Valid Parameters for Predictive State Representations

Britton Wolfe
Computer Science
Indiana University-Purdue University Fort Wayne
2101 Coliseum Blvd. E.
Fort Wayne, IN 46805
wolfeb@ipfw.edu

Abstract

Predictive state representations (PSRs) represent the state of a dynamical system as a set of predictions about future events. The parameters of a PSR model consist of several matrices and vectors, but not all values for those parameters result in valid PSR models. Our work starts with a general definition of what it means to be a valid PSR model and derives necessary and sufficient constraints for the model parameters to constitute a valid PSR. These same constraints also define the set of valid state vectors for a given PSR model, which we prove to be a convex set. We also derive a set of simplified constraints on the PSR parameters, and we prove that any PSR model has an equivalent parameterization that satisfies those simplified constraints. Lastly, we demonstrate one simple application of our constraints: preventing overflow or underflow of the PSR state as it changes over time.

Predictive state representations (PSRs) (Littman, Sutton, & Singh 2001) are a class of models that represent the state of a dynamical system as a set of predictions about future events. PSRs are capable of representing partially observable, stochastic dynamical systems, including any system that can be modeled by a finite partially observable Markov decision process (POMDP) (Singh, James, & Rudary 2004). There is evidence that predictive state is useful for generalization (Rafols *et al.* 2005) and helps to learn more accurate models than the state representation of a POMDP (Wolfe, James, & Singh 2005).

Both POMDPs and PSRs have constraints on the matrices that constitute their parameters. For POMDPs, the constraints on the parameters are well known: particular subsets of the parameters must form stochastic vectors (i.e., vector elements are between 0.0 and 1.0 and sum to 1.0). In contrast, the nature of the constraints on PSR parameters has not been explicitly studied. Thus, current algorithms for learning PSR models from data disregard the fact that there are constraints on the PSR parameters (e.g., James & Singh (2004), Wolfe, James, & Singh (2005), or Rosencrantz, Gordon, & Thrun (2004)). This can lead to invalid parameters and invalid state vectors that cause the PSR model to make predictions outside the range [0,1] of valid probabilities (Wolfe, James, & Singh 2005). This issue motivates the need to define the sets of valid parameters and state vectors so that future learning algorithms can find valid PSR parameters.

This work provides a set of necessary and sufficient constraints on the parameters for a class of PSRs. In addition, these same constraints define the set of valid state vectors for a PSR. Thus, the constraints could be used both for ensuring that a learning algorithm produces valid PSR parameters and for checking that the state vectors used during the model's operation are valid.

1 Background

This work deals with models of discrete-time dynamical systems which have a set of discrete actions A and a set of discrete observations O. At each time step x, the agent chooses some action $a_x \in A$ to execute and then receives some observation $o_x \in O$. A *history* is a possible sequence of actions and observations $a_1 o_1 a_2 o_2 \ldots a_\tau o_\tau$ from the beginning of time. A *test* is a sequence of possible future actions and observations $a_{\tau+1} o_{\tau+1} \ldots a_{\tau+k} o_{\tau+k}$, where τ is the current time step. The *prediction* for a test $t = a_{\tau+1} o_{\tau+1} \ldots a_{\tau+k} o_{\tau+k}$ from a history $h = a_1 o_1 \ldots a_\tau o_\tau$ 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

$$p(t|h) \stackrel{\text{def}}{=} \prod_{i=\tau+1}^{\tau+k} Pr(o_i | a_1, o_1, a_2, o_2, \ldots, a_i)$$

where a_x represents the event "a_x is the action at time x," and o_x represents the event "o_x is the observation at time x."

System-dynamics matrix: This work focuses upon linear PSRs (Littman, Sutton, & Singh 2001), a class of PSR models that we describe using the concept of a system-dynamics matrix, introduced by Singh, James, & Rudary (2004). A system-dynamics matrix $\mathcal{D}$ fully specifies a dynamical system, and 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 column is the prediction for that column's test from that row's history. Despite the fact that $\mathcal{D}$ has infinite size, it will have a finite rank n for a large class of systems, including POMDPs with a finite number of latent states; the rank n is no greater than the number of latent states in the POMDP (Singh, James, & Rudary 2004). For systems with a finite rank n, one can find a set Q of n linearly independent columns of $\mathcal{D}$ 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 is composed of the predictions for Q from the null history (the initial prediction vector), and the *parameters* $\mathcal{M}$ 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 parameters for a linear PSR are the m_t's for each one-step test (ao) and each one-step extension (aoq_i) of each core test $q_i \in Q$.

The state update procedure is to calculate $p(Q|hao)$ from $p(Q|h)$ after taking the action a and seeing the observation o from the history h. For any $q_i \in Q$, one can use the existing state vector $p(Q|h)$ and the parameters m_{ao} and m_{aoq_i} to calculate

$$p(q_i|hao) = \frac{p(aoq_i|h)}{p(ao|h)} = \frac{p^\top(Q|h)m_{aoq_i}}{p^\top(Q|h)m_{ao}}.$$

Let M_{ao} be the matrix with m_{aoq_i} as its i^{th} column. Then the state update in matrix form is

$$p^\top(Q|hao) = \frac{p^\top(Q|h)M_{ao}}{p^\top(Q|h)m_{ao}}.$$

From these parameters, one can calculate the m_t vector for any test $t = a_1o_1 \ldots a_ko_k$ as $M_{a_1o_1}M_{a_2o_2} \ldots M_{a_{k-1}o_{k-1}}m_{a_ko_k}$.

