Solving partially observable Markov decision processes via VFA

Michael L. Littman

Brown University

joint work with Leslie Kaelbling, Tony Cassandra, and Ron Parr (Berkeley)

Partially observable Markov decision processes

POMDP = MDP - GPS

Assume:

- environment is an MDP with finite states and actions
- rewards and transitions known
- agent cannot observe the state of the MDP directly

Agent is presented with probabilistic “observations” based on current state and action

Observations can be one to many with the states

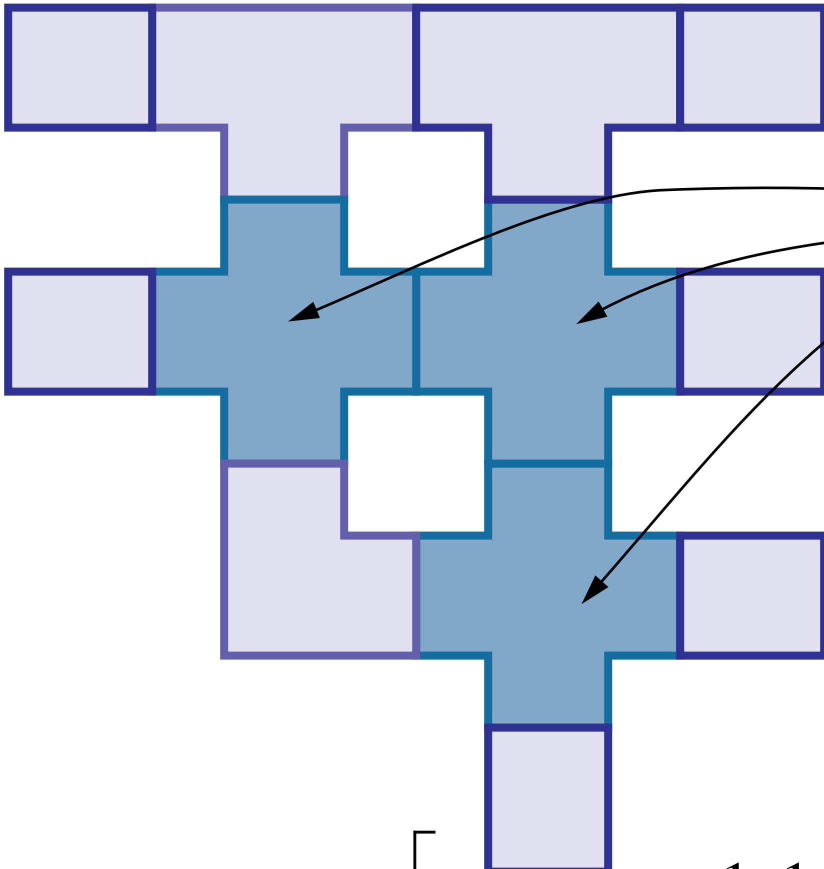
Summarizing observation histories

Agent's task is to choose actions to maximize reward given:

- model of the environment
- a distribution over initial states
- history of observations and actions

Histories are clumsy: summarize using "belief states"

