Building Incomplete but Accurate Models

Erik Talvitie, Britton Wolfe and **Satinder Singh**
Computer Science & Engineering
University of Michigan
{etalviti, bdwolfe, baveja}@umich.edu

Abstract

We consider an agent that seeks to make abstract predictions about the world by only distinguishing between certain features of observations. Making accurate abstract predictions of this form may not require a fully detailed model of the world, though in general it will require that the agent make *some* finer distinctions than those that interest it. We assume the agent has a partition of the observation space of a dynamical system induced by the features of interest. The goal of this paper is to find a minimal refinement of that partition such that a model of the refined system will make accurate predictions with respect to the features of interest. We provide algorithms and worst-case bounds on the difficulty of performing this task. Our results apply generally to all discrete dynamical systems.

1 Introduction and Preliminaries

An important capability for an artificial agent is that of building a model in order to answer questions about the world. An agent attempting to model a sufficiently complex environment may wish to limit the questions it will ask in hopes of simplifying its modeling task. This may be because the agent actually only requires the answers to a restricted set of questions or perhaps because the environment is so complex that the agent is computationally incapable of building a complete model.

In this work, we will consider discrete dynamical systems and we will think of "questions" as predictions about future events. In particular, we imagine there exists a finite set of actions $\mathcal{A}$ and a finite set of observations $\mathcal{O}$. At each time step i, an agent selects an action $a_i \in \mathcal{A}$ and the environment produces an observation $o_i \in \mathcal{O}$. We call a sequence of future actions and a sequence of observations that could result $t = a_1 o_1 ... a_k o_k$ a *test*. In the event that the agent performs the sequence of actions and actually observes the sequence of observations in a test, we say that the test *succeeded*. The agent uses its model of the world to make predictions of whether tests will succeed. That is, it asks questions of the form "If I perform the sequence of actions $a_1, a_2, ..., a_k$, what is the probability that I will see the sequence of observations $o_1, o_2, ..., o_k$?" A model that can answer all such questions can make *any* conditional prediction of the future (Littman, Sutton, & Singh 2002). After a sequence of past actions and observations (which we call a *history*) $h = a_1 o_1 ... a_j o_j$, we write the *true* prediction of test $t = a_{j+1} o_{j+1} ... a_{j+k} o_{j+k}$, determined by the physical system, as

$$p(t|h) \stackrel{\text{def}}{=} \Pr(o_{j+1} ... o_{j+k} | a_1 ... a_{j+k}, o_1 ... o_j). \quad (1)$$

We refer to a model M's prediction (which may or may not be correct) of test t at history h with $p_M(t|h)$.

We will let the set of all tests $\mathcal{T}$ be the set of all alternating sequences of actions and observations. The set of histories $\mathcal{H}$ is the set of all *possible* histories: $\mathcal{H} \stackrel{\text{def}}{=} \{t \in \mathcal{T} | p(t|\emptyset) > 0\}$, the set of action/observation sequences with positive probability at the null history. We call a model M *complete* if it makes a prediction for every test: for any history $h \in \mathcal{H}$ and test $t \in \mathcal{T}$, the model can be queried to find $p_M(t|h)$. We call a model *accurate with respect to a set of tests* $\bar{\mathcal{T}}$ if for any history $h \in \mathcal{H}$, and all $t \in \bar{\mathcal{T}}$, $p_M(t|h) = p(t|h)$. Our task is to build an incomplete model that is nevertheless accurate with respect to the questions of interest.

We will consider a specific way to limit the tests of interest, namely the widely-used practice of making predictions only about features of observations and the problem of building a model that is accurate with respect to predictions of those tests, but which may be simpler than a full model of the system. We will provide novel bounds on the worst-case difficulty of verifying accuracy and present two conceptual algorithms motivated by those results.

1.1 System Dynamics Matrix

Our results will make heavy use of a conceptual object, $\mathcal{D}$, called the system dynamics matrix (Singh, James, & Rudary 2004). $\mathcal{D}$ is an infinity-by-infinity matrix that describes a dynamical system. Each row corresponds to a history, each column a test. Each entry of the matrix $\mathcal{D}_{ij} \stackrel{\text{def}}{=} p(t_j|h_i)$, the prediction of test t_j at history h_i.

Though the system dynamics matrix is infinity-by-infinity, for large classes of interesting systems it has

finite rank, and thus can be specified using a finite number of parameters. We say a dynamical system has *linear dimension* n if its system dynamics matrix has rank n. The linear dimension is, in some sense, a measure of a system's complexity. For instance, a POMDP (Monahan 1982; Cassandra, Kaelbling, & Littman 1994) posits the existence of a finite set of hidden states and uses as its state the probability distribution over those hidden states. POMDPs have a linear dimension no greater than the number of hidden states (Singh, James, & Rudary 2004) so the belief state vector has at least as many entries as the linear dimension. A linear PSR (Littman, Sutton, & Singh 2002) uses as its state representation the predictions of a set of tests whose columns form a linear basis of the system dynamics matrix, and thus the predictive state vector has as many entries as the linear dimension. In this work when we say a system is "simpler" or "more compact" than another, we mean that it has a smaller linear dimension.

Because every finite-observation, finite-action, discrete-time dynamical system has a system dynamics matrix, our results concerning properties of $\mathcal{D}$ will apply generally to *all* such systems. Though because PSRs are defined directly by quantities in $\mathcal{D}$ they may be especially suited to take advantage of the results we present, this work remains agnostic with respect to the ultimate choice of representation.

