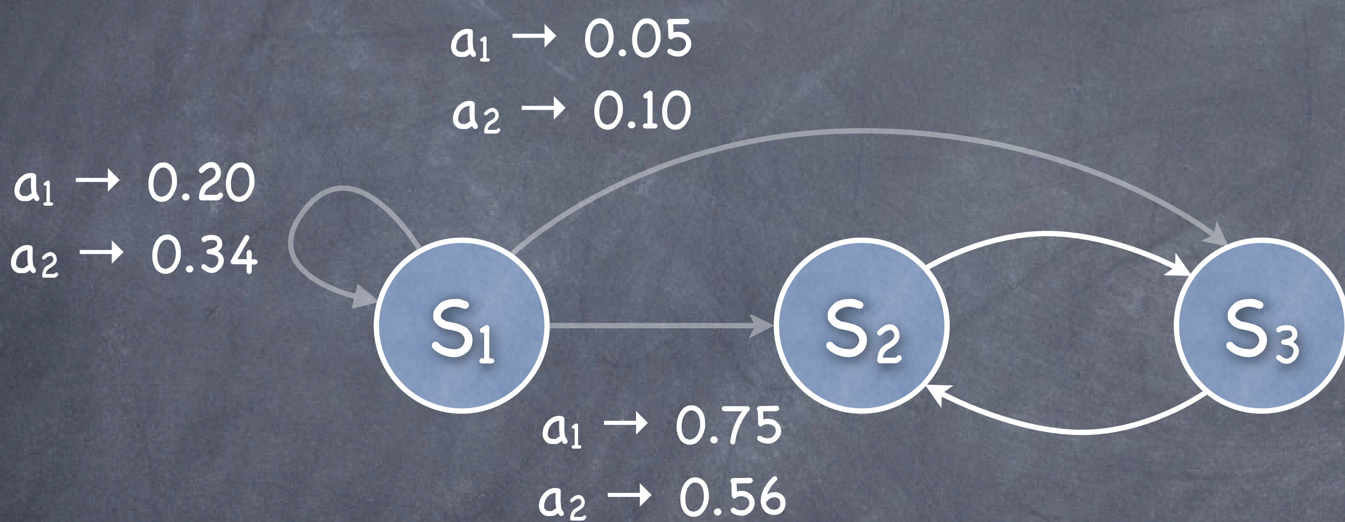
An Introduction to Predictive State Representations (PSR)

Rodrigo Ventura
yoda@isr.ist.utl.pt

Markov Decision Process

(MDP)