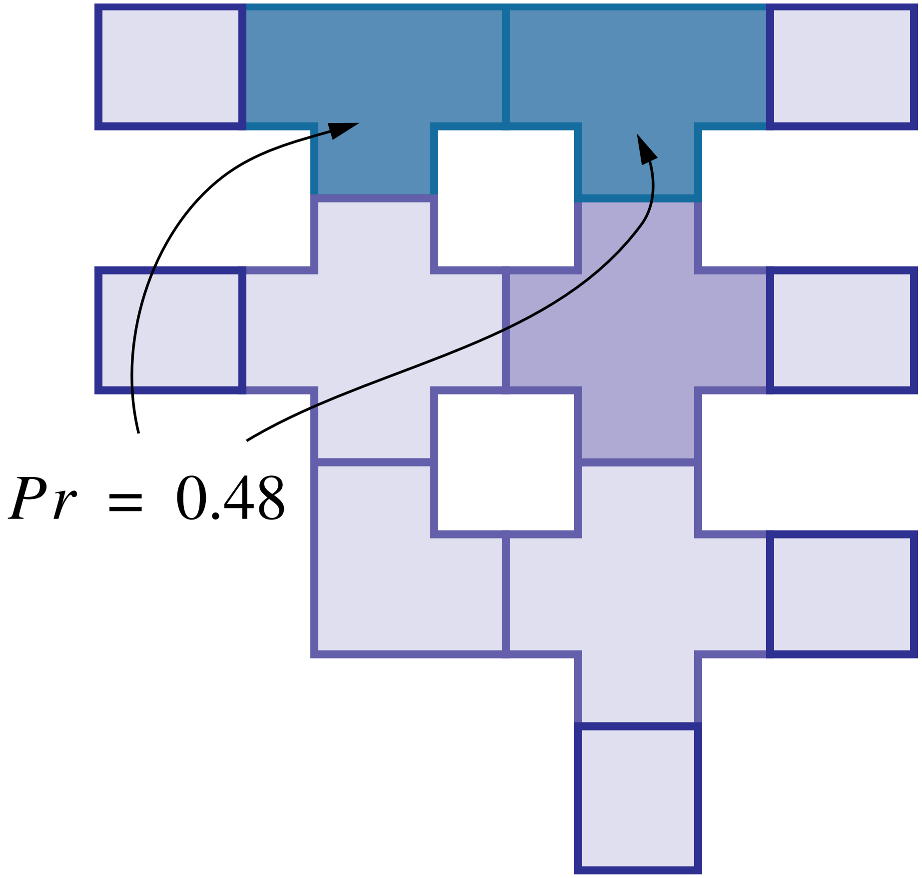
Belief state example: initial state



$$Pr = \frac{1}{3}$$

$$b = \left[0\ 0\ 0\ 0\ 0\ \frac{1}{3}\ \frac{1}{3}\ 0\ 0\ \frac{1}{3}\ 0\ 0 \right]$$

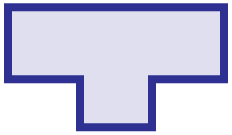
Belief state example: new state



Action:

Move North

Observation:



$$b = \begin{bmatrix} 0 & 0.48 & 0.48 & 0 & 0 & 0.01 & 0.02 & 0 & 0 & 0.01 & 0 & 0 \end{bmatrix}$$

Building a POMDP controller

Can construct SE easily, but what about the policy?

Good:

- optimal policy is a solution to continuous-space <u>MDP</u>

Bad:

- optimal policy is a solution to <u>continuous</u>-space MDP

Good:

- there is a compact representation for V_t

Bad:

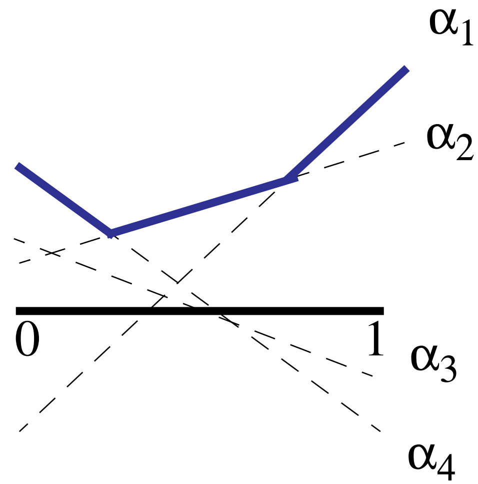
- the representation can grow explosively with t

Geometry of the t-step value function

- Max over a finite number of linear functions:

$$V_t(b) = \max_{\alpha \in \mathcal{V}_t} \alpha \cdot b$$

- Piecewise linear and convex



- V^* is convex but not necessarily piecewise linear

VFA for POMDPs

Even small problems very difficult to solve analytically

RL approach (similar to Chrisman and McCallum)

