European Workshop on Reinforcement Learning 2013

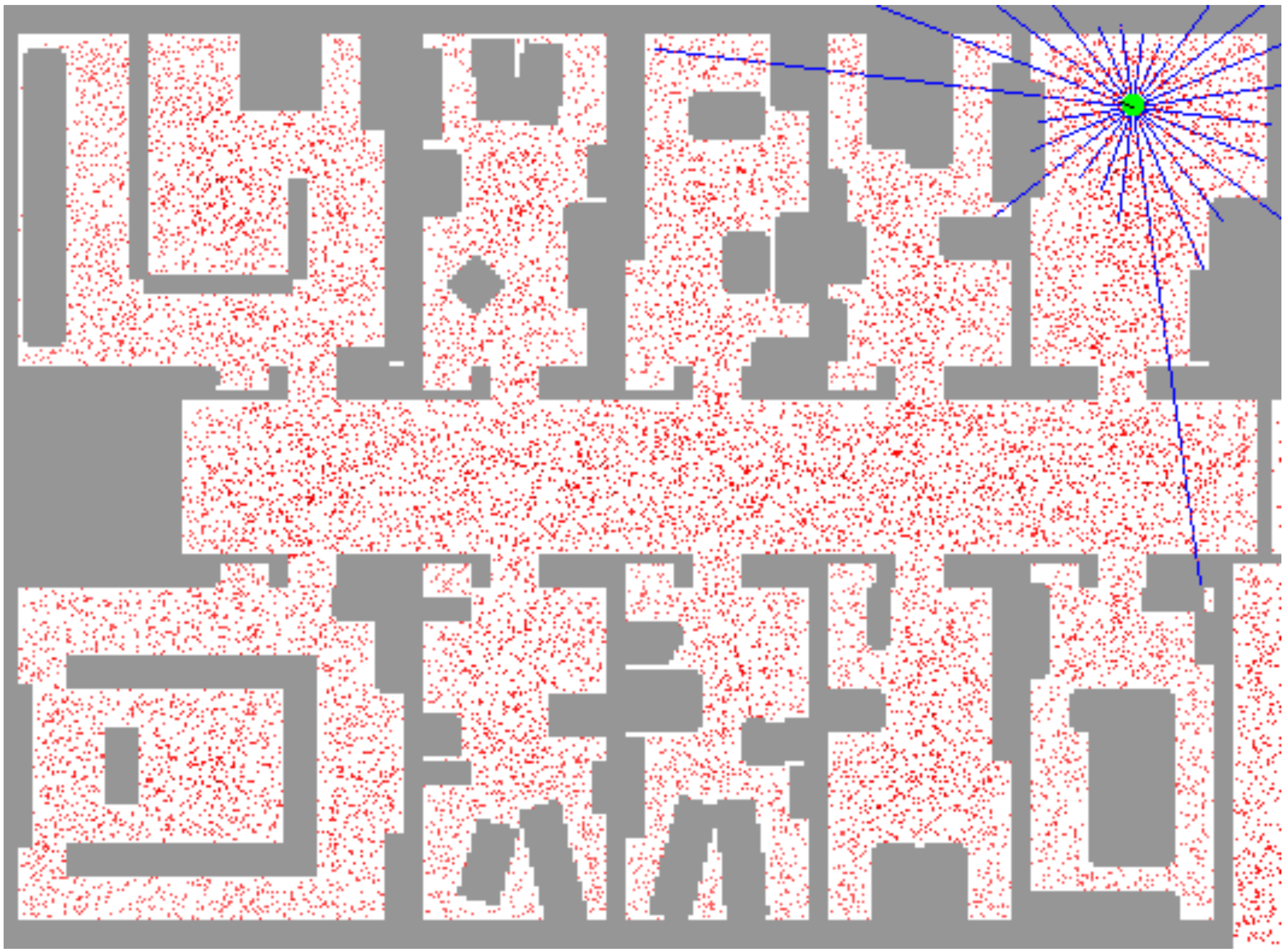
A POMDP Tutorial

Joelle Pineau
McGill University

(With many slides & pictures from Mauricio Araya-Lopez and others.)

August 2013

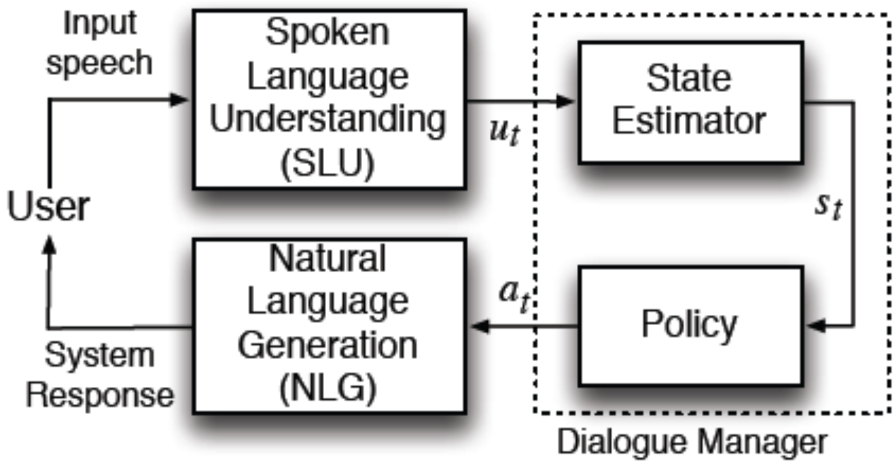
Sequential decision-making under uncertainty



http://www.cs.washington.edu/ai/Mobile_Robotics/mcl/animations/global-floor.gif

Sequential decision-making under uncertainty

http://mi.eng.cam.ac.uk/~sjy/papers/ygtw13.pdf

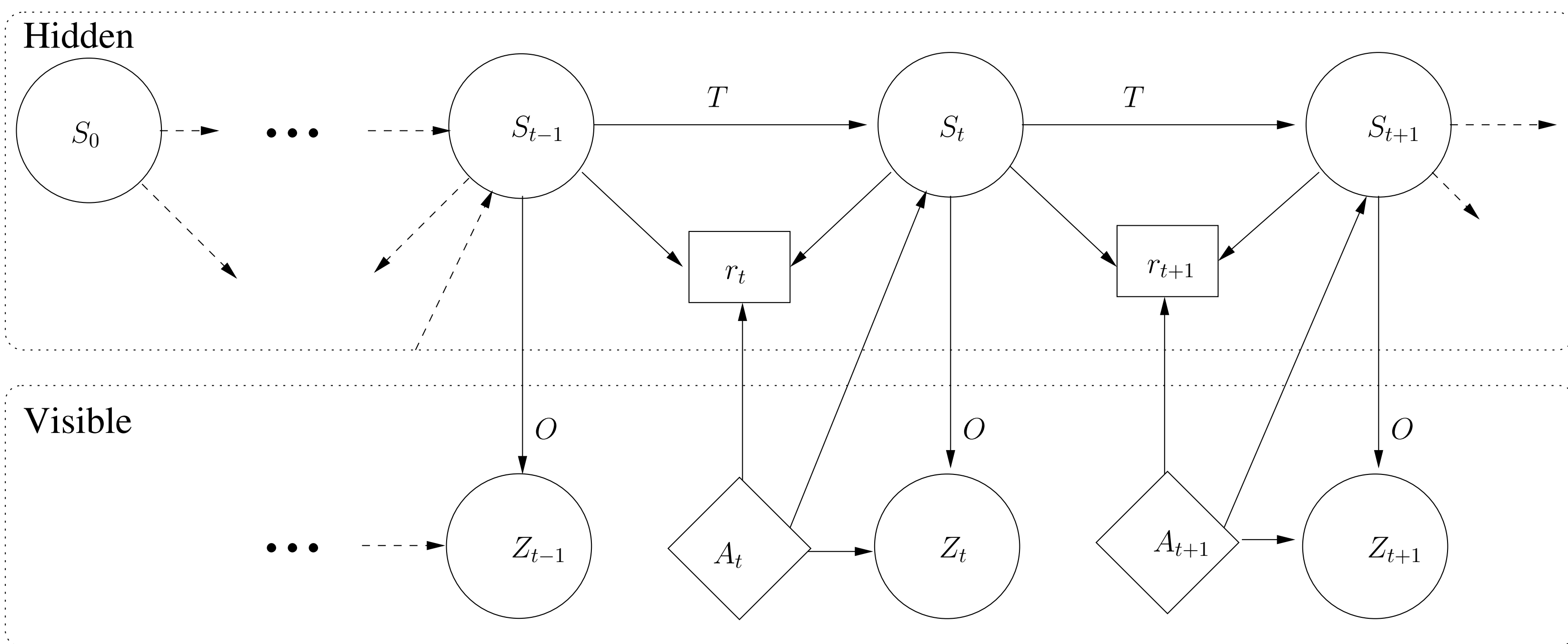


Observation	True State	Belief Entropy	Action	Reward
flo hello	request_begun	0.406	say_hello	100
flo what is like	start_meds	2.735	ask_repeat	-100
flo what time is it for will the	want_time	0.490	say_time	100
flo was on abc	want_tv	1.176	ask_which_station	-1
flo was on abc	want_abc	0.886	say_abc	100
flo what is on nbc	want_nbc	1.375	confirm_channel_nbc	-1
flo yes	want_nbc	0.062	say_nbc	100
flo go to the that pretty good what	send_robot	0.864	ask_robot_where	-1
flo that that hello be	send_robot_bedroom	1.839	confirm_robot_place	-1
flo the bedroom any i	send_robot_bedroom	0.194	go_to_bedroom	100
flo go it eight a hello	send_robot	1.110	ask_robot_where	-1
flo the kitchen hello	send_robot_kitchen	1.184	go_to_kitchen	100

http://www.cs.cmu.edu/nursebot

Partially Observable MDP

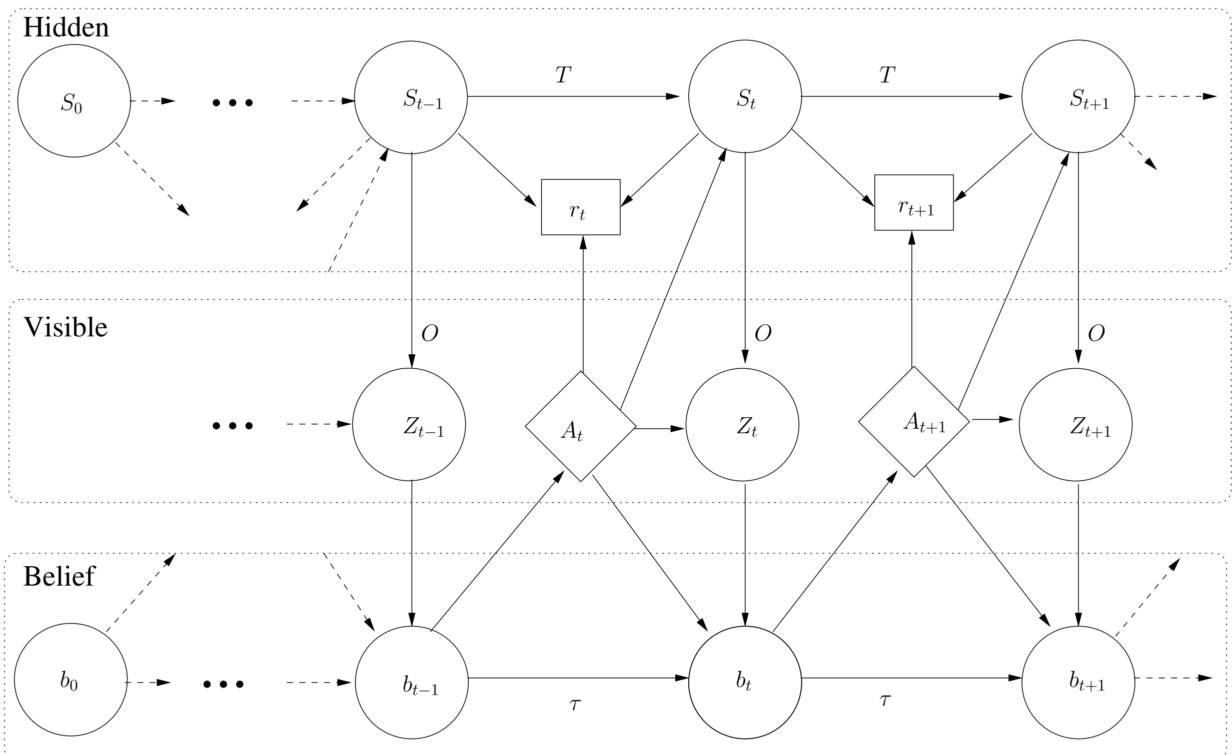
- POMDP defined by n-tuple *<S, A, Z, T, O, R>*,
  where *<S, A, T, R>* are same as in an MDP.
- States are hidden ➔ Observations (*Z*)
- Observation function $O(s,a,z) := Pr(z \mid s, a)$