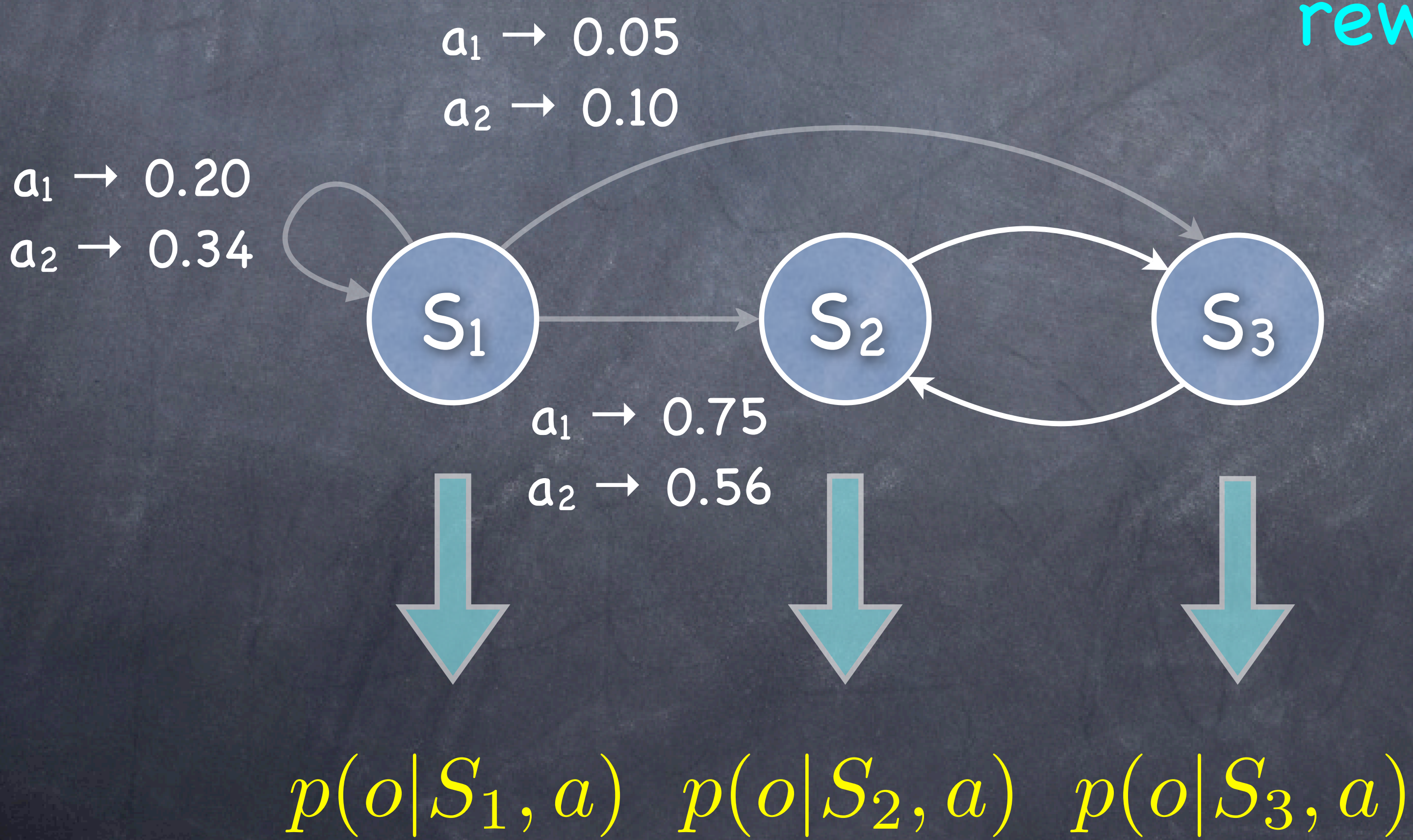
reward R(s)

$$T_1 = \begin{bmatrix} 0.20 & 0.75 & 0.05 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad T_2 = \begin{bmatrix} 0.34 & 0.56 & 0.10 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Partially Observable Markov Decision Process

(POMDP)

reward R(s,a)



POMDP formulation

action a

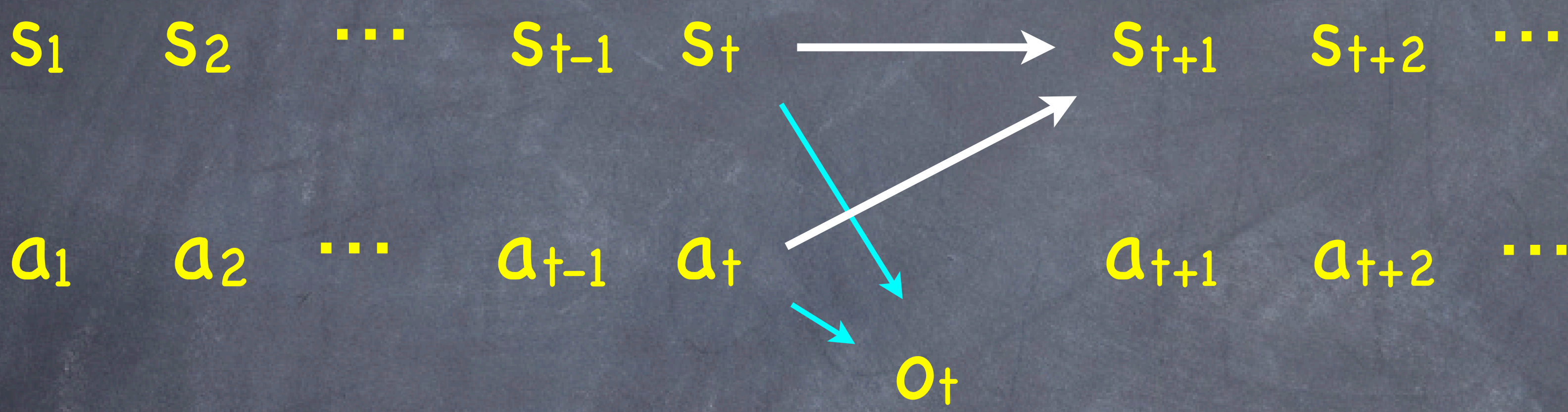
state s → state s'

