

MDPs...

- Make the problem precise (& simpler?)
- Yet keeps many interesting challenges
 - Preserves tradeoff between short-term and long-term consequences
 - Temporal credit assignment
 - Exploration vs. exploitation
 - Generalization across states (or learning from small amounts of experience).

So why rethink state?

Narrow vs. Broad competence

- MDPs/POMDPs very successful in OR/engineering/control
- Still lots of hard work left to do... especially in making RL more "off the shelf"
- My Goal: move towards old-fashioned AI? (build broadly competent, flexible, agents)

Knowledge in AI Systems

- AI systems tend to be brittle.
- MDP/POMDP representations while modeling uncertainty share that brittleness
- Relational extensions may help...

But many of these approaches are “linguistically inspired” using notions of objects, relations between objects... which while very meaningful and natural to humans may not be suited for computational agents.

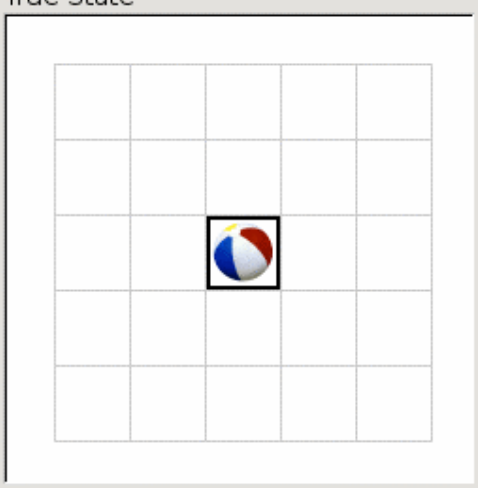
Goal: a KR expressed entirely in input-output terms... so that the knowledge learned or given is meaningful, verifiable, maintainable by the agent without human intervention

Knowledge/Models for RL/AI

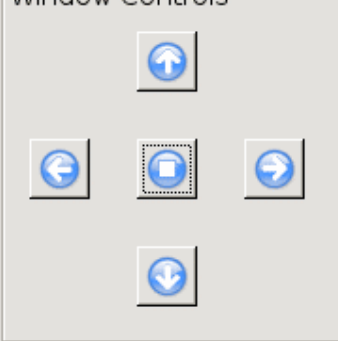
- Knowledge that is useful for achieving high reward
 - Is tweety a bird?
 - Can one sit on the object in front?
 - If I pick the phone and dial my home number what is the chance that my wife picks up?
 - Can block A be stacked on top of block B?
- Answers to questions (usually predictive)

File

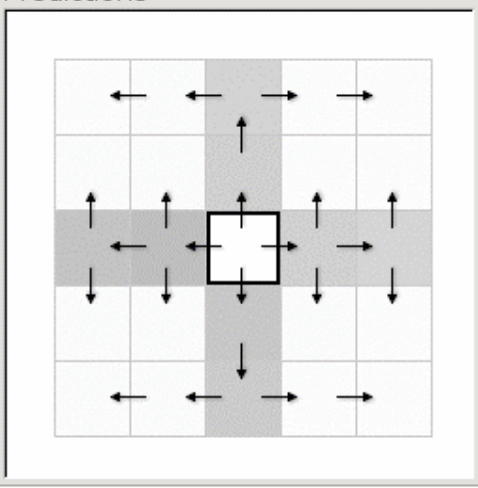
True State



Window Controls



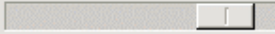
Predictions



Autopilot

Start

90



Tracking mode

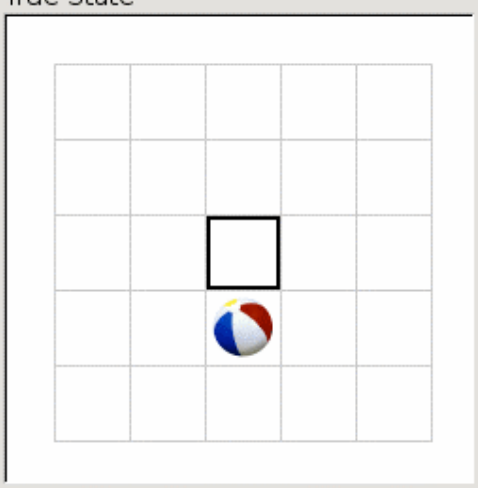
Core Tests

Test	Prob
Ub	0.78
UB	0.22
Db	0.74
UbUb	0.78

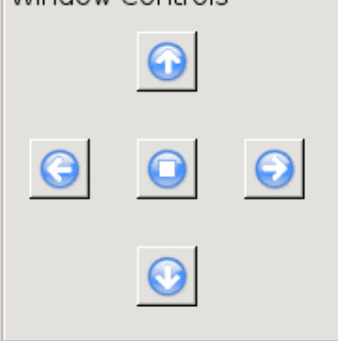
(no move)

File

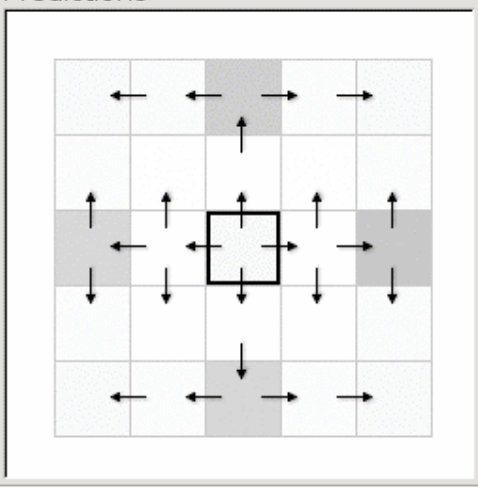
True State



Window Controls




Predictions



Autopilot

Start

90



Tracking mode

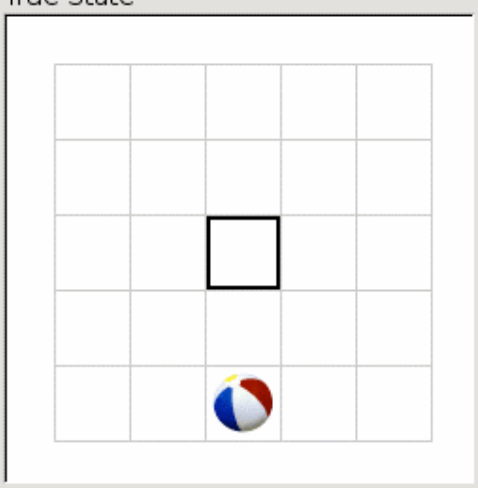
Core Tests

Test	Prob
Ub	1.00
UB	-0.00
Db	1.00
UbUb	0.78

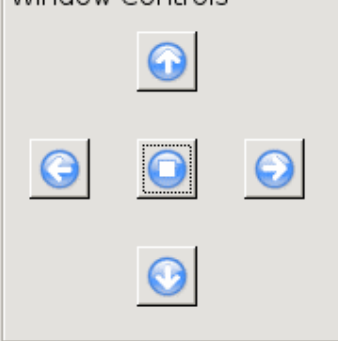
(no move)

File

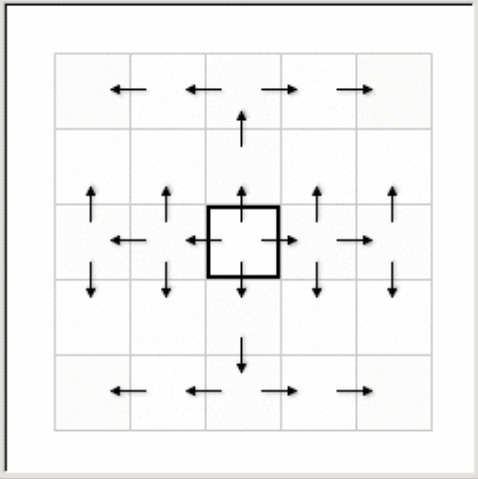
True State



Window Controls




Predictions



Autopilot

Start

90



Tracking mode

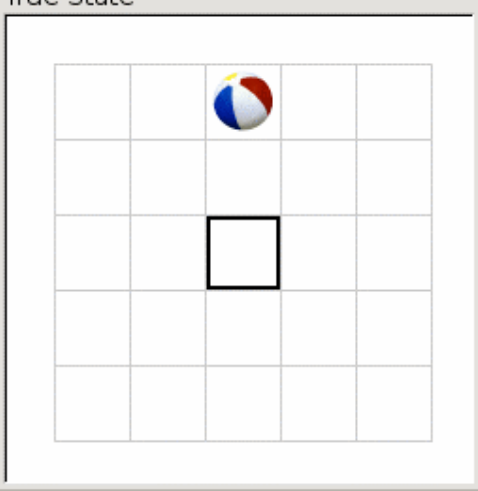
Core Tests

Test	Prob
Ub	0.99
UB	0.01
Db	0.99
UbUb	0.99

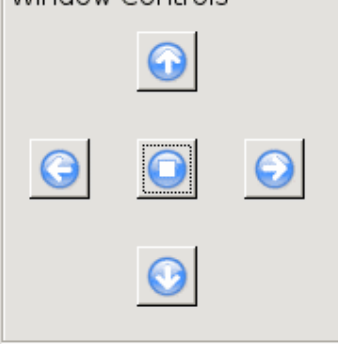
(no move)

File

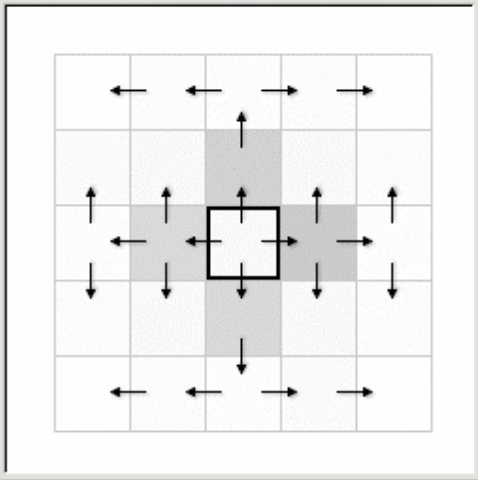
True State



Window Controls




Predictions



Autopilot

Start

90



Tracking mode

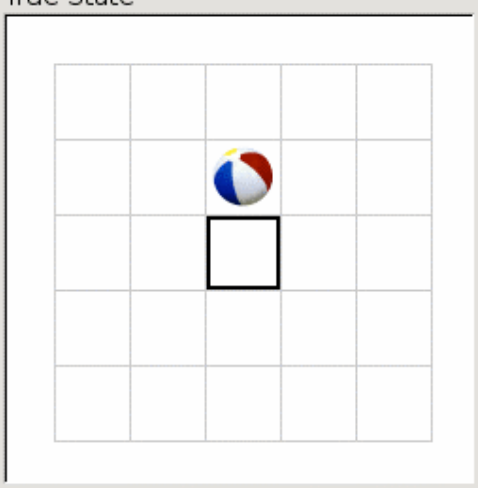
Core Tests

Test	Prob
Ub	0.79
UB	0.21
Db	0.82
UbUb	0.79

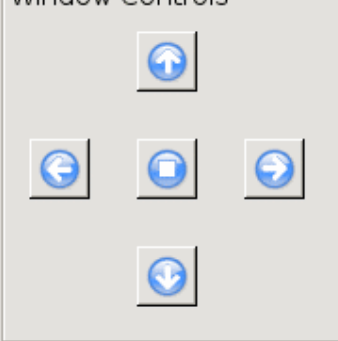
(no move)

File

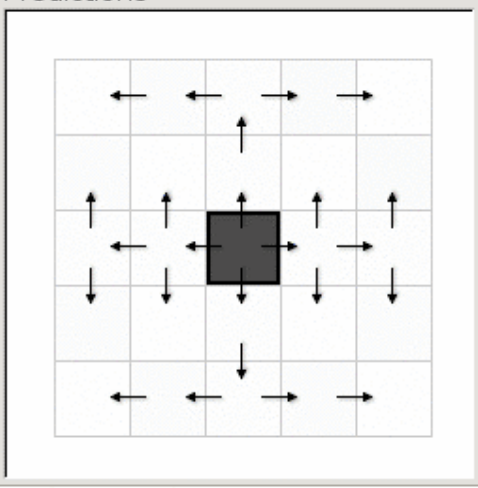
True State



Window Controls



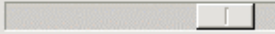
Predictions



Autopilot

Stop

90



Tracking mode

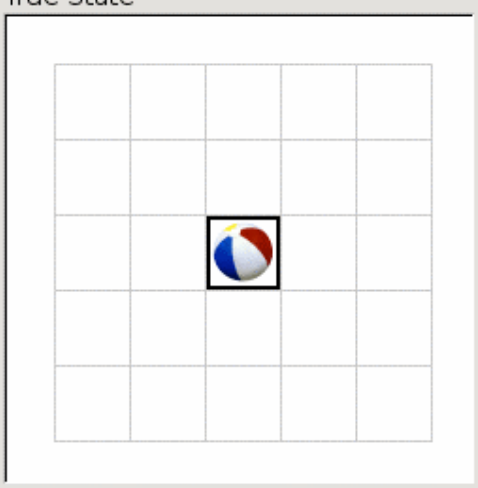
Core Tests

Test	Prob
Ub	0.99
UB	0.01
Db	0.99
UbUb	0.99

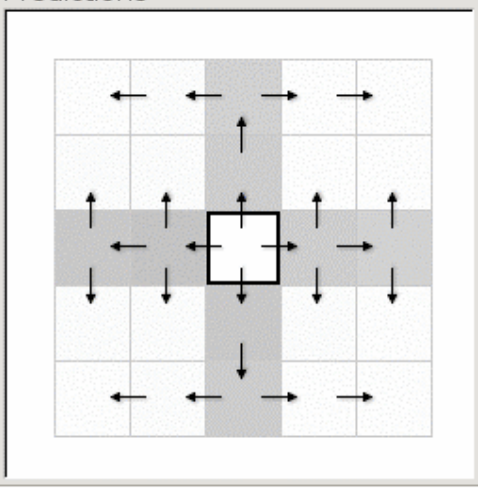
Autopilot started.

