathcal{P}_{\mathcal{H}}$, $\Phi\mathcal{P}_{\mathcal{T},\mathcal{H}}$, and $\Phi\mathcal{P}_{\mathcal{T},o^l,\mathcal{H}}\ \forall o^l \in \mathcal{O}$ from sample events $\mathcal{W}$ as follows:
   (a) Parse each event $w \in \mathcal{W}$ into a $\tau_i$ and history $h \in H_j$.
       (i) For each $\tau_i$ and $H_j$ pair, add column $i$ of $\Phi$ to column $j$ of $\Phi\hat{\mathcal{P}}_{\mathcal{T},\mathcal{H}}$, and similarly for all $\Phi\hat{\mathcal{P}}_{\mathcal{T},o^l,\mathcal{H}}$ matrices.
       (ii) Increment $[\hat{\mathcal{P}}_{\mathcal{H}}]_j$.
   (b) Normalize estimates by observation counts.
3: Compute:
   (a) $\mathbf{c}_0 = \Phi\hat{\mathcal{P}}_{\mathcal{T},\emptyset}$
   (b) $\mathbf{C}_{o^l} = \Phi\hat{\mathcal{P}}_{\mathcal{T},o^l,\mathcal{H}}(\Phi\hat{\mathcal{P}}_{\mathcal{T},\mathcal{H}})^\dagger\ \forall o^l \in \mathcal{O}$
   (c) $\mathbf{c}_\infty = (\Phi\hat{\mathcal{P}}_{\mathcal{T},\mathcal{H}})^\dagger\hat{\mathcal{P}}_H$
```

The algorithm generally assumes that data is provided in a batch. It is primarily designed to work with episodic domains where all events w_i start at the same state. However, if this is not the case, one can simply replace the equation *3-a* of the CPSR algorithm with $\mathbf{c}_* = \Phi\hat{\mathcal{P}}_{\mathcal{T},\mathcal{H}}\mathbf{1}_d$. In this situation, $\mathbf{c}_*$ would no longer correspond to a start state and instead would correspond to a distribution over feasible states. Thus prediction is still possible but will be subject to greater error. This error should decrease as the model receives updates, as the updates allow the prediction vector to converge to a true model state over time.

4. Theoretical Analysis

To simplify the analysis, assume that our test set is a core test set $\mathcal{Q}$. Therefore, random projections are applied on $\hat{\mathcal{P}}_{\mathcal{Q},\mathcal{H}}$ and $\hat{\mathcal{P}}_{\mathcal{Q},o,\mathcal{H}}$ matrices[1]. Define:

$$\mathbf{B}_o = \mathcal{P}_{\mathcal{Q},o,\mathcal{H}}(\mathcal{P}_{\mathcal{Q},\mathcal{H}})^\dagger, \quad \mathbf{b}_\infty = (\mathcal{P}_{\mathcal{Q},\mathcal{H}})^\dagger\mathcal{P}_{\mathcal{H}}. \tag{7}$$

Since $\mathcal{Q}$ is a core test set, the above is a TPSR representation (Boots et al., 2009; Rosencrantz et al., 2004). Assume we have enough histories in $\mathcal{H}$ such that matrices are full rank. Defining $\mathcal{P}_{\mathcal{Q},h}$ and $\mathcal{P}_{\mathcal{Q},o,h}$ to be the vectors containing the joint probabilities of all core tests and fixed history h, we have (by linearity of PSR):

$$\forall h : \mathcal{P}_{\mathcal{Q},o,h} = \mathbf{B}_o\mathcal{P}_{\mathcal{Q},h}, \quad \mathcal{P}_h = \mathbf{b}_\infty^T\mathcal{P}_{\mathcal{Q},h}. \tag{8}$$

One can thus think of finding the $\mathbf{B}_o$ and $\mathbf{b}_\infty$ parameters as regression problems, having the estimates of $\mathcal{P}_{\mathcal{Q},h}$'s as noisy input features. We also have noisy observations of the outputs $\mathcal{P}_{\mathcal{Q},o,h}$ and $\mathcal{P}_h$. Since the sample set is noisy both on the input and output values, direct regression in the original space might result in large estimation error. Therefore, we apply random projections to reduce the estimation error (variance) at the cost of a controlled approximation error (bias). Working in the compressed space also helps with the computation complexity of the algorithm.