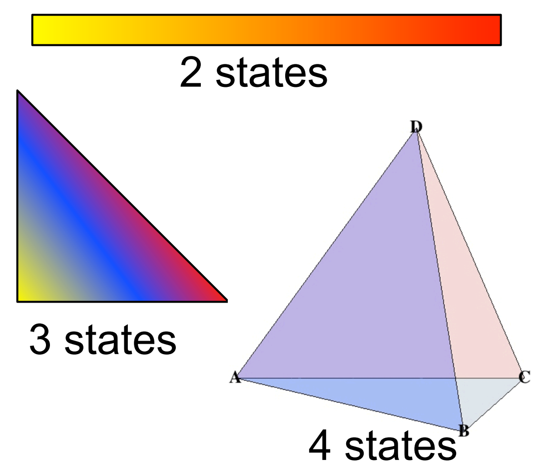
Belief-MDP (Astrom, 1965)

- Belief-state, b_t : Probability distribution over states, is a sufficient statistic of history $\{a_0, z_0, \ldots\ a_t, z_t\}$.

Karl Åström

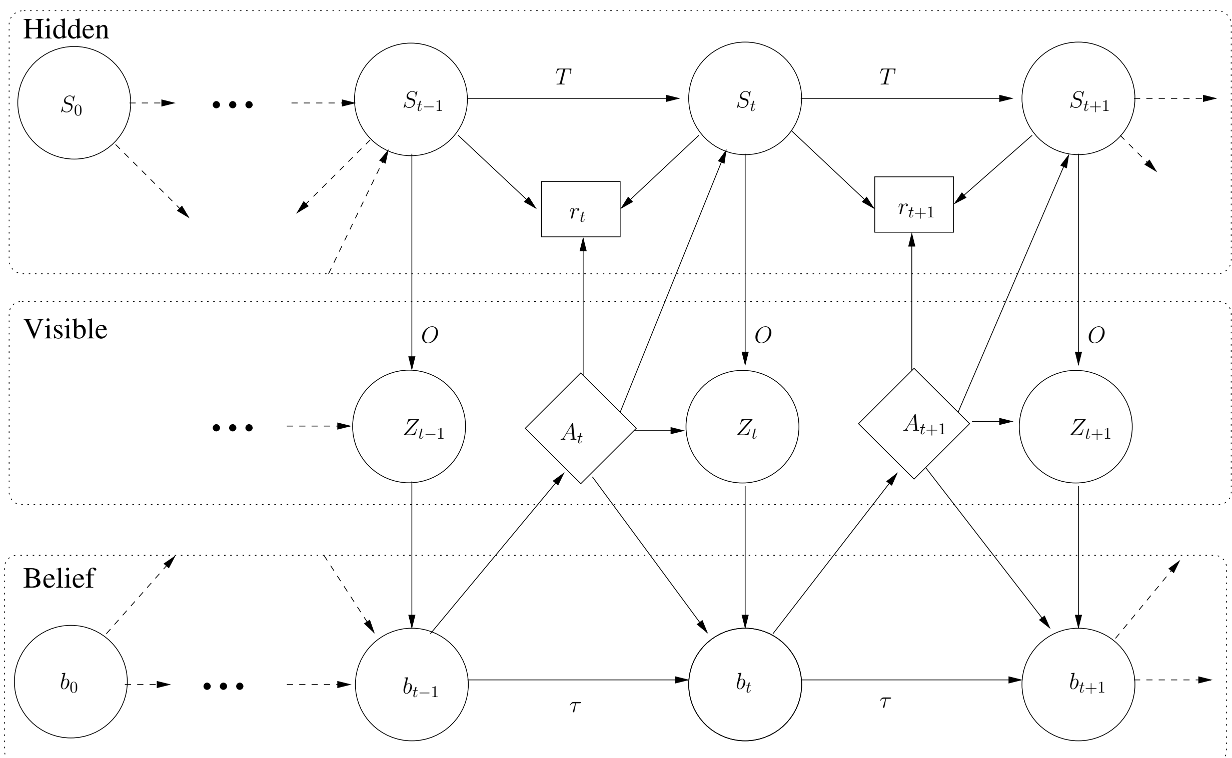


The belief simplex



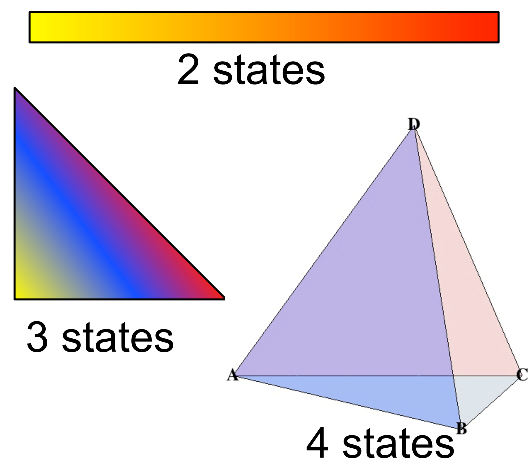
Belief-MDP (Astrom, 1965)

- Belief MDP: $\langle S, A, Z, T, O, R \rangle \rightarrow \langle \Delta, A, \tau, \rho, b_0 \rangle$
- Transition function: $\tau(b, a, b')$
- Expected reward: $\rho(b,a) = \sum_{s \in S} b(s) R(s,a)$



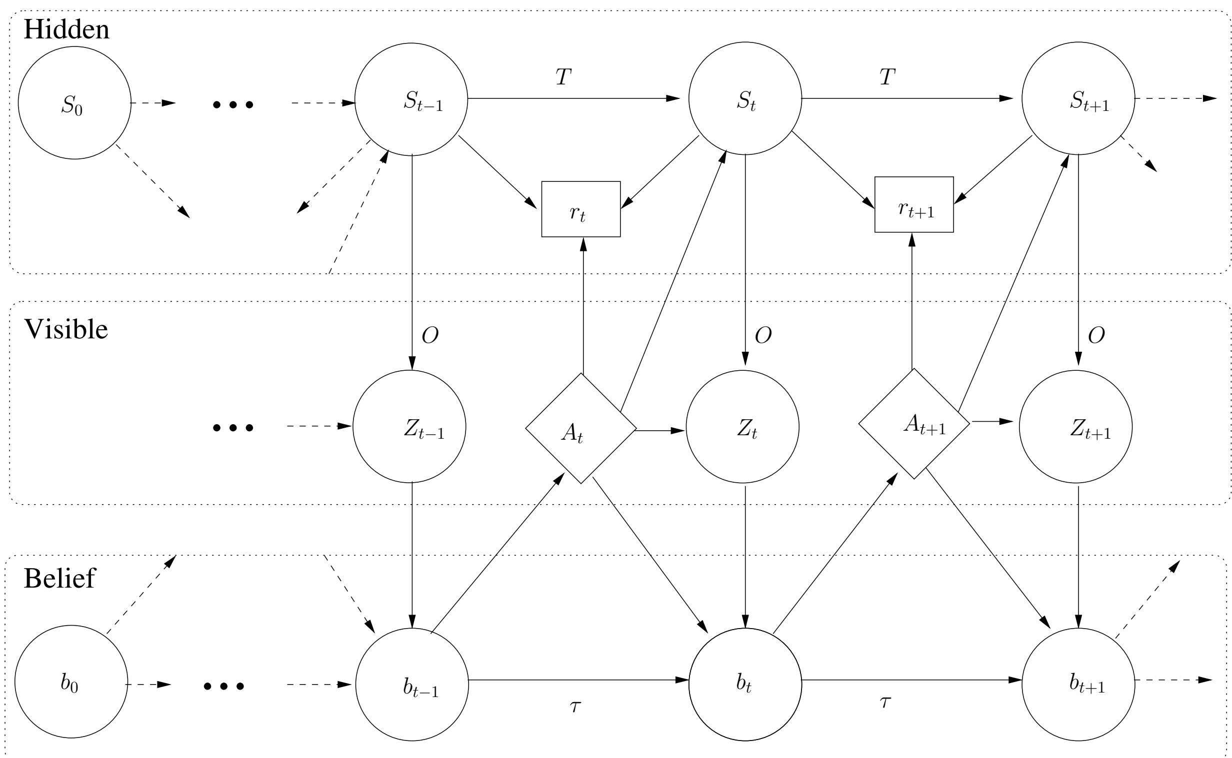
Karl Åström

The belief simplex



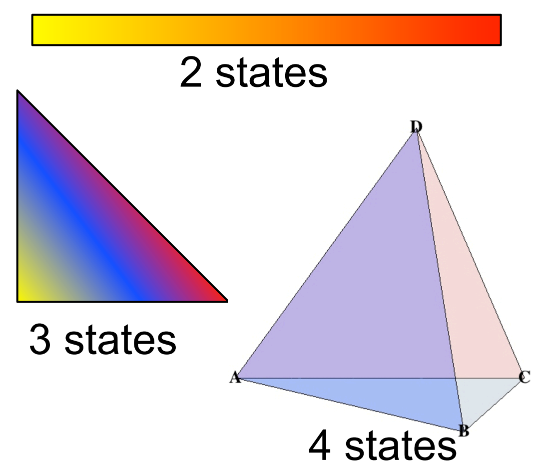
Belief-MDP (Astrom, 1965)

- Belief update: Bayes Rule!
- Value fn: Bellman's equation!
- Policy: $\pi : b \longrightarrow a$



