Approximate Predictive State Representations

Britton Wolfe
Computer Science and Engineering
University of Michigan
Ann Arbor, MI 48109
bdwolfe@umich.edu

Michael R. James
AI and Robotics Group
Toyota Technical Center
2350 Green Road, Ann Arbor, MI 48105
michael.r.james@gmail.com

Satinder Singh
Computer Science and Engineering
University of Michigan
Ann Arbor, MI 48109
baveja@umich.edu

ABSTRACT

Predictive state representations (PSRs) are models that represent the state of a dynamical system as a set of predictions about future events. The existing work with PSRs focuses on trying to learn exact models, an approach that cannot scale to complex dynamical systems. In contrast, our work takes the first steps in developing a theory of approximate PSRs. We examine the consequences of using an approximate predictive state representation, bounding the error of the approximate state under certain conditions. We also introduce factored PSRs, a class of PSRs with a particular approximate state representation. We show that the class of factored PSRs allow one to tune the degree of approximation by trading off accuracy for compactness. We demonstrate this trade-off empirically on some example systems, using factored PSRs that were learned from data.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Theory

Keywords

predictive state representations, reinforcement learning, factored models

1. INTRODUCTION

Predictive state representations (PSRs) [8] are a class of models that represent the state of a dynamical system as a set of predictions about the probability of future events. Much of the existing work with PSRs has focused on learning *exact* models of a system; such work will apply only to small systems. In order to scale PSRs to work with larger, complex systems, one must make approximations. These approximations can be in the model parameterization or in the state representation, both of which will be needed in order to learn a model of a complex system. This work addresses the need for a theory of approximation in PSRs. We show that one can represent predictive state using an approximation for which the error does not grow over time but remains bounded. We introduce a class of approximate models called *factored PSRs*. Factored PSRs address the need for modeling complex dynamical systems by allowing one to trade off model compactness for accuracy. We demonstrate empirically that one can learn reasonably accurate factored PSRs from a single trajectory of experience. Furthermore, even in the cases when it is possible to learn an exact model, learning a factored PSR can be faster because it can be significantly more compact than an exact model.

Cite as: Approximate Predictive State Representations, Britton Wolfe, Michael R. James and Satinder Singh, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 363-370.

2. BACKGROUND

We begin with an introduction to linear PSRs, which are a well-studied class of PSRs [8, 5, 12, 16] that form the basis of our work. First we will formalize the predictions that compose the state vector of a PSR.

Predictions: We assume that the agent is in a discrete-time environment with a set of discrete actions A and a set of n observation variables, each of which has a discrete, finite domain. At each time step x, the agent chooses some action $a_x \in A$ to execute and then receives some n-dimensional observation $\mathbf{o_x} = [o_x^1, o_x^2, \ldots o_x^n]$. A *history* is a possible sequence of actions and observations $a_1\mathbf{o_1}a_2\mathbf{o_2}\ldots a_\tau\mathbf{o}_\tau$ from the beginning of time. A *test* is a sequence of possible future actions and observations $a_{\tau+1}\mathbf{o}_{\tau+\mathbf{1}}\ldots a_{\tau+k}\mathbf{o}_{\tau+\mathbf{k}}$, where τ is the current time step. The *prediction* for a test $t = a_{\tau+1}\mathbf{o}_{\tau+\mathbf{1}}\ldots a_{\tau+k}\mathbf{o}_{\tau+\mathbf{k}}$ from a history $h = a_1\mathbf{o_1}\ldots a_\tau\mathbf{o}_\tau$ is defined as the probability of seeing the observations of t when the actions of t are taken from history h. Formally, this prediction is

$$p(t|h) \triangleq \prod_{i=\tau+1}^{\tau+k} Pr(\mathbf{o_i}|a_1, \mathbf{o_1}, \ldots a_\tau, \mathbf{o}_\tau, a_{\tau+1}, \mathbf{o}_{\tau+\mathbf{1}}, \ldots a_i)$$

where a_x represents the event "a_x is the action at time x," and $\mathbf{o_x}$ represents the event "$\mathbf{o_x}$ is the observation vector at time x."

System-dynamics matrix: We will describe linear PSRs using the concept of a system-dynamics matrix, introduced by Singh, James, and Rudary [12]. A system-dynamics matrix $\mathcal{D}$ fully specifies a dynamical system, and in turn any system completely defines some system-dynamics matrix $\mathcal{D}$. The matrix $\mathcal{D}$ has one row for each possible history (including the empty or null history ϕ) and one column for each possible test.[1] The entry in a particular row and

[1]The tests (and histories) can be arranged in length-lexicographical ordering to make a countable list.

column is the prediction for that column's test from that row's history. Despite the fact that $\mathcal{D}$ is $\infty \times \infty$, it will have a finite rank m for a large class of systems, including POMDPs with a finite number of latent states; the rank m is no greater than the number of latent states in the POMDP [12]. For systems with a finite rank m, one can find a set Q of m linearly independent columns such that all other columns are linearly dependent upon Q. The tests corresponding to these columns (also denoted Q) are called *core tests*. At any history h, the prediction for any test t is a *history-independent* linear function of the predictions for Q. In other words, the predictions for Q are a sufficient statistic for computing the prediction of any other test.

Linear PSR: A linear PSR represents the state of the system at h as the vector of predictions for Q from h. This vector is called the *prediction vector*, written as $p(Q|h)$. A *linear PSR* model $\mathcal{M}$ is composed of the predictions for Q from the null history (the initial predictive state), and the *update parameters* used to update the prediction vector as the agent moves to new histories. We use m_t to denote the history-independent vector of weights such that $\forall h : p(t|h) = p^\top(Q|h)m_t$; such an m_t exists for any t by definition of Q. The update parameters for a linear PSR are the m_t's for each one-step test $(a\mathbf{o})$ and each one-step extension $(a\mathbf{o}q_i)$ of each core test $q_i \in Q$. The update procedure is to obtain $p(Q|ha\mathbf{o})$ from $p(Q|h)$ after taking the action a and seeing the observation $\mathbf{o}$ from the history h. For any $q_i \in Q$, one can use the existing state vector $p(Q|h)$ and the update parameters $m_{a\mathbf{o}}$ and $m_{a\mathbf{o}q_i}$ to calculate $p(q_i|ha\mathbf{o}) = \frac{p(a\mathbf{o}q_i|h)}{p(a\mathbf{o}|h)} = \frac{p^\top(Q|h)m_{a\mathbf{o}q_i}}{p^\top(Q|h)m_{a\mathbf{o}}}$. The update parameters can also be used to calculate the m_t for any test t, enabling the PSR to make a prediction for any test.