Note that there is an inherent difference between our work and the TPSR framework. In TPSR, one seeks to find concise linear transformations of the observation matrices, whereas CPSR seeks to find good approximations in a compressed space (which cannot be linearly transformed to the original model). The following sections provide an analysis of the error induced by this compression and how the error prop-

[1]Note that projections from over-complete test sets with rank bigger than $|\mathcal{Q}|$ down to d dimensions can be achieved by first projecting to size $|\mathcal{Q}|$ and then projecting from $|\mathcal{Q}|$ to d. The combination of two random projection should be similar to one random projection but needs more detailed and complicated analysis.

agates through the application of several compressed operators.

4.1. Error of One Step Regression

There are several bounds on the excess risk of regression in compressed spaces (Maillard & Munos, 2009; Maillard et al., 2012; Fard et al., 2012). In this work, we assume the existence of a generic upper bound for the least squares regression. Assume we have a target function $f(\mathbf{x}) = \mathbf{x}^T w$ where $\mathbf{x}$ is in a k-sparse D-dimensional space. We observe an i.i.d. sample set $\{(\mathbf{x}_i, f(\mathbf{x}_i) + \eta_i)\}_{i=1}^n$, where η_i's are independent zero-mean noise terms for which the maximum variance is bounded by σ_η^2, and $\mathbf{x}_i$'s are sampled from distribution ρ. Let $\hat{f}_d(\mathbf{x})$ be the compressed least squares solution on this sample with a random projection of size d. We assume the existence of a generic upper bound function ϵ, such that with probability no less than $1-\delta$:

$$\|f(\mathbf{x}) - \hat{f}_d(\mathbf{x})\|_{\rho(\mathbf{x})} \leq \epsilon(n, D, d, \|\mathbf{w}\|\|\mathbf{x}\|_{\rho(\mathbf{x})}, \sigma^2, \delta), \quad (9)$$

where $\|g(\mathbf{x})\|_{\rho(\mathbf{x})} = \sqrt{\mathbb{E}_{\mathbf{x}\sim\rho}(g(\mathbf{x}))^2}$ is the weighted L^2 norm under the sampling distribution.

We make the following sparsity assumptions. For all h, $\mathcal{P}_{\mathcal{Q},h}$ and $\mathcal{P}_{\mathcal{Q},o,h}$ are k-sparse. Assuming that the empirical estimates of zero elements in these vectors are not noisy, for $\Delta_x = \hat{\mathcal{P}}_{\mathcal{Q},h} - \mathcal{P}_{\mathcal{Q},h}$ we have that Δ_x is k-sparse (similar argument for $\Delta_y = \hat{\mathcal{P}}_{\mathcal{Q},o,h} - \mathcal{P}_{\mathcal{Q},o,h}$). Finally, we assume that for all observations, $\mathbf{B}_o\Delta_x$ and $\mathbf{B}_o\Delta_y$ are k'-sparse.

In order to simplify the analysis, in this section we define our $\mathbf{C}_o$ matrices to be slightly different from the ones used in the described algorithm[2]. The results should only change very slightly if we use the original definitions.