File

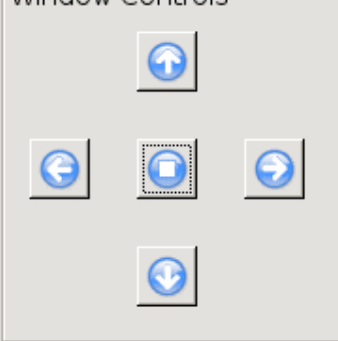
True State



Predictions



Window Controls



Autopilot

Start

90

Tracking mode

Core Tests

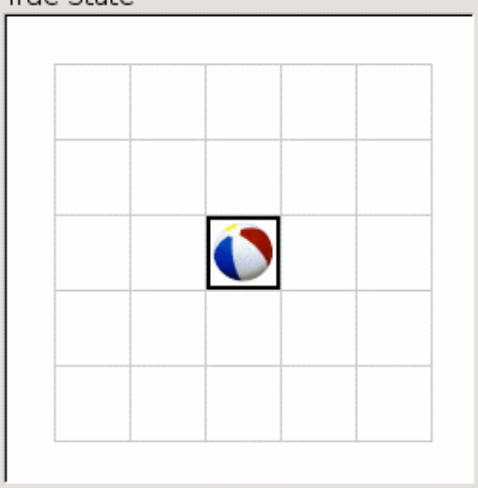
Test	Prob
Ub	0.77
UB	0.23
Db	0.74
UbUb	0.77

(no move)

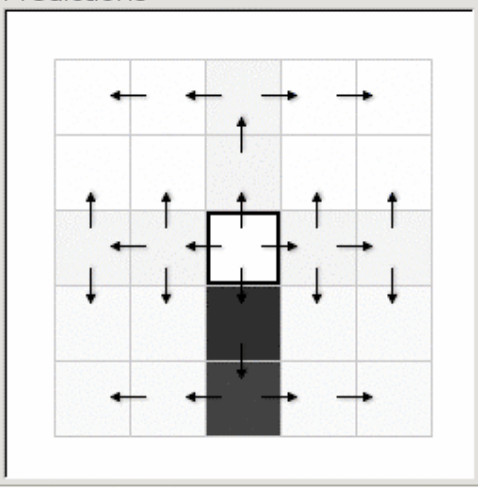
Rightward movement example

File

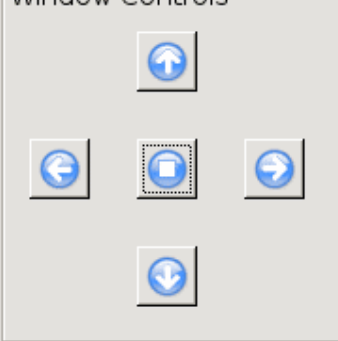
True State



Predictions



Window Controls



Autopilot

Start

90

Tracking mode

Core Tests

Test	Prob
Ub	0.95
UB	0.05
Db	0.14
UbUb	0.95

(no move)

File

True State

Predictions

Window Controls

Autopilot

Start

90

Tracking mode

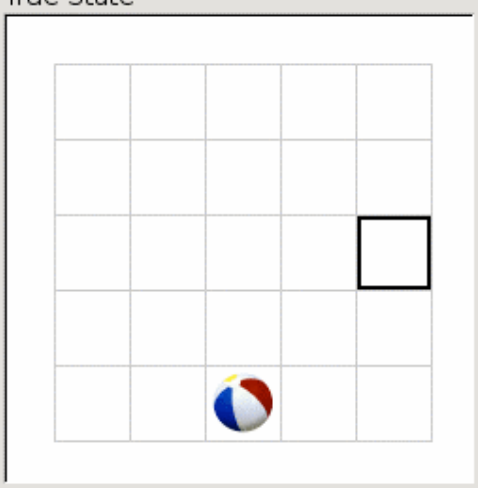
Core Tests

Test	Prob
Ub	1.00
UB	0.00
Db	0.98
UbUb	1.00

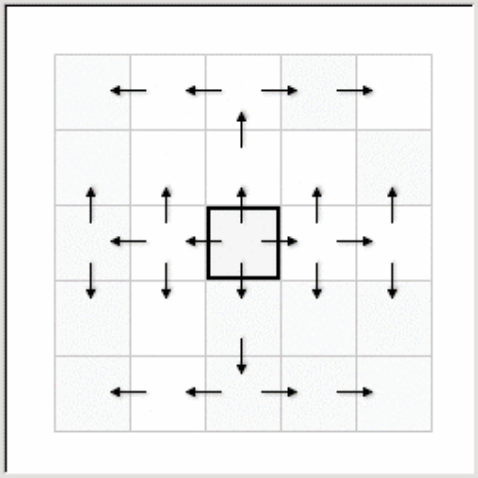
Move right

File

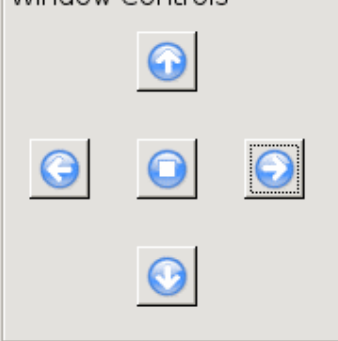
True State



Predictions



Window Controls



Autopilot

Start

90

Tracking mode

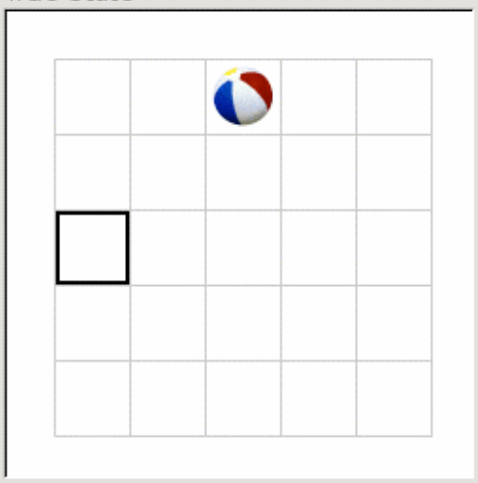
Core Tests

Test	Prob
Ub	1.00
UB	0.00
Db	0.98
UbUb	1.00

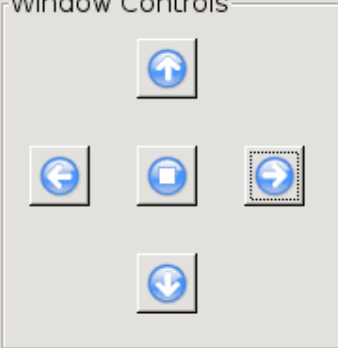
Move right

File

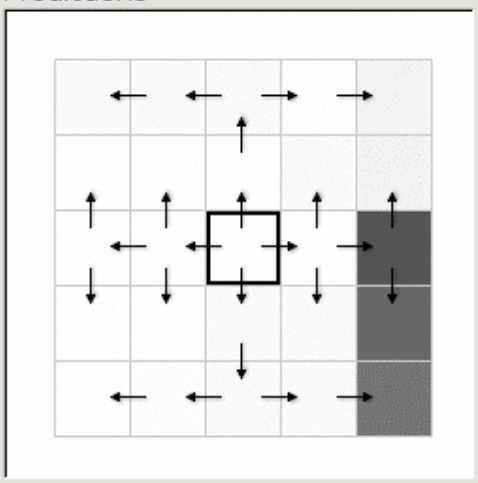
True State



Window Controls



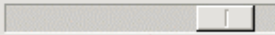
Predictions



Autopilot

Start

90



Tracking mode

Core Tests

Test	Prob
Ub	1.00
UB	0.00
Db	0.98
UbUb	0.98

Move right

Rethink state

- Think of states as answers to questions (i.e., predictions of outcomes of experiments one can do in the world)
 - Wallet's contents, Michael's location, presence of objects, ...
- Prior work
 - Learning deterministic FSA's (Rivest & Schapire, 1987); Multiplicity Automata (Beimel et al.)
 - Added stochasticity (Jaeger, 1999)
 - Added actions (PSR work, 2001)

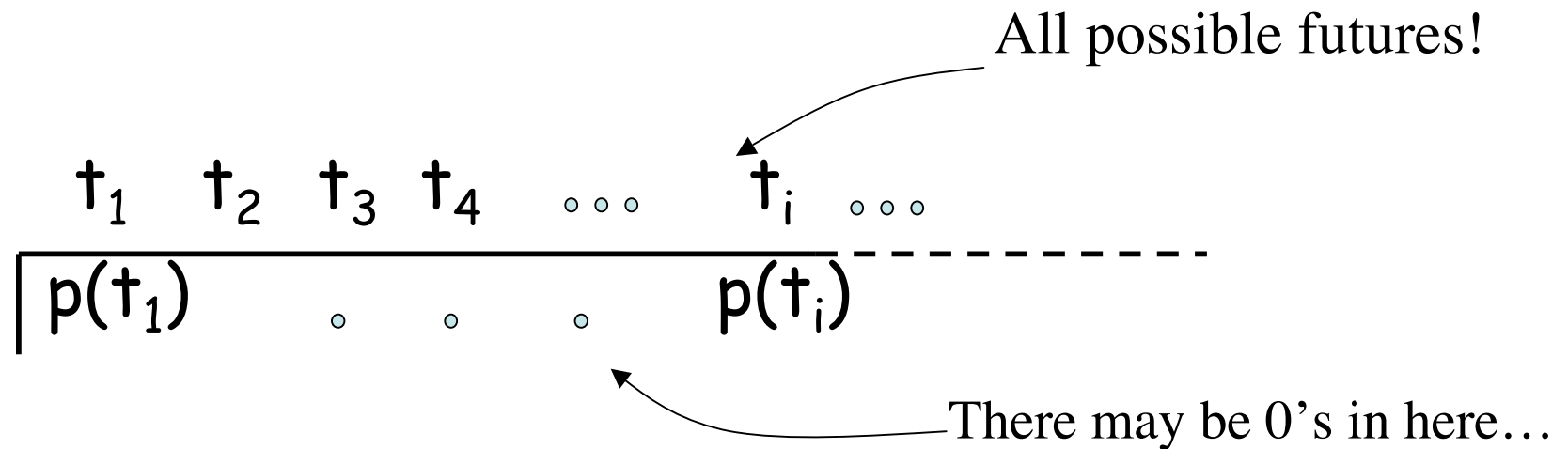
Which questions?

- What is a question (future, test)?
Uncontrolled system: a future is sequence of observations
 $t = o^1 o^2 \dots o^k$
Controlled system: a future is sequence of observations for a
sequence of actions $t = a^1 o^1 a^2 o^2 \dots a^k o^k$
- What is a (answer) prediction for a (question) future?
Uncontrolled system: $p(t) = \text{prob}(o_1=o^1, \dots, o_k=o^k)$
Controlled system:
 $p(t) = \text{prob}(o_1=o^1, \dots, o_k=o^k | a_1=a^1, \dots, a_k=a^k)$

We will show that this class of questions contains within it a subset whose answers are sufficient to model the state of interesting dynamical systems.

Discrete time, discrete observation, finite action systems

System Dynamics Vector



Mathematical construct that *IS* the system (not a model)

Any exact model of a system should be able to *generate* this vector

For both controlled and uncontrolled systems...

Lots of constraints on the entries of this vector

A “System” is a distribution over all futures...