3. APPROXIMATE STATE REPRESENTATION

For complex systems, the number of predictions that are required in the state vector (i.e. the number of core tests) is prohibitively large, so one must use some approximate, compact representation of the predictive state (e.g. product of factors). As with a full state, a model must update its approximate state after each time step. In general, the updated state will not be amenable to the desired compact representation, so the model must map the updated state to a compact updated state. This mapping can introduce some error in the state at each time step. In this section, we show that if the approximation error introduced at each time step is bounded, there exists a bound on the error of the predictive state vector that is independent of time (i.e. the approximation errors do not accumulate over time).

An approximate PSR consists of an approximate, compact state representation and some approximate model $\hat{\mathcal{M}}$ to update the state. We will use $\tilde{x}_\tau$ to denote the approximate predictive state at time τ; thus $\tilde{x}_\tau$ is shorthand for $\tilde{p}(Q|h_\tau)$, where h_τ is the history through time τ. The approximate state representation and model determine the approximate predictive state $\tilde{x}_\tau$ at each time τ in the following way (Figure 1): passing $\tilde{x}_{\tau-1}$ through the update process of $\hat{\mathcal{M}}$ for $a_\tau\mathbf{o}_\tau$ yields some predictive state $\hat{x}_\tau$. As mentioned above, $\hat{x}_\tau$ might not have the desired compact form, so the model must map $\hat{x}_\tau$ to some $\tilde{x}_\tau$ that will have the desired compact form; we let F be the function that maps $\hat{x}_\tau$ to $\tilde{x}_\tau$ for each τ. One can measure the quality of approximation by comparing $\tilde{x}_\tau$ with the true predictive state x_τ at time τ, which is determined by the true PSR model $\mathcal{M}$ (Figure 1). The primary result of this section is a bound on the error of $\tilde{x}_\tau$ that is independent of τ (Theorem 3) when $\hat{\mathcal{M}}$ is $\mathcal{M}$; of course, selecting an $\hat{\mathcal{M}}$ that is optimized for F will do no worse.

The results in this section rely heavily upon the work by Boyen and Koller [1] that bounds approximation error for latent-state models (e.g. POMDPs or DBNs). Latent-state models represent state as *belief state*, a distribution over a set of unobserved random variables called "latent states." After each time step τ the latent state model $\mathcal{P}$ calculates the new belief state σ_τ from the most recent action and observation and the old belief state $\sigma_{\tau-1}$. One can pass an approximate belief state $\tilde{\sigma}_{\tau-1}$ through the true belief state update $\mathcal{P}$ to get an estimate $\hat{\sigma}_\tau$ of the true σ_τ (Figure 1). Boyen and Koller [1] showed that the error $D(\sigma_\tau \parallel \hat{\sigma}_\tau)$ is a constant factor less than the error $D(\sigma_{\tau-1} \parallel \tilde{\sigma}_{\tau-1})$, where $D(v \parallel w)$ is the Kullback-Leibler (KL) divergence of two stochastic vectors v and w: $D(v \parallel w) \triangleq \sum_i v_i \log \frac{v_i}{w_i}$.

We use this fact about the belief state update to bound the predictive state error, employing a belief state approximation procedure that is *implicitly* defined by our approximate PSR. Note that the latent-state model is only used for analysis of the error bound, and any accurate latent-state model of the system may be used in calculating such a bound.

We associate each $\tilde{x}_\tau$ with some belief state $\tilde{\sigma}_\tau$ that implies the same future predictions as $\tilde{x}_\tau$ (Figure 1). Each $\tilde{x}_\tau$ and $\tilde{\sigma}_\tau$ are related through the *outcome matrix* U [12]: $\tilde{x}_\tau = U^\top\tilde{\sigma}_\tau$.[2] The U matrix has one row for each latent state and one column for each test in x_τ. The (i,j)th entry is the probability of test j succeeding when one starts in latent state i. Note that when $\hat{\mathcal{M}} = \mathcal{M}$, $\tilde{x}_\tau = U^\top\tilde{\sigma}_\tau$ implies $\hat{x}_{\tau+1} = U^\top\hat{\sigma}_{\tau+1}$ [8]. Thus, each predictive state in Figure 1 has a corresponding belief state.

Two steps remain to bound the error in $\tilde{x}_\tau$: 1) bound the error in $\tilde{\sigma}_\tau$ and 2) relate that to the error in $\tilde{x}_\tau$. The bound for $\tilde{\sigma}_\tau$ is given by Boyen and Koller [1], provided that F meets the following condition:

Definition A series $\tilde{x}_0, \tilde{x}_1, \tilde{x}_2, \ldots$ of approximate predictive states *incurs error* ϵ at each time step τ if for all τ, the implicit, approximate belief states $\tilde{\sigma}_\tau$ and $\hat{\sigma}_\tau$ corresponding to $\tilde{x}_\tau$ and $\hat{x}_\tau$ satisfy $D(\sigma_\tau \parallel \tilde{\sigma}_\tau) - D(\sigma_\tau \parallel \hat{\sigma}_\tau) \leq \epsilon$ (Figure 1).

For the rest of this section, we will assume that this condition is met. Now we state two results from Boyen and Koller [1]; Lemma 1 is used to define the rate of contraction for the bound on the error of $\tilde{\sigma}_\tau$ (Lemma 2).

(Boyen & Koller [1]) Lemma 1. *For any row-stochastic matrix C, let $\gamma_C \triangleq \min_{i_1,i_2} \sum_j \min(C_{i_1 j}, C_{i_2 j})$. Then for any stochastic vectors v and w, $D(C^\top v \parallel C^\top w) \leq (1 - \gamma_C)D(v \parallel w)$.*

(Boyen & Koller [1]) Lemma 2. *For any τ, $E_{h_\tau}[D(\sigma_\tau \parallel \tilde{\sigma}_\tau)] \leq \frac{\epsilon}{\gamma}$, where h_τ is the history through time τ and $\gamma \triangleq \min_a \gamma_{T_a}$, with T_a being the latent state transition matrix for action a.*

[2] We assume that at least one such $\tilde{\sigma}_\tau$ exists for each $\tilde{x}_\tau$. This is satisfied if each $\tilde{x}_\tau$ lies in the convex hull of possible predictive states.