Let A_i be the ith row of matrix A, and A_{-i} be matrix A with ith row removed. We have the following:

Theorem 1. *Let $\mathcal{H}$ be a large collection of sampled histories according to ρ, and let Φ be a random projection as described before. We observe noisy estimate $\hat{\mathcal{P}}_{\mathcal{Q},h} = \mathcal{P}_{\mathcal{Q},h} + \Delta_x$ of input and $\hat{\mathcal{P}}_{\mathcal{Q},o,h} = \mathcal{P}_{\mathcal{Q},o,h} + \Delta_y$ of the output, where elements of Δ_x and Δ_y are independent zero-mean random variables with maximum variance σ_x^2 and σ_y^2 respectively. For $1 \leq i \leq d$, define:*

$$\mathbf{u}_i = \Phi_i \hat{\mathcal{P}}_{\mathcal{Q},o,\mathcal{H}} (\Phi_{-i} \hat{\mathcal{P}}_{\mathcal{Q},\mathcal{H}})^\dagger.$$

Define $\mathbf{C}_o$ to be a $d \times d$ matrix such that:

$$(\mathbf{C}_o)_i = [\mathbf{u}_{i,1}, \mathbf{u}_{i,2}, \ldots, \mathbf{u}_{i,i-1}, 0, \mathbf{u}_{i,i}, \mathbf{u}_{i,i+1}, \ldots, \mathbf{u}_{i,d-1}].$$

[2]We do not use the ith feature for the ith regression to avoid dependence between the projection and the target weights.

Then with probability no less than $1-\delta$ we have:

$$\|\mathbf{C}_o(\Phi\mathcal{P}_{\mathcal{Q},h}) - \Phi\mathcal{P}_{\mathcal{Q},o,h}\|_{\rho(\mathbf{x})} \leq \sqrt{d}\epsilon(|\mathcal{H}|, |\mathcal{Q}|, d, L_o, \sigma_o^2, \delta/d), \quad (10)$$

where we define $\sigma_o^2 = \max_{i,j}(\Phi_{ij})^2(k\sigma_y^2 + k'\|\mathbf{B}_o\|\sigma_x^2)$ and $L_o = \max_i \|\Phi_i \mathbf{B}_o\|\|\mathcal{P}_{\mathcal{Q},h}\|_{\rho(\mathbf{x})}$.

Proof. We have:

$$\forall h : \Phi_i \mathcal{P}_{\mathcal{Q},o,h} = (\Phi_i \mathbf{B}_o)\mathcal{P}_{\mathcal{Q},h} \quad (11)$$

Therefore we have a linear target and by definition $\mathbf{u}_i$ is the COLS estimate with projection Φ_{-i}.

First we analyze the effective noise variance in the sample output. We have:

$$\begin{aligned}\Phi_i \hat{\mathcal{P}}_{\mathcal{Q},o,h} &= \Phi_i \mathcal{P}_{\mathcal{Q},o,h} + \Phi_i \Delta_y & (12)\\ &= \Phi_i \mathbf{B}_o(\hat{\mathcal{P}}_{\mathcal{Q},h} - \Delta_x) + \Phi_i\Delta_y & (13)\\ &= \Phi_i\mathbf{B}_o\hat{\mathcal{P}}_{\mathcal{Q},h} - \Phi_i\mathbf{B}_o\Delta_x + \Phi_i\Delta_y. & (14)\end{aligned}$$

And thus the $(\hat{\mathcal{P}}_{\mathcal{Q},h}, \Phi_i\hat{\mathcal{P}}_{\mathcal{Q},o,h})$ is the same as:

$$(\hat{\mathcal{P}}_{\mathcal{Q},h}, \Phi_i\mathbf{B}_o\hat{\mathcal{P}}_{\mathcal{Q},h} - \Phi_i\mathbf{B}_o\Delta_x + \Phi_i\Delta_y). \quad (15)$$

Since Δ_y is k-sparse and $\mathbf{B}_o\Delta_x$ is k'-sparse, the effective variance of the noise term is bounded by:

$$\max_j (\Phi_{ij})^2(k\sigma_y^2 + k'\|\mathbf{B}_o\|\sigma_x^2). \quad (16)$$

Maximization over i gives the σ_o^2 defined in the theorem and holds simultaneously for all i.