System Dynamics Matrix

	t_1	t_2	t_3	t_4	\dots	t_i	\dots
$h_1 = \phi$	$p(t_1)$		○	○	○	$p(t_i)$	-----
h_2							
h_3							
h_j	$p(t_1 h_j)$					$p(t_i h_j)$	
⋮							

Again, this construct **IS** the system (not a model)

Uncontrolled system: $t_i = o^1 o^2 \dots o^k$ $h_j = o_1 o_2 \dots o_n$
 $p(t_i|h_j) = \text{prob}(o_{n+1}=o^1, \dots, o_{n+k}=o^k | o_1 o_2 \dots o_n)$

Controlled system: $t_i = a^1 o^1 \dots a^k o^k$ $h_j = a_1 o_1 \dots a_n o_n$
 $p(t_i|h_j) = \text{prob}(o_{n+1}=o^1, \dots, o_{n+k} | a_1 o_1 \dots a_n o_n, a_{n+1}=a^1, \dots, a_{n+k}=a^k)$

tests or experiments...

System Dynamics Matrix

	t_1	t_2	t_3	t_4	\dots	t_i	\dots
$h_1 = \phi$	$p(t_1)$		\circ	\circ	\circ	$p(t_i)$	
h_2							
h_3							
h_j	$p(t_1 h_j)$					$p(t_i h_j)$	

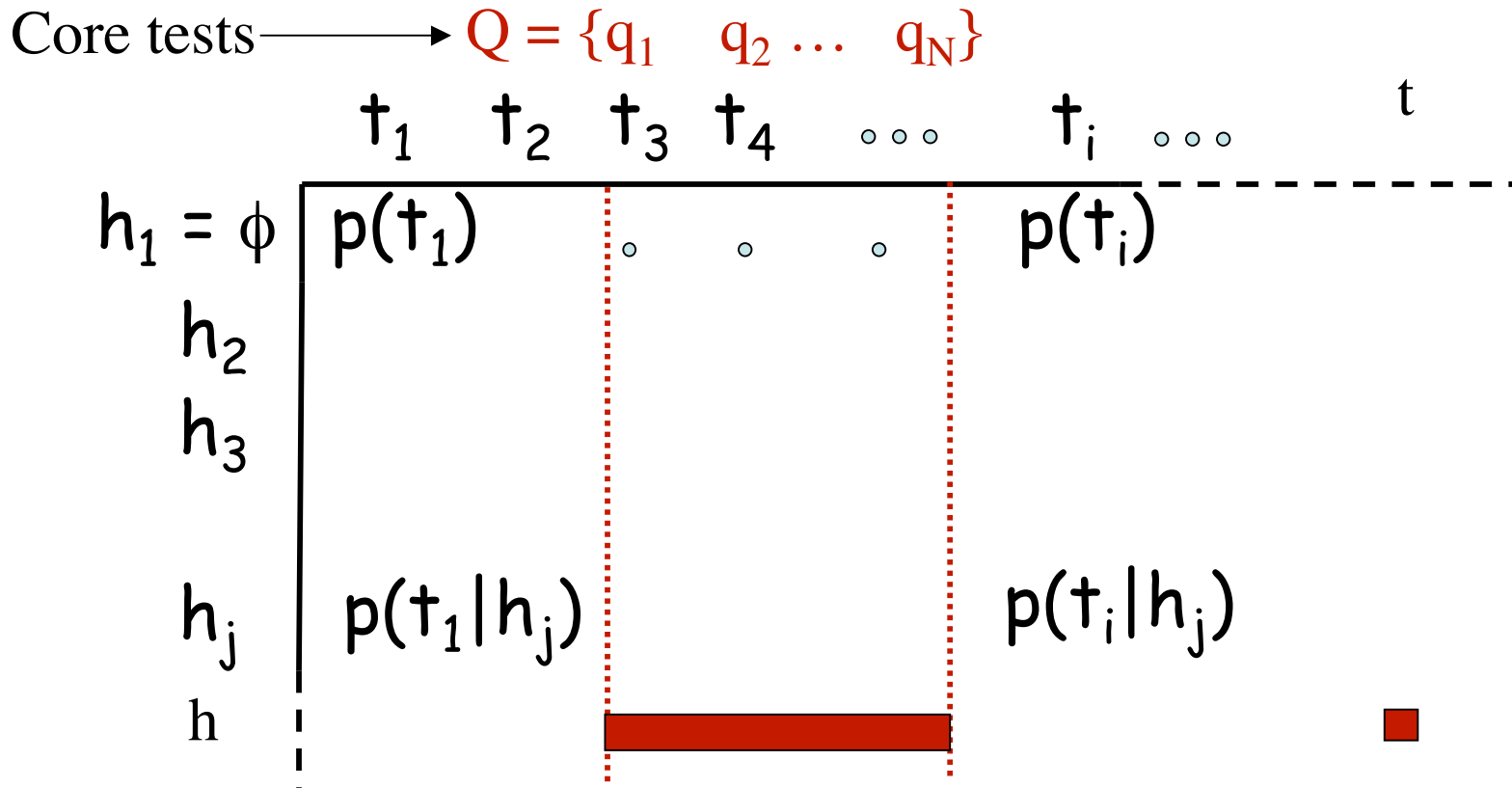
Only those histories that can happen

All rows are determined uniquely by the first row

Linear dimension of dynamical system is the **rank** (say N) of its system dynamics matrix (only consider finite rank systems here)

Any model must be able to **generate** this matrix

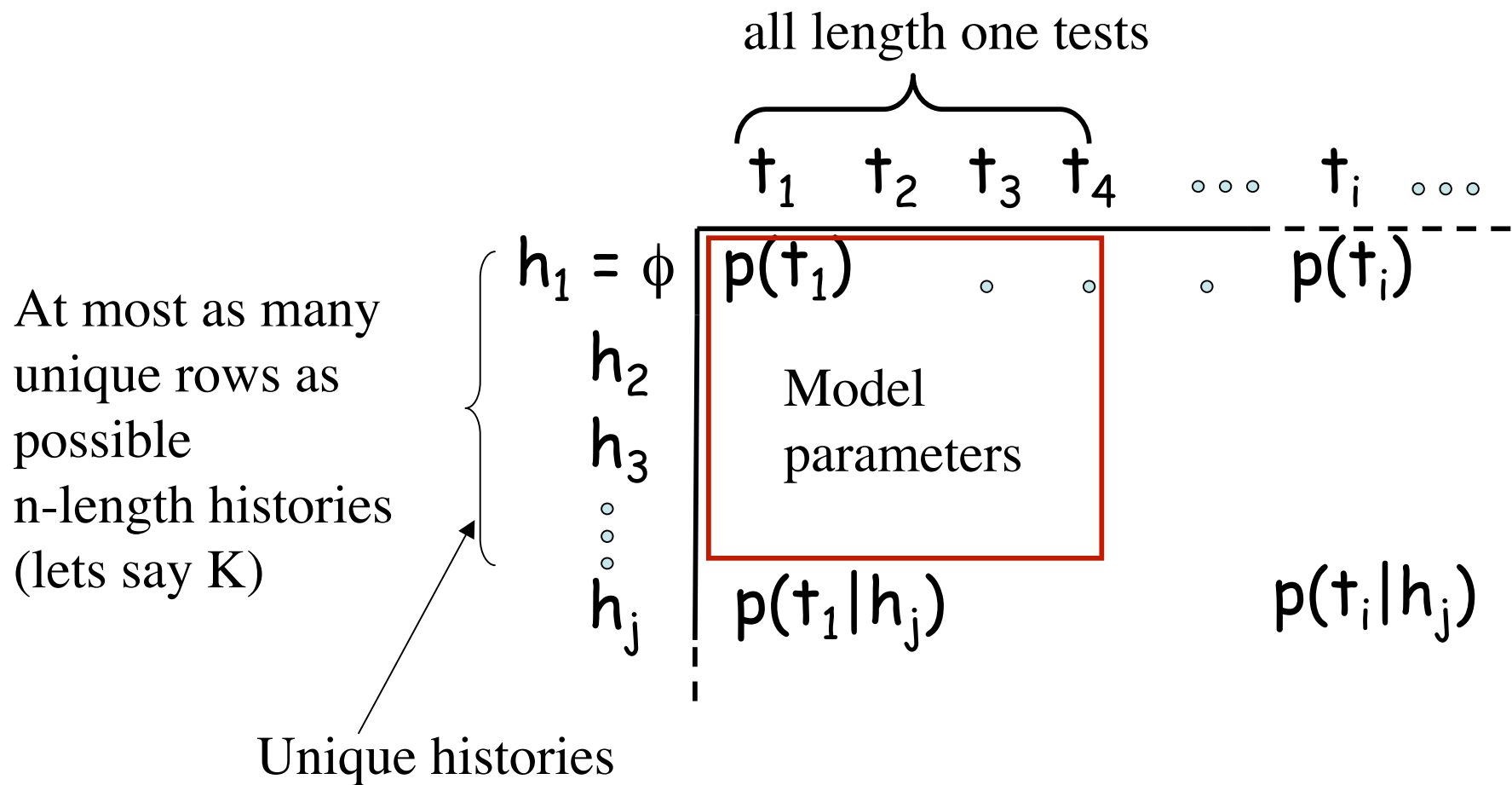
System Dynamics Matrix



$p(t|h) = p(Q|h)^T m_t$; note that m_t is independent of h !

Prediction for any test is *linear* combination of the predictions of the core tests $p(Q|h)$

n^{th} -order Markov Models



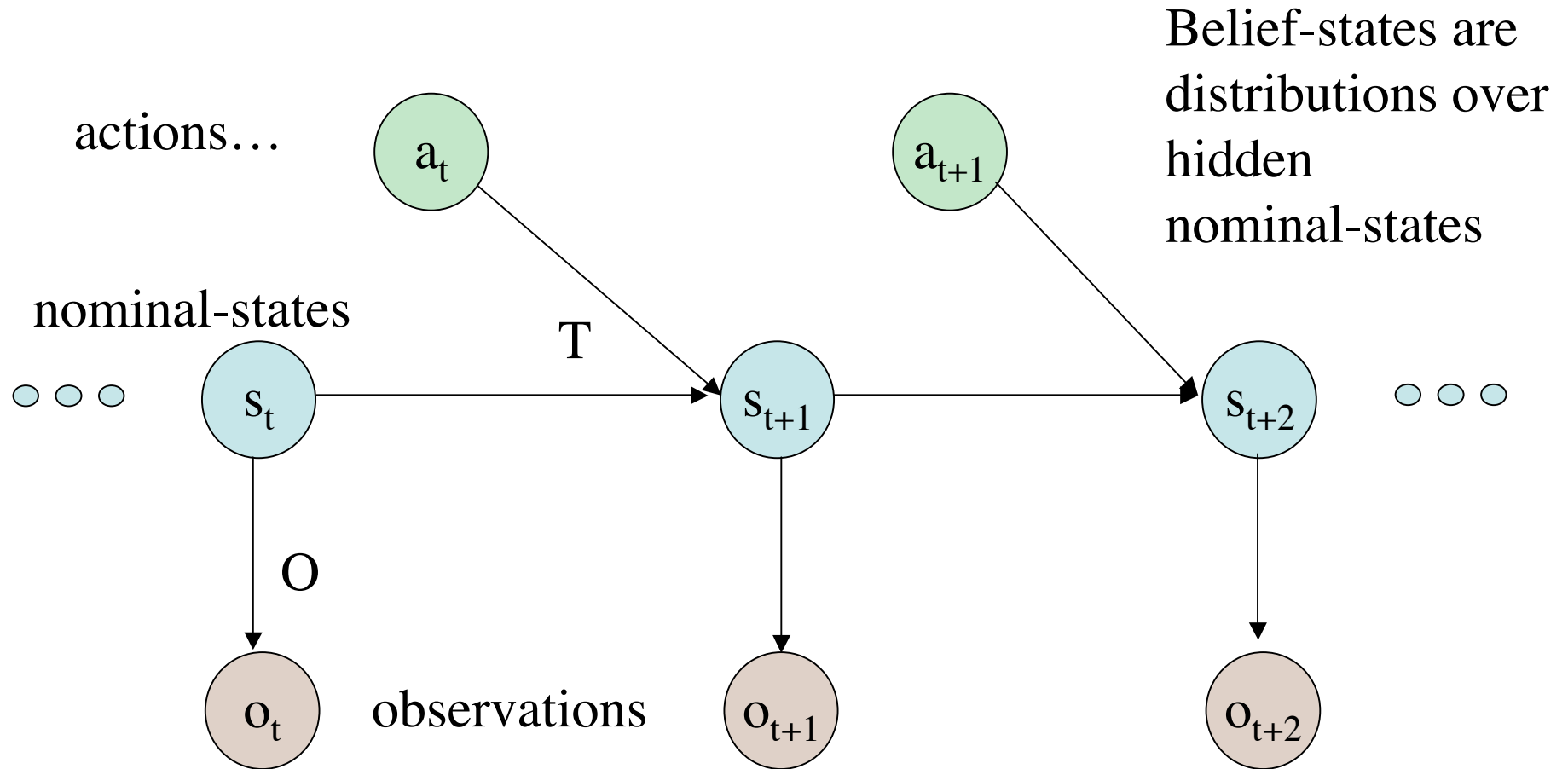
Theorem: All K -history Markov models are dynamical systems of linear-dimension $\leq K$

K-history Markov models...