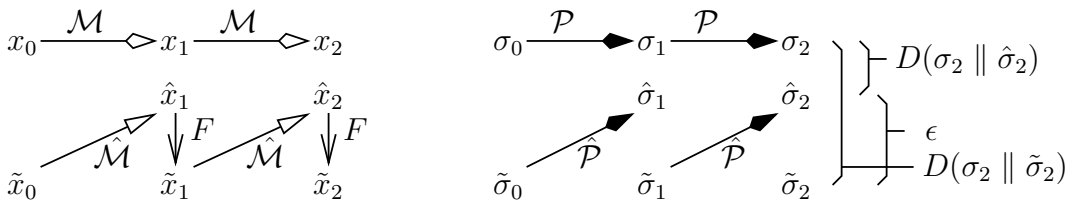


Figure 1: The approximation process. The belief states on the right correspond with the predictive states on the left. Arrow style and label indicates which model or function performs the transformation. The state updates (performed by the models $\mathcal{M}$, $\hat{\mathcal{M}}$, $\mathcal{P}$, and $\hat{\mathcal{P}}$), are each a function of the most recent action and observation (omitted for clarity). Note that each $\tilde{\sigma}_\tau$ is determined by its correspondence with $\tilde{x}_\tau$ and not necessarily as a function of $\hat{\sigma}_\tau$. The errors shown on the right side illustrate an F that incurs error ϵ at each time step.

Now we relate the error of $\tilde{\sigma}_\tau$ to that of $\tilde{x}_\tau$. Since $\tilde{x}_\tau$ is not necessarily a stochastic vector, one cannot directly use KL divergence to measure its error. However, $\tilde{x}_\tau$ implies a set of stochastic vectors, one for each unique action sequence of its core tests. The predictions for all of the possible tests with a given action sequence must sum to 1. Thus, one can partition the entries of $\tilde{x}_\tau$ according to their tests' action sequences. One can implicitly add a complement test to each partition, which succeeds if and only if no other test in that partition succeeds (assuming that the action sequence is taken).[3] Each partition (with the implicit complement test) then forms a stochastic vector. For simplicity, we will assume that all tests in the state vector fall in the same partition; our bound on the KL divergence of the single partition is easily extended to each of multiple partitions.

We define the error $\mathcal{E}(\tilde{x}_\tau)$ of $\tilde{x}_\tau$ as $D(y_\tau \parallel \tilde{y}_\tau)$, where y is just x augmented with the complement test prediction. To translate KL divergence of belief states to KL divergence of predictive state, we again appeal to the U matrix. Let V be the matrix formed by adding a column to U for the complement test. Since V is a stochastic matrix, Lemma 1 gives a contraction rate γ_V which is used in the bound on $\mathcal{E}(\tilde{x}_\tau)$. This bound (Theorem 3) is the main result of this section, showing that the error of the approximate predictive state does not grow without bound over time.

THEOREM 3. *For any τ, $E_{h_\tau}[\mathcal{E}(\tilde{x}_\tau)] \leq (1-\gamma_V)\frac{\epsilon}{\gamma}$.*

PROOF. By definition of V, $E_{h_\tau}[\mathcal{E}(\tilde{x}_\tau)] = E_{h_\tau}[D(V^\top \sigma_\tau \parallel V^\top \tilde{\sigma}_\tau)]$. By Lemma 1, this quantity is less than or equal to $(1-\gamma_V)E_{h_\tau}[D(\sigma_\tau \parallel \tilde{\sigma}_\tau)]$, which itself is less than or equal to $(1-\gamma_V)\frac{\epsilon}{\gamma}$ because of Lemma 2. □

In addition to being the first error bound for approximate predictive state representations, Theorem 3 has two potential advantages over the corresponding bound on belief state error (Lemma 2). First, the bound itself is smaller by a factor of γ_V. The second advantage arises because the ϵ, γ, and γ_V in our error bound are calculated based upon an accurate latent-state model of the system. Since our bound holds for *any* accurate latent-state model, it also holds for the accurate latent-state model that yields the smallest bound. Note that one does not actually need to figure out which accurate model will give the best error bound in order for that bound to apply.

[3]The complement test is an example of a set test [15], described in more detail later.

4. APPROXIMATE MODELS

To learn a PSR model of a complex system, one will need to employ both a compact, approximate parameterization and a compact, approximate state representation. Both of these approximations are possible with the *factored PSRs* described in this section. Factored PSRs are a class of approximate PSRs that allow one to trade off compactness and accuracy of the approximate model. Like DBNs, our factored PSRs use a factored form to compute the joint probability of a set of random variables. However, our random variables are observation dimensions, whereas DBNs' random variables also include latent state variables.

4.1 A Simple Approximation

We will begin by describing the most compact factored PSR: one that assumes that future observation dimensions are conditionally independent given history. This is a gross approximation that we will relax later, but it provides a simple illustration. We define a set of functions $\{g^i : 1 \leq i \leq n\}$ such that g^i selects the ith observation dimension from a history, test, or observation vector. For example, $g^i(t) = a_{\tau+1}o^i_{\tau+1}\ldots a_{\tau+k}o^i_{\tau+k}$ for $t = a_{\tau+1}\mathbf{o}_{\tau+1}\ldots a_{\tau+k}\mathbf{o}_{\tau+\mathbf{k}}$. Note that the test $g^i(t)$ does not specify values for the full observation vector but only the ith dimension; this leaves the other observation dimensions as wild cards, so $g^i(t)$ is a *set test* [15]. Set tests are so named because the prediction is the sum of the predictions of a set of tests: the set generated by filling in the wild cards with each possible observation value.

The assumption that the future observation dimensions are conditionally independent given the history h is expressed in the equation

$$\forall h, t : p(t|h) = \prod_{i=1}^{n} p(g^i(t)|h), \tag{1}$$

for any sequence or set test t. This equation suggests a method for compactly representing the predictive state $p(Q|h)$: for each core test $q_j \in Q$ and dimension i, maintain the prediction $p(g^i(q_j)|h)$. The compactness arises when core tests have the same action/observation sequences for some observation dimensions (i.e. when $g^i(q_{j_1}) = g^i(q_{j_2})$ for $j_1 \neq j_2$). In that case, the $g^i(q_j)$ predictions can be multiplied in different ways to compute predictions for exponentially many full-dimension tests.

This method starts with the set of tests Q and uses them to decide what predictions should be included in the approximate state. However, finding a set of core tests Q for the

whole system is the difficult learning problem we were trying to avoid by using an approximate model. We now show that one does not need to find Q but can determine which predictions of the form $g^i(t)$ to include in the approximate state vector *independently for each* i. The following theorem is the basis of this fact. It shows that under the assumption of Equation 1, the prediction of $g^i(t)$ is independent of part of history: the observation dimensions not selected by g^i.