Karl Åström

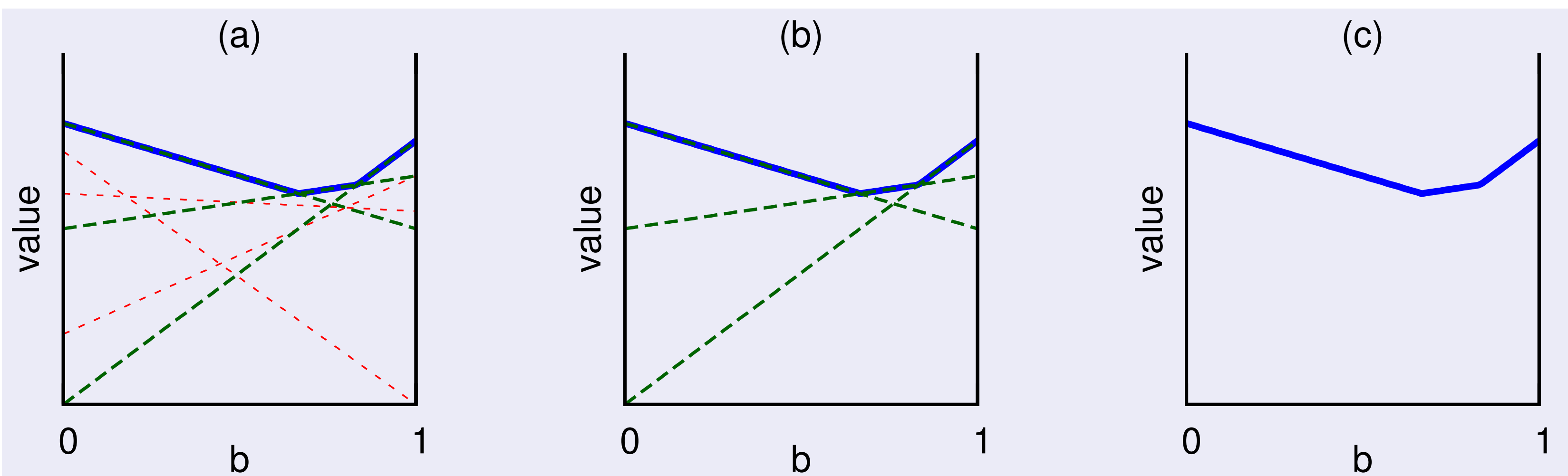
The belief simplex



Early POMDP solution methods

- Observe: Reward function is linear $\rho(b,a) = \sum_{s \in S} b(s) R(s,a)$
- $V_t^*(b)$ is therefore piecewise linear and convex.
- Set of hyper-planes, named α-vectors, represent value function $V(b) = \max_{\alpha \in \Gamma} \alpha \cdot b$

Edward Sondik



Early POMDP solution methods

- Observe: Reward function is linear $\rho(b,a) = \sum_{s \in S} b(s) R(s,a)$
- $V_t^*(b)$ is therefore piecewise linear and convex.
- Set of hyper-planes, named α-vectors, represent value function $V(b) = \max_{\alpha \in \Gamma} \alpha \cdot b$

Edward Sondik

Exact Solution Methods

Propagate, combine and prune hyperplanes using the Bellman Equation

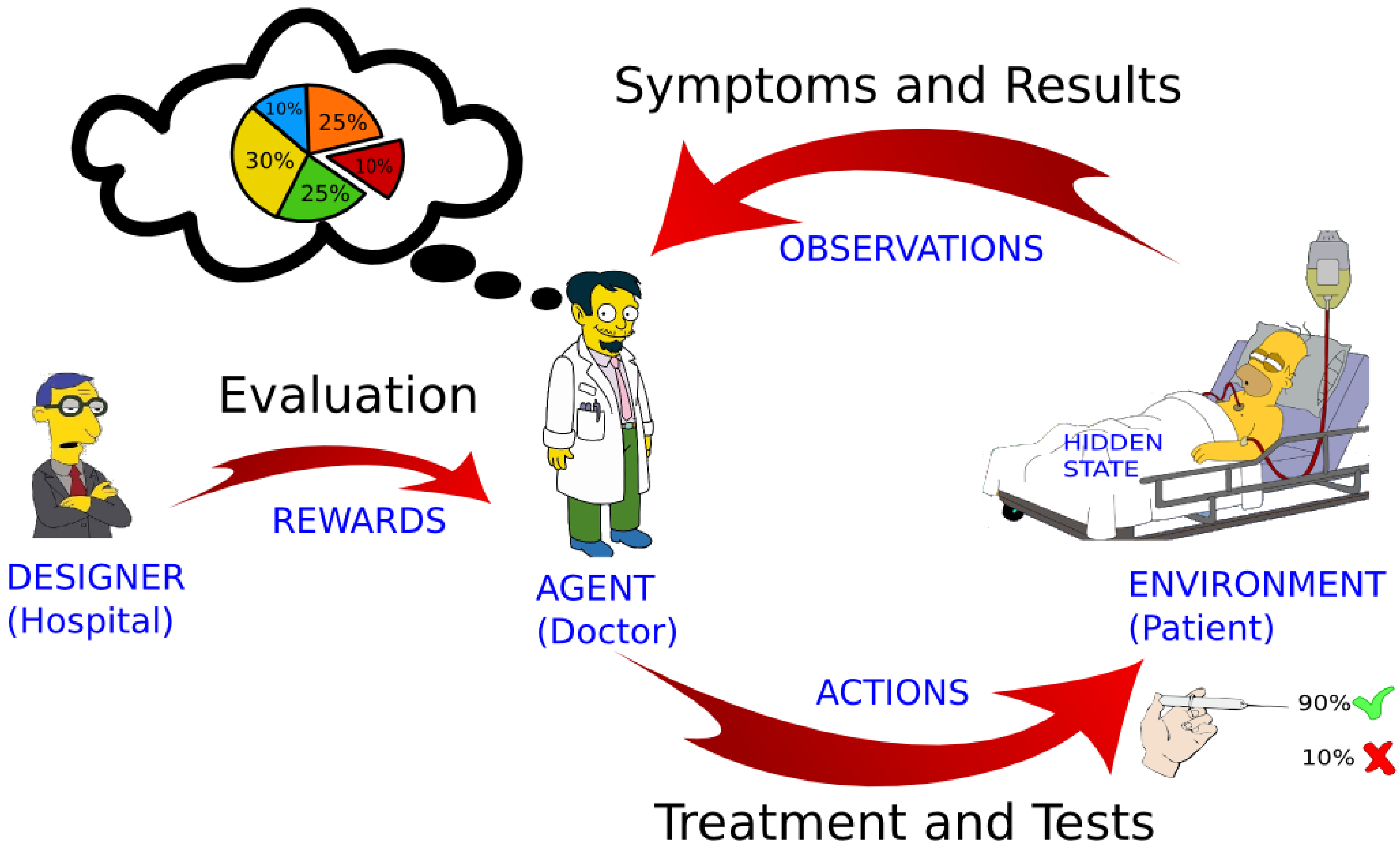
(Monahan, 1982)
Batch Enumeration

(Littman, 1996)
Witness Algorithm

(Cassandra, 1998)
Incremental Pruning

Problem: Exact solving POMDPs is PSPACE-Complete

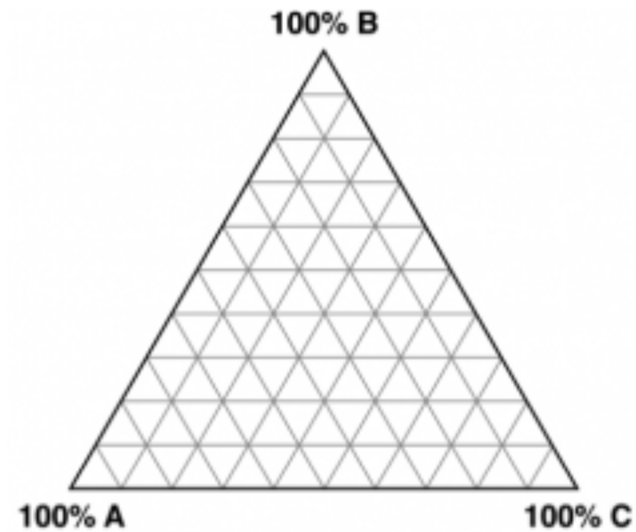
Solving small information-gathering problems



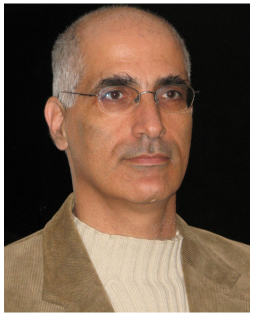
From Mauricio Araya-Lopez, JFPDA 2013.

Approximate belief discretization methods

- Apply Bellman over a discretization of the belief space.
- Various methods to select discretization (regular, adaptive).
- Polynomial-time approximate algorithm.
- Bounded error that depends on the grid resolution.



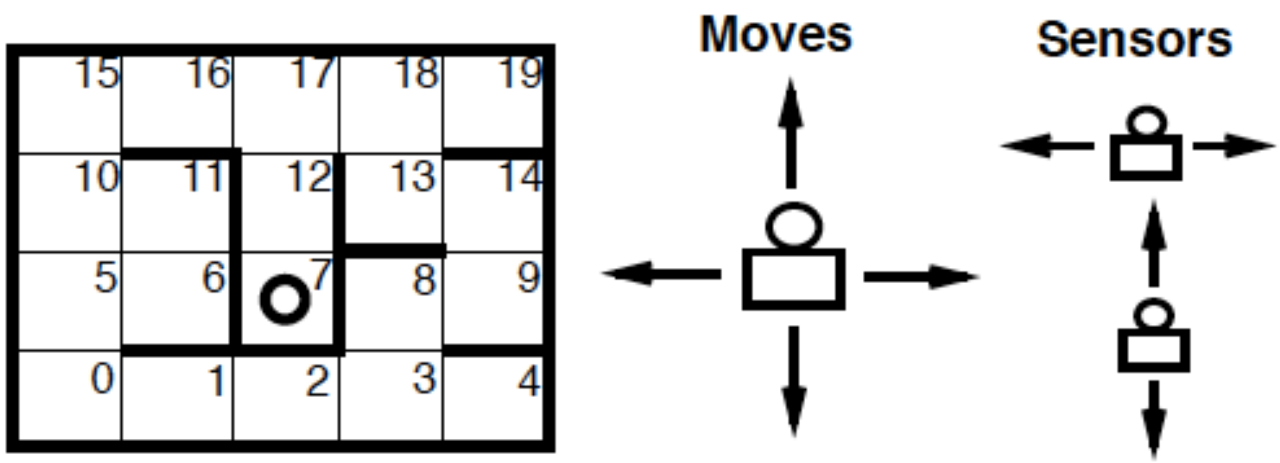
William Lovejoy



Ronen Brafman

Milos Hauskrecht

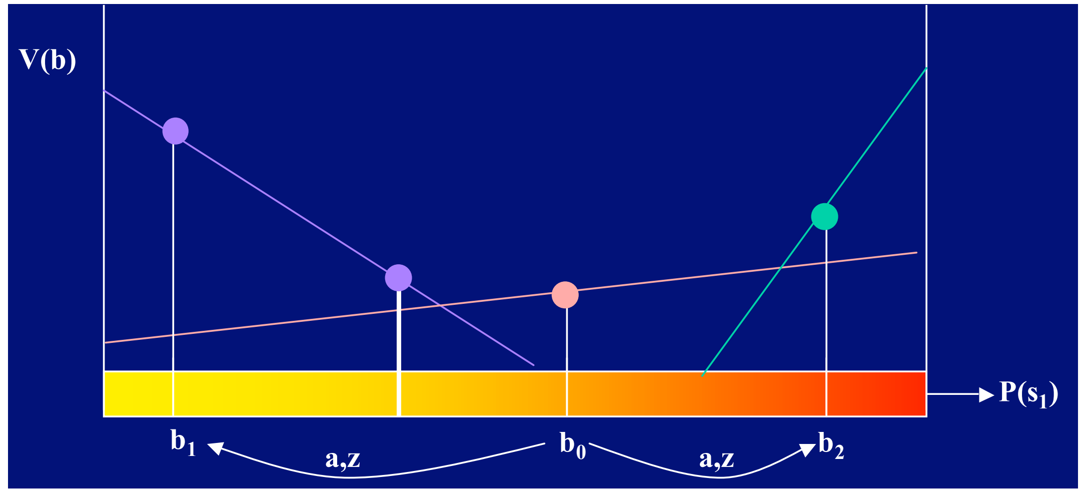
Gridworld POMDPs – Solved!



http://people.cs.pitt.edu/~milos/research/JAIR-2000.pdf

Point-based value iteration (Pineau et al., 2003)

- Select a small set of reachable belief points.
- Perform Bellman updates at those points, keeping value & gradient.
- Anytime and polynomial algorithm.
- Bounded error depends on density of points.