- **Theorem:** there exist dynamical systems of linear-dimension N that cannot be modeled by *any finite*-order Markov model

Consider a system in which the first observation determines which of two sub-systems is entered...

POMDPs...



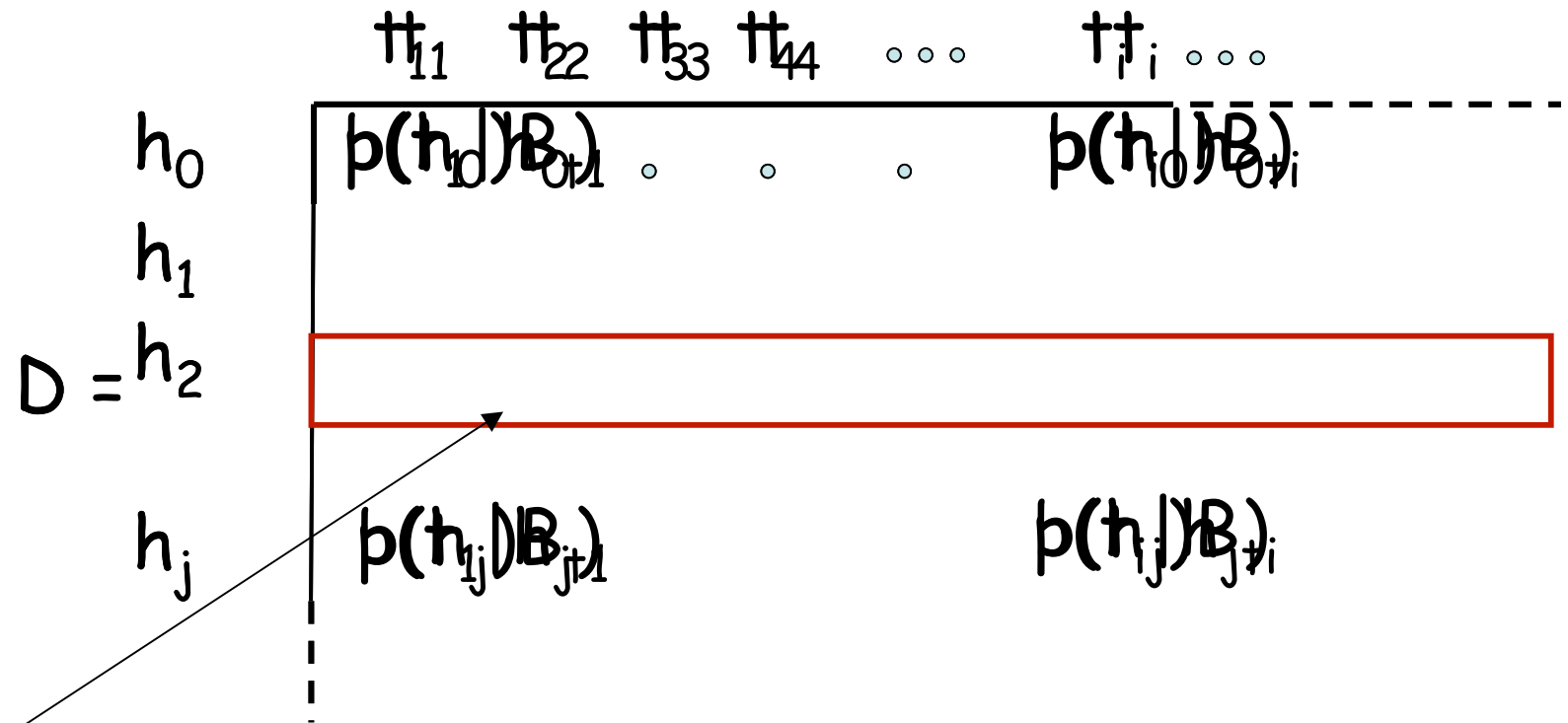
Learning POMDP models from data (EM)
does not work very well; almost no applications

POMDPs...

- n underlying or *nominal*-states
State representation for any history h (belief-state) $b(h)$ [a probability distribution over nominal-states]
- Update parameters
 - Transition probabilities T^a (one for every a); Observation probabilities $O^{a,o}$; (for every a,o)
 - Initial belief state $b(h_0)$
- $b(hao) = b(h)T^aO^{a,o}/Z = b(h) B^{a,o}/Z$
- For $t = a^1o^1...a^ko^k$; $p(t|h) = b(h) T^{a^1}O^{a^1o^1}...T^{a^k}O^{a^ko^k} = b(h) B_t$

predictions *linear* in belief-states

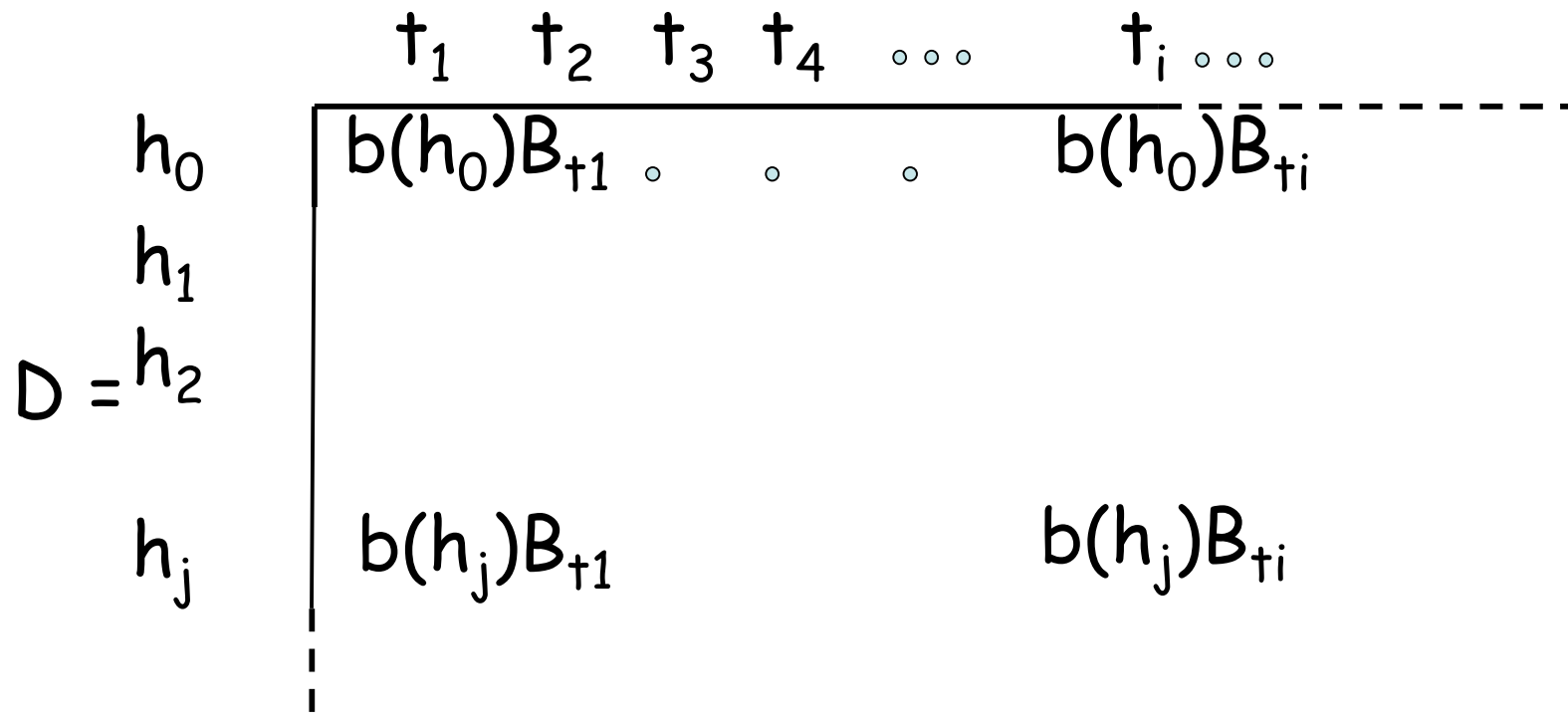
POMDPs



Linear combination with weights $b(h_2)$ of the rows corresponding to the n rows whose belief-states are *unit-basis*

• **Theorem:** Every POMDP with ' n ' nominal states is a dynamical system of linear-dimension $\leq n$

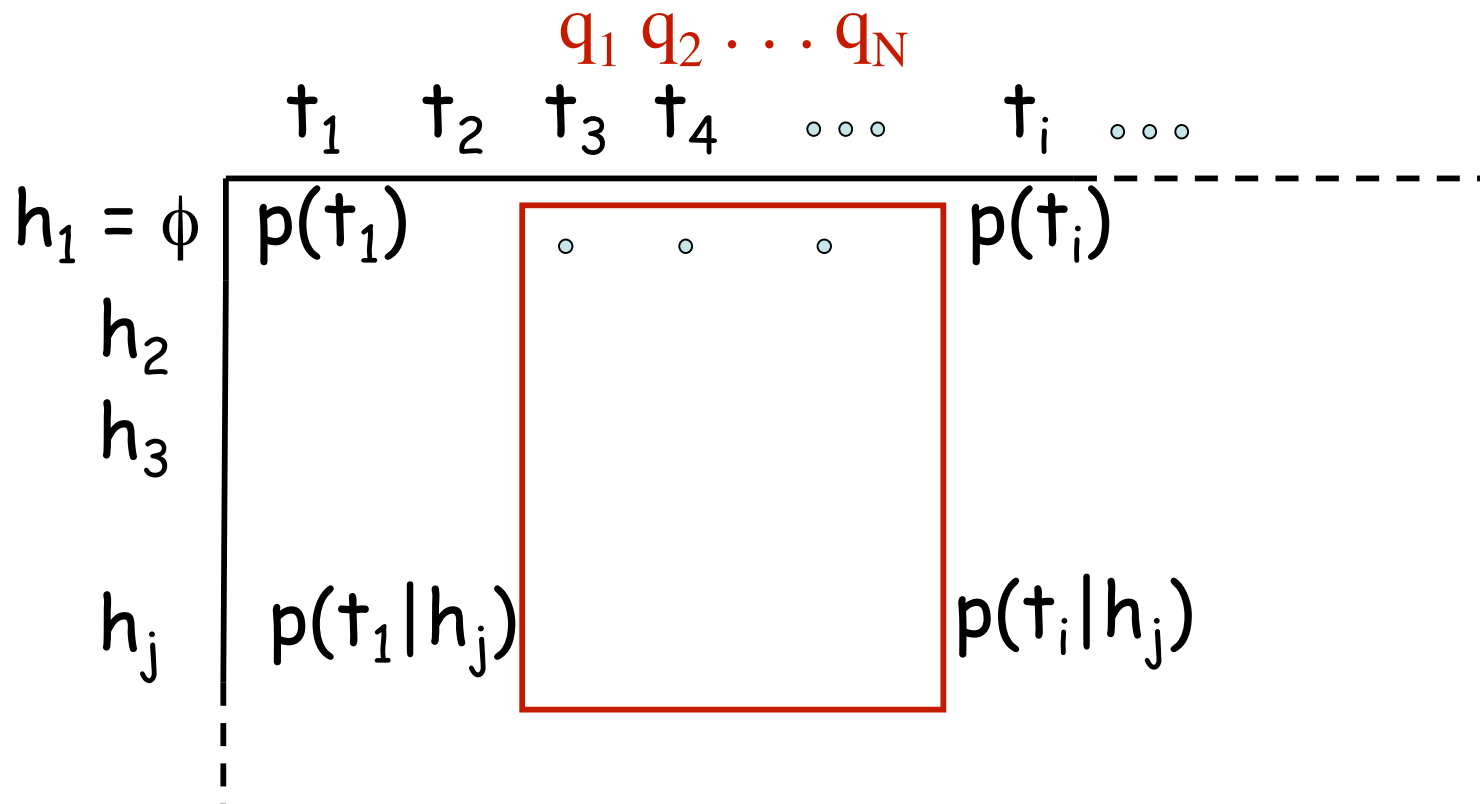
POMDPs



Theorem: there exist dynamical systems of finite linear-dimension that cannot be modeled by **any finite nominal-state** POMDP

Intuition: POMDP restricted to positive linear combinations...