THEOREM 4. *The statement* $\forall h, t : p(t|h) = \prod_{i=1}^{n} p(g^i(t)|h)$ *(Equation 1) implies* $p(g^i(t)|h) = p(g^i(t)|g^i(h))$ *for all histories* h *and tests* t.

PROOF. By the definition of prediction, $p(g^i(t)|h) = \frac{p(hg^i(t)|\phi)}{p(h|\phi)}$. We then apply Equation 1 to both numerator and denominator to get

$$\frac{\prod_{j=1}^{n} p(g^j(hg^i(t))|\phi)}{\prod_{j=1}^{n} p(g^j(h)|\phi)} = \frac{p(g^i(ht)|\phi)}{p(g^i(h)|\phi)} \cdot \prod_{j \neq i} \frac{p(g^j(h)|\phi)}{p(g^j(h)|\phi)}.$$

We factored out the case where $g^j(hg^i(t))$ is $g^j(ht)$ from the cases where it is $g^j(h)$ (i.e. $g^i(t)$ has no overlap with the jth dimension). Then

$$\frac{p(g^i(ht)|\phi)}{p(g^i(h)|\phi)} \cdot 1 = \frac{p(g^i(h)g^i(t)|\phi)}{p(g^i(h)|\phi)}$$

$$= \frac{p(g^i(t)|g^i(h))\; p(g^i(h)|\phi)}{p(g^i(h)|\phi)} = p(g^i(t)|g^i(h)).$$

□

Combining Theorem 4 with Equation 1 implies $p(t|h) = \prod_{i=1}^{n} p(g^i(t)|g^i(h))$. Thus, under the conditional independence assumption of Equation 1, each observation dimension can be modeled completely independently. That is, Equation 1 implies that a factored *model* can be used as well as a factored state representation. The factored model can be more compact than a non-factored model of the system (cf. Section 5), and it will make the same predictions as the non-factored model (under the assumption of Equation 1). This is the basis for our completely factored model.

A Completely Factored Model: Our completely factored model for a dynamical system with n observation dimensions consists of n linear PSRs $(\mathcal{M}^1 \ldots \mathcal{M}^n)$. To make the prediction for a multi-dimensional test t, one obtains the prediction for each $g^i(t)$ from the respective $\mathcal{M}^i$, multiplying those predictions together as in Equation 1 to get an estimate for $p(t|h)$. The $\mathcal{M}^i$ only makes predictions about observation dimension i, so it can ignore all the observations from history except dimension i (Theorem 4). One learns the model $\mathcal{M}^i$ by passing only dimension i of an agent's experience into a linear PSR learning algorithm such as in [16]. Thus the core tests and the state update procedure for $\mathcal{M}^i$ are restricted to dimension i: at history h, the prediction vector of $\mathcal{M}^i$ is $p(Q^i|g^i(h))$ for core tests Q^i that are set tests in dimension i. For the state update, $\mathcal{M}^i$ computes $p(q|g^i(ha\mathbf{o})) = p(q|g^i(h)ao^i) = \frac{p(ao^iq|g^i(h))}{p(ao^i|g^i(h))}$ for each $q \in Q^i$ upon taking action a and seeing observation $\mathbf{o}$.

The number of core tests for $\mathcal{M}^i$ is $\text{rank}(\mathcal{D}^i)$, where $\mathcal{D}^i$ is a system-dynamics matrix with one row for each $g^i(h)$ and one column for each $g^i(t)$; the entry in that row and column is $p(g^i(t)|g^i(h))$. Theorem 5 (below) shows that $\text{rank}(\mathcal{D}^i) \leq \text{rank}(\mathcal{D})$, where $\mathcal{D}$ is the system-dynamics matrix of the whole system. In practice, $\text{rank}(\mathcal{D}^i)$ can be much less than $\text{rank}(\mathcal{D})$, as illustrated in Section 5. Lower rank leads to smaller models, which are often easier to learn.

4.2 Factored PSRs

The completely factored model is based upon the strong assumption that the future observation dimensions are conditionally independent of each other given history (Equation 1). In this section, we describe *factored PSRs*, a generalization of the completely factored model that does not make such a strong independence assumption. Like the completely factored model, a factored PSR consists of n linear PSRs $(\mathcal{M}^1 \ldots \mathcal{M}^n)$, one for each observation dimension. A factored PSR is a generalization of the completely factored model because each $\mathcal{M}^i$ is allowed to model any subset of the observation dimensions that includes dimension i, rather than modeling just dimension i. We let f^i be the function that selects the observation dimensions for $\mathcal{M}^i$; when applied to a sequence of actions and observations, we define $f^i(a_1\mathbf{o_1} \ldots a_k\mathbf{o_k})$ as $a_1 f^i(\mathbf{o_1}) \ldots a_k f^i(\mathbf{o_k})$. One can view $f^i(h)$ as selecting somewhere between $g^i(h)$ and the full h.

The learning and state update procedures for a factored PSR with a given $(f^1 \ldots f^n)$ are the same as for the completely factored model, except each g^i is replaced with f^i.

For making predictions, we will still use $\mathcal{M}^i$ to calculate the probability of success for observation dimension i. Unlike the completely factored model, here $\mathcal{M}^i$ models more than just dimension i, so its predictions for dimension i can be conditioned upon other dimensions that it models, improving their accuracy (Theorem 7). To compute joint probabilities, we are faced with combining predictions from multiple $\mathcal{M}^i$ models. There are several possible ways to do this; here we present one possibility. To make the prediction for a test $t = a_1\mathbf{o_1} \ldots a_k\mathbf{o_k}$ from some history h, $\mathcal{M}^i$ will compute a probability for o^i_τ for each $1 \leq \tau \leq k$. The probability for o^i_τ is conditioned upon $f^i(h_{\tau-1})$ and the dimensions of $f^i(\mathbf{o}_\tau)$ that are less than i, where h_τ is $ha_1\mathbf{o_1} \ldots a_\tau\mathbf{o}_\tau$. Formally, this probability is $\frac{Pr(f^i([o^1_\tau o^2_\tau \ldots o^i_\tau])|f^i(h_{\tau-1})a_\tau)}{Pr(f^i([o^1_\tau o^2_\tau \ldots o^{i-1}_\tau])|f^i(h_{\tau-1})a_\tau)}$. The approximate prediction for $p(t|h)$ from the factored PSR is the product of these probabilities for each i and each τ.