Other Point-based Algorithms

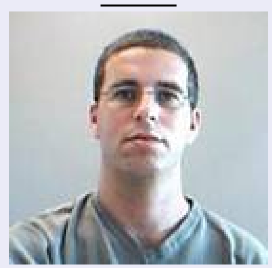
PERSEUS

(Spaan and Vlassis, 2005)

HSVI

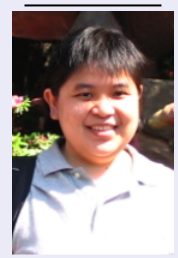
(Smith and Simmons, 2005)

FSVI



(Shani et al., 2007)

SARSOP

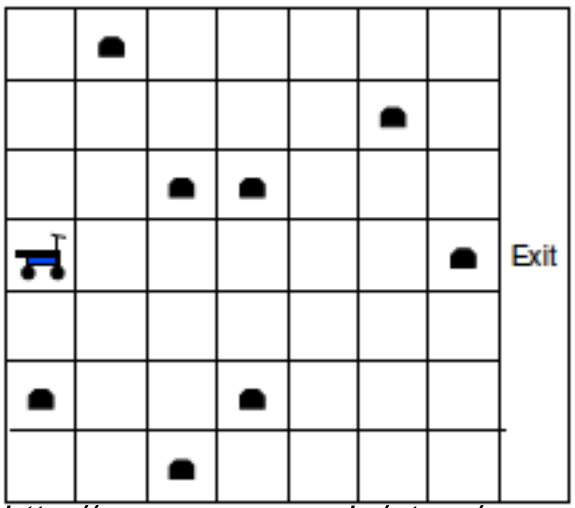


(Kurniawati et al., 2008)

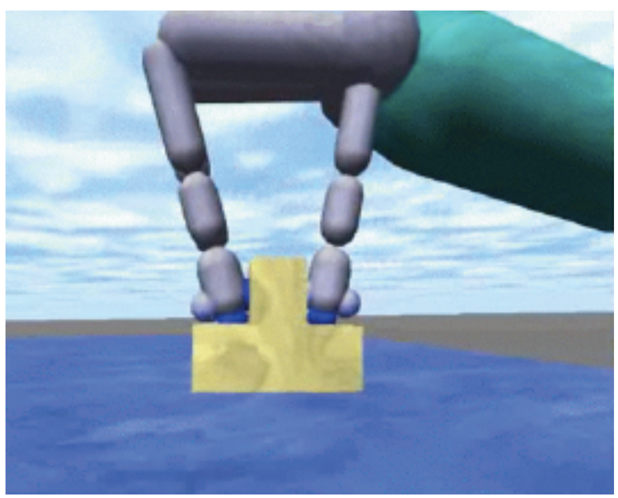
GapMin

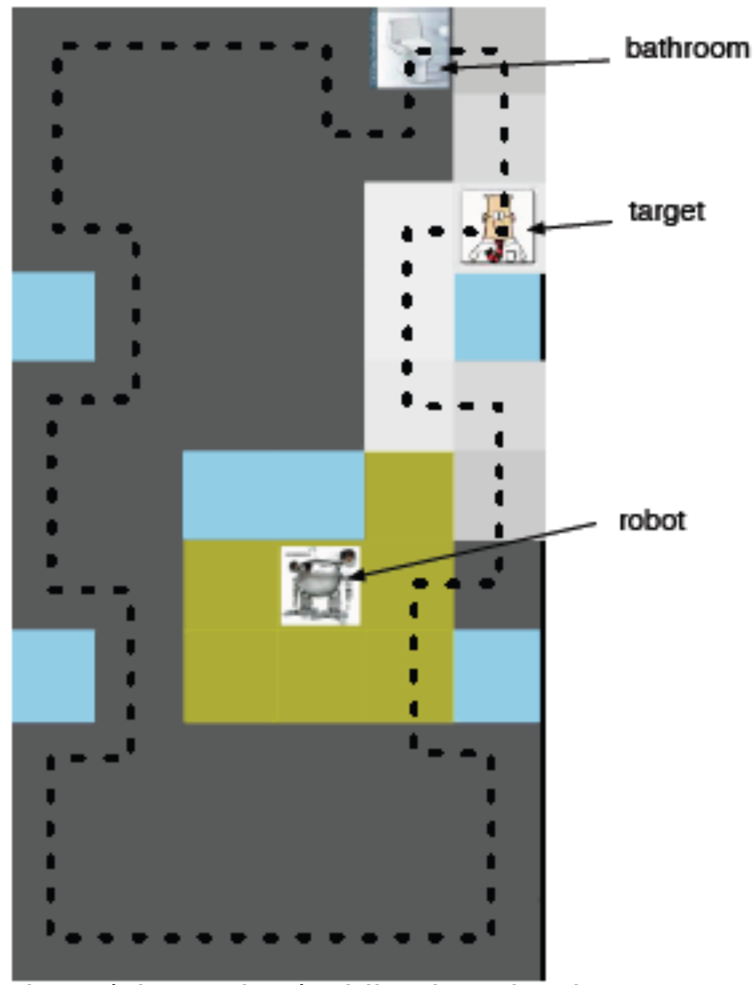
(Poupart et al., 2011)

POMDPs with thousands of states – SOLVED!



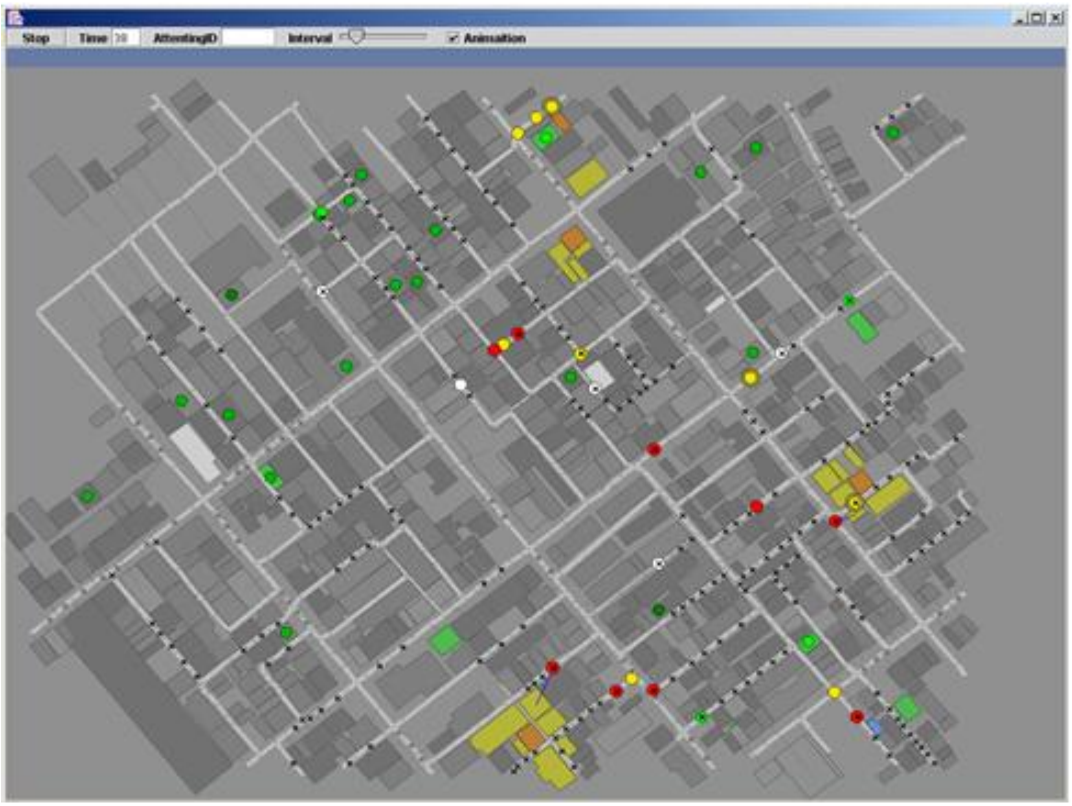
http://www.cs.cmu.edu/~trey/papers/smith04_hsvi.pdf





http://bigbird.comp.nus.edu.sg/~hannakur/publications.html

Harder problem: Robocup Rescue Simulation



From Sébastien Paquet et al., 2005

Highly dynamic environment.

Approx. 30 agents of 6 types

Partially observable state.

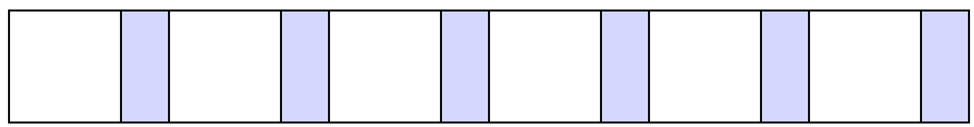
Real-time constraints on agents' response time.

Agents face unknown instances of the environment.

Online planning

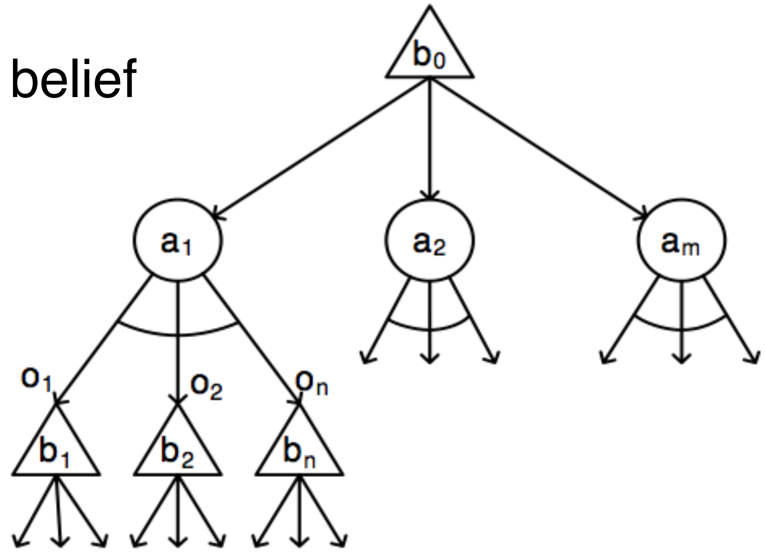
Offline:

Online:



Online search for POMDP solutions

Build an AND/OR tree of the reachable belief states, from the current belief ***b_0*** :



Approaches:

Branch-and-bound **Heuristic search** **Monte-Carlo Tree Search**

(Paquet, Tobin & **Chaib-draa**, 2005)