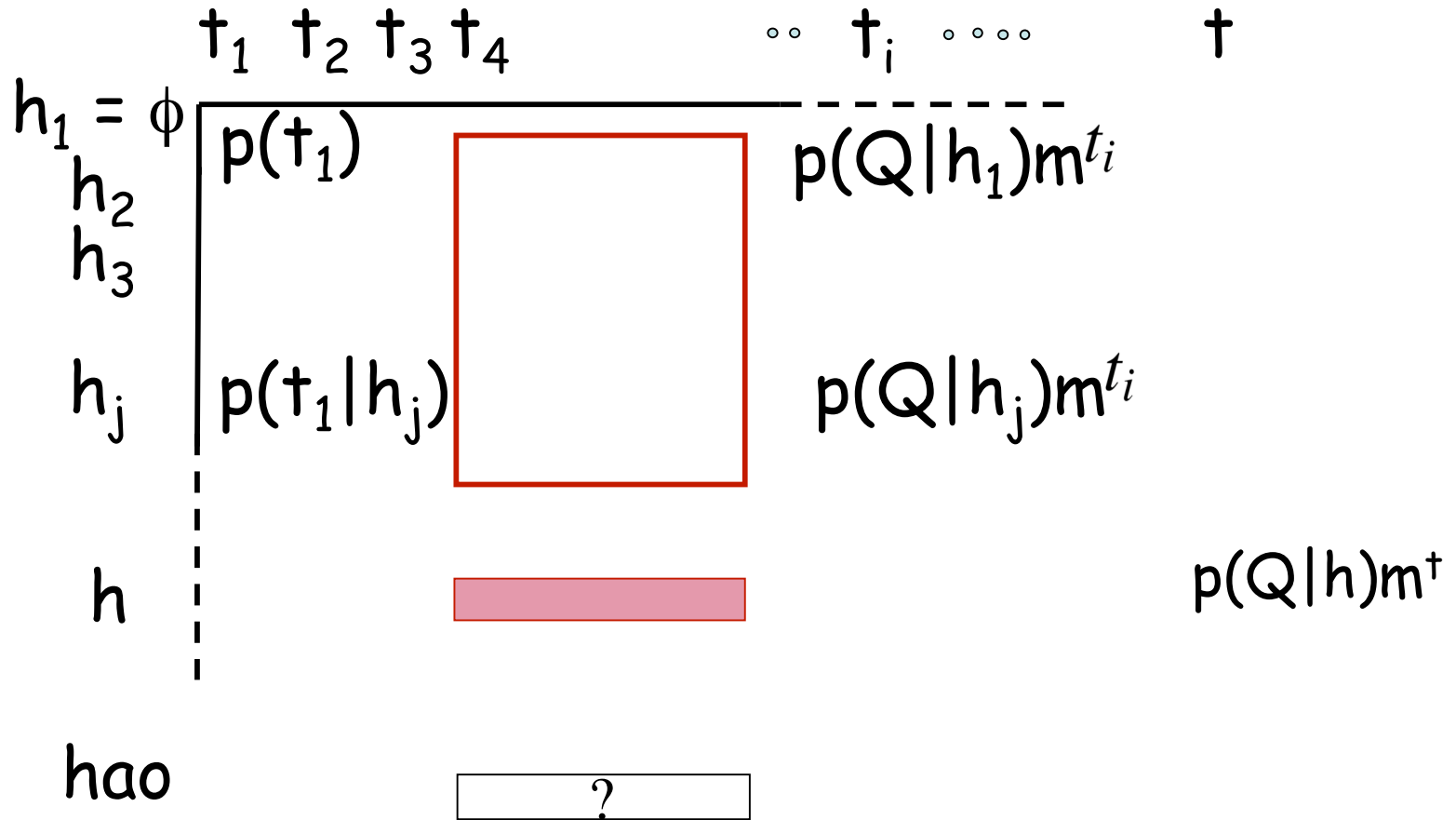
PSRs



Core tests $Q = \{q_1, q_2, \dots, q_N\}$

State representation: $p(Q|h) = [p(q_1|h) \dots p(q_N|h)]$

PSRs



$p(Q|h)$ is a sufficient statistic for history h

Updating Linear PSRs

- Update core test q_i on taking action a and observing o in history h

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

- Note: one only needs parameters for the one step extensions to the core tests!

$\{\forall a \in A, o \in O, q_i \in Q, m_{aoq_i}\}$ and $\{\forall a \in A, o \in O, m_{ao}\}$

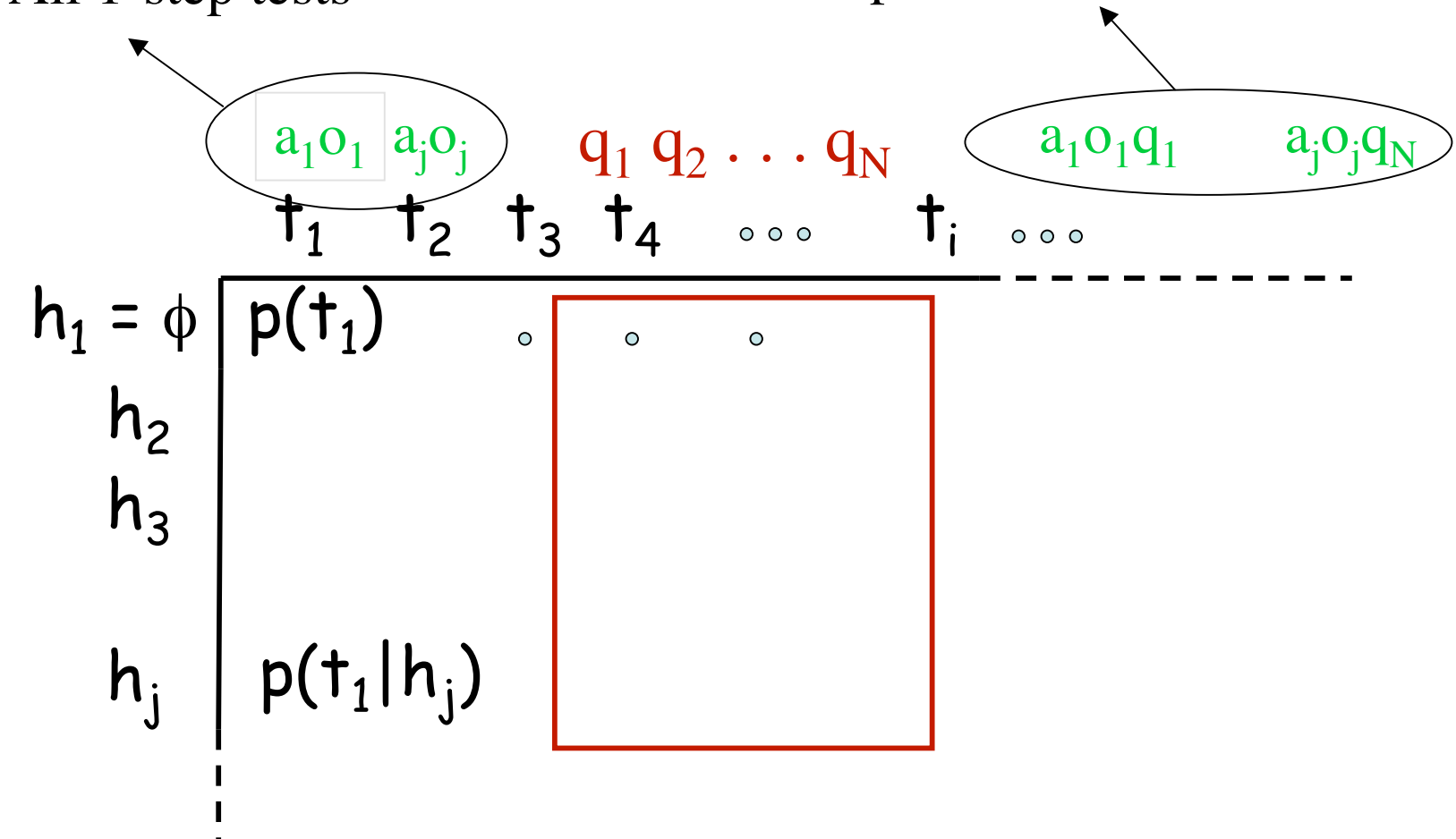
m's can have negative entries!!

model parameters

Update Parameters...

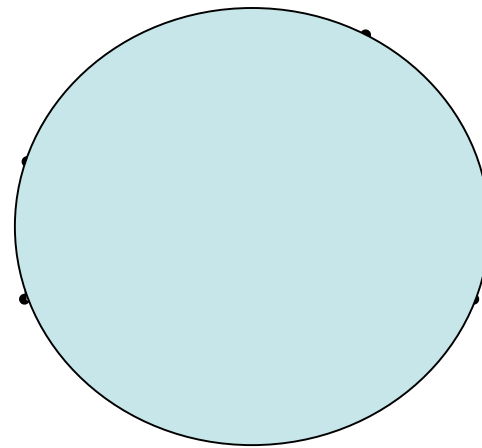
All 1-step tests

All 1-step extensions of core tests



Linear PSRs

- **Theorem:** Every discrete-time dynamical system of linear-dimension ' n ' is *equivalent* to a linear PSR with ' n ' core tests



Ok, where are we?

- Defined system dynamics matrix
- Defined linear PSRs

Result:

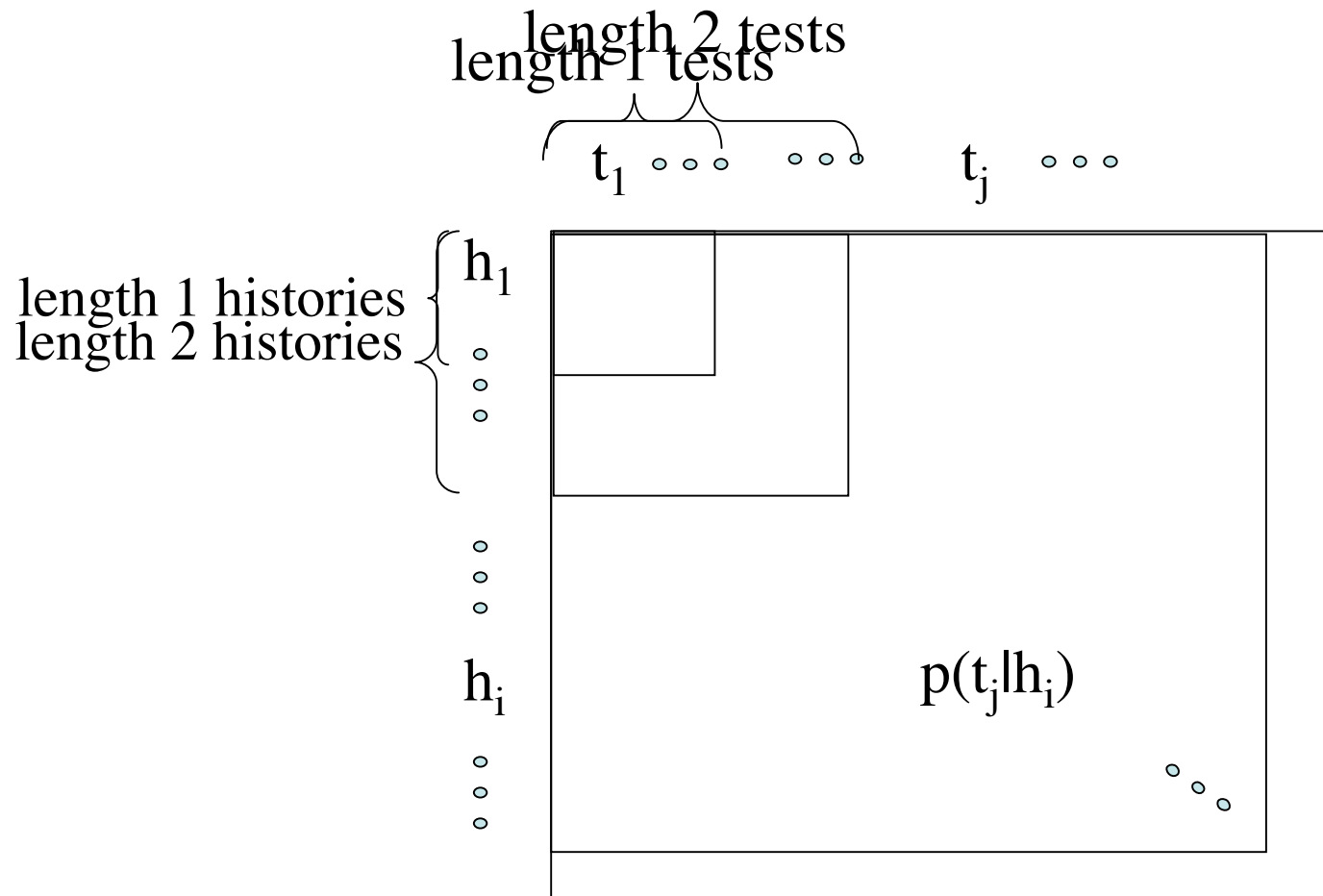
K-history Markov model \leftarrow K-nominal-state
POMDPs \leftarrow K-test linear PSRs $=$
dynamical systems of linear-dimension K

(applies to both controlled and uncontrolled systems)

Discovery & Learning in PSRs

- Discovery
 - Determine core tests given experience data
- Learning
 - Determine update parameters given core tests and experience data
- Discovery & Learning
 - Do both from experience data

Discovering PSR tests



If rank stops changing - you are done?

In practice Yes - In theory No!