We now apply the union bound to Equation 9: With probability no less than $1-\delta$, for all $1 \leq i \leq d$:

$$\|\mathbf{u}_i(\Phi_{-i}\mathcal{P}_{\mathcal{Q},h}) - \Phi_i\mathcal{P}_{\mathcal{Q},o,h}\|_{\rho(\mathbf{x})} \leq \epsilon(|\mathcal{H}|, |\mathcal{Q}|, d, L_o, \sigma_o^2, \delta/d). \quad (17)$$

Note that by our definition of $\mathbf{C}_o$ we have that $\mathbf{u}_i(\Phi_{-i}\mathcal{P}_{\mathcal{Q},h}) = (\mathbf{C}_o)_i(\Phi\mathcal{P}_{\mathcal{Q},h})$, which immediately gives the theorem by combining the error bounds on each row. $\square$

For large $|Q| \gg d$, using the properties on the distribution of the maximum of i.i.d. normal variables, we have with high probability $\max_{i,j}(\Phi_{ij})^2 = O((\log|Q|)/d)$ (Fisher & Tippett, 1928). The norm of $\mathbf{B}_o$ is largely problem dependent, but it is likely to be close to 1 in many cases. Assuming $k' = k$, we see that the effective variance term is a $(k\log|Q|)/d$ factor of the original variance. Thus, setting $d = O(k\log|Q|)$ should suffice to control the effective noise variance.

The effectiveness of the compressed regression is largely dependent on how the L_o term behaves compared to the norm of the target values. We refer the reader to the discussions in Maillard & Munos (2009) and Maillard et al. (2012) on the $\|\mathbf{w}\|\|\mathbf{x}\|_{\rho(\mathbf{x})}$ term. If the target functions are smooth, then we expect an analysis similar to that of Fard et al. (2012) to prove that projections of size logarithmic in $|Q|$ are enough to outperform the baseline predictor (constant output). A full discussion on such analysis is beyond the scope of this paper.

4.2. Error of The Compressed Normalizer

The $\mathbf{c}_\infty$ operator is the normalization operator for the compressed space. Therefore, for any history h, $\mathbf{c}_\infty^T \Phi \mathcal{P}_{\mathcal{Q},h}$ should equal $\mathcal{P}_h$. The following theorem provides a bound over the error of such prediction:

Theorem 2. *Let $\mathcal{H}$ be a large collection of sampled histories according to ρ. We observe noisy estimate $\hat{\mathcal{P}}_{\mathcal{Q},\mathcal{H}} = \mathcal{P}_{\mathcal{Q},\mathcal{H}} + \Delta_x$ of input and $\hat{\mathcal{P}}_{\mathcal{H}} = \mathcal{P}_{\mathcal{H}} + \Delta_z$ of the output, where elements of Δ_x and Δ_z are independent zero-mean random variables with maximum variance σ_x^2 and σ_z^2 respectively. Define $\mathbf{c}_\infty = (\Phi_i \hat{\mathcal{P}}_{\mathcal{Q},\mathcal{H}})^\dagger \hat{\mathcal{P}}_{\mathcal{H}}$. Then with probability no less than $1-\delta$ we have:*

$$\left\|\mathbf{c}_\infty^T(\Phi \mathcal{P}_{\mathcal{Q},h}) - \mathcal{P}_h\right\|_{\rho(\mathbf{x})} \leq \epsilon(|\mathcal{H}|, |\mathcal{Q}|, d, L_\infty, \sigma_\infty^2, \delta),$$

where we define effective noise $\sigma_\infty^2 = \sigma_z^2 + \sigma_x^2 \|\mathbf{b}_\infty\|^2$ and $L_\infty = \|\mathbf{b}_\infty\| \|\mathcal{P}_{\mathcal{Q},h}\|_{\rho(\mathbf{x})}$.