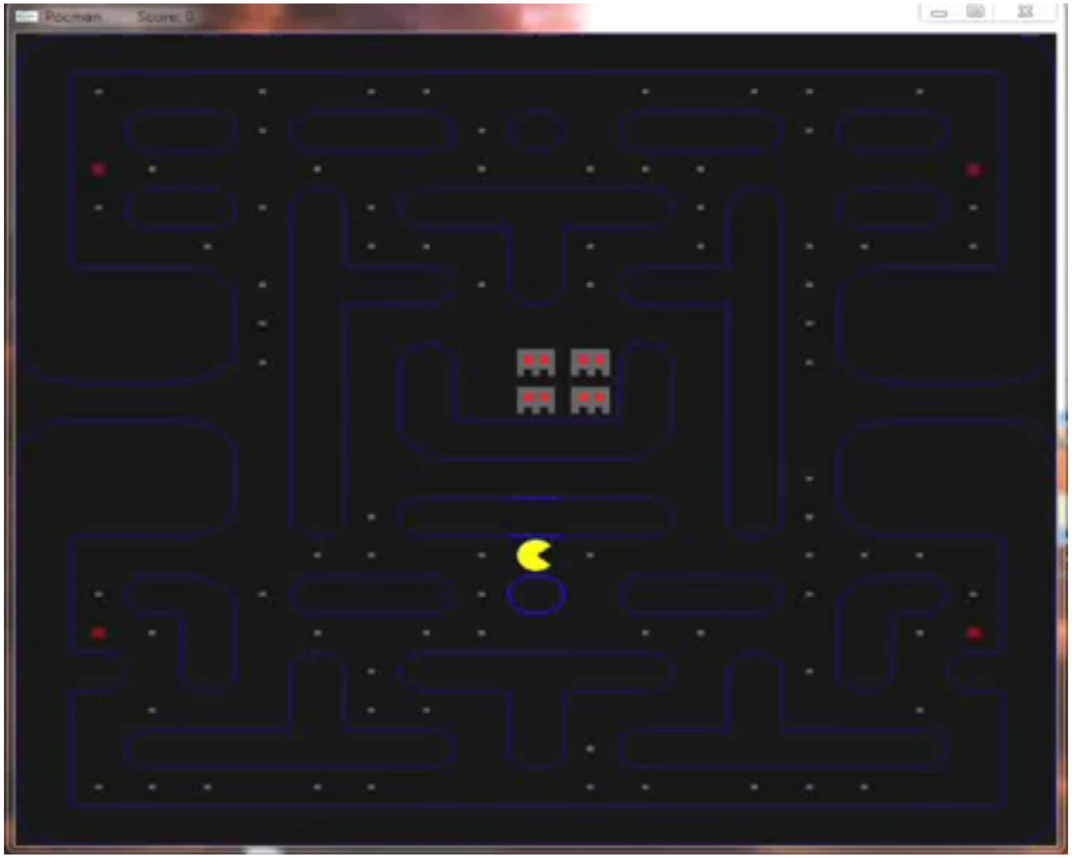
(**Ross**, Pineau, Paquet, Chaib-draa, 2008)


(**McAllester** & **Singh**, 1999)


(**Silver** & **Veness**, 2010)

Are we done yet?

- Pocman problem: $|S|=10^{56}$, $|A|=4$, $|Z|=1024$.



http://www0.cs.ucl.ac.uk/staff/D.Silver/web/Applications.html

There is no simulator for this!

http://www.tech.plym.ac.uk/SoCCE/CRNS/staff/fbroz/

http://web.mit.edu/nickroy/www/research.html

Learning POMDPs from data

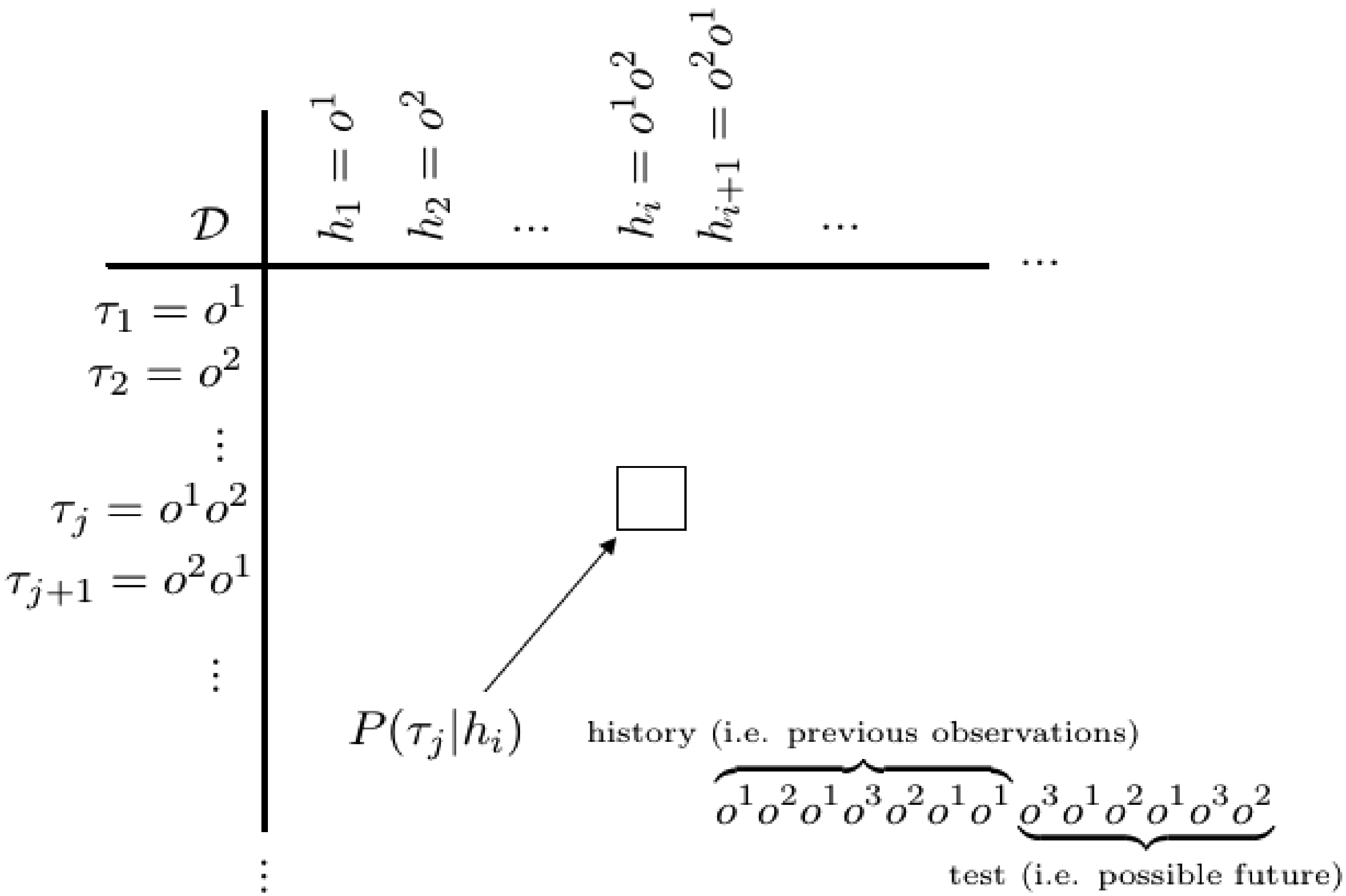
- Expectation-maximization
  - » Online nested EM (Liu, Liao & Carin, 2013)
  - » Model-free RL as mixture learning (Vlassis & Toussaint, 2009)
- History-based methods
  - » U-Tree (McCallum, 1996)
  - » MC-AIXI (Veness, Ng, Hutter, Uther & Silver, 2011)
- Predictive state representations
  - » PSRs (Littman, Sutton, Singh, 2002)
  - » TPSRs (Boots & Gordon, 2010)
  - » Compressed PSRs (Hamilton, Fard & Pineau, 2013)
- Bayesian learning
  - » Bayes-Adaptive POMDP (Ross, Chaib-draa & Pineau, 2007)
  - » Infinite POMDP (Doshi-Velez, 2009)

Compressed Predictive State Representations (CPSRs)

Goal: Efficiently learn a model of a dynamical system using time-series data, when you have:

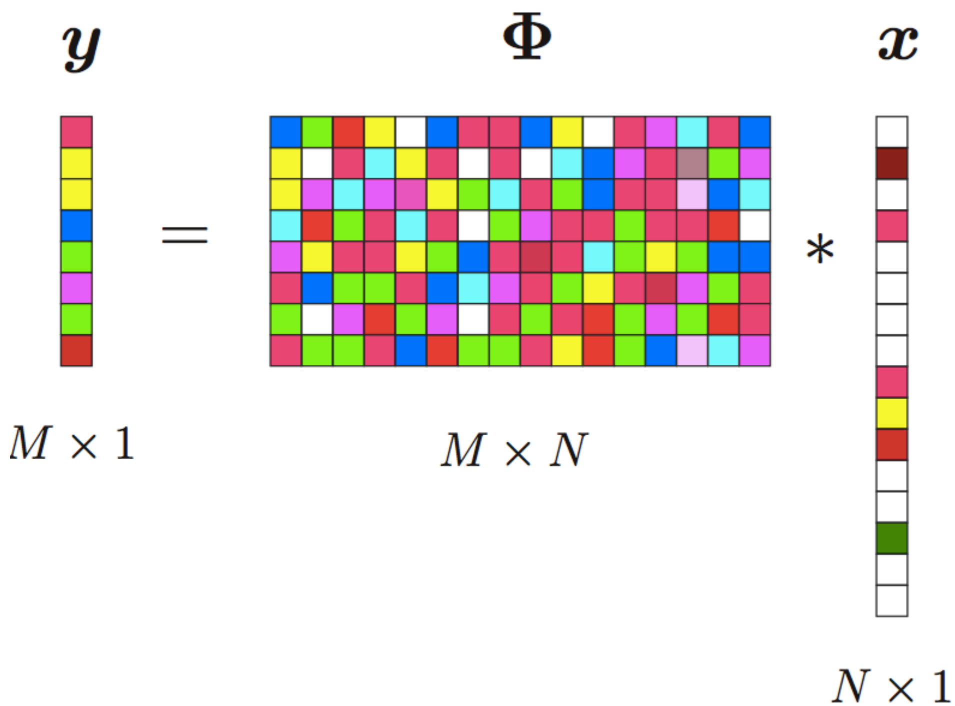
- large discrete observation spaces;
- partial observability;
- sparsity.

The PSR systems dynamics matrix



Sparsity

- Assume that only a subset of tests is possible given any history h_i.
- Sparse structure can be exploited using **random projections.**



CPSR Algorithm

Algorithm

- Obtain compressed estimates for sub-matrices of $\mathcal{D}$, $\Phi\mathcal{P}_{\mathcal{T},\mathcal{H}}$, $\Phi\mathcal{P}_{\mathcal{T},o^l,\mathcal{H}}$s, and $\mathcal{P}_{\mathcal{H}}$ by sampling time series data.
  - Estimate $\Phi\mathcal{P}_{\mathcal{T},\mathcal{H}}$ in compressed space by adding ϕ_i to column j each time t_i observed after h_i (Likewise for $\Phi\mathcal{P}_{\mathcal{T},o^l,\mathcal{H}}$s).
- Compute CPSR model:
  - $\mathbf{c}_0 = \Phi\hat{\mathcal{P}}(\tau|\emptyset)$
  - $\mathbf{C}_o = \Phi\mathcal{P}_{\mathcal{T},o^l,\mathcal{H}}(\Phi\mathcal{P}_{\mathcal{T},\mathcal{H}})^+$
  - $\mathbf{c}_\infty = (\Phi\mathcal{P}_{\mathcal{T},\mathcal{H}})^+\hat{\mathcal{P}}_H$

State definition and necessary equations

- $\mathbf{c}_0$ serves as initial prediction vector (i.e. state vector).
- Update state vector after seeing observation with
  - $\mathbf{c}_{t+1} = \frac{\mathbf{C}_o\mathbf{C}_t}{\mathbf{C}_\infty\mathbf{C}_o\mathbf{c}_t}$
- Predict k-steps into the future using
  - $P(o^j_{t+k}|h_t) = \mathbf{b}_\infty\mathbf{C}_{o^j}(\mathbf{C}_\star)^{k-1}\mathbf{C}_t$ where $\mathbf{C}_\star = \sum_{o^i\in\mathcal{O}}\mathbf{C}_{o_i}$.