1.2 Observation Aggregation

In this work we will consider a simple, yet natural and common, mechanism for limiting the tests of interest: the agent will choose not to distinguish between certain observations, effectively as if the agent were only interested in certain *features* of each observation. The features of interest may involve the reward signal of some control task as in, for instance, McCallum (1995), Hoey & Poupart (1995), and Ravindran (2004), or they may be more general, for instance if the agent is specifically assigned a prediction task, or if some features are expected to be useful across multiple tasks.

A set of features of interest induces a partition $\hat{\mathcal{O}}$ of the observation space $\mathcal{O}$, which we call the *partition of interest*. Hereafter, we assume the partition of interest to be given. Each element $O \in \hat{\mathcal{O}}$ is a set of *primitive observations* the agent does not distinguish (because they share values for the features of interest). We call O a *set observation*. Not distinguishing between certain observations will clearly render some action/observation sequences indistinguishable. As such, a partition over observations induces partitions $\hat{\mathcal{T}}$ and $\hat{\mathcal{H}}$ over $\mathcal{T}$ and $\mathcal{H}$, respectively. Elements $T \in \hat{\mathcal{T}}$ and $H \in \hat{\mathcal{H}}$ are sequences of actions and set observations and we call them *set tests* and *set histories*, respectively, following Wingate *et al.* (2007). Note that, because we only aggregate observations, all primitive tests (resp. histories) contained within a set test (resp. history) will have the same action sequence. We will often refer to $\hat{\mathcal{T}}$ as the *set tests of interest*.

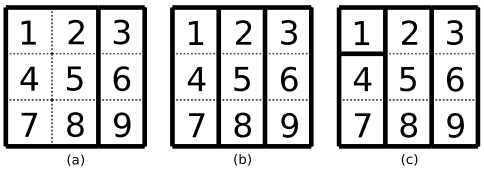

Figure 1: The 3x3 grid world. See text for description.

Because the agent is interested in making predictions only about observation features, it requires predictions only for set tests $T \in \hat{\mathcal{T}}$. That is, it requires an *incomplete* model that provides answers for the predictions $p(T|h)$, for $h \in \mathcal{H}$ and $T \in \hat{\mathcal{T}}$.

A common approach to creating such a model is to build a complete and accurate model $\hat{M}$ of the *aggregate system* such that for any set test T and *set* history H, $p_{\hat{M}}(T|H) = p(T|H)$. $\hat{M}$ can then be used for the agent's purposes by defining $p_{\hat{M}}(T|h) \stackrel{\text{def}}{=} p_{\hat{M}}(T|H) = p(T|H)$ where H is the set history that contains h.

Since $\hat{M}$ models the system at an aggregate level, it may be significantly more compact than the primitive system. However, because $\hat{M}$ does not distinguish primitive histories, it will not necessarily be *accurate* with respect to $\hat{\mathcal{T}}$. To make accurate predictions we require that $p_{\hat{M}}(T|h) = p(T|h)$. Since $p_{\hat{M}}(T|h) = p(T|H)$, where $h \in H \in \hat{\mathcal{H}}$, this implies that the partition $\hat{\mathcal{O}}$ must have the property that for any history $h \in H \in \hat{\mathcal{H}}$ and $T \in \hat{\mathcal{T}}$, $p(T|h) = p(T|H)$. Of course in general $\hat{\mathcal{O}}$ need not satisfy this property. It may be necessary to make more distinctions than $\hat{\mathcal{O}}$ in order to accurately predict the desired tests. So, we seek an *accurate refinement* of $\hat{\mathcal{O}}$.

Definition 1. *An* accurate refinement $\hat{\mathcal{O}}'$ *of the partition of interest* $\hat{\mathcal{O}}$ *is a refinement such that*

$$p(T|H') = p(T|h) \,\forall\, T \in \hat{\mathcal{T}}, H' \in \hat{\mathcal{H}}', h \in H' \qquad (2)$$

where $\hat{\mathcal{T}}$ *is the partition over* $\mathcal{T}$ *induced by* $\hat{\mathcal{O}}$ *and* $\hat{\mathcal{H}}'$ *is the partition over* $\mathcal{H}$ *induced by* $\hat{\mathcal{O}}'$.

Such a refinement always exists because this property is trivially true of the primitive system. In fact, there must exist at least one *minimal* accurate refinement.

Definition 2. *A* minimal, accurate refinement *of* $\hat{\mathcal{O}}$ *is an accurate refinement* $\hat{\mathcal{O}}'$ *such that any coarsening of* $\hat{\mathcal{O}}'$ *is not an accurate refinement of* $\hat{\mathcal{O}}$.

In the following example we shall see that for some systems, minimal and non-minimal accurate refinements can induce drastically different linear dimensions in the resulting aggregate systems.

Example: Consider the 3x3 grid world pictured in Figure 1a. At the beginning of an episode, the agent is placed on a random square. The agent has a movement

action for each cardinal direction (n, e, s, w) and it observes the label (numbers 1-9) of the square it moves into. If the agent attempts to move off of the grid, no movement occurs.

We imagine that the agent is interested in tests distinguishing the right column from the rest of the grid, as indicated by the bold lines in Figure 1a. This partition is not accurate with respect to its own set tests because, for instance, $p(e3|n2) = 1$ but $p(e3|n1) = 0$. The minimal accurate refinement of this aggregation $\hat{\mathcal{O}}'$ is pictured in Figure 1b. It is easy to check that the linear dimension of the system aggregated according to $\hat{\mathcal{O}}'$ is 3. Now consider $\hat{\mathcal{O}}''$, a further refinement of $\hat{\mathcal{O}}'$ that splits the left column $\{1, 4, 7\}$ into $\{1\}$ and $\{4, 7\}$ (as pictured in Figure 1c). Clearly $\hat{\mathcal{O}}''$ is still accurate with respect to the set tests of interest. However, though it may not be obvious from inspection, its aggregate system has a linear dimension of 9, the same as the primitive system. On a general $k \times k$ grid, this would represent a quadratic increase of the linear dimension induced by the minimal accurate refinement.