Proof. Similar to Theorem 1, we have $\mathcal{P}_h = \mathbf{b}_\infty^T \mathcal{P}_{\mathcal{Q},h}$ for all h. Therefore we have a linear target and by definition $\mathbf{c}_\infty$ is the COLS estimate with projection Φ. We have:

$$\begin{aligned} \hat{\mathcal{P}}_h &= \mathcal{P}_h + \Delta_z = \mathbf{b}_\infty^T \mathcal{P}_{\mathcal{Q},h} + \Delta_z && (18) \\ &= \mathbf{b}_\infty^T \hat{\mathcal{P}}_{\mathcal{Q},h} - \mathbf{b}_\infty^T \Delta_x + \Delta_z. && (19) \end{aligned}$$

Thus the effective variance is bounded by the σ_∞^2 defined in the theorem. We complete the proof by an application of the bound in Equation 9. $\square$

4.3. Error Propagation

Once we have the one step errors of compressed operators, we can analyze the propagation of errors as we concatenate the operators. Define $o^{1:t} = o^1 o^2 \dots o^t$. We would like to bound the error between $P\{ho^{1:T}\}$ and our prediction $\mathbf{c}_\infty \mathbf{C}_{o^T} \mathbf{C}_{o^{T-1}} \dots \mathbf{C}_{o^1} \mathbf{c}_h$, where $\mathbf{c}_h = \Phi \mathcal{P}_{\mathcal{Q},h}$. Since the theorems in the previous sections were in terms of a fixed measure ρ, we have to make distributional assumptions to simplify the derivations. Assume that if we sample h from ρ, for all o, $\mathcal{P}_{\mathcal{Q},o,h}$ has the same distribution as $\mathcal{P}_{\mathcal{Q},h}$. Thus, by induction $\|f(\mathcal{P}_{\mathcal{Q},o^{1:t},h})\|_\rho = \|f(\mathcal{P}_{\mathcal{Q},h})\|_\rho$.

For all t, define $\mathbf{e}_t^h$ such that $\mathbf{C}_{o^t} \mathbf{C}_{o^{t-1}} \dots \mathbf{C}_{o^1} \mathbf{c}_h = \mathcal{P}_{\mathcal{Q},o^{1:t},h} + \mathbf{e}_t^h$. Also let ϵ_t be the bound of Theorem 1 for $\mathbf{C}_{o^t}$.

For a fixed t assume that $\|\mathbf{e}_t^h\|_\rho \leq \alpha_t$. After applying the $(t+1)$th compressed operator we have:

$$\begin{aligned} \|\mathbf{e}_{t+1}^h\|_\rho &= \|\mathbf{C}_{o^{t+1}} \mathbf{C}_{o^t} \mathbf{C}_{o^{t-1}} \dots \mathbf{C}_{o^1} \mathbf{c}_h - \mathcal{P}_{\mathcal{Q},o^{1:t+1},h}\|_\rho \\ &= \|\mathbf{C}_{o^{t+1}}(\mathcal{P}_{\mathcal{Q},o^{1:t},h} + \mathbf{e}_t^h) - \mathcal{P}_{\mathcal{Q},o^{1:t+1},h}\|_\rho \\ &\leq \|\mathbf{C}_{o^{t+1}} \mathcal{P}_{\mathcal{Q},o^{1:t},h} - \mathcal{P}_{\mathcal{Q},o^{1:t+1},h}\|_\rho + \|\mathbf{C}_{o^{t+1}} \mathbf{e}_t^h\|_\rho \\ &\leq \|\mathbf{C}_{o^{t+1}} \mathcal{P}_{\mathcal{Q},h} - \mathcal{P}_{\mathcal{Q},h}\|_\rho + \|\mathbf{C}_{o^{t+1}}\| \|\mathbf{e}_t^h\|_\rho && (20) \\ &\leq \epsilon_t + \alpha_t \|\mathbf{C}_{o^{t+1}}\|. && (21) \end{aligned}$$

Line 20 uses the distribution assumption discussed above. We can see that there is an additive error after each compressed operator, and also a multiplicative part that amplifies the error of the previous steps. Assuming that ϵ_∞ is the bound of Theorem 2, one can apply a similar analysis to the compressed normalization operator to obtain an additive error of ϵ_∞ and the amplification of the previous error by $\|\mathbf{c}_\infty\|$.