Theory overview

Error of the CPSR parameters

With probability no less than $1-\delta$ we have:

$$\left\|\mathbf{C}_o(\Phi\mathcal{P}_{\mathcal{Q},h})-\Phi\mathcal{P}_{\mathcal{Q},o,h}\right\|_{\rho(\mathbf{x})}\leq\sqrt{d}\epsilon(|\mathcal{H}|,|\mathcal{Q}|,d,L_o,\sigma_o^2,\delta/d)$$

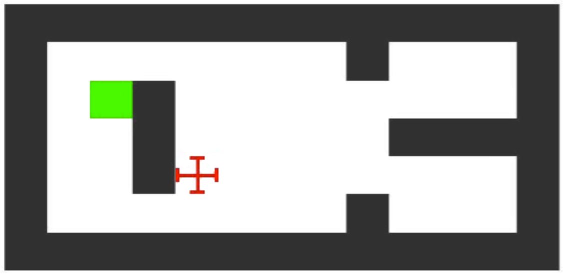
Error propagation

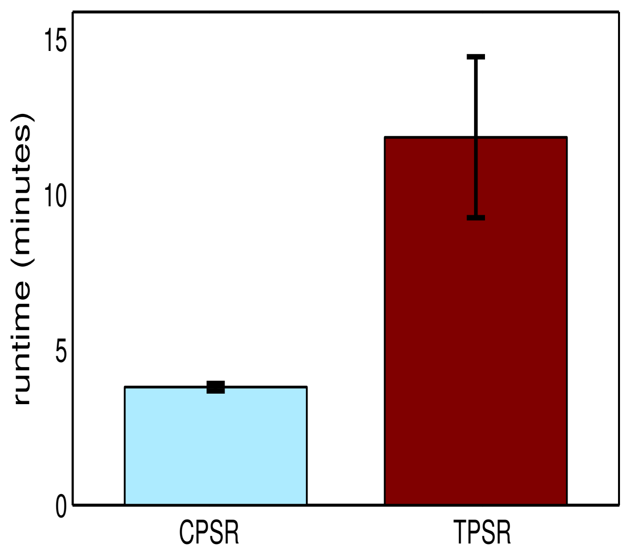
The total propagated error for T steps is bounded by $\epsilon(c^T-1)/(c-1)$.

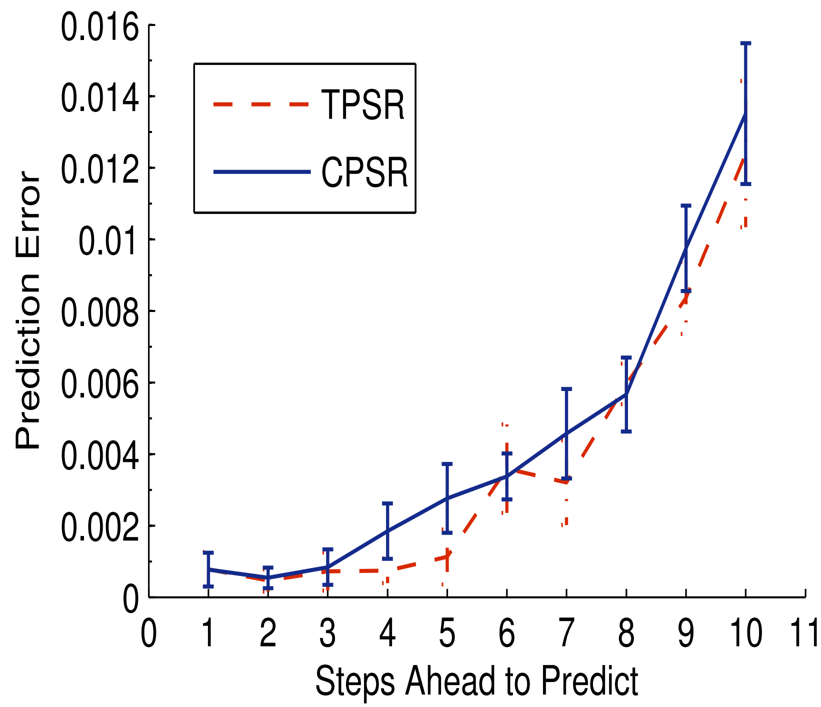
Projection size

A projection size of $d=O(k\log|Q|)$ suffices in a majority of systems.

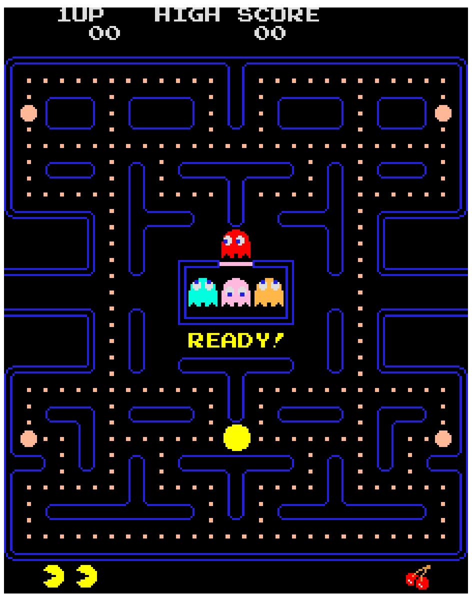
GridWorld Results

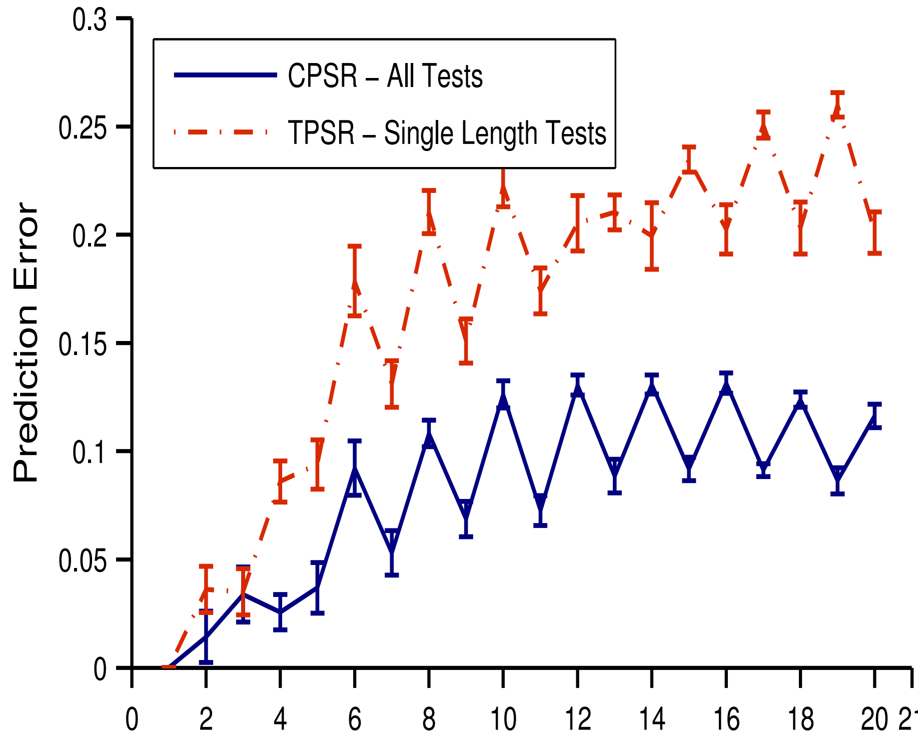






Poc-Man Results





Learning POMDPs from data

- Expectation-maximization
  - » Online nested EM (Liu, Liao & Carin, 2013)
  - » Model-free RL as mixture learning (Vlassis & Toussaint, 2009)
- History-based methods
  - » U-Tree (McCallum, 1996)
  - » MC-AIXI (Veness, Ng, Hutter, Uther & Silver, 2011)
- Predictive state representations
  - » PSRs (Littman, Sutton, Singh, 2002)
  - » TPSRs (Boots & Gordon, 2010)
  - » Compressed PSRs (Hamilton, Fard & Pineau, 2013)
- Bayesian learning
  - » Bayes-Adaptive POMDP (Ross, Chaib-draa & Pineau, 2007)
  - » Infinite POMDP (Doshi-Velez, 2009)

Bayesian learning: POMDPS

Estimate POMDP model parameters using Bayesian inference:

- ***T*:** Estimate a posterior $\phi^a_{ss'}$ on the incidence of transitions $s \rightarrow_a s'$.
- ***O*:** Estimate a posterior ψ^a_{sz} on the incidence of observations $s' \rightarrow_a z$.
- ***R*:** Assume for now this is known (straight-forward extension.)

Goal: Maximize expected return under partial observability of (s, ϕ, ψ).

<u>This is also a POMDP problem</u>:

- S' : physical state $(s \in S)$ + information state (ϕ, ψ)
- T' : describes probability of update $(s, \phi, \psi) \rightarrow_a (s', \phi', \psi')$
- O' : describes probability of observing count increment.

Bayes-Adaptive POMDPs

[Ross et al. JMLR'11]

- $S' = S \times \mathbb{N}^{|S|^2|A|} \times \mathbb{N}^{|S||A||Z|}$
- $A' = A$
- $Z' = Z$
- $Pr(s', \phi', \psi' | s, \phi, \psi, a, z) =$
$$\frac{\phi^a_{ss'}}{\sum_{s'' \in S} \phi^a_{ss''}} \frac{\psi^a_{s'z}}{\sum_{z' \in Z} \psi^a_{s'z}} I(\phi', \phi + \delta^a_{ss'}) I(\psi', \psi + \delta^a_{s'z})$$
- $R'(s, \phi, \psi, a) = R(s, a)$

Learning = Tracking the hyper-state

A solution to this problem is an optimal plan to act and learn!

Bayes-Adaptive POMDPs: Belief tracking

Assume S, A, Z are discrete. Model ϕ, ψ using Dirichlet distributions.

Initial hyper-belief: $b_0(s, \phi, \psi) = b_0(s)\, I(\phi=\phi_0)\, I(\psi=\psi_0)$

where $b_0(s)$ is the initial belief over original state space

$I(\,)$ is the indicator function

(ϕ_0, ψ_0) are the initial counts (prior on T, O)

Updating b_t defines a mixture of Dirichlets, with $O(|S|^{t+1})$ components.

In practice, approximate with a particle filter.