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

2 The Parameter Constraints

We call a linear PSR *valid* if its predictions satisfy the axioms of probability. The prediction for any test t from any history[2] h is equal to the ratio of two predictions from the null history ϕ:

$$p(t|h) = \frac{p(ht|\phi)}{p(h|\phi)}.$$

Thus, if a linear PSR's predictions from the null history satisfy the axioms of probability, then all of its predictions will satisfy the axioms of probability, making the model valid. This definition of valid does not place any constraints upon the state vector of the model (e.g., that the state vector should correspond to predictions about core tests). Thus, the results we develop apply to both linear PSRs and transformed PSRs (TPSRs) (Rosencrantz, Gordon, & Thrun 2004). TPSRs have the same parametric form as linear PSRs, but the state vector is allowed to be a linear function of core tests' predictions (rather than the predictions themselves, as in a linear PSR). Hereafter, we use "PSR" to refer generically to a linear PSR or a TPSR, and we use s_h to refer to the state vector of the PSR at history h. We use t to refer to an arbitrary test $a_1o_1 \ldots a_ko_k$ of length k, where the subscripts indicate the time step relative to the current time.

Given parameters $\mathcal{M}$ and initial state s_ϕ, the prediction for t from ϕ is defined as

$$\begin{aligned} p(t|\phi) &\stackrel{\text{def}}{=} s_\phi^\top m_t \\ &= s_\phi^\top M_{a_1o_1}M_{a_2o_2} \ldots M_{a_{k-1}o_{k-1}}m_{a_ko_k}. \end{aligned}$$

We say that the parameters $\mathcal{M}$ are *valid* if and only if there exists some vector s_ϕ such that the predictions given $\mathcal{M}$ and s_ϕ satisfy the axioms of probability.

There are three sets of constraints on the predictions in order for them to be valid. The domains of the variables in these (and subsequent) constraints are as follows: k : integers; t : sequences of length k; a variables: actions; o variables: observations. The first set of constraints requires the predictions to be non-negative:

$$\forall k \geq 1,\ t, \qquad p(t|\phi) \geq 0. \tag{1}$$

Second, the predictions for all tests with a particular action sequence must sum to one, since those predictions form a multinomial distribution over the observation sequences of the same length as the action sequence:

$$\forall k \geq 1,\ a_1a_2 \ldots a_k, \quad \sum_{o_1o_2 \ldots o_k} p(a_1o_1 \ldots a_ko_k|\phi) = 1. \tag{2}$$

These constraints are necessary and sufficient to ensure that the predictions for any action sequence satisfy the

[2]Predictions from impossible histories are not defined.

axioms of probability. However, predictions for different action sequences are related when one action sequence is a prefix of the other. To ensure consistency among such predictions, Equation 3 requires that for each sequence t, the predictions (given some action a) for all possible observations o after t sum to the prediction for t itself:

$$\forall k \geq 0,\ t,\ a, \qquad p(t|\phi) = \sum_o p(tao|\phi). \quad (3)$$

Equations 1, 2 and 3 are necessary and sufficient constraints for a set of parameters to be valid. However, each set of constraints is infinite, since t and k can each take on an infinite number of values. The remainder of this section reduces each of Equations 2 and 3 to an equivalent, *finite* set of constraints, given Equation 1.

Theorem 1. *Given Equation 3, Equation 2 is equivalent to*

$$\forall a, \qquad s_\phi^\top \sum_o m_{ao} = 1 \quad (4)$$

Proof. We start with the left hand side of Equation 2 and show that it reduces to the left hand side of Equation 4 (given Equation 3). Thus, the left-hand sides will equal 1 for exactly the same parameters.

For any $k \geq 1$ and any $a_1 \dots a_k$,

$$\begin{aligned} &\sum_{o_1 \dots o_k} p(a_1 o_1 a_2 o_2 \dots a_k o_k|\phi) \\ &= \sum_{o_1 \dots o_{k-1}} \sum_{o_k} s_\phi^\top M_{a_1 o_1} \dots M_{a_{k-1} o_{k-1}} m_{a_k o_k} \\ &= \sum_{o_1 \dots o_{k-1}} s_\phi^\top M_{a_1 o_1} \dots M_{a_{k-2} o_{k-2}} m_{a_{k-1} o_{k-1}} \end{aligned}$$

where the last step applies Equation 3 for $t = a_1 o_1 a_2 o_2 \dots a_{k-1} o_{k-1}$. Repeatedly applying Equation 3 for each prefix $a_1 o_1 \dots a_{k-i} o_{k-i}$ reduces the sum to

$$\sum_{o_1} s_\phi^\top m_{a_1 o_1},$$

which is the left hand side of Equation 4. □

The second part of this section reduces Equation 3 to a finite set of constraints. In particular, the upcoming Theorem 3 proves that checking Equation 3 for a *finite* set of sequences (instead of all sequences t) is sufficient, given the other constraints. The constraints of Equation 3 for the sequences $t = hao$ that are one-step extensions of a single sequence h are