So, the goal of this paper is to take any discrete dynamical system and any partition of interest $\hat{\mathcal{O}}$ and find a minimal accurate refinement, $\hat{\mathcal{O}}'$. An aggregate model built using this refinement will accurately predict the tests of interest and can be more compact than a complete model of the system. We will now state our main result, and prove it in subsequent sections.

1.3 Main Theorem

We will develop a procedure which splits set observations in $\hat{\mathcal{O}}$ by searching for pairs of histories h and h' contained within the same set history $H \in \hat{\mathcal{H}}$ for which there exists some set test $T \in \hat{\mathcal{T}}$ with $p(T|h) \neq p(T|h')$. Such a pair of histories must be split apart in order to achieve accuracy.

Our main result will be that this search procedure need only check finitely many histories and tests in order to guarantee accuracy.

Theorem 1. *For every discrete dynamical system with linear dimension n and observation aggregation $\hat{\mathcal{O}}$, a minimal, accurate refinement $\hat{\mathcal{O}}'$ of $\hat{\mathcal{O}}$ can be found by checking only predictions involving tests of length less than or equal to n and histories of length less than n^2.*

2 Relationship to Homomorphisms

The accuracy criterion we have defined is closely related to recent work in homomorphisms for model minimization. In this section we will briefly discuss the connection, finding that an accurate refinement is a weaker notion than a homomorphism. Specifically, we will show that an accurate refinement of the partition of interest need not be a homomorphism of the primitive system.

Much of the work on homomorphisms has been done in the setting of Markov Decision Processes (MDP). MDP homomorphisms (Ravindran 2004) essentially address the question "What states and actions can be combined, while still allowing accurate predictions about the reward signal?" Wolfe & Barto (2006) generalized the criterion, much as we are interested in doing, to allow for arbitrary observation features of interest, rather than focusing on just the reward signal. Wolfe & Barto focused on the special case of MDPs, however, so their algorithms do not apply to our more general setting.

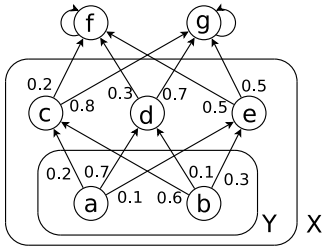

Figure 2: MDP example from Theorem 2. Multiple arrows coming out of one state indicate a stochastic transition with the probability distribution indicated. States a and b are initial states with equal probability.

The homomorphisms framework was extended to arbitrary discrete dynamical systems with PSR homomorphisms (Soni & Singh 2007). PSR homomorphisms allow for history-dependent observation *and* action aggregations. Since we only consider history-*in*dependent observation aggregations, we are working with a special case of the class of partitions allowed by PSR homomorphisms. However, Soni & Singh provide no algorithms for *finding* PSR homomorphisms. More importantly, the PSR homomorphism criteria are stronger than is strictly necessary for our purposes. Specialized to our setting, a refinement $\hat{\mathcal{O}}'$ of the partition of interest is a homomorphism of the primitive system if

$$p(T'|H') = p(T'|h) \; \forall T' \in \hat{\mathcal{T}}', H' \in \hat{\mathcal{H}}', h \in H'. \quad (3)$$

That is, it must be accurate *not only* with respect to the set tests of interest $\hat{\mathcal{T}}$, but *also* with respect to *its own* set tests $\hat{\mathcal{T}}'$. If $\hat{\mathcal{O}}'$ satisfies this property, we call the aggregate system with respect to $\hat{\mathcal{O}}'$ a homomorphic image of the primitive system.

We will now present an example in which a refinement $\hat{\mathcal{O}}'$ is accurate with respect to $\hat{\mathcal{T}}$ (satisfies Equation 2) and is not a homomorphism (does not satisfy 3).

Theorem 2. *A minimal, accurate refinement with respect to a partition of interest $\hat{\mathcal{O}}$ over the observations of a dynamical system need not necessarily be a homomorphism of that system.*

Proof. Consider the uncontrolled MDP in Figure 2 and imagine that the partition of interest distinguishes f and g from other observations (and each other) with the rest in one set observation $X = \{a, b, c, d, e\}$. Now

consider a refinement of this initial partition in which c, d, and e are also distinguished, but a and b are aggregated into the set observation $Y = \{a, b\}$. This refinement is *not* accurate with respect to its own set tests. For instance: $p(c|a) = 0.2$ and $p(c|b) = 0.6$ even though $a, b \in Y$. By inspection, however, it is easy to see that this refinement *is* accurate with respect to the tests of interest, most notably because $p(Xf|a) = p(Xf|b) = p(Xf|Y) = 0.3$.

In fact, though it may not be obvious from inspection, this refined system has a linear dimension of 4, while the primitive system (the only refinement that is a homomorphism) has a linear dimension of 5. Thus the system aggregated according to this refinement is more compact than the minimal homomorphic image that is accurate with respect to the tests of interest. □

Our main results concern the process of finding a minimal accurate refinement, though later we will present an algorithm for finding a homomorphism, and see that there may be reasons to prefer it, even at the potential cost of compactness.

3 Finding an Accurate Refinement

As discussed in Section 1.3, our refinement procedure will search for violations of the accuracy criterion. These will consist of pairs of histories h and h', both contained within the same set history $H \in \hat{\mathcal{H}}$ for which there exists a set test $T \in \hat{\mathcal{T}}$ with $p(T|h) \neq p(T|h')$. However, identifying such a pair of histories is not, in itself, very informative with respect to how to refine the partition $\hat{\mathcal{O}}$. Knowing that h and h' must be split apart tells us only that we need to split at least one pair of observations at corresponding time steps in the two histories, but not which pair. That said, for some pairs of histories, it *is* clear. If h and h' differ only at one time step, then the pair of observations at that time step must be split apart.