- simulation
- one-step sampled backups
- linear function (vector) for each Q function

$$Q(b, a) = q_a \cdot b$$

- V is piecewise linear and convex

$$V(b) = \max_a Q(b, a) = \max_a q_a \cdot b$$

Test environments

Problem	Source	States	Actions	Obs	Noise
Tiger	Cassandra, et al., 1994	2	3	2	O
Shuttle	Chrisman, 1992	8	3	5	T/O
Paint	Kushmerick, et al., 1993	4	4	2	T/O
4x3	Parr & Russell, 1995	11	4	6	T
4x4	Cassandra, et al., 1994	16	4	2	-
Cheese	McCallum, 1993	11	4	7	-
Hallway1	Littman, et al., 1995	57	5	21	T/O
Hallway2	Littman, et al., 1995	89	5	17	T/O

Linear VFA solved all, given some parameter twiddling

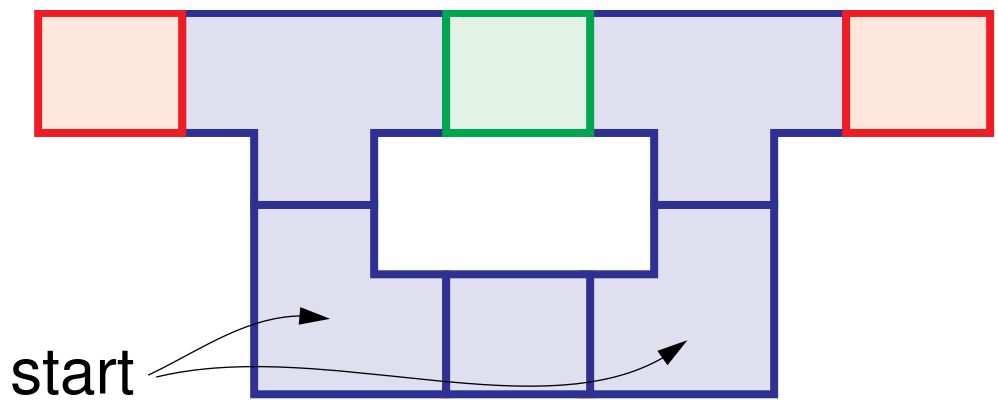
Results described in detail in proceedings paper and TR

Linear not always sufficient

For some problems, we know the optimal value function and the pwlc approximations were much simpler.

Some problems have optimal policies that *cannot* be represented with linear Q functions

Argument similar to why *xor* is not linearly separable



PWLC learning

Set of vectors, Q_a, for each Q function

$$Q_a(b) = \max_{\alpha \in Q_a} \alpha \cdot b$$

Learning only changes the α with largest dot product (competitive update rule)

Solves some (linearly non-representable) problems

Prone to “losing” vectors during learning

Soft maxes (Parr, Russell)

Set of vectors, $\mathcal{V}$, parameterize the function

$$V(b) = \left(\sum_{\alpha \in \mathcal{V}} (\alpha \cdot b)^k \right)^{1/k}$$

Acts like max as k gets big (like L_k norm)

Increase k during learning

Works! Though sensitive to parameter settings

VFA POMDP approaches

We've experimented with many variations:

- simulation vs. value iteration at random points
- sample vs. full vs. gradient backups
- direct vs. residual gradient
- representing Q vs representing V

Results qualitatively similar---no clear winner

VFA POMDP issues

- does it help to use convexity?
- is there a better soft max out there?
- computation time to make a backup
- focus on the important parts of the space
- range of initial random values matters
- may help to initialize with Q values or overestimates
- exploration very difficult
- Bellman equation has multiple solutions
- parameter setting!
- optimal policy simpler to represent than optimal VF?