$$\forall a, o, a', \qquad p(hao|\phi) = \sum_{o'} p(haoa'o'|\phi). \quad (5)$$

To describe the set of sequences one needs to check, we use the following definitions:

$$u_h^\top \stackrel{\text{def}}{=} s_\phi^\top M_{a_1 o_1} M_{a_2 o_2} \dots M_{a_k o_k}$$

for any sequence $h = a_1 o_1 \dots a_k o_k$. Two properties of the u vectors worth noting are $u_\phi = s_\phi$ and $p(hao|\phi) = u_h^\top m_{ao}$. We define a set of histories $\{h_1, \dots, h_j\}$ to be a *history basis* if and only if $u_{h_1}, \dots, u_{h_j}$ are linearly independent and, for any h, u_h is linearly dependent upon $u_{h_1}, \dots, u_{h_j}$. We use H^* to denote an arbitrary history basis.

Lemma 2. *Equation 5 holds for all histories if and only if it holds for all $h \in H^*$.*

Proof. The "only if" direction is trivial ("all histories" includes those in H^*). For the "if" direction, let $H^* = \{h_1, \dots, h_j\}$, and let U be the matrix with i^{th} row equal to $u_{h_i}^\top$. For any history h, there exists a vector w_h such that $u_h^\top = w_h^\top U$ (because u_h is linearly dependent upon the rows of U).

Then for any a, o,

$$\begin{aligned} p(hao|\phi) &= s_\phi^\top M_{a_1 o_1} M_{a_2 o_2} \dots M_{a_k o_k} m_{ao} \\ &= u_h^\top m_{ao} = w_h^\top U m_{ao} \\ &= w_h^\top \begin{bmatrix} - & u_{h_1}^\top & - \\ - & u_{h_2}^\top & - \\ & \vdots & \\ - & u_{h_j}^\top & - \end{bmatrix} m_{ao} = w_h^\top \begin{bmatrix} p(h_1 ao|\phi) \\ p(h_2 ao|\phi) \\ \vdots \\ p(h_j ao|\phi) \end{bmatrix} \end{aligned}$$

by definition of the u's. Using the premise that Equation 5 holds for all $h_i \in H^*$, this is equal to

$$\begin{aligned} &= \sum_{o'} w_h^\top \begin{bmatrix} p(h_1 aoa'o'|\phi) \\ p(h_2 aoa'o'|\phi) \\ \vdots \\ p(h_j aoa'o'|\phi) \end{bmatrix} \\ &= \sum_{o'} w_h^\top U M_{ao} m_{a'o'} = \sum_{o'} u_{hao}^\top m_{a'o'}, \end{aligned}$$

where the last step uses the fact that $u_{hao}^\top = u_h^\top M_{ao} = (w_h^\top U) M_{ao}$. The resulting sum is just $\sum_{o'} p(haoa'o'|\phi)$, which completes the proof that Equation 5 for all $h \in H^*$ implies Equation 5 for all histories h. □

The following theorem reduces Equation 3 to a finite set of constraints. The set is finite because H^* is never larger than the length of the u vectors (which is also the length of the m_{ao} vectors).[3]

Theorem 3. *Given Equation 4, Equation 3 is equivalent to Equation 5 for all $h \in H^*$.*

Proof. Lemma 2 proved that Equation 5 for all $h \in H^*$ is equivalent to Equation 5 for all histories. Equation 5 for all h (including the null history) is exactly the same

[3]This is because one cannot construct a set of more than n linearly independent vectors when the vectors have length n.

as Equation 3 for all t of length at least 1. The empty test is the only test of length less than 1, and Equation 4 ensures that Equation 3 is satisfied for that test. □

Theorems 1 and 3 prove that the following constraints are necessary and sufficient for PSR parameters to be valid.

$$\forall k \geq 1,\ t, \qquad p(t|\phi) \geq 0 \quad (1)$$

$$\forall a, \qquad s_\phi^\top \sum_o m_{ao} = 1 \quad (4)$$

$$\forall h \in H^*, a, o, a', \quad p(hao|\phi) = \sum_{o'} p(haoa'o'|\phi) \quad (6)$$

3 Simplification of Constraints

In this section, we replace Equation 6 with a set of constraints that does not depend upon a history basis H^*. This leads to the following set of simplified constraints:

$$\forall k \geq 1,\ t, \qquad p(t|\phi) \geq 0 \quad (1)$$

$$\forall a, \qquad s_\phi^\top \sum_o m_{ao} = 1 \quad (4)$$

$$\forall a, o, a', \qquad m_{ao} = M_{ao} \sum_{o'} m_{a'o'}. \quad (7)$$

The simplified constraints are still sufficient for determining valid PSR parameters, but they are not necessary. Nevertheless, Theorem 8 (in Section 3.2) proves that for *any* valid PSR parameters, there exists a set of PSR parameters of the same size that make the same predictions and satisfy the simplified constraints. Thus, one loses no modeling power by using the simplified constraints, even though they are not strictly necessary.