Lemma 1. *Consider two primitive observations $o_1, o_2 \in O \in \hat{\mathcal{O}}$. If there exists $a \in \mathcal{A}$, $T \in \hat{\mathcal{T}}$ and primitive histories $h_1, h_2 \in \mathcal{H}$ such that $p(T|h_1ao_1h_2) \neq p(T|h_1ao_2h_2)$ then in any accurate refinement $\hat{\mathcal{O}}'$ of $\hat{\mathcal{O}}$, $\exists O_1, O_2 \in \hat{\mathcal{O}}'$ such that $o_1 \in O_1$, $o_2 \in O_2$, and $O_1 \neq O_2$.*

Proof. In any refinement that does not split o_1 and o_2, $h_1ao_1h_2$ and $h_1ao_2h_2$ will belong to the same set history. Since they have different predictions for T, such a refinement could not possibly be accurate. □

Using the following result, we will show that it will be sufficient to compare only pairs of histories that differ in one time step. Throughout the remainder of the paper we use the notation h^i to signify the i-length prefix of a history h and h^{-j} to signify the j-length suffix of h. That is, if $h = a_1o_1...a_ko_k$, $h^i = a_1o_1...a_io_i$ and $h^{-j} = a_{k-j+1}o_{k-j+1}...a_ko_k$. Also, $h^0 = h^{-0} = \emptyset$ and $h^k = h^{-k} = h$.

Lemma 2. *Consider two primitive histories with the same action sequence $h = a_1o_1...a_ko_k$ and $h' = a_1o'_1...a_ko'_k$. Let $h_i = h^ih'^{-(k-i)}$. If for all $i \in \{0, 1, ..., k\}$, $p(T|h_i) = p(T|h_{i+1})$ then $p(T|h) = p(T|h')$.*

Proof. The result is easy to see by making a series of transformations from h to h', swapping only one observation at a time. For any T, $p(T|h) = p(T|h_k) = p(T|h_{k-1}) = ... = p(T|h_0) = p(T|h')$. □

By the contrapositive of this result, we can conclude that for any pair of histories h and h' of length k with $p(T|h) \neq p(T|h')$ there exists *another* pair of histories $h^ia_io_ih'^{-(k-i-1)}$ and $h^ia_io'_ih'^{-(k-i-1)}$ that also disagree on T. These two histories differ on only one time step. So, in searching for a split, it suffices to consider every pair of primitive observations o_1 and o_2 in each set observation O and check for all h_1, h_2, a, and T, whether $p(T|h_1ao_1h_2) = p(T|h_1ao_2h_2)$. Of course there are infinitely many such histories and tests. It will be the work of later sections to prove that, in fact, we need perform only finitely many of these checks.

A *violation* for observations o_1 and o_2 is a tuple $\langle T, a, h_1, h_2\rangle$ such that $T \in \hat{\mathcal{T}}$, $a \in \mathcal{A}$, $h_1 \in \mathcal{H}$, $h_2 \in \mathcal{T}$, and $p(T|h_1ao_1h_2) \neq p(T|h_1ao_2h_2)$. We will say a pair of observations o_1, o_2, is *comparable* if there exists some history h and some action a such that $p(hao_1|\emptyset) > 0$ and $p(hao_2|\emptyset) > 0$. If all observations are comparable, then the pair-wise relation "Have no violations" is an equivalence relation. If we define our refinement such that each set observation is exactly one equivalence class according to this relation, we will have split every observation that *must* be split in *any* accurate refinement, and no more:

Lemma 3. *If all observations are comparable and $\hat{\mathcal{O}}'$ is the set of equivalence classes induced by the relation "Have no violations," then $\hat{\mathcal{O}}'$ is a minimal, accurate refinement of $\hat{\mathcal{O}}$. Furthermore, there is no accurate refinement of $\hat{\mathcal{O}}$ that induces an aggregate system with a smaller linear dimension than that induced by $\hat{\mathcal{O}}'$.*

Proof. Accuracy and minimality are direct consequences of Lemmas 2 and 1, respectively. Furthermore, Lemma 2 implies that *any* accurate refinement must be a refinement of $\hat{\mathcal{O}}'$, since it must make at least the distinctions made by $\hat{\mathcal{O}}'$. Since the columns and rows of the aggregate system dynamics matrix induced by $\hat{\mathcal{O}}'$ will be sums of columns and rows of the system dynamics matrix induced by any refinement of $\hat{\mathcal{O}}'$, refinement can never reduce rank, which gives us the result. □

Incomparable observations will have only a minor impact on finding a minimal, accurate refinement, though they can affect the linear dimension of the aggregate system that results. For a brief discussion, see Appendix A. For the remainder, we will assume all observations are comparable and turn to the work of proving

that we can find a minimal accurate refinement using only finitely many violation checks.

3.1 Bounding Test Length

We will start by proving the first part of Theorem 1: for a dynamical system with linear dimension n, we need only consider tests of length less than or equal to n in our violation search. In fact, we will prove a slightly tighter bound. We will need the concept of the *set test matrix*. This is an infinity-by-infinity matrix like the system dynamics matrix, and each entry is a prediction. In the set test matrix, however, columns correspond to set tests only (the rows still correspond to primitive histories). The rank of this matrix, $\bar{n}$, can be at most n and must be at least $\hat{n}$, the rank of the aggregate system dynamics matrix (which has set tests for columns and set histories for rows). The set test matrix also has the following useful property, which we will present without proof:

Lemma 4. *There exists a linear column basis of the set test matrix consisting entirely of columns corresponding to set tests of length $\bar{n}$ or less.*

This allows us to consider only finitely many tests in our violation search.