Results on Learning & Discovery

Problem	Core Tests	Num. Act.	Num. Obs.	Reset Learning	Myopic Learning	Discovery Worked?
Tiger	2	3	2	0.000001	0.0000043	Yes
Paint	2	4	2	0.0000001	0.00001	Yes
Float-reset	5	2	2	0.000001	0.0001	Yes
Cheese Maze	11	4	7	0.000005	0.00037	Yes
Network	7	4	2	0.0004	0.00083	Yes
Bridge Repair	5	12	5	0.00015	0.0034	Yes
Shuttle	7	3	5	0.00026	0.027	Yes
4x3 Maze	10	4	6	0.00027	0.066	Yes

Nonlinear PSRs

- Suppose we allow non-linear predictions and non-linear updates?
- Nonlinear core tests $X=\{x_1, x_2, \dots, x_w\}$; sufficient statistic of history *but smaller in size than the linear-dimension of the dynamical system*
- State representation $p(X|h)$ for history h
- Prediction for test 't', $p(t|h) = f_+(p(X|h))$ for some nonlinear function 'f' (independent of 't')
- Update process

$$p(x_i|h_{ao}) = \frac{p(aox_i|h)}{p(ao|h)} = \frac{f_{aox_i}(p(X|h))}{f_{ao}(p(X|h))}$$

Beyond Linear PSRs

- Nonlinear PSRs
 - Exponential compression over Linear PSRs and POMDPs in some deterministic systems (Rudary & Singh, NIPS 2003)
- PSRs for continuous systems

Predictive Linear Gaussian (PLG)

- The distribution over the next “N” observations given current history is Gaussian.

$$Z_t = [Y_t, Y_{t+1}, \dots, Y_{t+n}]$$

$$Z_t | h_t \sim N(\mu_t, \Sigma_t)$$

- The predictive state is the *mean* and *covariance* matrix of the Gaussian
- The N+1st observation is computed as a linear function of the next N observations.

$$Y_{t+n+1} = g^T Z_t + \eta_{n+t+1}$$

$$\text{Cov}[Z_t, \eta_{n+t+1}] = C \quad \eta_{n+t+1} | h_t \sim N(0, \sigma^2)$$

PLG vs. LDS

Theorem: Every linear dynamical system (LDS) with dimension 'n' can be modeled as a PLG with dimension 'n'. (Kalman Filters)

A PLG has no hidden variables

We can derive consistent learning algorithms for a PLG

PLG Learning vs. EM for LDSs

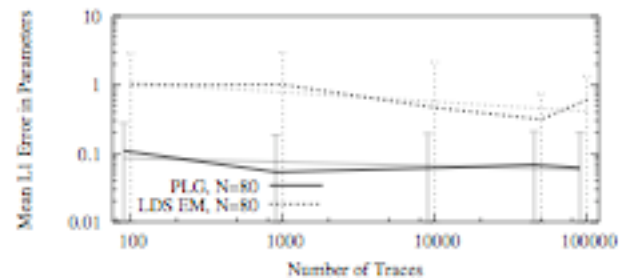
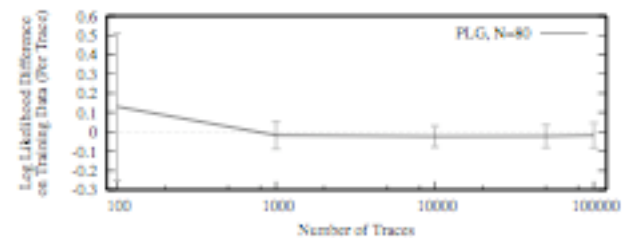
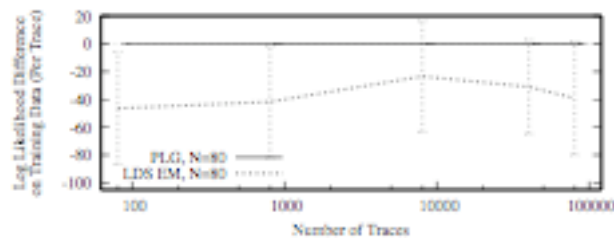
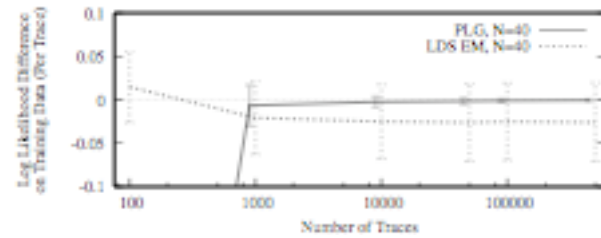
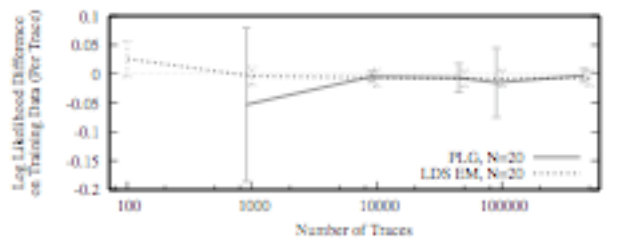


Illustration of PLG

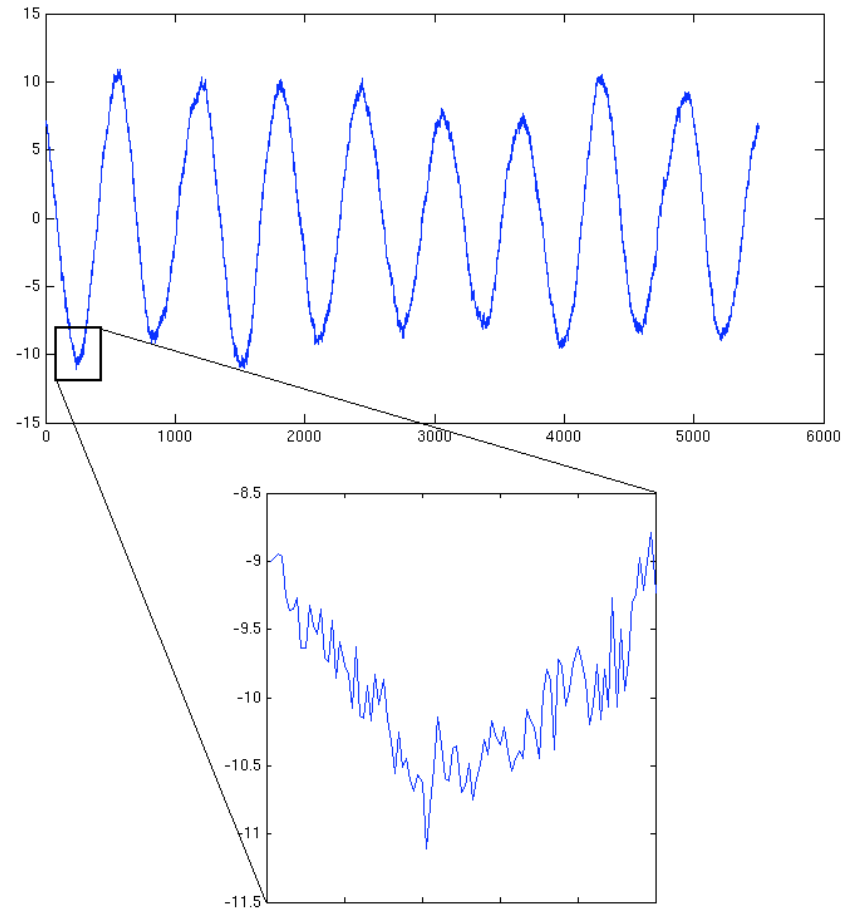
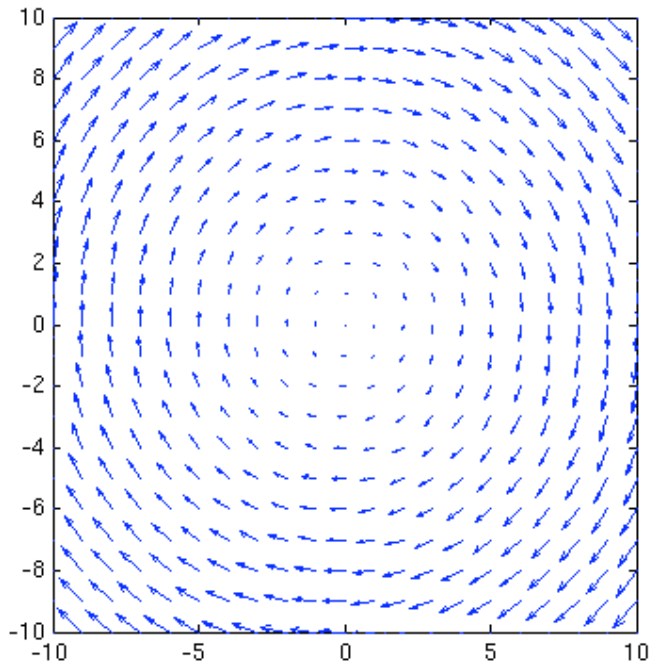


Illustration of PLG

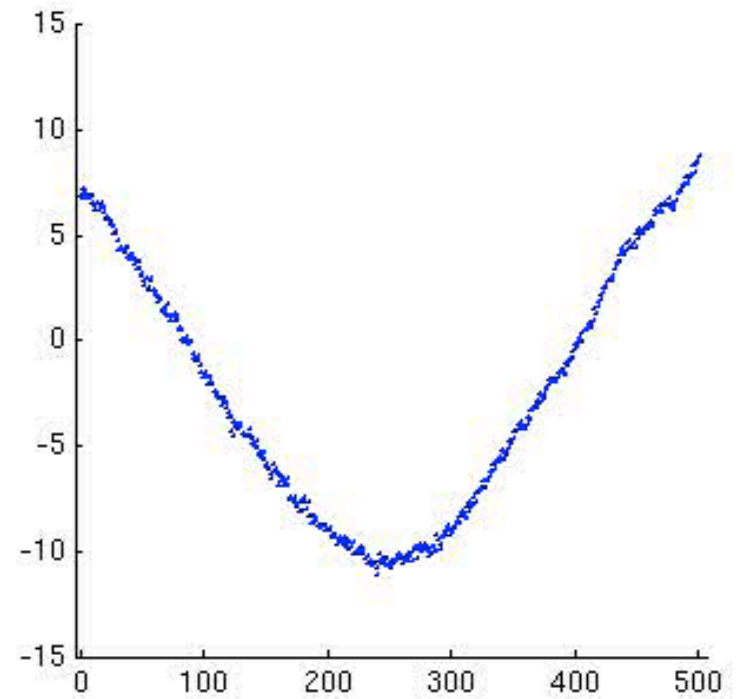
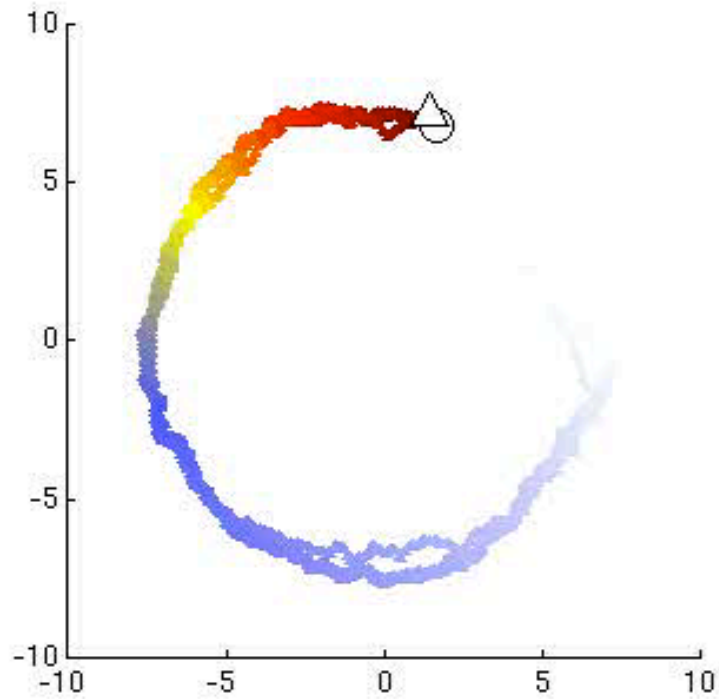
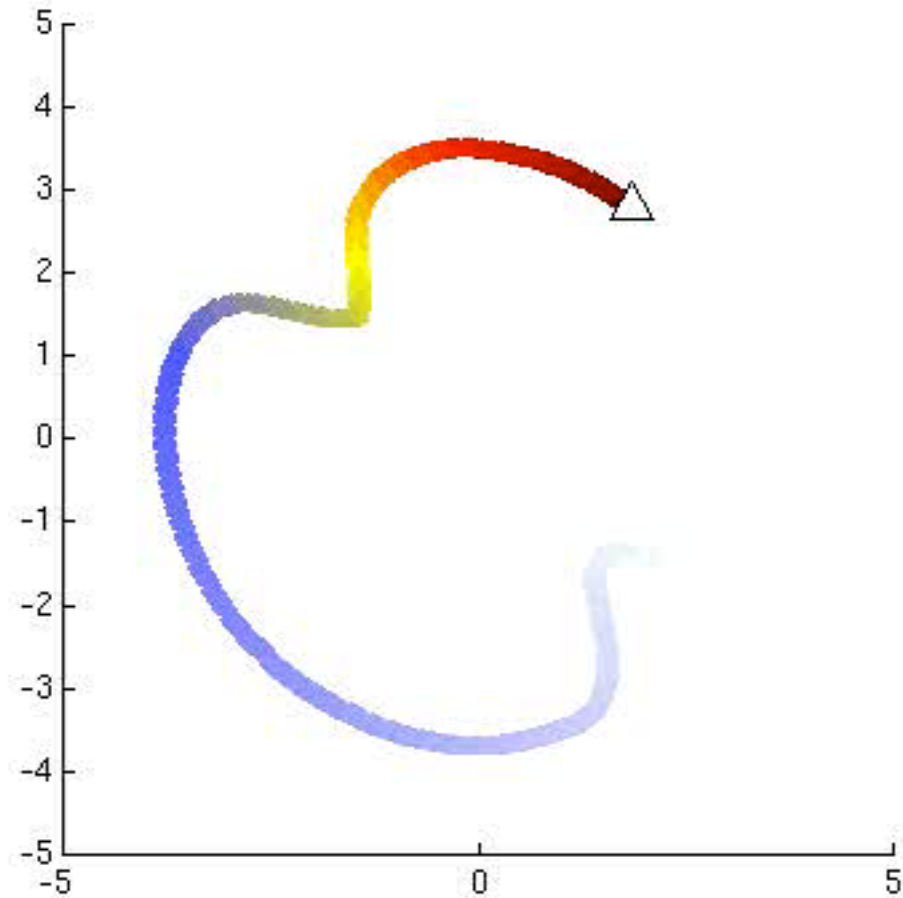
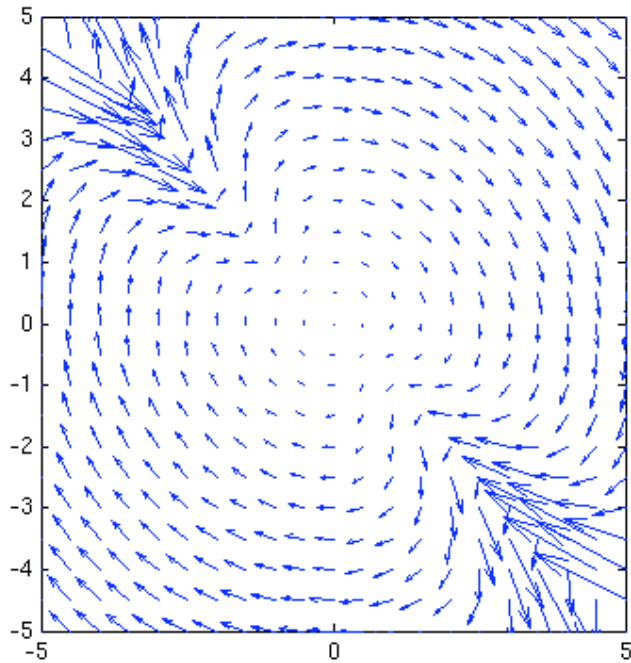