transition prob $T(s,a,s')$

observation prob: $O(s,a,o)$

reward: $R(s,a)$

Bayesian dependencies



$$T(s_t, a_t, s_{t+1}) = P(s_{t+1} | s_t, a_t)$$

$$O(a_t, s_t, o_t) = P(o_t | s_t, a_t)$$

POMDP parameters

- Transition matrix for each action a

$$[T_a]_{ij} = P(s_{t+1} = j | s_t = i)$$

- Observation (diagonal) matrix for each action-observation pair (a,o)

$$[O_{ao}]_{ii} = P(o_t = o | s_t = i, a_t = a)$$

Belief vector

- belief vector: $b(h) = [\ b_1 \dots b_k\ ]$

  probability of being in each state

  given an history h of the system

  $$h = [\ a^1 o^1 \dots a^k o^k\ ]$$

Belief update

- belief update: given an action a and an observation o, the belief vector is updated as

$$b(hao) = \frac{b(h)T_a O_{ao}}{b(h)T_a O_{ao}\mathbf{1}}$$

- NOTE that b(h) encodes all information about the history → concept of state

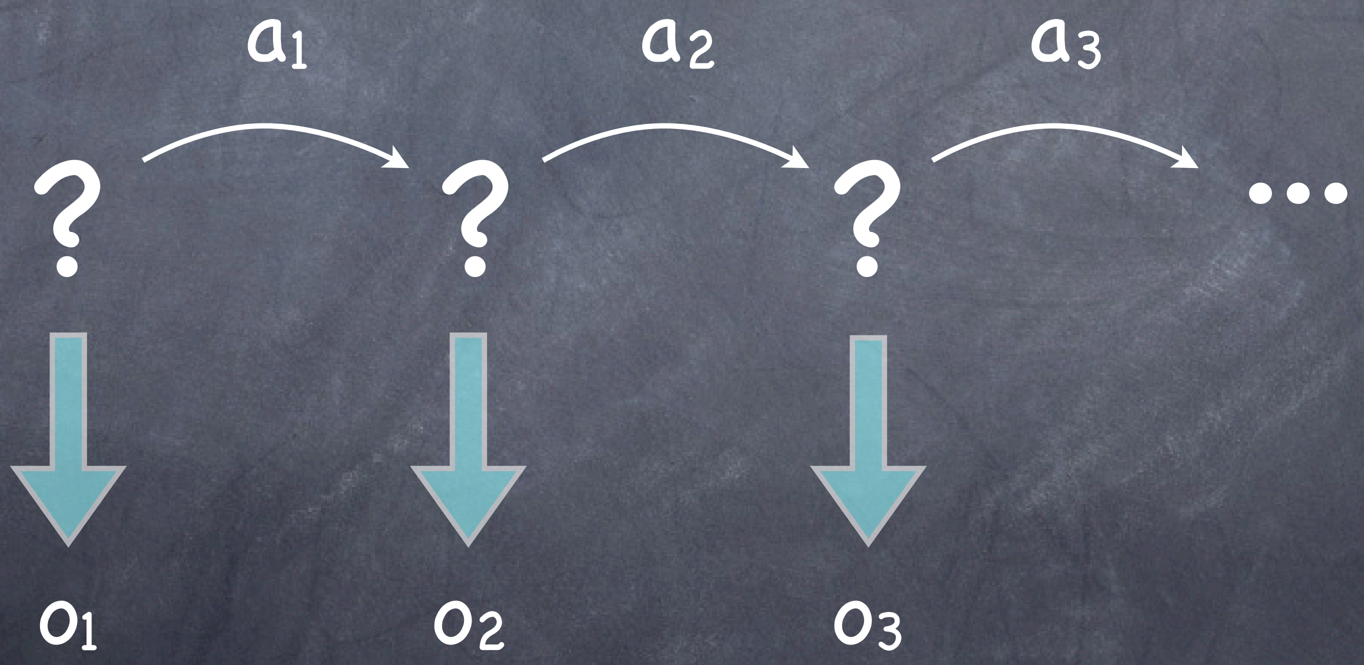
POMDP

- are OK if we know a priori the structure of the system
e.g., the state is the position of the robot

- but in unstructured, dynamic environments, it is cumbersome to learn a POMDP model
e.g., the real world?