Lemma 5. *If a violation $\langle T, a, h_1, h_2\rangle$ exists for observations $o_1, o_2 \in O \in \hat{\mathcal{O}}$ then a violation $\langle T', a, h_1, h_2\rangle$ exists for o_1 and o_2 with length$(T') \leq \bar{n}$.*

Proof. If $\bar{Q}$ is a set of tests whose columns form a basis for the set test matrix and for some pair of histories $p(\bar{q}|h) = p(\bar{q}|h')$ for all $\bar{q} \in \bar{Q}$, then $p(T|h) = p(T|h')$ for any T. By Lemma 4, there exists a $\bar{Q}$ consisting of tests length $\bar{n}$ or less. The result immediately follows. □

3.2 Bounding History Length

In this section we will complete the proof of Theorem 1 by showing that we need only consider histories of length less than n^2 in our search for violations. For the purposes of this argument, we will introduce a transformation of the system dynamics matrix $\mathcal{D}$, which we will call $\mathcal{D}^2$. Each row in $\mathcal{D}^2$ corresponds to *a pair* of histories $\langle h_1, h_2\rangle$, and each column a pair of tests $\langle t_1, t_2\rangle$. Each entry $\mathcal{D}^2_{\langle h_1,h_2\rangle,\langle t_1,t_2\rangle} \stackrel{\text{def}}{=} p(t_1|h_1)p(t_2|h_2)$. It is straightforward to bound the rank of $\mathcal{D}^2$:

Lemma 6. *If rank$(\mathcal{D}) = n$, then rank$(\mathcal{D}^2) \leq n^2$.*

Proof. Let Q be a set of n tests corresponding to columns that form a basis for $\mathcal{D}$. Then for any test t and history h, $p(t|h) = \sum_{q\in Q} p(q|h)m_t(q)$, where $m_t(q)$ is some scalar weight. Then

$$\begin{aligned}\mathcal{D}^2_{\langle h_1,h_2\rangle,\langle t_1,t_2\rangle} &= p(t_1|h_1)p(t_2,h_2)\\ &= \sum_{q\in Q}\sum_{q'\in Q} p(q|h_1)p(q'|h_2)m_{t_1}(q)m_{t_2}(q')\\ &= \sum_{\langle q,q'\rangle\in Q\times Q} \mathcal{D}^2_{\langle q,q'\rangle,\langle h_1,h_2\rangle} m_{\langle t_1,t_2\rangle}(\langle q,q'\rangle).\end{aligned}$$

So the columns corresponding to $Q \times Q$ are a column basis of $\mathcal{D}^2$and $rank(\mathcal{D}^2) \leq |Q \times Q| = n^2$. □

Using this fact we can prove a general result about the system dynamics matrix which will lead directly to the bound we seek.

Theorem 3. *Let the linear dimension of a system be n and consider two primitive histories with the same action sequence: $h_1 = a_1o_1^1...a_ko_k^1$ and $h_2 = a_1o_1^2...a_ko_k^2$. If for some test t $p(t|h_1) \neq p(t|h_2)$, then there exists a subsequence $\{i_1, i_2, ...i_j\}$ of $\{1, ..., k\}$ such that $j < n^2$ and $p(t|a_{i_1}o^1_{i_1}a_{i_2}o^1_{i_2}...a_{i_j}o^1_{i_j}) \neq p(t|a_{i_1}o^2_{i_1}a_{i_2}o^2_{i_2}...a_{i_j}o^2_{i_j})$.*

Proof. If $k < n^2$, this result holds trivially, so let us assume $k \geq n^2$. We will work with a $(k+1) \times (k+1)$ matrix, X, constructed from entries of $\mathcal{D}^2$, defined entry-wise for all $0 \leq i, j \leq k$:

$$X_{ij} \stackrel{\text{def}}{=} p(h_1^{-j}t|h_1^i)p(h_2^{-j}|h_2^i) - p(h_2^{-j}t|h_2^i)p(h_1^{-j}|h_1^i).$$

Note that the rows of X are a subset of the rows of $\mathcal{D}^2$ and the columns are weighted sums of the columns of $\mathcal{D}^2$. Therefore, $rank(X) \leq rank(\mathcal{D}^2) \leq n^2$.

If an entry X_{ij} is zero, we have

$$\frac{p(h_1^{-j}t|h_1^i)}{p(h_1^{-j}|h_1^i)} = \frac{p(h_2^{-j}t|h_2^i)}{p(h_2^{-j}|h_2^i)} \tag{4}$$

and, by Bayes' rule,

$$p(t|h_1^ih_1^{-j}) = p(t|h_2^ih_2^{-j}). \tag{5}$$

Conversely, if $p(t|h_1^ih_1^{-j}) \neq p(t|h_2^ih_2^{-j})$ then $X_{ij} > 0$.