We define the *size* of a set of parameters as the length of the m vectors (which is also the length of the u vectors and the dimensions of the M matrices). Using this definition, the following theorem establishes a relationship between Equations 6 and 7.

Theorem 4. *For a set of parameters with size n, if the size of a history basis H^* is n, then Equation 7 is equivalent to Equation 6.*

Proof. For a given h, Equation 6 is equal to

$$\forall a, o, a', \qquad u_h^\top m_{ao} = \sum_{o'} u_h^\top M_{ao} m_{a'o'}.$$

For a history basis $H^* = \{h_1, \ldots, h_n\}$, let U be the $n \times n$ matrix with i^{th} row equal to $u_{h_i}^\top$. Then Equation 6 in matrix form is equal to

$$\forall a, o, a', \qquad U m_{ao} = U M_{ao} \sum_{o'} m_{a'o'}.$$

Left-multiplying by U^{-1} yields Equation 7. (By definition of history basis, U is invertible.) □

To prove that replacing Equation 6 with Equation 7 loses no modeling power (Theorem 8), we take the following steps: Section 3.1 proves that there always exists an equivalent set of parameters with size equal to the size of a history basis; then Section 3.2 uses those equivalent parameters to construct equivalent parameters that satisfy Equation 7 and have the same size as the original parameters.

3.1 Existence of Minimal Parameters

The remaining discussion uses some additional definitions and notation. For brevity, we include the initial state vector in the definition of a set of parameters for the rest of Section 3. We use $\mathcal{M}$ to refer to a set of parameters $\{u_\phi, m_{ao}, M_{ao} : \forall a, o\}$, and we use $\mathcal{Z}$ to refer to another set of parameters $\{y_\phi, z_{ao}, Z_{ao} : \forall a, o\}$, where the z's and Z's correspond to m's and M's. Two sets of parameters are *equivalent* if they make the same predictions for any test; note that equivalent parameters need not be the same size. A *test basis* for a set of parameters $\mathcal{M}$ is a set of tests $\{t_1, \ldots, t_k\}$ such that the vectors $\{m_{t_1}, \ldots, m_{t_k}\}$ are linearly independent and form a linear basis of the m_t vectors (for all t). The *rank* of parameters is the smaller of the history basis size and the test basis size. Parameters are *minimal* if their size equals their rank. The use of the term "minimal" is justified by the fact that when a set of parameters is minimal, there is no equivalent set of parameters of smaller size (proven in the Appendix).

The primary result of this section is that, for any set of parameters, there exists an equivalent, minimal set of parameters (Theorem 7). We use the following two lemmas to prove this result.

Lemma 5. *Let $\mathcal{M}$ be a set of parameters with size n that has a history basis H of size k. Then there exists an equivalent set of parameters $\mathcal{Z}$ of size k.*

Proof. The first piece of the proof defines $\mathcal{Z}$ in terms of $\mathcal{M}$. Let $H = \{h_1, \ldots, h_k\}$ be a history basis for $\mathcal{M}$, and let U be the matrix with i^{th} row equal to $u_{h_i}^\top$. Since U has rank k, it has k linearly independent columns, which we assume are the first k columns (without loss of generality). Let Ψ be the matrix equal to the first k columns of U. Because the columns of Ψ form a basis of $\mathbb{R}^k$, there exists a matrix X such that

$$\Psi X = U.$$

For any history h, let $w_h^\top$ be the vector such that $u_h^\top = w_h^\top U$. Such a w_h exists because the rows of U are a basis for all the u_h.

The parameters $\mathcal{Z}$ are

$$y_\phi^\top \stackrel{\text{def}}{=} w_\phi^\top \Psi$$

$$z_{ao} \stackrel{\text{def}}{=} X m_{ao}$$

$$Z_{ao} \stackrel{\text{def}}{=} X M_{ao} X^\dagger$$

where $X^\dagger$ is the Moore-Penrose pseudoinverse of X.

The $\mathcal{Z}$ have size k by construction, but it remains to show that they are equivalent to $\mathcal{M}$. To do so, let y_h be the analogue to the u_h in the parameters $\mathcal{M}$. That is, for any h, a, o, the vector y_{hao} is defined recursively as $y_h Z_{ao}$; the base case y_ϕ is defined as part of $\mathcal{Z}$. To prove parameter equivalence, we first show that $y_h^\top = w_h^\top \Psi$ for any history h, using induction on the length of h. The base case $h = \phi$ follows from the definition of y_ϕ. The inductive step is to prove that $y_{hao}^\top = w_{hao}^\top \Psi$ for an arbitrary hao:

$$\begin{aligned} y_{hao}^\top &= y_h^\top Z_{ao} \\ &= w_h^\top \Psi X M_{ao} X^\dagger \qquad \text{(by inductive hyp.)} \\ &= w_h^\top U M_{ao} X^\dagger \\ &= u_h^\top M_{ao} X^\dagger \\ &= u_{hao}^\top X^\dagger \\ &= w_{hao}^\top U X^\dagger \\ &= w_{hao}^\top \Psi X X^\dagger \\ &= w_{hao}^\top \Psi. \end{aligned}$$

The last step here uses the fact that $(XX^\dagger)^\top = (X^\top)^\dagger X^\top$ is the matrix of the orthogonal projection onto the image of X. Since X has full row rank, this image is $\mathbb{R}^k$, so $(XX^\dagger)^\top = I_k = XX^\dagger$.