Predictive State Representations (PSR)

- How about removing the (hidden) states?



$$a^1o^1a^2o^2a^3o^3\ldots$$

- a k-length test

  $t = a^1 o^1 \ldots a^k o^k$

  $p(t) = \text{prob}(O^1=o^1,\ldots, O^k=o^k \mid A^1=a^1,\ldots, A^k=a^k)$

- infinite system-dynamics vector d
  given an ordering of tests t_1, t_2, t_3, ...

  $d = [\ d_1\ d_2\ d_3\ \ldots\ ]$

  where $d_i = p(t_i)$

- System-dynamics matrix D

$D_{ij} = p(\ t_j \mid h_i\ )$

i.e., probability of test

$t_j = a^1 o^1 \ldots a^n o^n$

given a history

$h_i = a_1 o_1 \ldots a_m o_m$

	t_1	$\cdots$	t_j	$\cdots$
$h_1 = \phi$	$p(t_1 \mid h_1)$		$p(t_j \mid h_1)$	
h_2				
$\vdots$				
h_i	$p(t_1 \mid h_i)$		$p(t_j \mid h_i)$	
$\vdots$				

linear dimension

$\Leftrightarrow$

rank of D

- a POMDP with k states → D of rank ≤ k

$$P(t_j|h_i) = \sum_s P(t_j|s)P(s|h)$$

B **U** **=** **D**

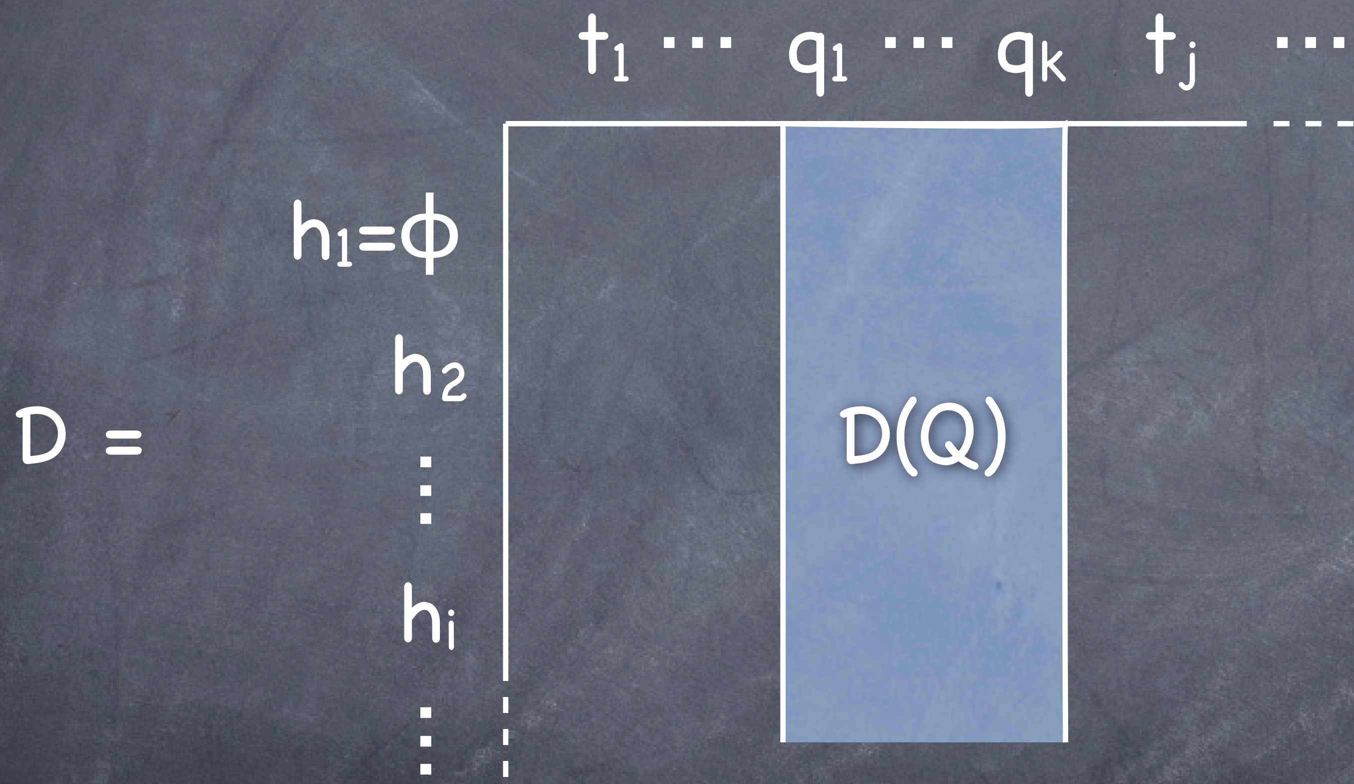
B: columns s_1 … s_k; rows h_1, ⋮, h_i, ⋮ with entries $b(h_1)$, ⋮, $b(h_i)$, ⋮ — $\infty \times k$