This prediction method comes from applying the chain rule of probability to the set $\{o^i_\tau : 1 \leq \tau \leq k, 1 \leq i \leq n\}$ in ascending order of τ, and in ascending order of i within each τ. The choice of ascending i within each τ was arbitrary; another order will yield a different estimate for $p(t|h)$, in general. One can obtain estimates for different orderings from the same factored PSR, combining those estimates to get an overall prediction.

Choosing f^i: There is a trade-off to consider when choosing f^i: having f^i select more of the observation vector can lead to better predictions, but having f^i select less of the observation vector can decrease the model size, which typically makes it easier to learn. One extreme choice is $f^i(h) = h$, which makes $\mathcal{M}^i$ a model for the whole system. The other extreme choice is $f^i = g^i$, which completely ignores information in the observation dimensions other than i. We prove two results about moving between these extremes: Theorem 5 shows that decreasing the scope of f^i will never increase the number of core tests for $\mathcal{M}^i$, and thus never increase the size of the model. Theorem 7 shows that increasing the scope of f^i will not make $\mathcal{M}^i$ less accurate in predicting the future of observation dimension i.

To prove these results, we first formalize the notion of

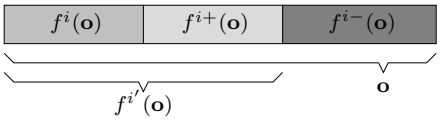


Figure 2: How the f functions select from o (the entire rectangle).

scope, illustrated in Figure 2. We say that $f^{i'}$ *has larger scope than* f^i (written $f^i \subseteq f^{i'}$) if $f^i(\mathbf{o})$ is a sub-vector of $f^{i'}(\mathbf{o})$ for any $\mathbf{o}$. Theorem 5 proves that decreasing the scope of f^i will never increase the number of core tests for $\mathcal{M}^i$.

THEOREM 5. *For f^i and $f^{i'}$ with respective system-dynamics matrices $\mathcal{D}^i$ and $\mathcal{D}^{i'}$, if $f^i \subseteq f^{i'}$, then* $\text{rank}(\mathcal{D}^i) \leq \text{rank}(\mathcal{D}^{i'})$.

PROOF. This proof describes matrices V and W such that $\mathcal{D}^i = V\mathcal{D}^{i'}W$, which implies $\text{rank}(\mathcal{D}^i) \leq \text{rank}(\mathcal{D}^{i'})$. The matrix V will combine the rows of $\mathcal{D}^{i'}$, yielding an intermediate matrix $\mathcal{X}$ of predictions that has rows for the histories of $\mathcal{D}^i$ and columns for the tests of $\mathcal{D}^{i'}$. The matrix V exists because the row for any history $f^i(h)$ in $\mathcal{X}$ is equal to a linear combination of rows of $\mathcal{D}^{i'}$. This can be seen by conditioning upon the observation dimensions of h selected by $f^{i'}$ but not by f^i, which we denote by f^{i+} (Figure 2).

The column for any test $f^i(t)$ in $\mathcal{D}^i$ is equivalent to a sum of columns of $\mathcal{X}$ because $f^i(t)$ is a set test in $\mathcal{X}$. It has wild cards for the f^{i+} observations at each time step of t. Therefore, there exists a matrix W such that $\mathcal{D}^i = \mathcal{X}W$, which equals $V\mathcal{D}^{i'}W$. □

As mentioned earlier, applying this theorem with $f^{i'}(h) = h$ proves that $\text{rank}(\mathcal{D}^i) \leq \text{rank}(\mathcal{D})$ for any f^i. (Recall that $\text{rank}(\mathcal{D}^i)$ and $\text{rank}(\mathcal{D})$ are the numbers of core tests needed for $\mathcal{M}^i$ and a full model $\mathcal{M}$, respectively.) Section 5 shows that the empirical estimate for $\text{rank}(\mathcal{D}^i)$ can be significantly less than that of $\text{rank}(\mathcal{D}^{i'})$, which supports making the scope of f^i small.

On the other hand, having f^i with a larger scope can give better predictions; the intuition is that conditioning one's predictions upon more of history cannot yield worse predictions, in expectation. Specifically, an f^i with larger scope will not increase the expected KL-divergence of the predictions given $f^i(h)$ from the predictions given the full history h (Theorem 7). To prove this, we use the following fact:

LEMMA 6. *For random variables Y, B, C, $E_{B,C}[D(Y|B,C \parallel Y) - D(Y|B, C \parallel Y|B)] \geq 0$.*

PROOF. The expected value given here is equal to the KL-divergence $D(Pr(B,C,Y) \parallel Pr(Y)Pr(B)Pr(C|Y,B))$, which is always non-negative. To see this equivalence, note that the expected value is equal to

$$\sum_{b,c} Pr(b,c)\left[\left(\sum_y Pr(y|b,c)\log\frac{Pr(y|b,c)}{Pr(y)}\right) - \left(\sum_y Pr(y|b,c)\log\frac{Pr(y|b,c)}{Pr(y|b)}\right)\right]$$

$$= \sum_{y,b,c} Pr(y,b,c)\left[\log\frac{Pr(y|b,c)}{Pr(y)} - \log\frac{Pr(y|b,c)}{Pr(y|b)}\right]$$

$$= \sum_{y,b,c} Pr(y,b,c)\log\frac{Pr(y|b)}{Pr(y)}\frac{Pr(b)Pr(c|y,b)}{Pr(b)Pr(c|y,b)}$$

$$= \sum_{y,b,c} Pr(y,b,c)\log\frac{Pr(y,b,c)}{Pr(y)Pr(b)Pr(c|y,b)}$$

which is the KL-divergence mentioned above. □

The following theorem compares the expected difference between two KL-divergences: the divergence (from the accurate predictions) of predictions conditioned upon f^i, and the divergence of predictions conditioned upon $f^{i'}$; when $f^i \subseteq f^{i'}$, the expected divergence with $f^{i'}$ is no greater.