We finish the analysis by assuming that the norm of all $\mathbf{C}_o$ terms and the $\mathbf{c}_\infty$ term is bounded by c, and that all ϵ_t and the ϵ_∞ term are bounded by ϵ. The total propagated error is thus bounded by $\epsilon(c^T - 1)/(c-1)$.

5. Empirical Results

To complete our analysis, we consider the empirical performance of the CPSR algorithm on two synthetic domains, called *GridWorld* and *PocMan*. Both domains were originally defined as POMDPs; we consider here the non-controllable version (i.e. only observations). The goal therefore is to learn a good predictive representation of the domains, rather than attempt to plan using these representations. Throughout our experiments, the agents were given fixed exploration policies, and only the sequences of observations were recorded. The CPSR algorithm was then used to construct models, which produce probability distributions over observation sequences.

5.1. GridWorld

The *GridWorld* domain is a 5×12 grid maze (see Figure 1) where the agent must navigate towards a goal state. The primary difficulty in this domain is that the agent receives aliased observations, designating whether or not there is an adjacent wall in any of the four cardinal directions. The domain also has the added complication that state-to-state transitions are stochastic (0.2 transition noise probability). The primary purpose of this experiment is to illustrate the computational efficacy of the CPSR method compared to an uncompressed naive TPSR approach.

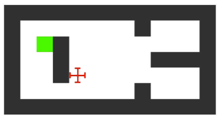

Figure 1. Image shows the structure of the GridWorld maze. The + target is the start state and the shaded green square is the goal.

Method and Evaluation: In all trials, the learning algorithm was given a sample of 1000 training runs to learn a model representation of the domain. We compare the model accuracy and algorithm runtime for building TPSR and CPSR models. In both the CPSR and TPSR cases, tests of length at most four were used and the final model dimension was set to 30.

The quality of the models were evaluated according to their T-step prediction performance. That is, each model was asked to produce probability distributions for the terminal observation at each time step (up to the history bound). In other words, the models produced must predict what observation would occur at each time-step (up to some bound) using only the initial PSR state (i.e. without performing any updates).

In all models, these predictions were computed using equation (3) and the prediction performance was evaluated on 10 test sets. The model predictions $\hat{P}(o_k^j|h_0)$ were then compared to Monte-Carlo rollout predictions $\tilde{P}(o_k^j|h_0)$ (produced using knowledge of the underlying state dynamics) according to the prediction error $\|\hat{P}(o_k^j|h_0) - \tilde{P}(o_k^j|h_0)\|_2$.

Results: As shown in Figure 2, the CPSR algorithm was able to produce competitive predictions, compared to the TPSR algorithm, showing that the compression does not have detrimental effects on model quality. Moreover, as shown in Figure 3, performing the estimation in the compressed space greatly reduced the runtime of the algorithm.

5.2. PocMan

The *PocMan* domain is a partially observable variant of the popular video game Pac-Man (Silver & Veness, 2010). Like in the video game, the goal here is to collect randomly distributed food pellets while navigating in a 17×19 maze and avoiding coming in contact with any of four ghosts. Unlike the video game, however, the agent does not have knowledge of the full environment state and only has access to a set of lo-

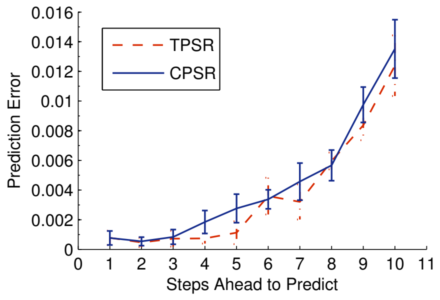

Figure 2. Graph shows T-step prediction error of the CPSR and TPSR in the GridWorld domain. The TPSR outperforms the CPSR algorithm by a slight margin; however, it is unlikely that this small difference would lead too a significant advantage in practice.