We now prove that $\mathcal{Z}$ and $\mathcal{M}$ make the same predictions. For any h, a, o, the prediction $p(hao|\phi)$ according to $\mathcal{Z}$ is $y_h^\top z_{ao} = w_h^\top \Psi X m_{ao}$ (using $y_h^\top = w_h^\top \Psi$). This is equal to $w_h^\top U m_{ao} = u_h^\top m_{ao}$, which is the prediction $p(hao|\phi)$ according to $\mathcal{M}$. Thus, $\mathcal{Z}$ and $\mathcal{M}$ are equivalent. □

Lemma 6. *Let $\mathcal{M}$ be a set of parameters with size n with a test basis T of size k. Then there exists an equivalent set of parameters $\mathcal{Z}$ of size k.*

Proof. The proof is similar to that of Lemma 5. The first piece of the proof defines $\mathcal{Z}$ in terms of $\mathcal{M}$. Let $T = \{t_1, \dots, t_k\}$ be a history basis for $\mathcal{M}$, and let M be the matrix with i^{th} column equal to m_{t_i}. Since M has rank k, it has k linearly independent rows, which we assume are the first k rows (without loss of generality). Let Ψ be the matrix equal to the first k rows of M. Because the rows of Ψ form a basis of $\mathbb{R}^k$, there exists a matrix X such that

$$X\Psi = M.$$

For any test t, let w_t be the vector such that $m_t = M w_t$. Such a w_t exists because the columns of M are a basis for all the m_t.

The parameters $\mathcal{Z}$ are

$$\begin{aligned} y_\phi^\top &\stackrel{\text{def}}{=} u_\phi^\top X \\ z_{ao} &\stackrel{\text{def}}{=} \Psi w_{ao} \\ Z_{ao} &\stackrel{\text{def}}{=} X^\dagger M_{ao} X \end{aligned}$$

where $X^\dagger$ is the Moore-Penrose pseudoinverse of X.

The $\mathcal{Z}$ have size k by construction, but it remains to show that they are equivalent to $\mathcal{M}$. To do so, let z_t be the analogue to the m_t in the old parameters. That is, the vector z_{aot} is defined recursively as $Z_{ao} z_t$. To prove parameter equivalence, we first show that $z_t = \Psi w_t$ for any test t, using induction on the length of t. The base case $t = ao$ follows from the definition of z_{ao}. The inductive step is to prove that $z_{aot} = \Psi w_{aot}$ for an arbitrary aot:

$$\begin{aligned} z_{aot} &= Z_{ao} z_t \\ &= X^\dagger M_{ao} X \Psi w_t \qquad \text{(by inductive hyp.)} \\ &= X^\dagger M_{ao} M w_t \\ &= X^\dagger M_{ao} m_t \\ &= X^\dagger m_{aot} \\ &= X^\dagger M w_{aot} \\ &= X^\dagger X \Psi w_{aot} \\ &= \Psi w_{aot}. \end{aligned}$$

The last step here uses the fact that $X^\dagger X$ is the matrix of the orthogonal projection onto the image of $X^\top$. Since $X^\top$ has full row rank, this image is $\mathbb{R}^k$, so $X^\dagger X = I_k$.

We now prove that $\mathcal{Z}$ and $\mathcal{M}$ make the same predictions. For any test t, the prediction $p(t|\phi)$ according to $\mathcal{Z}$ is $y_\phi^\top z_t = u_\phi^\top X \Psi w_t$ (using $z_t = \Psi w_t$). This is equal to $u_\phi^\top M w_t = u_\phi^\top m_t$, which is the prediction $p(t|\phi)$ according to $\mathcal{M}$. Thus, $\mathcal{Z}$ and $\mathcal{M}$ are equivalent.

□

Theorem 7. *For any set of parameters there is an* equivalent, minimal *set of parameters.*

Proof. We prove this by describing a procedure to iteratively transform a set of parameters into an equivalent, minimal set of parameters. Let $\mathcal{Z}$ be the parameters on the current iteration, and let k be the smaller of (1) the size of a history basis for $\mathcal{Z}$ and (2) the size of a test basis for $\mathcal{Z}$. If k is equal to the size of $\mathcal{Z}$, then $\mathcal{Z}$ are minimal parameters, so the procedure is done. Otherwise, the procedure transforms $\mathcal{Z}$ into equivalent parameters of size k using the method of Lemma 5 (if the history basis is smaller) or Lemma 6 (if the test basis is smaller). Those transformed parameters are used in the next iteration.

The correctness of this procedure follows from these facts: it only returns when it finds minimal parameters;

it always returns (because the size of the parameters is strictly decreasing on each iteration); and equivalence to the original parameters is preserved with every transformation. □

Because minimal parameters satisfy the condition of Theorem 4 (i.e., the size of a history basis is equal to the size of the parameters), minimal parameters satisfy the simplified constraints (Equations 1, 4, and 7). Thus, *for every set of valid PSR parameters, there is an equivalent set of parameters that satisfies the simplified constraints.*

3.2 Same-Size Equivalent Parameters

At this point, the only parameters we have proven to satisfy the simplified constraints are minimal parameters. The following theorem goes beyond this, proving that there are non-minimal parameters that satisfy the simplified constraints. Specifically, for *any* valid parameters of *any* size, there exists equivalent parameters that satisfy the simplified constraints and have *the same size* as the original parameters. Thus, one need not change the size of the parameters when searching to satisfy the simplified constraints, even if the original parameters are not minimal.