Bayes-Adaptive POMDPs: Belief tracking

Different ways of approximating $b_t(s, \phi, \psi)$ via particle filtering:

1. Monte-Carlo sampling (MC)

2. K most probable hyper-states (MP)

3. Risk-sensitive filtering with weighted distance metric:

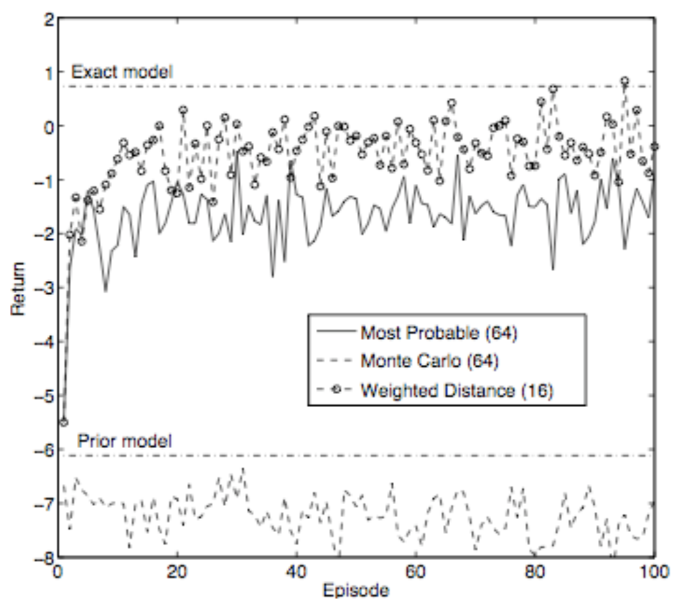
$$\sup_{\alpha \in \Gamma_t, s \in S} |V_t^\alpha(s,\phi,\psi) - V_t^\alpha(s,\phi',\psi')| \leq \frac{2\gamma||R||_\infty}{(1-\gamma)^2} \sup_{s,s' \in S, a \in A} \Bigg[D_S^{sa}(\phi,\phi') + D_Z^{s'a}(\psi,\psi') + \frac{4}{\ln(\gamma^{-e})} \left(\frac{\sum_{s'' \in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{(\mathcal{N}_\phi^{sa}+1)(\mathcal{N}_{\phi'}^{sa}+1)} + \frac{\sum_{z \in Z} |\psi^a_{s'z} - \psi'^a_{s'z}|}{(\mathcal{N}_\psi^{s'a}+1)(\mathcal{N}_{\psi'}^{s'a}+1)} \right) \Bigg]$$

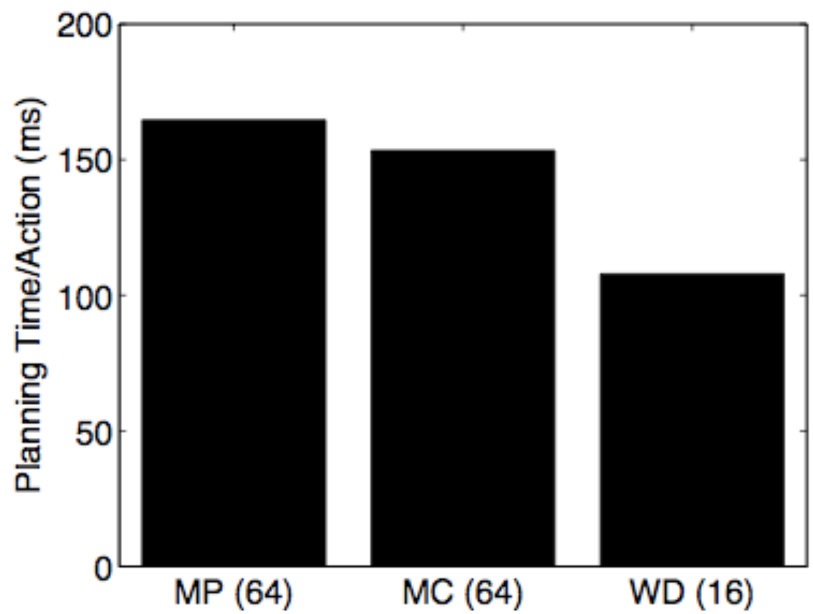
Bayes-Adaptive POMDPs: Preliminary results

Follow domain: A robot must follow one of two individuals in a 2D open area. Their identity is not observable. They have different (unknown) motion behaviors.

Learn $\phi^1 \sim \mathrm{Dir}(\alpha_1^1, \ldots, \alpha_K^1)$, $\phi^2 \sim \mathrm{Dir}(\alpha_1^2, \ldots, \alpha_K^2)$, a motion model of each person.

Bayesian POMDP results:

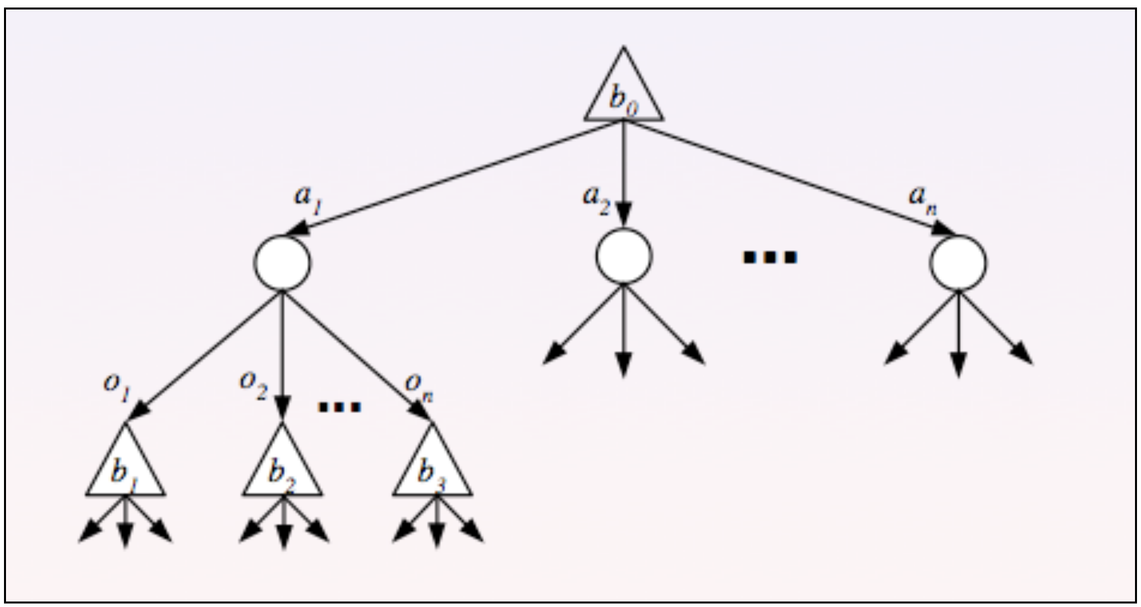




Learning is achieved, if you track the important hyper-beliefs.

Bayes-Adaptive POMDPs: Planning

- Receding horizon control to estimate the value of each action at current belief, b_t.
  - Usually consider a short horizon of reachable beliefs.
  - Use pruning and heuristics to reach longer planning horizons.

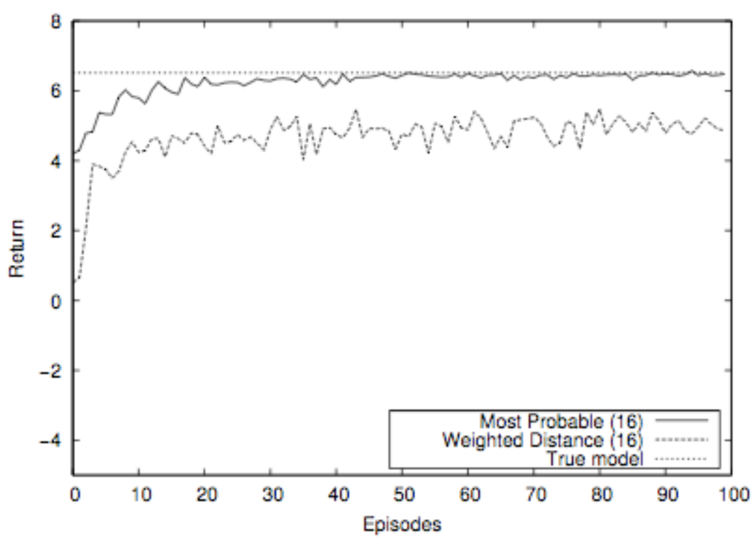


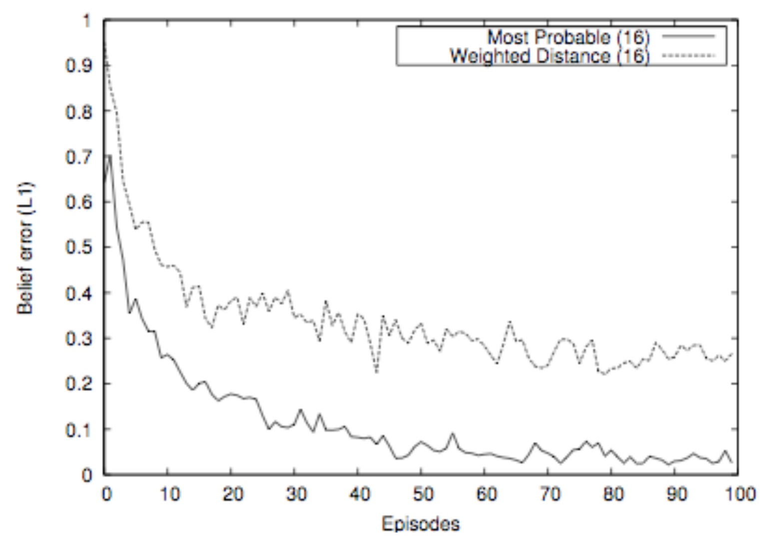
Bayes-Adaptive POMDPs: Preliminary results

RockSample domain *[Smith&Simmons, 2004]*: A robot must move around its environment in order to gather samples of “good” rocks, while avoiding “bad” rocks.

Learn $\psi \sim \mathrm{Dir}(\,\alpha_1^{\,1}, \ldots, \alpha_d^{\,1}\,)$, the accuracy of the rock quality sensor.

Results:





Again, learning is achieved, converging to the optimal solution.

In this case, *most probable* particle selection is better.

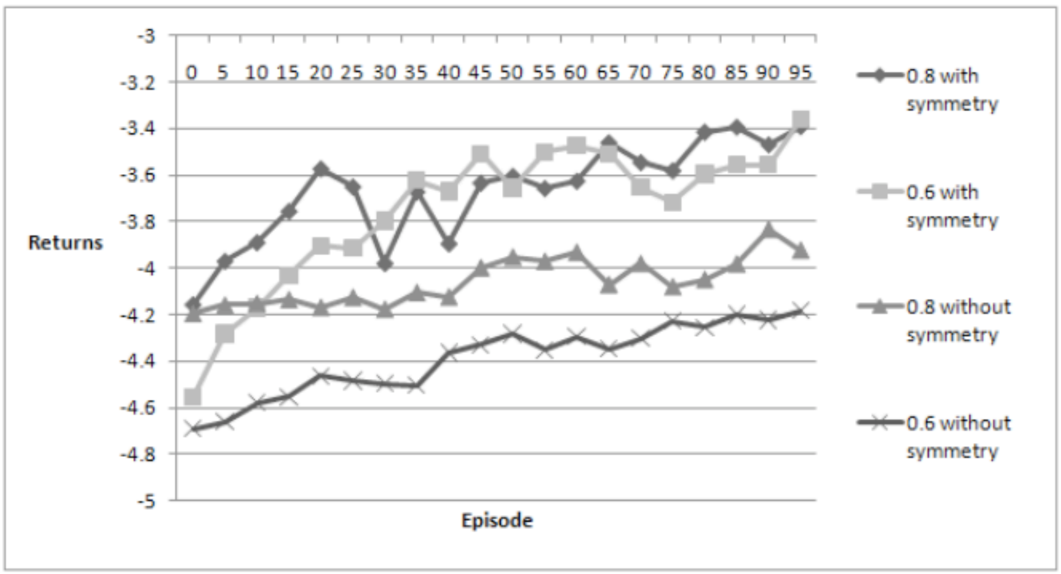
Case study #1: Dialogue management

[Png & Pineau ICASSP'11]

Estimate *O(s,a,z)* using Bayes-adaptive POMDP.

– Reduce number of parameters to learn via hand-coded symmetry.

– Consider both a good prior (ψ=0.8) and a weak prior (ψ=0.6)



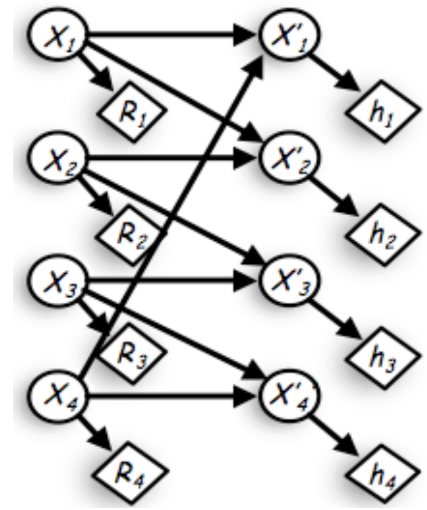
Empirical returns show good learning. Using domain-knowledge to constrain the structure is more useful than having accurate priors.