Since $p(t|h_1) \neq p(t|h_2)$, it must be that $X_{i(k-i)} > 0$ for any i. If $X_{ij} = 0$ for all $j < k - i$, then X would have full rank: $k + 1 > n^2$, a contradiction. As such, there is some $j < k - i$ such that $X_{ij} > 0$. That is, if $p(t|h_1) \neq p(t|h_2)$ and $k \geq n^2$, we can find another, shorter pair of histories that disagree on t by removing the same time-steps from both h_1 and h_2. Specifically:

$$\begin{aligned}&p(t|a_1o_1^1...a_io_i^1a_{k-j+1}o^1_{k-j+1}...a_ko_k^1) \neq\\ &\qquad p(t|a_1o_1^2...a_io_i^2a_{k-j+1}o^2_{k-j+1}...a_ko_k^2).\end{aligned}$$

The resulting subsequence of indices has length $k' = i + j$. If $k' \geq n^2$, we simply repeat the argument to remove more substrings until we reach a subseqence with length less than n^2. □

Lemma 7. *If a violation $\langle T, a, h_1, h_2\rangle$ exists for observations $o_1, o_2 \in O \in \hat{\mathcal{O}}$ then a violation $\langle T, a, h_1', h_2'\rangle$ exists for o_1 and o_2 with length$(h_1'ao_1h_2') < n^2$.*

Proof. This follows immediately from Theorem 3 when we note that, since $h_1ao_1h_2$ and $h_1ao_2h_2$ only differ in one time-step, the subsequences that disagree on T must still contain that step, otherwise they would be equal. As such, they are a violation for o_1 and o_2. □

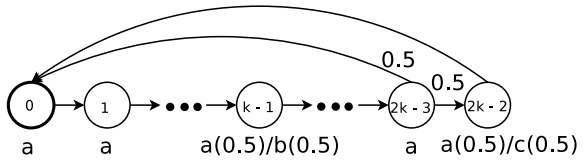

Figure 3: HMM example from Section 3.2. The observation (distribution) associated with each state is next to the state. State 0 is the initial state.

This completes the proof of Theorem 1, our main result. It may be possible to tighten these bounds in terms of other parameters of the primitive system. For instance, we have, as of yet, no example which requires histories of length greater than $\bar{n}^2$. The quadratic dependence on n, however, is necessary in the worst case, as we now demonstrate.

Theorem 4. *For any n there exists a dynamical system with linear dimension at least n and partition of interest $\hat{\mathcal{O}}$ such that a violation exists, and the shortest pair of histories involved in a violation have $O(n^2)$ length.*

Proof. Consider the family of uncontrolled POMDPs (or HMMs), indexed by k, of the form given in Figure 3. The linear dimension is equal to the number of hidden states, $n = 2k - 1$. Imagine that the partition of interest distinguishes a from b and c, but aggregates b and c into the set observation X. This partition is not accurate with respect to its own set tests because $p(a^{k-1}X|a^{k(n-1)}b) = 0.25$ and $p(a^{k-1}X|a^{k(n-1)}c) = 0.5$. In fact, it is possible to show that there is no history h' shorter than $a^{k(n-1)}$ such that $p(b|h') > 0$ and $p(c|h') > 0$. Therefore, the shortest history at which a violation occurs has length $k(n-1)+1 = \frac{n+1}{2}(n-1)+1 = \frac{1}{2}(n^2-1)+1$. □

3.3 One-Pass Algorithm

Theorem 1 in hand, we can describe in detail a conceptual refinement procedure, which we call the one-pass algorithm. It is an exhaustive search for violations $p(T|h_1ao_2h_2) \neq p(T|h_1ao_2h_2)$ over all $O \in \hat{\mathcal{O}}$, $o_1, o_2 \in O$, $a \in \mathcal{A}$, $T \in \hat{\mathcal{T}}$ with $length(T) \leq \bar{n}$, and $h_1, h_2 \in \mathcal{H}$ with $length(h_1)+length(h_2) < n^2-1$. Once all possible violations have been accounted for, equivalence classes are found and serve as the refinement.

Note that, in its pure form, the one-pass algorithm is computationally daunting. Even putting aside that the linear dimension of most problems of real interest will be enormous in itself, the number of tests and histories to be checked grows *exponentially* in the length to be considered. Thus in the worst case, even for systems of moderate complexity, finding all violations is entirely impractical. However, in many cases we would expect to see violations with much shorter tests and histories. For instance, if the linear basis of the set test matrix consists of short tests, then correspondingly we need only look at short tests. If the system is more densely connected than the example in Theorem 4, then we would expect to see violations at shorter histories. In the next section we will develop an alternative algorithm which is better equipped to take advantage of these circumstances when they occur, by being more opportunistic in its choices of splits.

4 Iterative Splitting

In this section we will consider a variation of the one-pass algorithm which, rather than searching for all violations before splitting, produces a refinement as soon as it can do so "safely." A safe split is one that only splits observation pairs that have a violation. It will often be possible to make a safe split without finding *all* violations. The iterative algorithm looks at increasingly long histories and tests (up to length n^2 and $\bar{n}$, respectively) searching for a safe split. Once it finds one, it treats the set tests of the new refinement as if *they* were the tests of interest and begins again. Because a safe split always exists (the one-pass algorithm always makes a safe split), this algorithm is guaranteed to stop at a refinement that is accurate with respect to *its own* set tests, a homomorphism of the primitive system. Such a refinement is also accurate with respect to the tests of interest. It need not, however, be a *minimal* accurate refinement, as we saw in Theorem 2.

In exchange for the loss of the minimality guarantee, the iterative algorithm can enjoy substantial improvements in the length of tests and histories that must be considered, since intermediate splits can cause new violations, as we shall see in the following example.

Example: Imagine a $k \times k$ grid world analogous to our previous 3×3 grid world example. In order to find a violation for observations 1 and 2, the one-pass algorithm must look at histories and tests that have a combined length of $k-1$ (because it takes one step to see observation 1 or 2 and $k-2$ steps to reach the right side from square 2 and not from square 1). In comparison, the iterative algorithm would find, by looking at 1-step histories and 1-step tests, that the $(k-1)$th column must be split off because it is possible to reach the kth column from those squares in one step, and not from the others. In the next iteration, again checking only one-step tests and one-step histories, it would discover that the $(k-2)$th column should be split off, and so on until the minimal accurate refinement was found (one set observation per column).