Illustration of Kernel PLGs



Summary

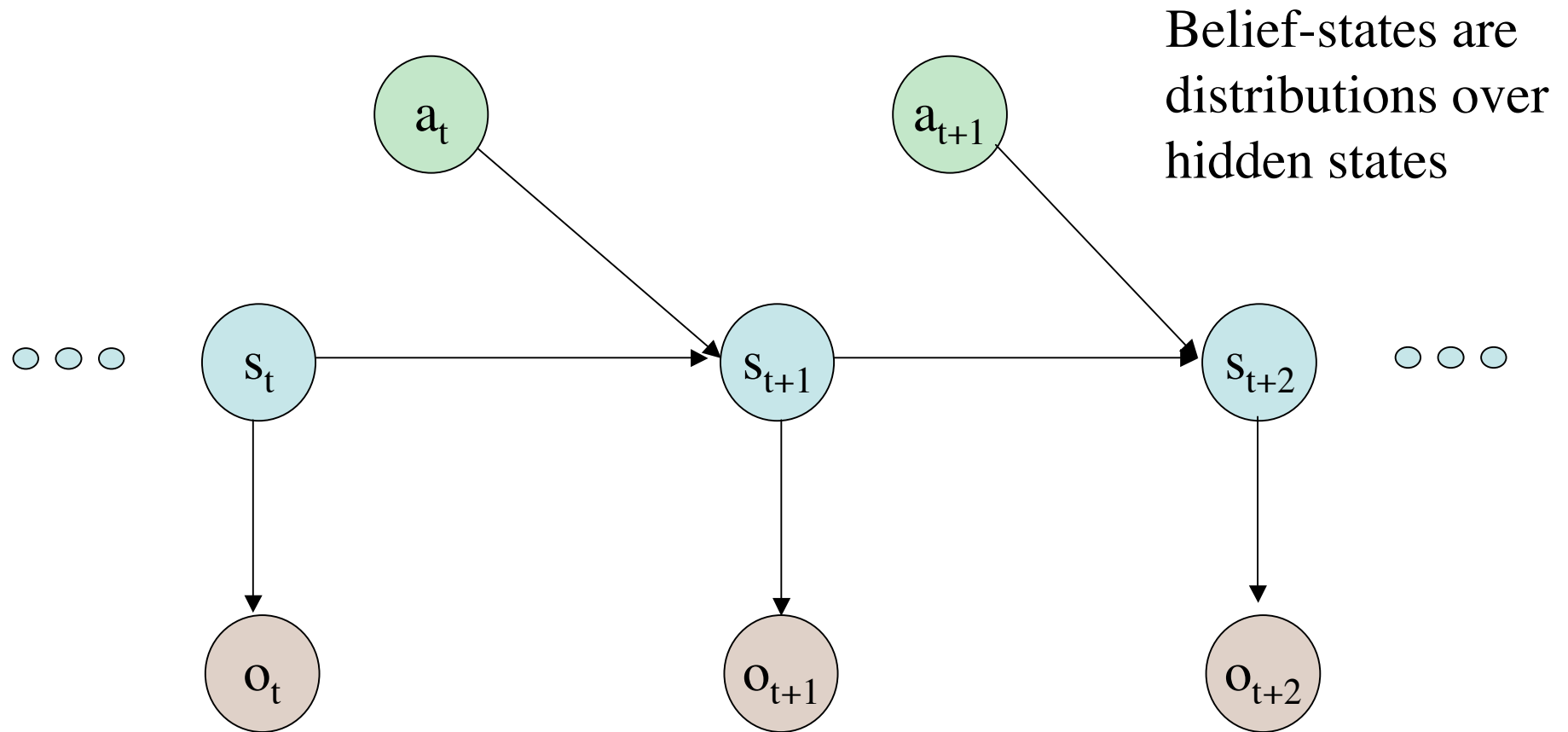
- Knowledge expressed entirely in observable quantities
 - is possible
 - is no less compact than at least unstructured traditional (latent variable) representations
 - may be more efficiently learnable/plannable/maintainable...
- So far: sufficient representations
- Next: efficient (perhaps structured) observable representations

Conclusion

- MDPs are great!
- We are making progress in going beyond MDPs (in states, actions & *rewards*)
- Lots of work to be done...

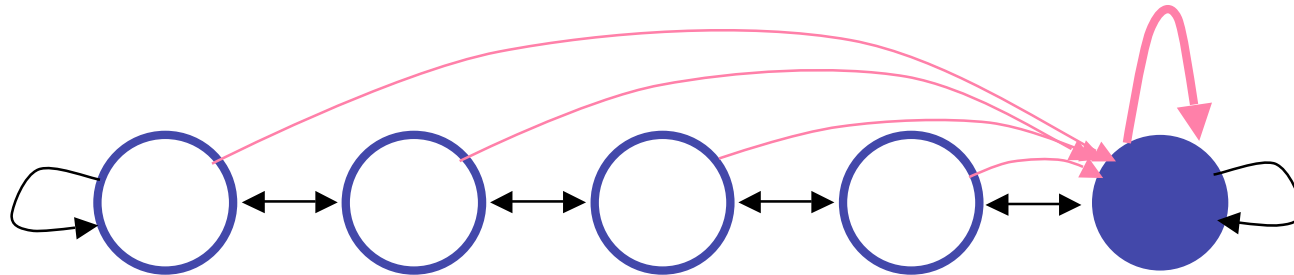
Leftover Slides

Graphical Model for POMDPs



Learning POMDP models from data (EM)
does not work very well; almost no applications

Float/Reset

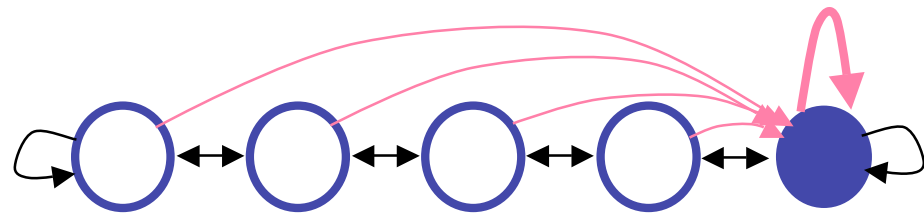


Float: Random walk.

Reset: Go right, observe 1 if already there.

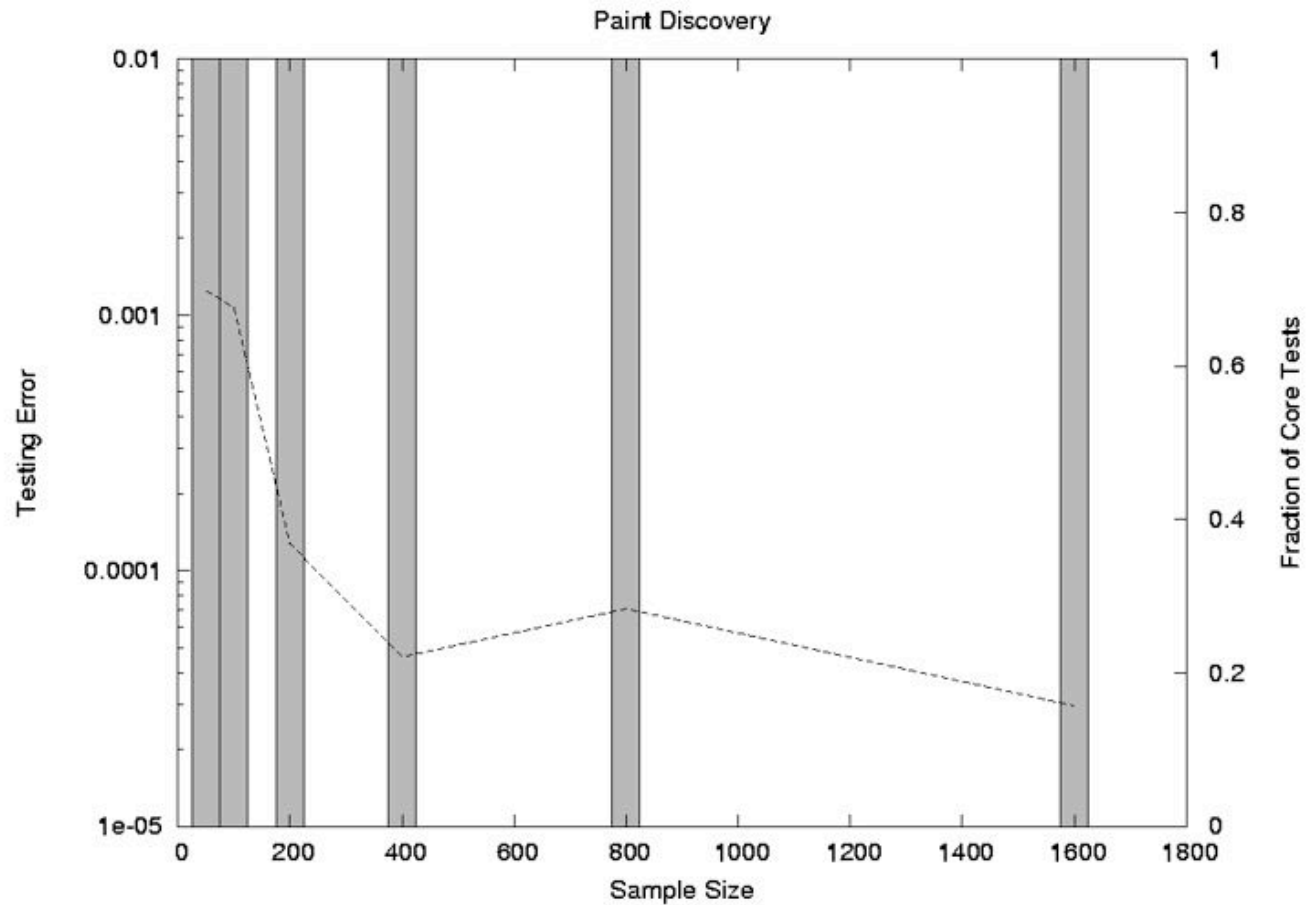
Float/Reset (Linear PSR)