Theorem 8. *For* any *valid parameters, there exist equivalent parameters of the same size that satisfy Equation 7.*

Proof. Let n be the size of the original parameters. There exists some equivalent minimal parameters $\mathcal{M} = \{u_\phi, m_{ao}, M_{ao} : \forall a, o\}$ (Theorem 7), which satisfy Equation 7. This proof uses the minimal parameters to construct a set of equivalent parameters $\mathcal{Z} = \{y_\phi, z_{ao}, Z_{ao} : \forall a, o\}$ that have size n and satisfy Equation 7. These parameters are just the minimal parameters padded with 0's. Specifically,

$$y_\phi \stackrel{\text{def}}{=} \begin{bmatrix} u_\phi \\ 0 \end{bmatrix} \quad z_{ao} \stackrel{\text{def}}{=} \begin{bmatrix} m_{ao} \\ 0 \end{bmatrix} \quad Z_{ao} \stackrel{\text{def}}{=} \begin{bmatrix} M_{ao} & 0 \\ 0 & 0 \end{bmatrix}.$$

The dimensions of the 0 vectors and matrices are determined by the fact that these new parameters have size n. One property of these parameters worth noting explicitly is that for any sequence t,

$$z_t = \begin{bmatrix} m_t \\ 0 \end{bmatrix}.$$

This can be easily verified from the definition of $\mathcal{Z}$.

There are two things to prove about $\mathcal{Z}$:

- *The $\mathcal{Z}$ parameters are equivalent to the minimal parameters $\mathcal{M}$.* For any test t, the prediction $p(t|\phi)$ according to $\mathcal{Z}$ is

$$y_\phi^\top z_t = [u_\phi^\top \quad 0] \begin{bmatrix} m_t \\ 0 \end{bmatrix} = u_\phi^\top m_t$$

  which is equivalent to the prediction made by $\mathcal{M}$.

- *The $\mathcal{Z}$ parameters satisfy Equation 7:*

$$\forall a, o, a', \qquad z_{ao} = Z_{ao} \sum_{o'} z_{a'o'} \tag{7}$$

  This can be seen with the following derivation:

$$\begin{aligned} Z_{ao} \sum_{o'} z_{a'o'} &= \begin{bmatrix} M_{ao} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \sum_{o'} m_{a'o'} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} M_{ao} \sum_{o'} m_{a'o'} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} m_{ao} \\ 0 \end{bmatrix}. \end{aligned}$$

  This last step uses the fact that the minimal parameters $\mathcal{M}$ satisfy Equation 7. The resulting vector is equal to z_{ao}, by definition.

□

Theorem 8 proves that even though the simplified constraints (Equations 1, 4, and 7) are not strictly necessary, one does not lose modeling power by requiring that the PSR parameters satisfy those constraints.

3.3 The Non-negativity Constraints

While Equations 4 and 7 specify a finite set of constraints, there remains one non-negativity constraint for each test t (Equation 1): $u_\phi^\top m_t \geq 0$. For a fixed set of parameters, these non-negativity constraints define a cone of acceptable initial states u_ϕ (with a point at the origin). The non-negativity constraint for each test t (potentially) adds a face to this cone. We are exploring a connection between the non-negativity constraints and the geometric problem of determining if a cone is closed under matrix multiplication (the matrices in our case would be the M_{ao}'s). As of yet, we have not found a way to reduce Equation 1 to a finite set of constraints in general. However, for parameters that satisfy the k^{th}-order Markov property, one only needs to check the non-negativity constraints for tests of length k or less (proof omitted). Thus, in practice, one would approximate the full set of non-negativity constraints by checking the subset of constraints for all tests shorter than some fixed length. Checking non-negativity constraints for long tests is unlikely to be helpful because as the length of the test t increases, the number of matrix multiplications to compute m_t increases, magnifying numerical precision errors in the parameters.

4 Constraints for State Vectors

Up to this point, we have described Equations 1, 4, and 6 as constraints on PSR parameters. However, the same equations can also be viewed as constraints on the set of valid state vectors for a given set of parameters. In this section, a set of parameters $\mathcal{M}$ will refer to the M_{ao} matrices and m_{ao} vectors, but not the initial state u_ϕ. We say that a vector u_ϕ is a *valid state vector* for a given a set of parameters $\mathcal{M}$ if u_ϕ and $\mathcal{M}$ together satisfy Equations 1, 4, and 6. While these equations were derived in terms of the initial state vector u_ϕ and predictions from the null history, they are also necessary and sufficient constraints for the PSR state vector at any history h. That is, the constraints on the set of all predictions $\{p(t|\phi) : \forall t\}$ from the null history also apply to the predictions $\{p(t|h) : \forall t\}$ from any history h.