Ultimately, of course, when the iterative algorithm reaches a stopping point, it must still check tests and histories of the same length as the one-pass algorithm in order to verify that there are no violations. So, it is appropriate to think of the iterative algorithm as a "fail-fast" style algorithm, which will tend to rule out non-solutions quickly, but which may still take a long time to verify a solution once it is ultimately found.

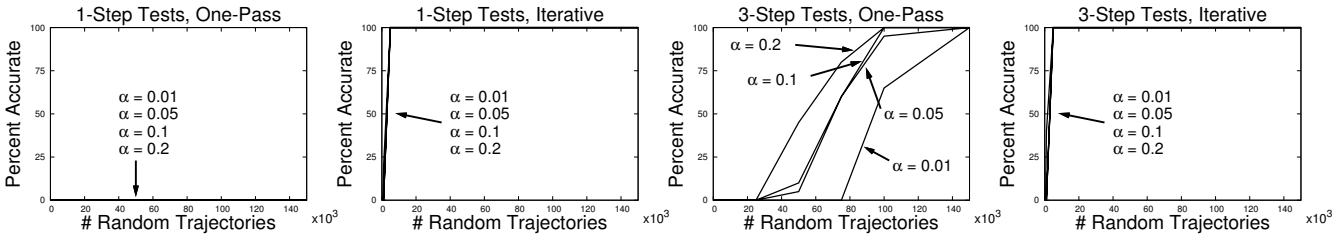

Figure 4: Percentage of accurate refinements (out of 20) for different test lengths in the 5x5 grid world domain.

5 Experiments

There are a number of practical challenges to overcome when applying the principles we have described to a modeling problem. First and foremost, in most settings we will not have access to true entries of the system dynamics matrix. Instead, we will estimate prediction probabilities from data, and compare estimated values using the chi-square test of homogeneity. Secondly, it will be impractical to perform the violation search with histories and tests of the length required to guarantee accuracy because in most systems of interest the linear dimension n will be large and furthermore n is not known *a priori*. For many systems, however, a minimal, accurate refinement can be found by checking short tests and histories. In practice we will select a maximum test length l^t_{max} and a maximum history length l^h_{max} for our violation search.

In the following experiments we had the agent explore the environment using a uniform random policy in a number of distinct trajectories. Periodically we would freeze data collection, and apply both splitting algorithms, recording the refinement found. Then, restoring the original aggregation, data collection would resume. We compare the performance of the one-pass algorithm to the iterative algorithm, and explore the effect of different choices of α, the significance level for the chi-square tests used to compare table entries.

5.1 Grid World

In our first set of experiments, we consider our running grid world example, in this case a 5×5 grid. The primitive system has 25 observations and a linear dimension of 25. The minimal refinement accurate with respect to whether the agent will observe the right-most column results in a system with 5 aggregate observations (one for each column) and a linear dimension of 5.

In Figure 4 we present the percentage of refinements found that were accurate (out of 20 runs) compared to the number of trajectories seen for four choices of α (0.01, 0.05, 0.1, and 0.2), and two choices of l^t_{max} (1 and 3). In all cases we set $l^h_{max} = 1$. As predicted, the one-pass algorithm is unable to find an accurate refinement using 1-step tests, while the iterative algorithm can. Even with 3 step tests, when both algorithms are capable of finding an accurate refinement, the iterative algorithm seems to be significantly more data-efficient, regardless of the choice of α. This is because splits at one iteration can cause new violations in subsequent iterations, even without acquiring any new data.

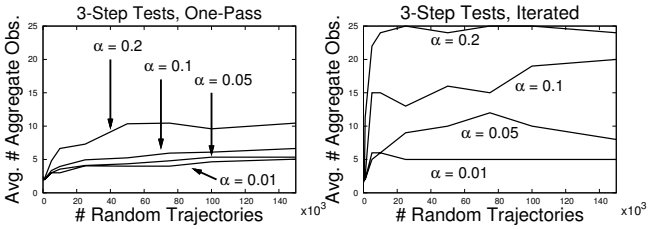

Figure 5: Avg. number of aggregates (out of 20 runs) in the 5×5 grid world.

In Figure 5 we see the average number of aggregate observations found (out of 20 runs) compared to the number of trajectories seen for the same four choices of alpha (and 3-step tests). The effect of α is unsurprising: higher values of α tend to cause over-splitting while lower values require more data to make the necessary distinctions. The over-splitting effect is especially pronounced with the iterative algorithm, again because spurious splits in early iterations can propagate into later ones.

5.2 Machine Maintenance

Our second set of experiments is in a slightly modified version of the Machine Maintenance domain presented by Cassandra (1998). In this domain the agent is in charge of a manufacturing machine with k components. The components can be in any of 4 states of disrepair. At every time step, the agent can choose from four actions. If it chooses to *replace* the machine all components are reset to excellent condition. If it *repairs* the machine each component upgrades its status with some probability. In either of these cases, the agent observes only that the machine has been serviced. If it *inspects* the machine the current status of all components is revealed (in the original formulation, each component produced a stochastic signal of its status). Finally, if the agent chooses to *manufacture*, the machine produces a product, stochastically good or bad (determined by the components' states), and the components downgrade their status with some probability.