THEOREM 7. *For $f^i \subseteq f^{i'}$ and any subset $\mathbf{X}$ of future observations, $E_H[D(\mathbf{X}|H \parallel \mathbf{X}|f^i(H)) - D(\mathbf{X}|H \parallel \mathbf{X}|f^{i'}(H))] \geq 0$, where H is the random variable for history through some time step.*

PROOF. We use Z to denote the random variable $D(\mathbf{X}|H \parallel \mathbf{X}|f^i(H)) - D(\mathbf{X}|H \parallel \mathbf{X}|f^{i'}(H))$; this makes Z a function of the random variable H. We use the following functions to denote the random variables corresponding to different parts of H: `acts` selects the actions and F^i, $F^{i'}$, F^{i+}, and F^{i-} each select some observation dimensions, detailed in Figure 2. We let f^i and $f^{i'}$ be realizations of the random variables $(\texttt{acts}, F^i)$, and $(\texttt{acts}, F^i, F^{i+})$, respectively, consistent with their definitions above. First we apply iterated expectations: $E_H[Z] = E_{\texttt{acts},F^i}[E_{F^{i+},F^{i-}}[Z|\texttt{acts}, F^i]]$. We now show that the inner expectation is always non-negative, so the whole expression is non-negative. For a given f^i,

$$E_{F^{i+},F^{i-}}[Z|f^i] = E_{F^{i+},F^{i-}}[D(\mathbf{X}|f^i, F^{i+}, F^{i-} \parallel \mathbf{X}|f^i) - D(\mathbf{X}|f^i, F^{i+}, F^{i-} \parallel \mathbf{X}|f^i, F^{i+})].$$

One can apply Lemma 6 directly to this expression to show that it is non-negative, using the following mapping: $Y \triangleq \mathbf{X}|f^i$, $B \triangleq F^{i+}$, and $C \triangleq F^{i-}$. □

5. EXPERIMENTAL RESULTS

This section describes our results on learning factored PSRs to model three domains of varying complexity. The data for the first two domains comes from simulations, while the data for the last domain comes from cameras overlooking a section of a highway.

5.1 Simulated Domains

The simulated domains allow us to provide some simple empirical illustrations of our theorems on two example systems. The first system is smaller to permit a comparison with exact predictions, while the second system is an example where approximate models and state representations are required for model learning. Both systems highlight the trade-off between the compactness and accuracy of a factored PSR; we compare a "baseline" model that uses $f^i = g^i$ with an "augmented" model that uses an $f^{i'}$ with larger scope. As part of the approximation scheme for the learned models, the entries of the prediction vector were clipped to fall in $[\epsilon, 1]$ after each time step of the testing sequence. Each learned model was evaluated throughout a testing sequence of length 2^{14} to account for any compounding of the approximation error over time.

The first system is a grid world (Figure 3) in which the agent can move any of the four cardinal directions and observes the colors of the adjacent floor tiles (or a wall) in each

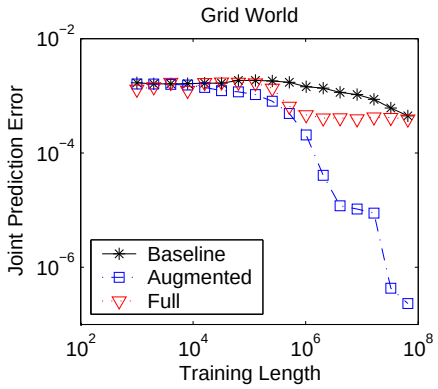


Figure 4: Grid world: median joint prediction error over 50 trials.

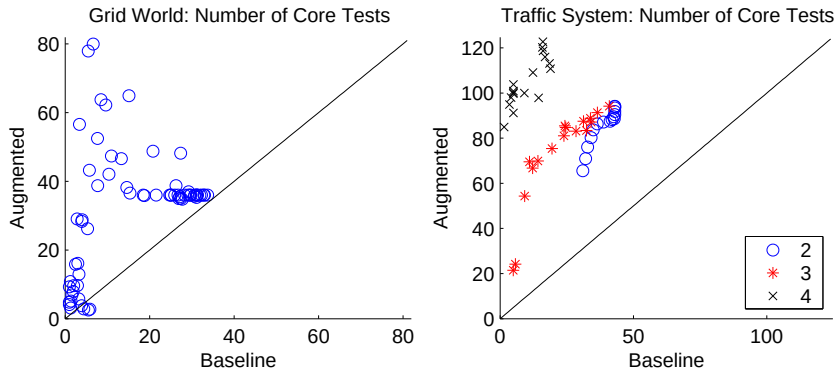


Figure 5: Average number of core tests for the augmented versus baseline models (grid world on the left and simulated traffic system on the right). One point is plotted for each $\mathcal{M}^i$ and training sequence length; the marker shapes indicate congestion threshold for the simulated traffic system. The line $x = y$ illustrates how the augmented model usually has more tests.

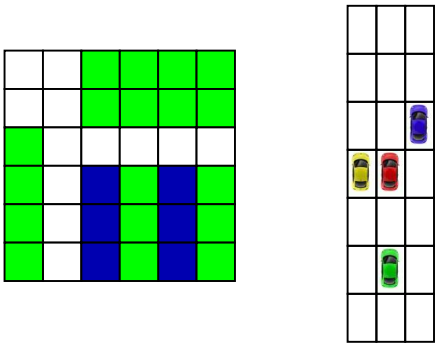


Figure 3: Grid world and simulated traffic domains.

of the four cardinal directions (a four-dimensional observation). Our baseline f^i selects just the ith observation dimension, while the augmented $f^{i'}$ also selects the dimension corresponding to the tile 90 degrees counterclockwise from the ith dimension. Each component model $\mathcal{M}^i$ was learned using a variant of the suffix-history algorithm [16] applied to f^i (or $f^{i'}$) of the agent's single experience trajectory.

To evaluate the predictive accuracy of our model, we used the mean squared error of one-step predictions, where the tests evaluated at time τ are those with action a_τ and any observation [16]. We bounded each estimated marginal prediction of the factored forms in $[0, 1]$. For the augmented model, we used five random orderings of the observation dimensions to get five estimates for $p(t|h)$; the median of these estimates was used as the model's prediction. As expected (Theorem 7), the augmented model makes better predictions than the baseline model (Figure 4); for the larger training sizes, this difference is several orders of magnitude. The augmented model also makes better predictions than a single linear PSR for the whole system learned using suffix-history (Figure 4). This illustrates that more data is required to learn a reasonably accurate full model than to learn a reasonably accurate, more compact, approximate model. The increased accuracy of the augmented model over the baseline model requires more core tests (Figure 5), as suggested by Theorem 5. This demonstrates that the factored PSR architecture allows one to choose f^i to trade off model simplicity and accuracy.