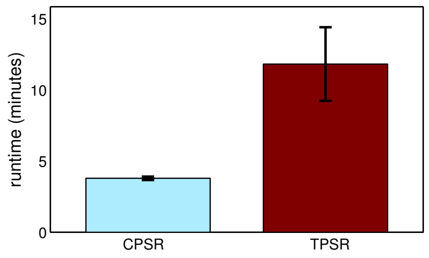

Figure 3. Graph shows runtimes for the TPSR and CPSR algorithms. Runtime here includes both estimation and the construction of the model. The CPSR algorithm is able to build a model in significantly less time compared to the TPSR algorithm due to the use of compressed estimation.

cal observations. This domain is especially interesting as its large state space ($|S| \approx 10^{56}$) and observation space ($|\mathcal{O}| = 2^{10}$) make it intractable for conventional EM style algorithms (Silver & Veness, 2010). Experimentation in this domain illustrates how the CPSR method is able to produce higher quality models due to the fact that it performs estimation in the compressed space and can, therefore, include more information in its estimates.

Method and Evaluation: The algorithms were once again given 1000 trials to learn a model representation. However, due to the extremely large observation space, the CPSR method is able to include more information in its empirical estimates. Specifically, the CPSR algorithm is able to include tests of all lengths in its estimates, whereas the TPSR algorithm is only able to include tests of length 1 while not exceeding memory limits (8 GB in this case).

We compared quality of models produced by a CPSR with all tests and a TPSR with tests of length 1. Again, the T-step prediction error was used as a measure of quality.

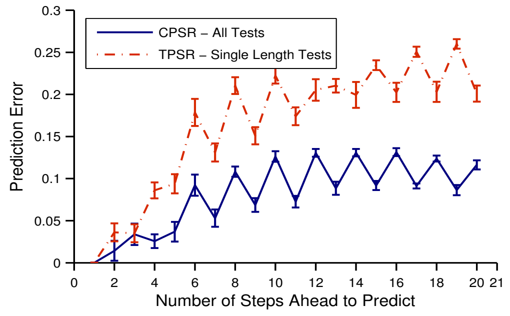

Figure 4. Graph shows T-step prediction error in the *PocMan* domain. The CPSR algorithm has far lower error due to its ability to include more tests. The oscillation present is due to periodicity in the domain and is not an artifact of the algorithms.

Results: As shown in Figure 4, the CPSR algorithm was able to learn a more accurate model due to its ability to include longer tests. This effect will be strongest in domains, such as *PocMan*, that are strongly partially observable and that have very large observation spaces. Of course, a domain specific feature mapping technique could be used to reduce to memory load for the TPSR algorithm. However, such a feature mapping would not be general and would require expert construction. The benefit of the CPSR approach is that the random projections are agnostic with respect to the domains characteristics.

In this experiment, we did not directly compare empirical runtimes, since the algorithms did not use the same tests. However, for simplicity, we set the compressed dimension of the CPSR model to be equal to the number of tests, $|\mathcal{T}|$, used by the TPSR algorithm. We also set this as our final model dimension (the average final dimension was 140). Thus, by design, the runtime for the algorithms is asymptotically equal at $O(|\mathcal{O}||H||T|^2)$ and they use comparable resources.

6. Discussion

The CPSR algorithm provides a memory efficient technique for learning compressed representations of dynamic domains with large observation spaces using batch data. It provides theoretical guarantees for the fidelity of the compression and shows good empirical performance. One particular advantage is that it relieves researchers of the burden of determining an appropriate compact representation. Selecting a compact latent state representation, in the case of HMMs/POMDPs, is a difficult problem; some recent work shows promise, but is much more expensive computationally than the methods we present here (Veness et al., 2011). Similarly, selecting a compact set of tests to use when learning PSR representations is a difficult issue, and currently, there are no principled approaches beyond enumerating observed tests with fixed bounds on test length, and applying spectral methods to reduce the dimension. The benefit of the CPSR algorithm is that a much greater number of possible tests can be considered without affecting the space complexity of the learning process.