Constraints on the set of valid state vectors could be used after the PSR parameters have been learned and an agent is using the model to make predictions. In particular, due to errors in the parameters, the state of the PSR can drift outside the space of valid state vectors. By defining the space of valid state vectors, an agent could detect such drifting and project the state vector back into the valid space. (This is analogous to renormalizing the belief state of a POMDP model.) This projection can utilize the fact that the space of valid state vectors for a PSR model is convex, as the following theorem proves.

Theorem 9. *For any set of parameters $\mathcal{M}$, the space of PSR state vectors that are valid for those parameters is convex.*

Proof. This theorem follows immediately from the constraints of Equations 1, 4, and 6. Each of those constraints is *linear* in the state vector u_ϕ, because any prediction $p(\cdot|\phi)$ is a linear function of u_ϕ. As mentioned above, the space of valid initial state vectors u_ϕ is also the space of valid state vectors at any history. This space is defined by linear equations and linear inequalities. Therefore, it is a convex space. □

4.1 Experiments

This section presents results from experiments that adjust the PSR state vector after each state update. The adjustment is based upon the constraints

$$\forall a, \qquad s_\phi^\top \sum_o m_{ao} = 1. \tag{4}$$

After every time step, we take the updated state s and compute the adjusted state s_+ as

$$s_+ = \frac{s}{\frac{1}{|A|}\sum_a s^\top \sum_o m_{ao}}. \tag{8}$$

This adjustment ensures that the sum of all the predictions for length-one tests will be $|A|$, which is implied by Equation 4.

This scaling of the state vector prevents the entries in the state vector from overflowing or underflowing (i.e., magnitudes growing too large or all dwindling to 0.0) as the state is updated after each time step. Once the state overflows or underflows, the model is unable to make predictions. To test the effect of adjusting the state vector, we learned linear PSR models using the suffix-history algorithm and the POMDP domains from Wolfe, James, & Singh (2005). For each domain, we learned 30 PSR models for each of several amounts of training data, ranging from 10 time steps up through 40 million time steps. After learning the models, we evaluated their predictions for 50000 time steps. Table 1 presents the fraction of the models for each domain that either overflowed or underflowed their state vectors during the evaluation. As seen in the table, the overflow/underflow problem happens a significant fraction of the time. In contrast, *when scaling the state vectors* of the same models according to Equation 8, *none of the models suffered from overflow or underflow*.

Domain	Linear PSRs	TPSRs
tiger	0.079710	0.318841
paint	0.040580	0.275362
shuttle	0.392754	0.956522
network	0.459420	0.739130
cheese	0.411594	0.521739
bridge	0.078261	0.594203
4x3 maze	0.214493	0.666667

Table 1: The fraction of learned models that suffered from overflow or underflow during the 50000 time steps of model evaluation. Adjusting the state vector based upon our derived constraints eliminated *all* overflow or underflow for both linear PSRs and TPSRs.

Because each element of a linear PSR state should be a probability, another way to prevent the overflow/underflow problem is to clip the entries of the state vector in the range [ϵ, 1.0] for some very small ϵ. Clipping the state vector in this way is a limited application of the non-negativity and marginal probability constraints. As when scaling the state vector, clipping the state vector after every time step also prevents overflow or underflow in all models.

We used a mean-squared error measure to evaluate the models' predictions, the same measure used by Wolfe, James, & Singh (2005). For small amounts of training data, clipping led to more accurate predictions than scaling in three of the seven domains (up to one order of magnitude lower error). Otherwise, clipping and scaling were comparable. Combining both methods did not produce any significant benefit over clipping.

As mentioned early in Section 2, the constraints we derived apply to TPSRs as well as to linear PSRs. With TPSRs, the clipping method for adjusting state is not applicable, since the state vector is a linear function of predictions. However, the constraints of Equation 4 still apply to TPSRs, so scaling the state vector is a valid method for preventing overflow/underflow. We ran the same experiments with the TPSRs as with the linear PSRs. Table 1 presents the fraction of TPSR models for each domain that suffered from overflow or underflow when the state vector was not adjusted. The overflow/underflow problem is even more prevalent with TPSRs than with linear PSRs, yet when scaling the state vectors of the same TPSR models, none of the models suffered from overflow or underflow.[4]

5 Summary and Future Work

We derived constraints that are necessary and sufficient for parameters of a linear PSR or TPSR model to make valid predictions. For a given set of parameters, these same constraints also define the set of valid state vectors, which we proved to be a convex set. We also derived a set of simplified constraints, proving that, for any valid parameters, there exist equivalent parameters of the same size that satisfy the simplified constraints. We demonstrated that using our constraints to adjust the state vectors of linear PSRs and TPSRs prevents overflow and underflow that otherwise occurs frequently.

Another possible use of our constraints would be to incorporate them into a learning algorithm for PSR parameters. Current algorithms ignore any constraints on the PSR parameters, in part because the constraints were not well-defined prior to this work (e.g., James & Singh (2004), Wolfe, James, & Singh (2005), or Rosencrantz, Gordon, & Thrun (2004)). The current algorithms use matrix pseudo-inverses to find the parameters that best fit a set of estimated predictions. However, due to noise in the estimated predictions, the best-fit parameters do not necessarily constitute a valid PSR model. In future work, our constraints could be incorporated into a learning algorithm that solves a constrained optimization problem: find the parameters that best fit the estimated predictions *subject to* the constraints. While this would not be as simple as computing a matrix pseudo-inverse, it would potentially yield more accurate models.