- F O; R O
- F O R O
- F O F O R O
- F O F O F O R O

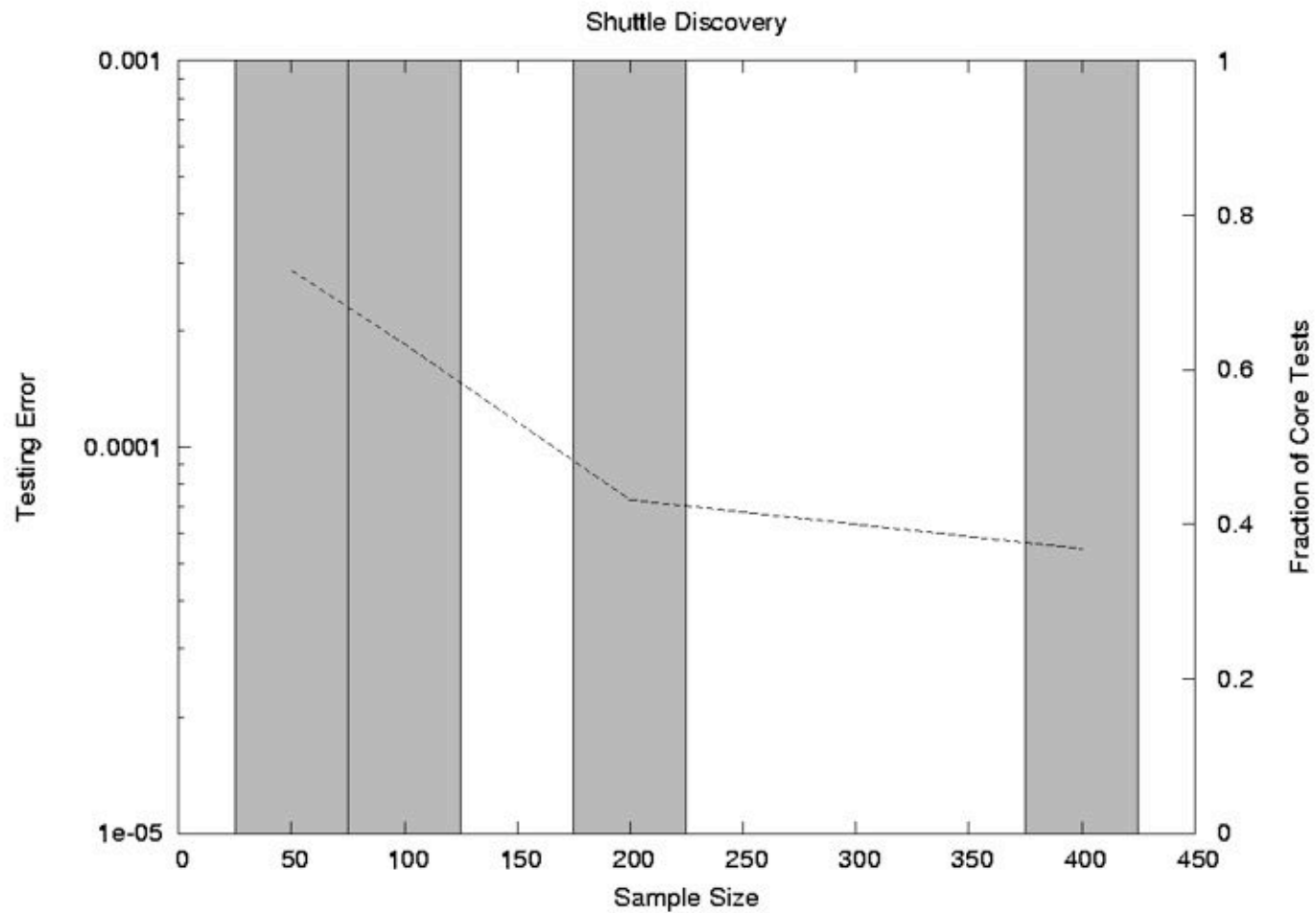


$$\begin{aligned}
 p(\text{F O F O F O R O} | h \text{ F O}) &= 0.25 p(\text{R O} | h) \\
 &\quad - 0.0625 p(\text{F O R O} | h) \\
 &\quad + 0.750 p(\text{F O F O R O} | h)
 \end{aligned}$$

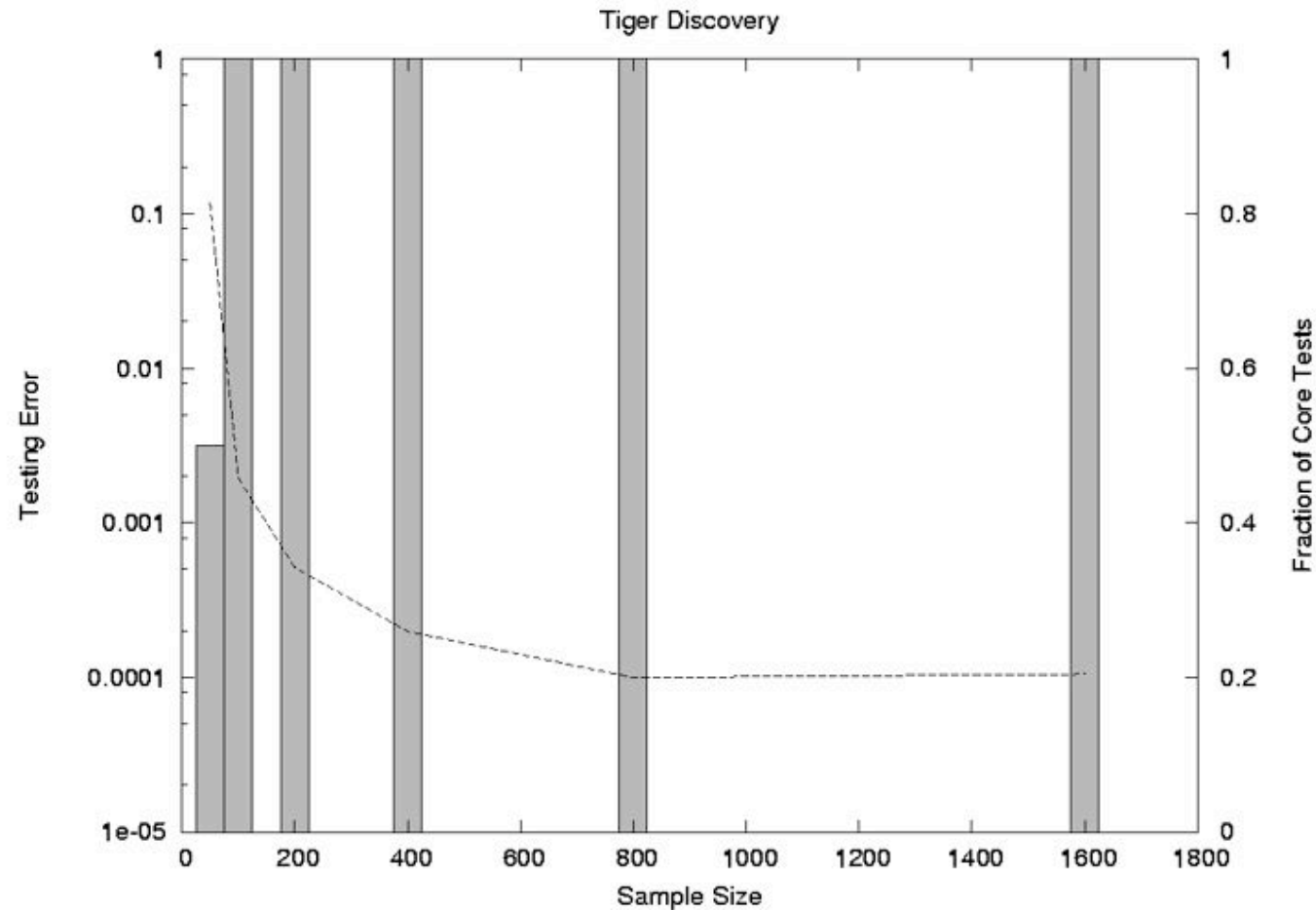
Learning & Discovery in Paint



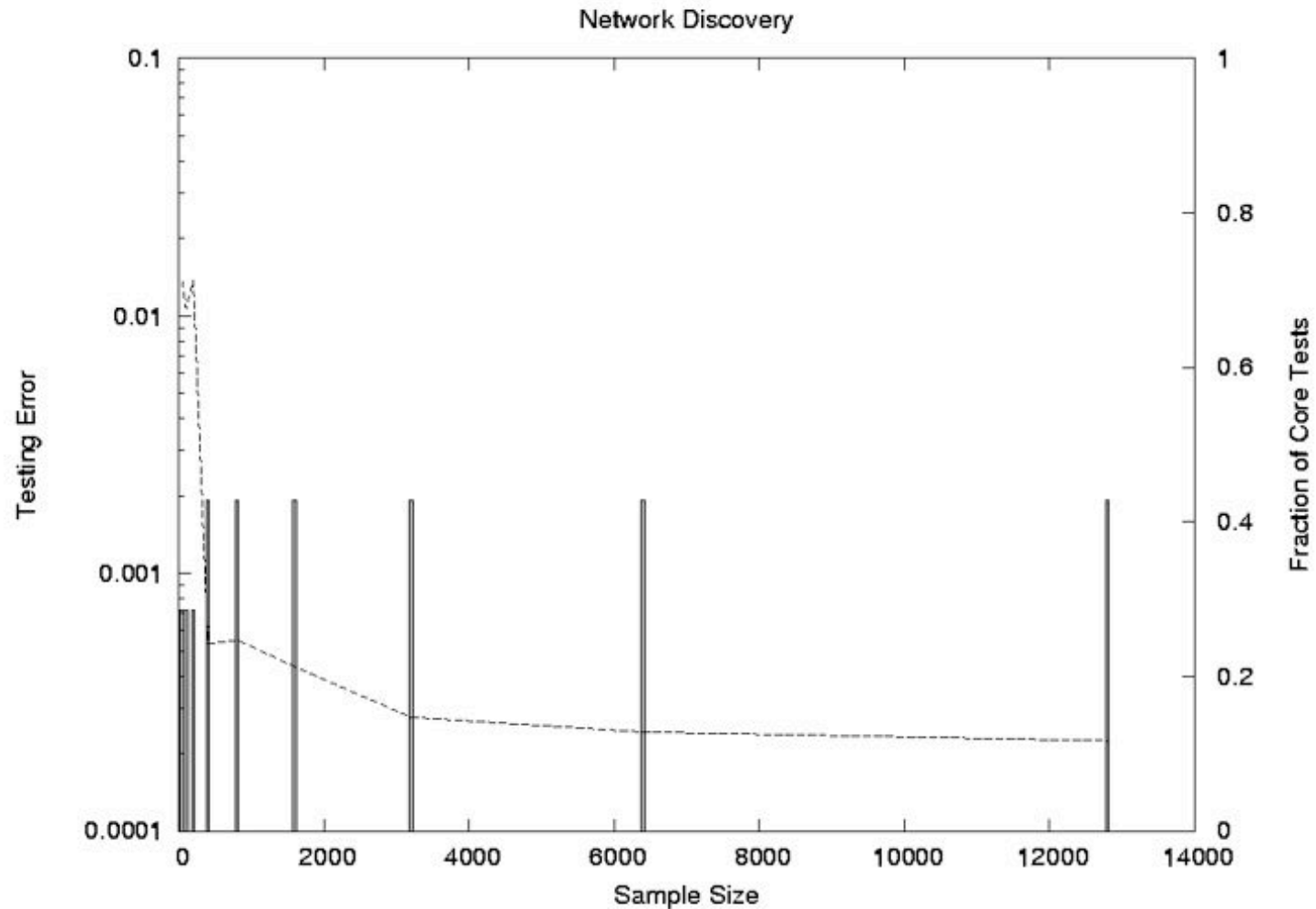
Learning & Discovery in Shuttle



Learning & Discovery in Tiger



Learning & Discovery in Network



PSRs - a definition

- Test $t = \{a_1 o_1 a_2 o_2 \cdots a_n o_n\}$
- Predictions for test t
$$p(t|h) = \text{Pr}(o_1 o_2 \cdots o_n | h a_1 a_2 \cdots a_n)$$
- A core set of tests $Q = \{t_1, t_2, \dots, t_m\}$
- State at history h : $p(Q|h) = [p(t_1|h) \dots p(t_m|h)]$
- There exists Q , such that $p(Q|h)$ is *sufficient statistic* for all histories h !
- \Rightarrow For arbitrary test t , $p(t|h) = f_t(p(Q|h))$