Our second system simulates one direction of a three-lane freeway (Figure 3). This system simulates the problem faced by an agent in a vehicle whose goal is to track and predict the movements of the other vehicles around it. The agent is given the current positions of all cars within some range, as would be returned from a radar system with multiple sensors. However, some simplifications were introduced. Cars take discrete positions in this system, and the field of view is defined relative to our agent's vehicle: the agent can see all three lanes for three spaces in front and behind its current location. Since our work focuses on modeling rather than control, our agent simply maintains a constant velocity in the middle lane. At each time step, the agent observes only the positions of each other car in its field of view (i.e. velocities are not given as observations); each car corresponds to a different observation dimension.

Cars enter the field of view stochastically, contingent upon a fixed congestion threshold θ: no new cars enter the field of view if there are already θ cars in the field of view. Each car that enters the field of view is assigned an unused observation dimension that remains unchanged until it disappears from the field of view. Each car has a default velocity that it will maintain unless the car in front of it was going too slowly at the last time step. In that case, the car will change lanes or slow down, depending on traffic in the neighboring lanes.

Learning a single PSR for the whole system was intractable because the prediction vector would have size combinatorial in the number of cars: a conservative lower bound on this size is 125,000 for the congestion threshold of 4. Figure 5 shows that our component models $\mathcal{M}^i$ use much smaller state vectors; they automatically exploited the structure among the observation dimensions that results from the spatially localized interaction between cars.

Our baseline f^i just selects the position of car i; the augmented $f^{i'}$ also selects the position of the car directly in

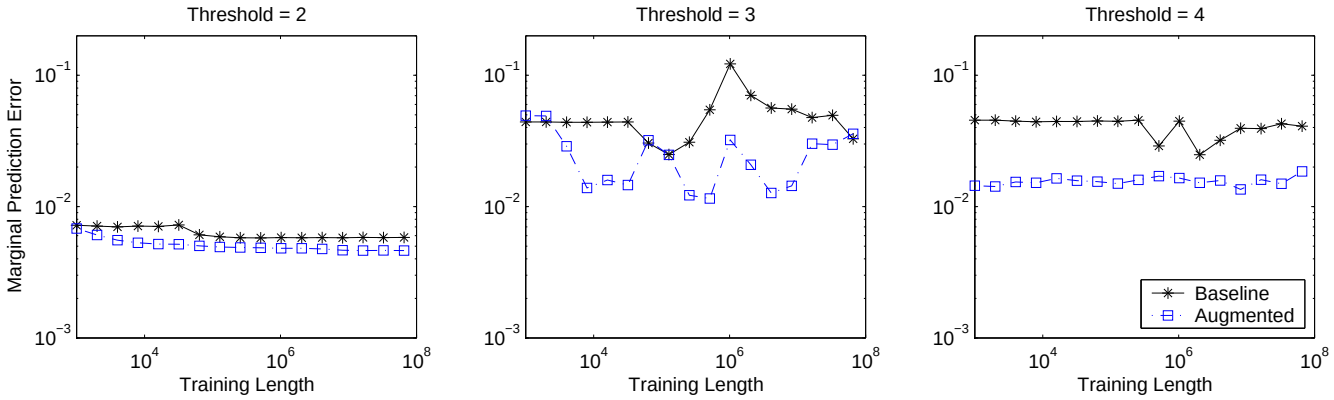


Figure 6: Simulated traffic system: median marginal prediction error over 50 trials versus training length. The title denotes congestion threshold.

front of i. The factored PSR for this system will consist of multiple copies of one linear PSR $\mathcal{M}$, since each observation dimension corresponds to a car of initially unknown default velocity. Each time a new car i enters the field of view, a new copy $\mathcal{M}^i$ of $\mathcal{M}$ will be initialized. When car i leaves the field of view, $\mathcal{M}^i$ is discarded. To train $\mathcal{M}$, we take the agent's single training sequence and divide it up into multiple overlapping trajectories: a trajectory begins when a car enters the field of view and ends the first time the car exits the field of view. The trajectories for car i are passed through f^i (or $f^{i'}$) and then given to a variant of the reset algorithm [5] to learn the linear PSR $\mathcal{M}$.

Because the joint observation space was so large, we evaluated the accuracy of the marginal predictions for each car: at each time τ we used $\mathcal{M}^i$ to get an estimate $\hat{p}(i,\tau)$ of the likelihood of car i moving to its actual next space. Our error measure is the mean over τ and i of $(1.0 - \hat{p}(i,\tau))^2$. Note that 0.0 error is not always attainable, since not even the full history will always be sufficient to predict the next position of each car. Overall, as with our first example, the augmented model made better predictions (Figure 6) but required more core tests (Figure 5).

5.2 Real-world Traffic Data

Our third domain is a real-world version of the simulated traffic domain presented above. We used the data collected by the Next Generation SIMulation (NGSIM) project [14] that captures traffic movement along approximately 500 meters of six-lane freeway in the San Francisco Bay area [13]. This is a rich data set, useful for analyzing many different aspects of traffic movement (e.g. [9, 2]).

Our model is built to predict traffic movements relative to a given *reference car* (i.e. the car in which the model would be employed). We assume that observations are only possible within some field of view that is defined relative to the front, center point of the reference car. In our experiments, the field of view extends 15 feet on either side, 100 feet behind, and 150 feet in front. Figure 7 shows the field of view of some car at one time point of the data. One can see that, even though traffic lanes are clearly present, the cars' positions in the lanes vary significantly.

The factored PSR consists of one component model for each car. The observations of the component model were discretized, instantaneous, relative accelerations in the x (side-to-side) and y (forward-and-back) dimensions. Acceleration values were measured in feet per second squared. The y acceleration was discretized into bins of width 1; the data ranged from approximately -10 to 10. The x acceleration was discretized into 3 bins: $(-\infty, -20)$, $[-20, 20]$, and $(20, \infty)$. The threshold of 20 was selected after noting a strong correlation in the data between lane changes and spikes in x acceleration of magnitude 20 or greater.