It is worth noting that the CPSR algorithm does not reduce the dimension of histories. There is a trade-off here: we can reduce either the dimension of tests, or of histories; reducing both would likely alter the sparsity of the matrices. Reducing the dimension of tests is probably preferable since in general, the space of all possible tests is greater than, or equal to, the space of all histories (in any domain with a defined start state it will be combinatorially larger) (Singh et al., 2004). Moreover, a number of methods already exist to reduce the memory burden induced by the number of histories, such as the improved batch algorithm (Boots & Gordon, 2011), and the use of large set sizes in indicative events (which are the sets of histories used in the estimation). Most importantly for CPSRs, the matrices representing the learned parameters are all of size $d \times d$, where d is the reduced dimension of tests ($d = |\mathcal{T}|$ if no compression is performed).

Throughout this paper, we described the CPSR algorithm in the uncontrolled case (i.e. only observations, no actions) primarily for simplicity of exposure. It is worth noting that the CPSR approach is fully compatible with work on planning using TPSRs (Boots et al., 2009; Boots & Gordon, 2011). One simply substitutes the CPSR states for the corresponding TPSR states. This is useful as planning algorithms can be developed independently of the type of PSR approach being used.

In conclusion, the CPSR algorithm provides a novel lightweight algorithm for efficiently learning models of dynamic systems in large discrete observation spaces. We have used it to learn an effective representation in the *PocMan* domain, which has on the order of 10^{56} states and 2^{10} observations. We believe that the type of sparsity required for CPSR is present in a large number of dynamic systems. We are now considering extensions of the algorithm for continuous spaces.

Acknowledgements

The authors would like to thank Doina Precup, Yuri Grinberg and Sylvie Ong for helpful discussions on this work, and Joel Veness and David Silver for support on the PocMan domain. Financial support for this research was provided by the NSERC Discovery grant and USRA programs.

References

Baraniuk, R. and Wakin, M. Random projections of smooth manifolds. *Foundations of Computational Mathematics*, 9:51–77, 2009.

Boots, B. and Gordon, G. An online spectral learning algorithm for partially observable dynamical systems. In *Association for the Advancement of Artificial Intelligence*, 2011.

Boots, B., Siddiqi, S., and Gordon, G. Closing the learning-planning loop with predictive state representations. In *Proceedings of Robotics: Science and Systems VI*, 2009.

Fard, M.M., Grinberg, Y., Pineau, J., and Precup, D. Compressed least-squares regression on sparse spaces. In *Association for the Advancement of Artificial Intelligence*, 2012.

Fisher, R.A. and Tippett, L.H.C. Limiting forms of the frequency distribution of the largest or smallest member of a sample. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 24. Cambridge Univ Press, 1928.

Hsu, D., Kakade, S., and Zhang, T. A spectral algorithm for learning hidden markov models. In *Conference on Learning Theory*, 2008.

Kaelbling, L., Littman, M., and Cassandra, A. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

Maillard, O.A. and Munos, R. Compressed least-squares regression. In *Advances in Neural Information Processing Systems*, 2009.

Maillard, O.A., Munos, R., et al. Linear regression with random projections. *Journal of Machine Learning Research*, 2012.

Rabiner, Lawrence R. A tutorial on hidden markov models and selected applications in speech recognition. In Waibel, Alex and Lee, Kai-Fu (eds.), *Readings in speech recognition*, pp. 267–296. 1990.

Rosencrantz, M., Gordon, G., and Thrun, S. Learning low dimensional predictive representations. In *Proceedings of the twenty-first International Conference on Machine learning*, 2004.

Silver, D. and Veness, J. Monte-carlo planning in large pomdps. *In Advances in Neural Information Processing Systems*, 2010.

Singh, S., James, M.., and Rudary, M. Predictive state representations: a new theory for modeling dynamical systems. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 2004.

Veness, J., Ng, K.S., Hutter, M., Uther, W., and Silver, D. A monte-carlo AIXI approximation. *Journal of Artificial Intelligence Research*, 40, 2011.

Wiewiora, E. *Modeling probability distributions with predictive state representations.* PhD thesis, University of California at San Diego, 2007.