U: columns t_1 … t_j; rows s_1 … s_k with entries $p(t_j|s_i)$ — $k \times \infty$

D: columns t_1 … t_j; rows h_1, ⋮, h_i, ⋮ with entries $D_{ij} = \Sigma_k \, b_k(h_i)\, p(t_j|s_k)$ — $\infty \times \infty$

The rank of D is, at most, k

- Consider a system-dynamics matrix D of rank k

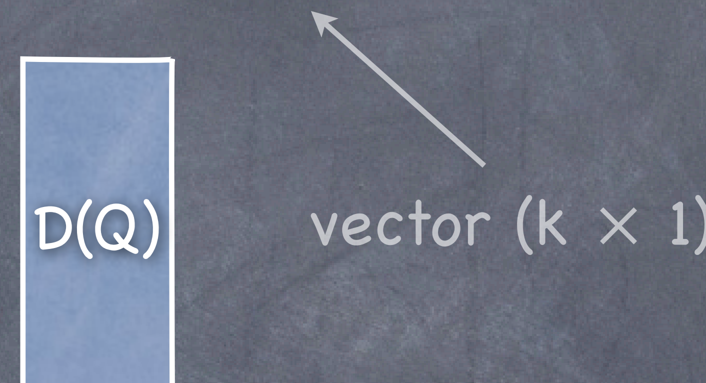


core tests (linearly independent columns of D):