Appendix

This appendix proves that minimal parameters have no equivalent set of parameters of smaller size (Theorem 11). The following lemma is used in that proof.

Lemma 10. *A minimal set of parameters with size n has a history basis of size n and a test basis of size n.*

[4]The TPSR error is comparable to the error of the linear PSRs reported by Wolfe, James, & Singh (2005). Their error was sometimes lower than our linear PSR error because, in addition to clipping the state vector, they hand-selected the learning algorithm's rank-estimation parameter. Our experiments used a parameter value automatically chosen by cross-validation.

Proof. A history basis can have size no less than n: such a history basis would imply that the rank of the parameters was less than n, making the parameters non-minimal. A history basis can have size no greater than n because the u_h vectors are n-dimensional, and there does not exist a set of more than n linearly independent vectors in $\mathbb{R}^n$. Thus, the history basis must have size exactly n. The same argument applies to the test basis, because the m-vectors are also n-dimensional. □

Theorem 11. *If a set of parameters $\mathcal{M}$ is minimal, there is no equivalent set of parameters of smaller size.*

Proof. Let $\mathcal{M} = \{u_\phi, m_{ao}, M_{ao} : \forall a, o\}$ be a minimal set of parameters with size n, and let $H = \{h_1, \dots, h_n\}$ and $T = \{t_1, \dots, t_n\}$ be history and test bases, respectively. Let U be the matrix with i^{th} row equal to $u_{h_i}^\top$, and let M be the matrix with i^{th} column equal to m_{t_i}. Finally, define the matrix

$$P \stackrel{\text{def}}{=} \begin{bmatrix} p(h_1t_1|\phi) & p(h_1t_2|\phi) & \cdots & p(h_1t_k|\phi) \\ p(h_2t_1|\phi) & p(h_2t_2|\phi) & \cdots & p(h_2t_k|\phi) \\ \vdots & \vdots & \ddots & \vdots \\ p(h_kt_1|\phi) & p(h_kt_2|\phi) & \cdots & p(h_kt_k|\phi) \end{bmatrix}$$

where the predictions are computed according to the parameters $\mathcal{M}$. Then

$$UM = P.$$

Since M is an $n \times n$, full-rank matrix (Lemma 10), it is invertible, so

$$U = PM^{-1}.$$

Then P must also have rank n, because $\text{rank}(A\ B) \leq \min(\text{rank}(A), \text{rank}(B))$ for any matrices A and B. Specifically, U has rank n (Lemma 10), so the rank of P must be at least n. Since it is an $n \times n$ matrix, it has rank exactly n.

Consider a set of equivalent parameters $\mathcal{Z} = \{y_\phi, z_{ao}, Z_{ao} : \forall a, o\}$ with size n'. We show that n' is no less than n, the size of the minimal parameters. Let Y be the matrix with rows equal to the y_{h_i} vectors, and let Z be the matrix with columns equal to the z_{t_i} vectors. Because $\mathcal{M}$ and $\mathcal{Z}$ are equivalent,

$$YZ = P.$$

Since P has rank n, each of Y and Z must also have rank at least n. This means that the size n' of the parameters $\mathcal{Z}$ must be at least n (because each of Y and Z is $n' \times n'$). This completes the proof that the equivalent parameters have size no less than n. □

References

James, M. R., and Singh, S. 2004. Learning and discovery of predictive state representations in dynamical systems with reset. In Brodley, C. E., ed., *Proceedings of the* 21^{st} *International Conference on Machine Learning*, 417–424. ACM.

Littman, M. L.; Sutton, R.; and Singh, S. 2001. Predictive representations of state. In Dietterich, T. G.; Becker, S.; and Ghahramani, Z., eds., *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, 1555–1561. MIT Press.

Rafols, E. J.; Ring, M. B.; Sutton, R.; and Tanner, B. 2005. Using predictive representations to improve generalization in reinforcement learning. In Kaelbling, L. P., and Saffiotti, A., eds., *Proceedings of the* 19^{th} *International Joint Conference on Artificial Intelligence*, 835–840. Professional Book Center.

Rosencrantz, M.; Gordon, G.; and Thrun, S. 2004. Learning low dimensional predictive representations. In Brodley, C. E., ed., *Proceedings of the* 21^{st} *International Conference on Machine Learning*, 88–95. ACM.

Singh, S.; James, M. R.; and Rudary, M. 2004. Predictive state representations: A new theory for modeling dynamical systems. In Chickering, M., and Halpern, J., eds., *Proceedings of the* 20^{th} *Conference on Uncertainty in Artificial Intelligence*, 512–519. AUAI Press.

Wolfe, B.; James, M. R.; and Singh, S. 2005. Learning predictive state representations in dynamical systems without reset. In Raedt, L. D., and Wrobel, S., eds., *Proceedings of the* 22^{nd} *International Conference on Machine Learning*, 985–992. ACM.