We imagine that the agent would be primarily interested in whether the machine produces *good* or *bad*

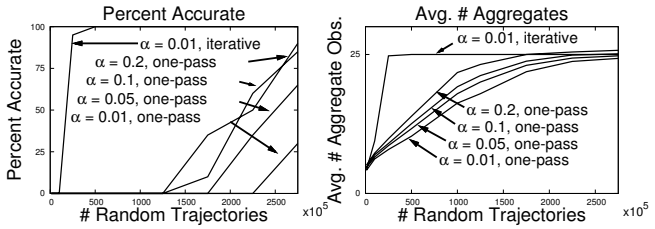

Figure 6: Results in the machine maintenance domain.

products. So, we will consider an initial aggregation with four aggregates: $\{good\}$, $\{bad\}$, $\{serviced\}$, and $\{excellent, fair, poor, broken\}^k$, where the last aggregation contains all 4^k possible machine states that could be observed after an inspection. In order to make predictions at this level, one need not actually know the full machine state. For instance, if any component is broken, the machine will always produce *bad* products. If no component is broken, it suffices to merely know *how many* components are in each state.

We performed our experiments on a 5 component machine. Thus, the primitive system has linear dimension of $4^5 = 1024$, enough to pose a significant challenge to model-building techniques based on either POMDPs or PSRs. However, the aggregate system that considers only how many components are in each state and whether any component is broken has a more manageable linear dimension of 22.

In our experiments we set $l^t_{max} = 2$ and $l^h_{max} = 1$. At the beginning of each trajectory each component's state was chosen uniformly randomly. In Figure 6 we can see that, as with the grid world domain, the iterative splitting algorithm with a small α significantly outperforms the one-pass algorithm, in this case requiring an order of magnitude fewer trajectories to find an accurate refinement. Even so, the amount of data necessary is quite high. In part this is due to our choice of a uniform random exploration policy, which will take a long time to collect sufficient data on low-probability events. In the future it may be possible to guide exploration to focus on trajectories that will most help the violation search.

6 Conclusions

An agent seeking to accurately predict set tests with respect to a partition over observations may have to build a model using a refinement of that partition. We have bounded the difficulty of finding such a refinement in terms of the complexity of the primitive system and presented two conceptual algorithms based on those results. We also presented experimental results that are consistent with our theoretical expectations.

Of course the brute-force algorithms we have presented are not themselves applicable to many domains of interest for reasons we have discussed. However, we hope that issues and insights raised in this analysis will pave the way to more practicable methods. It will be critical to study notions of approximate accuracy in which perhaps not all violations are found, heuristics or biases (perhaps based on domain knowledge) to improve the search for violations, and incorporation of more expressive forms for the questions of interest, including action abstraction and temporal abstraction.

Acknowledgments

Erik Talvitie was supported under a National Science Foundation Graduate Research Fellowship. Britton Wolfe and Satinder Singh were supported by NSF grant 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 NSF.

A Incomparable Observations

Recall that a pair of observations o_1, o_2 are *comparable* if there exists some history h and action a such that $p(hao_1|\emptyset) > 0$ and $p(hao_2|\emptyset) > 0$. Observations that are *incomparable* are notable because, by definition, they cannot have a violation and can, as a result, cause intransitivities in the "Have no violations" relation.

First note that, by Lemma 2, whether two incomparable observations are distinguished cannot affect accuracy, since they cannot cause a violation. Furthermore, it is possible to identify incomparable observations in the course of the violation search by the following result, which we present without proof (the argument is similar in form to that of Theorem 4).

Lemma 8. *If $p(ao_1|h) \cdot p(ao_2|h) = 0$ for all histories h with length$(h) < n^2$, then $p(ao_1|h) \cdot p(ao_2|h) = 0$ for all histories h of any length.*

Thus, it is possible to construct a minimal, accurate refinement by respecting all comparisons that can be made and otherwise grouping incomparable observations arbitrarily. One procedure for achieving this would first group together all pairs of observations that *are* comparable and have no violations. Two such sets can be merged if all members of one set are incomparable with all members of the other set (otherwise there is a violation between a member of one set and a member of the other set). We simply merge pairs of sets until no more merges are available. The result will be accurate (since no two observations with a violation will be grouped together) and minimal (since every pair of sets has at least one violation and thus, cannot be merged).

This simple procedure could result in any of several refinements, depending on how incomparable sets are merged. All of them will be minimal, accurate refinements. However, they *may* differ in the linear dimension of the aggregate system they induce, which is, in some sense, our true decision criterion. Heuristics for grouping incomparable observations in order to produce a compact aggregate model will be an important direction for future research.

References

Cassandra, A. R.; Kaelbling, L. P.; and Littman, M. L. 1994. Acting optimally in partially observable stochastic domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence, (AAAI)*, volume 2, 1023–1028.

Cassandra, A. R. 1998. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. Ph.D. Dissertation, Brown University.

Hoey, J., and Poupart, P. 2005. Solving pomdps with continuous or large discrete observation spaces. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1332–1338.

Littman, M.; Sutton, R.; and Singh, S. 2002. Predictive representations of state. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 1555–1561.

McCallum, A. K. 1995. *Reinforcement Learning with Selective Perception and Hidden State*. Ph.D. Dissertation, Rutgers University.

Monahan, G. E. 1982. A survey of partially observable markov decisions processes: Theory, models, and algorithms. *Management Science* 28(1):1–16.

Ravindran, B. 2004. *An Algebraic Approach to Abstraction in Reinforcement Learning*. Ph.D. Dissertation, University of Massachusetts, Amherst, MA.

Singh, S.; James, M. R.; and Rudary, M. R. 2004. Predictive state representations: A new theory for modeling dynamical systems. In *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference (UAI)*, 512–519.

Soni, V., and Singh, S. 2007. Abstraction in predictive state representations. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence, (AAAI)*. To appear.

Wingate, D.; Soni, V.; Wolfe, B.; and Singh, S. 2007. Relational knowledge with predictive state representations. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2035–2040.

Wolfe, A. P., and Barto, A. G. 2006. Decision tree methods for finding reusable MDP homomorphisms. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence, (AAAI)*.