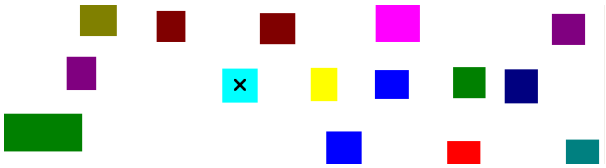
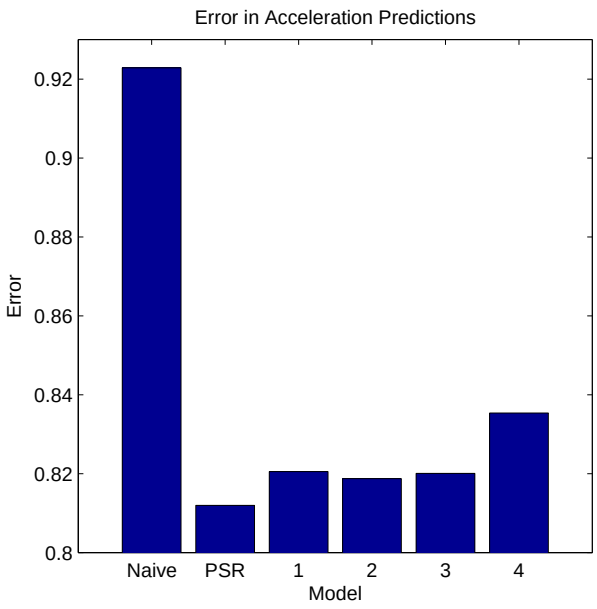


Figure 7: An overhead snapshot from a time step of the NGSIM traffic data. The rectangles are cars, and the direction of travel is toward the right of the page. The field of view is defined relative to the car marked by the 'x'.

The data that we used to train our model consists of trajectories of relative accelerations: for each reference car, we obtain one trajectory of data each time a car passes through the field of view (or if a car enters the field of view and remains there until the end of the data). We used each car in the training data as a reference car, adding all the associated trajectories into the training set for our PSR. We used the NGSIM I-80 traffic data from the 4:00–4:15 time period for training. In both testing and training, we subsampled the given data with a period of 1 second (rather than the 1/10 second interval of the data files).

For testing, we used a subset of the 5:00–5:15 data from the NGSIM I-80 traffic data; the testing data represents approximately one minute of real time. We evaluated our model on each trajectory for each reference car in the testing data. Our evaluation uses the same error measure as in the simulated traffic system, except that in this case the model predicts the likelihood of the actual next acceleration (rather than the actual next position, as used for evaluation with the simulated system). Predictions about position are easily calculated from the predictions of acceleration and the last observed position and velocity.

We compared our factored PSR against two other classes



Model	Error
PSR	0.81198427
2	0.81875077
3	0.82011353
1	0.82056525
4	0.83541472
Naive	0.92289034

Figure 8: NGSIM traffic data: mean squared error of the different models. The numbers 1 through 4 indicate the order of the respective Markov models.

of models. The first is a "naive" model that predicts that the acceleration in one second will be the same as the last observed acceleration. The second class of models are k^{th} order Markov models for $1 \leq k \leq 4$, which were trained on the same data we used to train the factored PSR.

Figure 8 compares the error for each of these models on our testing data. The naive model does much worse than the other models. The poor performance of the fourth-order Markov model is likely due to data sparsity in the training set (i.e. not all length-four histories are seen in the data). The other Markov models and the PSR obtained similar error, with the PSR achieving the lowest error.

6. DISCUSSION AND CONCLUSIONS

Assuming statistical independence is an approximation technique commonly used to make computation tractable (e.g. [4, 11, 10]). To our knowledge, our work is the first to use such an independence assumption with predictive state models. This is a first step in using graphical models techniques to scale PSRs to complex systems.

Examples of leveraging independence in the multi-agent literature include graphical games [6] and graphical multi-agent MDPs [3]. These approaches are most beneficial when each agent is only affected by a small subset of the other agents. This is in contrast to our traffic systems, where each agent (i.e. car) can be affected by any of the other agents, but only in certain contexts. Multi-agent influence diagrams (MAIDs) [7] could potentially exploit this context-specific independence; however, their focus is representing and solving games, where our focus is learning a model.

Our work has taken the first steps toward establishing a theory of approximation for PSRs. We have provided a bound on the error of an approximate PSR's state. We have introduced factored PSRs, a class of approximate PSRs that allow one to trade off model compactness and accuracy, and we have demonstrated the viability of learning factored PSRs for systems where learning an exact linear PSR is intractable.

Acknowledgments

This work is supported in part by the National Science Foundation under Grant Number IIS-0413004. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

7. REFERENCES

[1] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Proceedings of UAI 1998*, pages 33–42, 1998.

[2] E. Brockfeld and P. Wagner. Validating microscopic traffic flow models. In *Proceedings of Intelligent Transportation Systems Conference*, pages 1604–1608, 2006.

[3] D. Dolgov and E. Durfee. Graphical models in local, asymmetric multi-agent markov decision processes. In *Proceedings of AAMAS 2004*, pages 956–963, 2004.

[4] B. E. Engelhardt, M. I. Jordan, and S. E. Brenner. A graphical model for predicting protein molecular function. In *Proceedings of ICML 2006*, pages 297–304, 2006.

[5] M. R. James and S. Singh. Learning and discovery of predictive state representations in dynamical systems with reset. In *Proceedings of ICML 2004*, pages 417–424, 2004.

[6] M. J. Kearns, M. L. Littman, and S. P. Singh. Graphical models for game theory. In *Proceedings of UAI 2001*, pages 253–260, 2001.

[7] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. In *Proceedings of IJCAI 2001*, pages 1027–1036, 2001.

[8] M. L. Littman, R. S. Sutton, and S. Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems 14*, pages 1555–1561, 2002.

[9] X. Lu and B. Coifman. Highway traffic data sensitivity analysis. Technical Report UCB-ITS-PRR-2007-3, University of California, Berkeley, 2007.

[10] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In AAAI-98 Workshop on Learning for Text Categorization, 1998.

[11] R. Sandberg, G. Winberg, C.-I. Branden, A. Kaske, I. Ernberg, and J. Coster. Capturing Whole-Genome Characteristics in Short Sequences Using a Naive Bayesian Classifier. *Genome Res.*, 11(8):1404–1409, 2001.

[12] S. Singh, M. R. James, and M. Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of UAI 2004*, pages 512–519, 2004.

[13] U.S. Federal Highway Administration. Interstate 80 freeway dataset. http://www.tfhrc.gov/about/06137.htm.

[14] U.S. Federal Highway Administration. Next generation simulation project. http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.

[15] D. Wingate, V. Soni, B. Wolfe, and S. Singh. Relational knowledge with predictive state representations. In *Proceedings of IJCAI 2007*, pages 2035–2040, 2007.

[16] B. Wolfe, M. R. James, and S. Singh. Learning predictive state representations in dynamical systems without reset. In *Proceedings of ICML 2005*, pages 985–992, 2005.