Can we infer this structure from data?

Case study #2: Learning a factored model

[Ross & Pineau. UAI'08]

- Consider a factored model, where both the graph structure and transition parameters are unknown.



Guestrin et al. JAIR'03

- Bayesian POMDP framework:

  $S' = S \times (G, \theta_G)^{|A|}$

  $A' = A$

  $Z' = S$

  $T'(s, G, \theta_G, a, s', G', \theta'_{G'}) = Pr(s' \mid s, G, \theta_G, a)\, Pr(G', \theta'_{G'} \mid G, \theta_G, s, a, s')$

  – Approximate posterior $Pr(G_a \mid h)$ using a particle filter.

  – Maintain exact posterior $Pr(\theta_G \mid G_a)$ using Dirichlet distributions.

  – Solve the planning problem using online forward search.

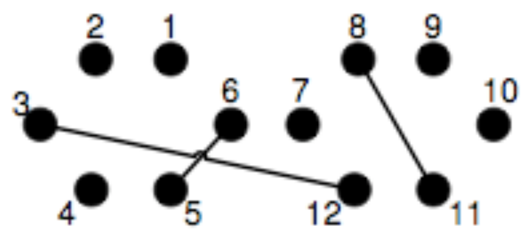
Case study #2: Learning a factored model

Network administration domain *[Guestrin et al. JAIR'03]*

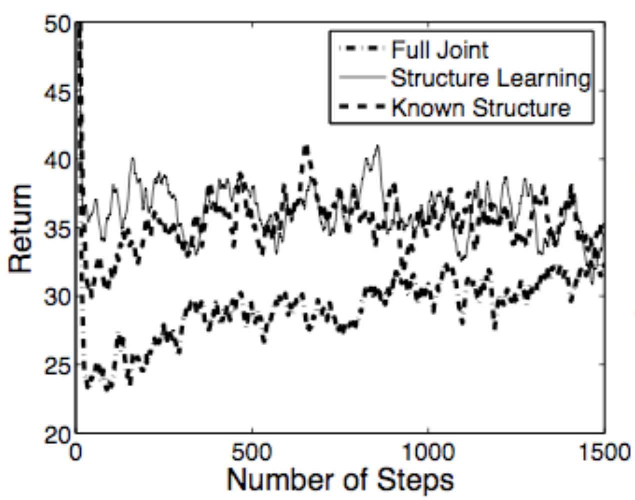
» Two-part densely connected network.

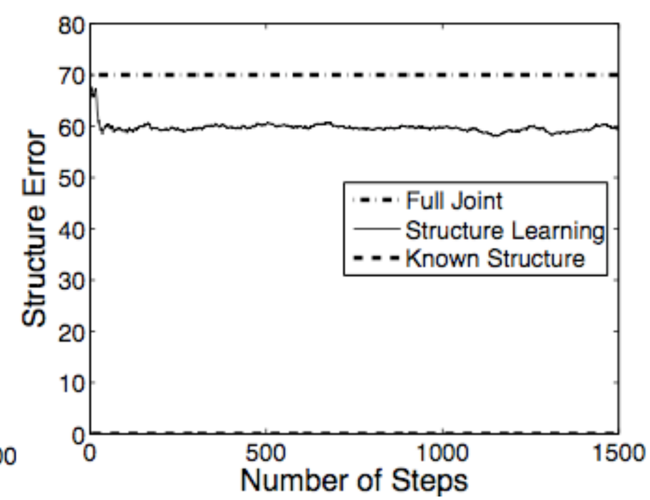
» Each node is a machine (on/off state).

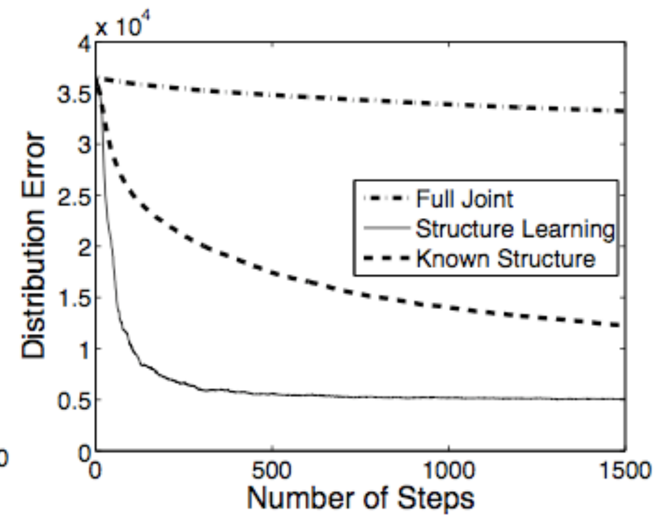
» *Actions*: reboot any machine, do nothing.



Bayesian RL results:





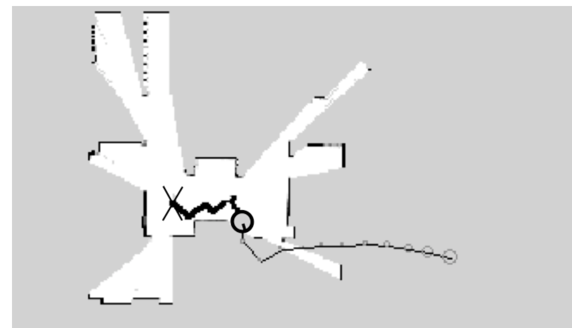


Learning the structure and parameters **simultaneously** improves performance.

Case study #3: Multitasking SLAM

[Guez & Pineau. ICRA'10]

- SLAM = Simultaneous Localization and Mapping
  - » One of the key problems in robotics.
  - » Usually solved with techniques such as EM.



- Active SLAM = Simultaneous Planning, Localization, and Mapping
  - » Can be cast as a POMDP problem.
  - » Often, greedy exploration techniques perform best.

- Multitasking SLAM = Active SLAM + other simultaneous task

  e.g. target following / avoidance

Case study #3: Multitasking SLAM

Case study #3: Multitasking SLAM

Decision-theoretic framework:

State space: $S = X \times M \times P$
- X = set of possible trajectories taken
- M = set of possible maps
- P = set of additional planning states

Actions: $A = D \times \theta$
- D = forward displacement
- θ = angular displacement

Observations: $Z = L \times U$
- L = laser range-finder measurements
- U = odometry reading

Learning (state estimation):

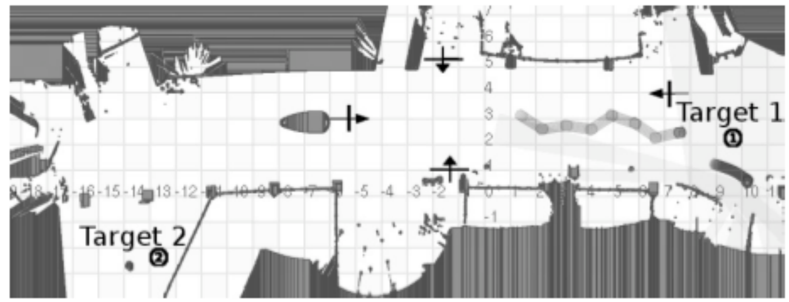
Approximated using a Rao-Blackwellized particle filter.

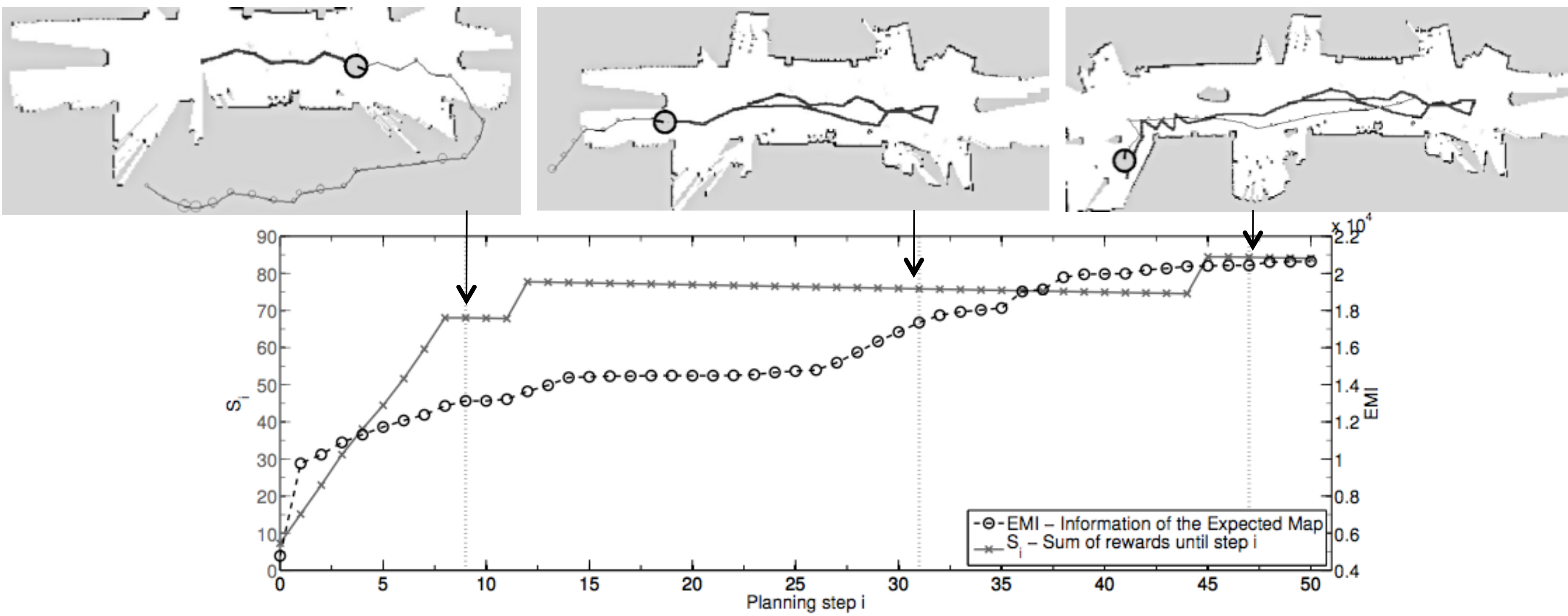
$$p(x_{1:t}, m \mid l_{1:t}, u_{0:t}) = p(m \mid x_{1:t}, l_{1:t})\, p(x_{1:t} \mid l_{1:t}, u_{0:t})$$

Planning: Online forward search + **Deterministic motion planning algorithm (e.g. RRTs) to allow deep search.**

Case study #3: Multitasking SLAM

Target following experiment:





Beyond the standard POMDP framework

- Policy search for POMDPs (Hansen, 1998; Meuleau et al. 1999; Ng&Jordan, 2000; Aberdeen&Baxter, 2002; Braziunas&Boutilier, 2004)
- Continuous POMDPs (Porta et al. 2006; Erez&Smart 2010; Deisenroth&Peters 2012)
- Factored POMDPs (Boutilier&Poole, 1996; McAllester&Singh, 1999; Guestrin et al., 2001)
- Hierarchical POMDPs (Pineau et al. 2001; Hansen&Zhou, 2003; Theocharous et al., 2004; Foka et al. 2005; Toussaint et al. 2008; Sridharan et al. 2008)
- Dec-POMDPs (Emery-Montemerlo et al. 2004; Szer et al. 2005; Oliehoek et al. 2008; Seuken&Zilberstein, 2007; Amato et al., 2009; Kumar&Zilberstein 2010; Spaan et al. 2011)
- Mixed Observability POMDPs (Ong et al., RSS 2005)
- ρPOMDPs (Araya et al., NIPS 2010)
- …