$Q = \{q_1, \ldots, q_k\}$

- Thus, for any test t, one can write

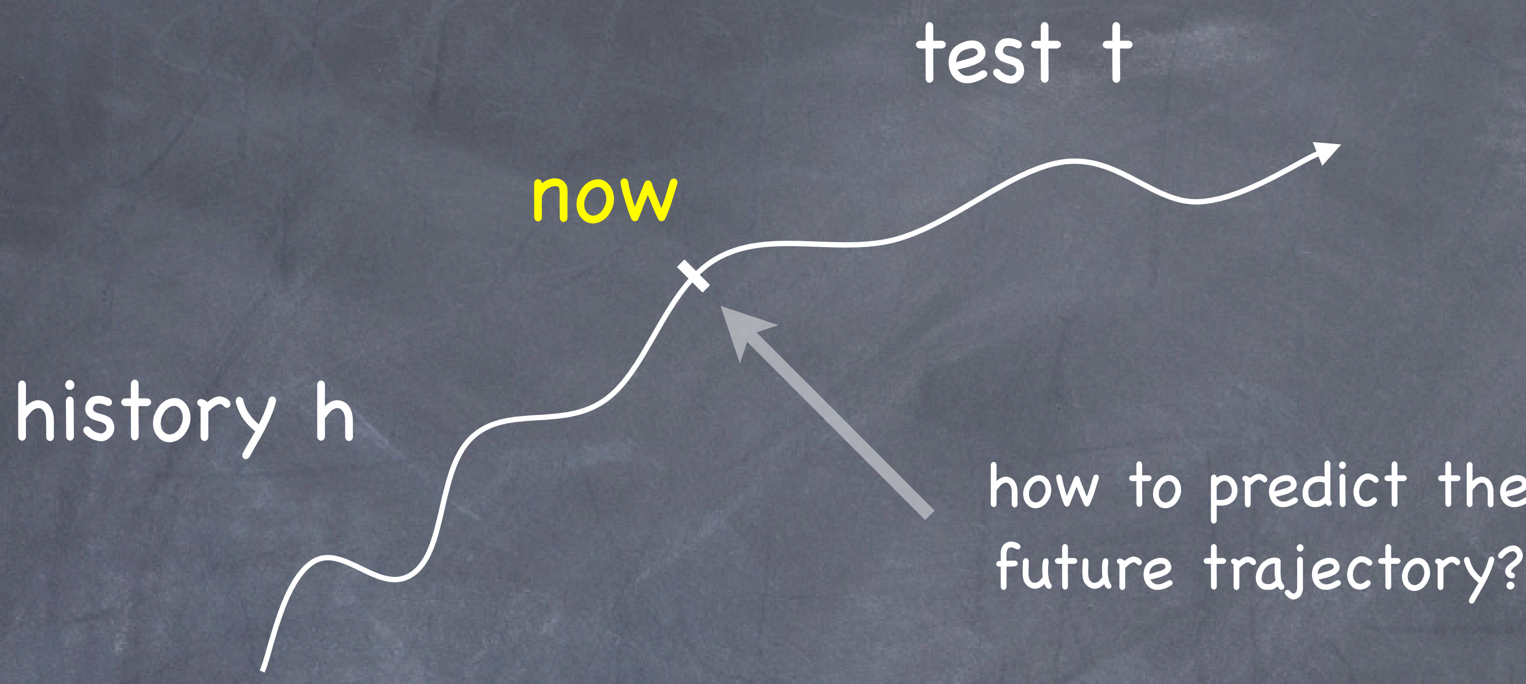
$$D(t) = D(Q)\, m_t$$



- In particular, $p(t|h) = p(Q|h)\, m_t$

this defines a Linear PSR

p(Q|h) substitutes the concept of state



- $p(Q|h)$ encodes all information about h
- $p(t|h) = f_t(\ p(Q|h)\ )$, f_t independent of h
- $p(Q|h)$ is a <u>sufficient statistic</u> of h for $p(t|h)$
- $p(Q|h)$ is a <u>P</u>redictive <u>S</u>tate <u>R</u>epresentation

- To compute any element of the system-dynamic D

$$D_{ij} = p(\ t_j \mid h_i\ )$$

- we can use

$$p(t|h) = p(Q|h)\ m_t$$

- and therefore we just need $P(Q|h)$ and m_t

- $p(Q|h)$ can be computed using:

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

this can be combined into

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

and used iteratively from $p(Q|\phi)$

- m_t can be computed, for a given test

$$t = a^1 o^1 \cdots a^n o^n$$

using this expression:

$$m_t = M_{a^1 o^1} \cdots M_{a^{n-1} o^{n-1}} \; m_{a^n o^n}$$

- Model parameters of a PSR:

$$\{m_{aoq}\}_{a\in\mathcal{A},o\in\mathcal{O},q\in\mathcal{Q}}$$

M_{ao} →

$$\{m_{ao}\}_{a\in\mathcal{A},o\in\mathcal{O}}$$

$$p(Q|\phi)$$

Non-linear PSR

- a smaller amount of tests $N=\{n_1, \ldots, n_c\}$

  such that $p(t|h) = f_t(p(N|h))$

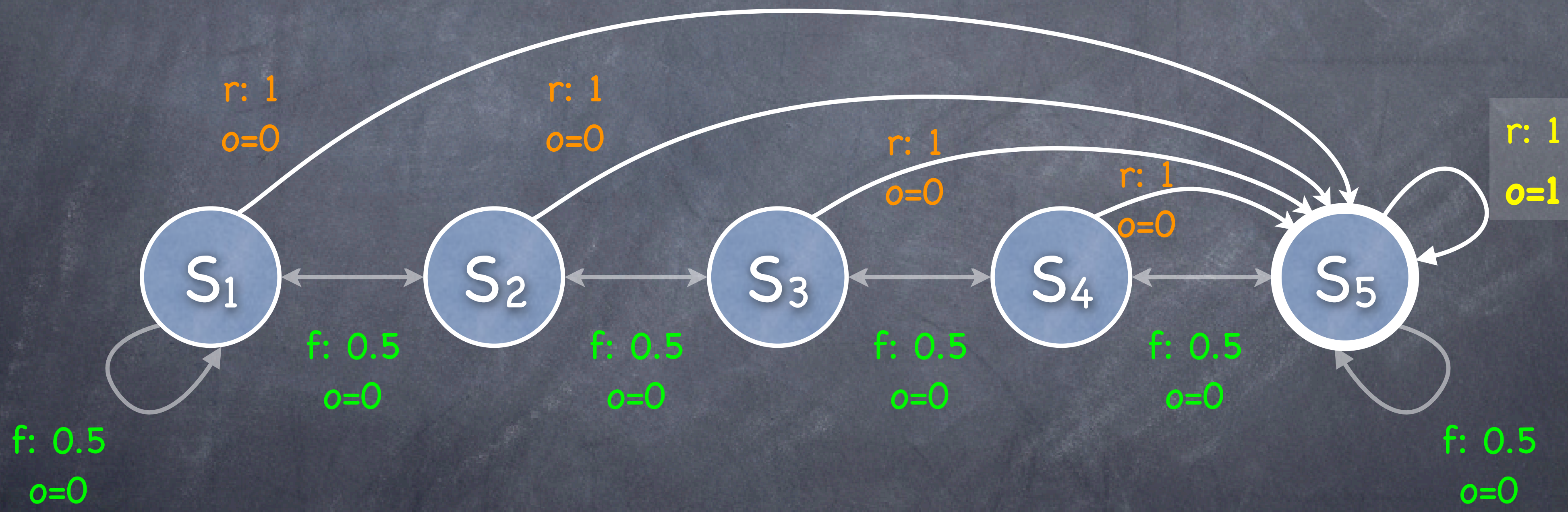
  for some nonlinear function f_t

- the update equation becomes

$$p(n_i|hao) = \frac{f_{aon_i}(p(N|h))}{f_{ao}(p(N|h))}$$

An example...

- The float-reset problem (POMDP)



linear PSR: 5 core tests

non-linear PSR: 2 core tests

- linear PSR: 5 core tests:

$$P(q_i|h) = \sum_s P(q_i|s)P(s|h)$$

r1
f0 r1
f0 f0 r1
f0 f0 f0 r1
f0 f0 f0 f0 r1

s	P(r1|s)	P(f0r1|s)	P(f0f0r1|s)	P(f0f0f0r1|s)	P(f0f0f0f0r1|s)
S_5	1	0.5	0.5	0.375	0.375
S_4	0	0.5	0.25	0.375	0.25
S_3	0	0	0.25	0.125	0.25
S_2	0	0	0	0.125	0.0625
S_1	0	0	0	0	0.0625

- nonlinear PSR: 2 core tests are enough

r1
f0 r1

$$p(t|h) = f_t(\ p(N|h)\ )$$

instead of

$$p(t|h) = p(Q|h)\ m_t$$

Progresses: learning

- Learning a PSR model implies:
  - discovery of the core tests
    e.g., incremental construction of a prediction matrix $P(Q_T|Q_H)$
    [Q_T = core tests; Q_H = core histories]
  - learning the model parameters
    e.g., using $m_t = P^{-1}(Q_T|Q_H)\, P(t|Q_H)$
    for $t \in \{ ao, aoq_1, \ldots, aoq_k \}$

Progresses: planning

- Reward as an additional observarion

  $t = a^1(r^1o^1) \ldots a^k(r^ko^k)$

  - example 1: extending POMDP-IP (based on value iteration)
  - example 2: extending Q-learning with function approximation, and P(Q|h) as state representation

Progresses: continuous observations

- Autoregressive (AR):
  observation depends on n previous ones
- Kalman filter (KF):
  hidden state vector
- Predictive Linear-Gaussian (PLG):
  - distribution of the n next observations is a sufficient statistic of the history
  (+) observation noise may be correlated
  (+) less parameters, more accurate than KF

Pointers

- http://www.eecs.umich.edu/~baveja/PSRmainpage.html
- http://scholar.google.com/   :)

People

- Satinder Singh
- Michael Littman
- Sebastian Thrun
- Peter Stone
- Michael Bowling
- … among others

Q&A