POMDPs (Ch. 17.4-17.6)

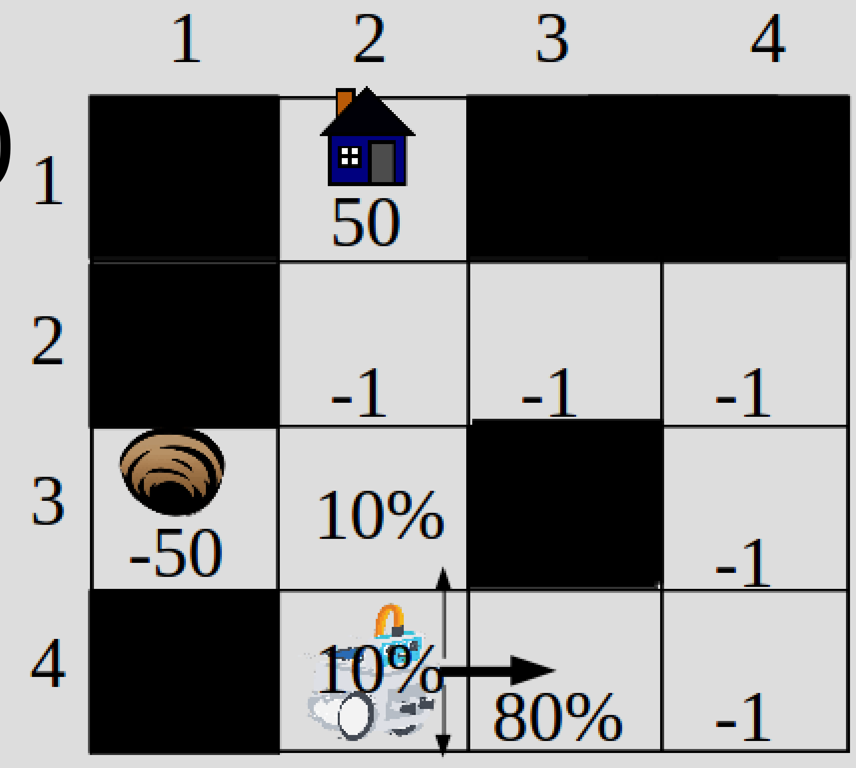
Markov Models		Do we have control over the state transitons?	
		NO	YES
Are the states completely observable?	YES	**Markov Chain**	**MDP** Markov Decision Process
	NO	**HMM** Hidden Markov Model	**POMDP** Partially Observable Markov Decision Process

Markov Decision Process

Recap of Markov Decision Processes (MDPs):

Know:
- Current state (s)
- Rewards for states (R(s))

Uncertain:
- Result of actions (a)



POMDPs

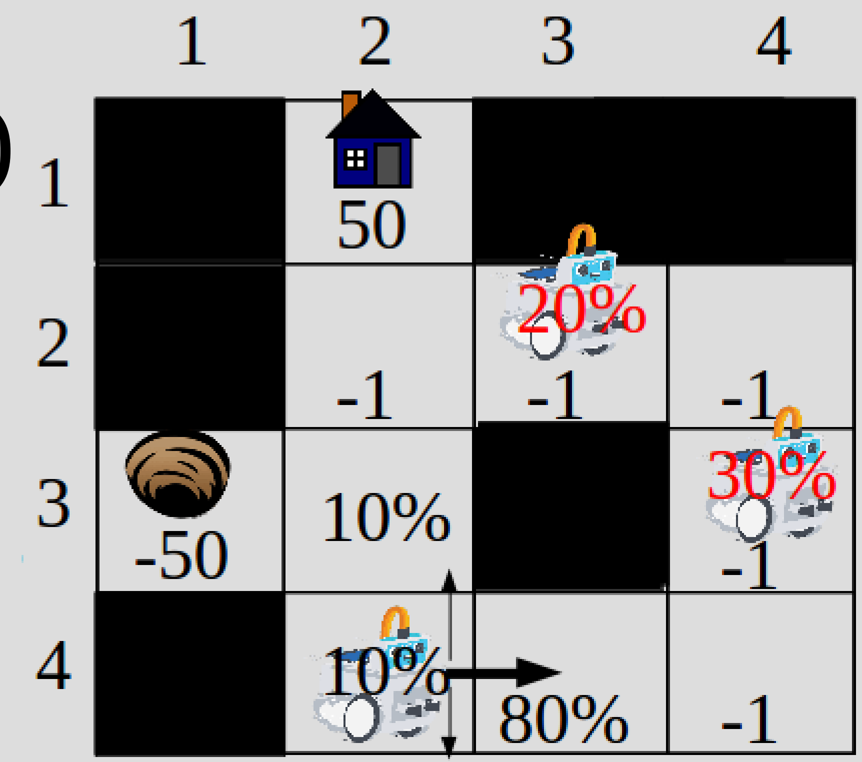
Today we look at Partially Observable MDPs:

Know:

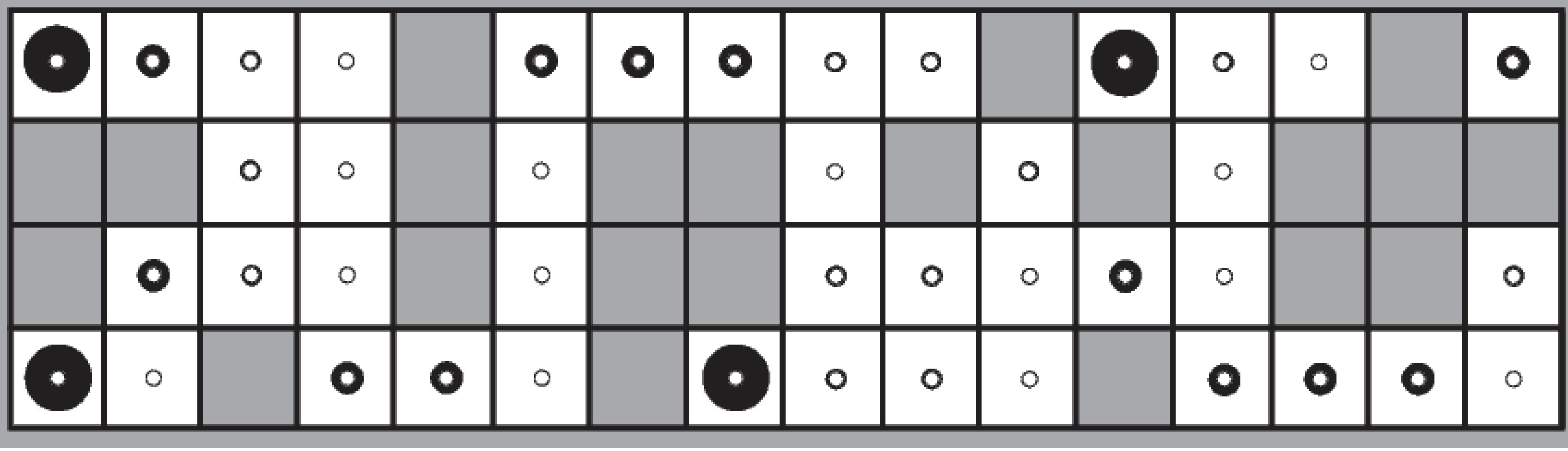
- ~~Current state (s)~~
- Rewards for states (R(s))

Uncertain:

- Current state (s)
- Result of actions (a)

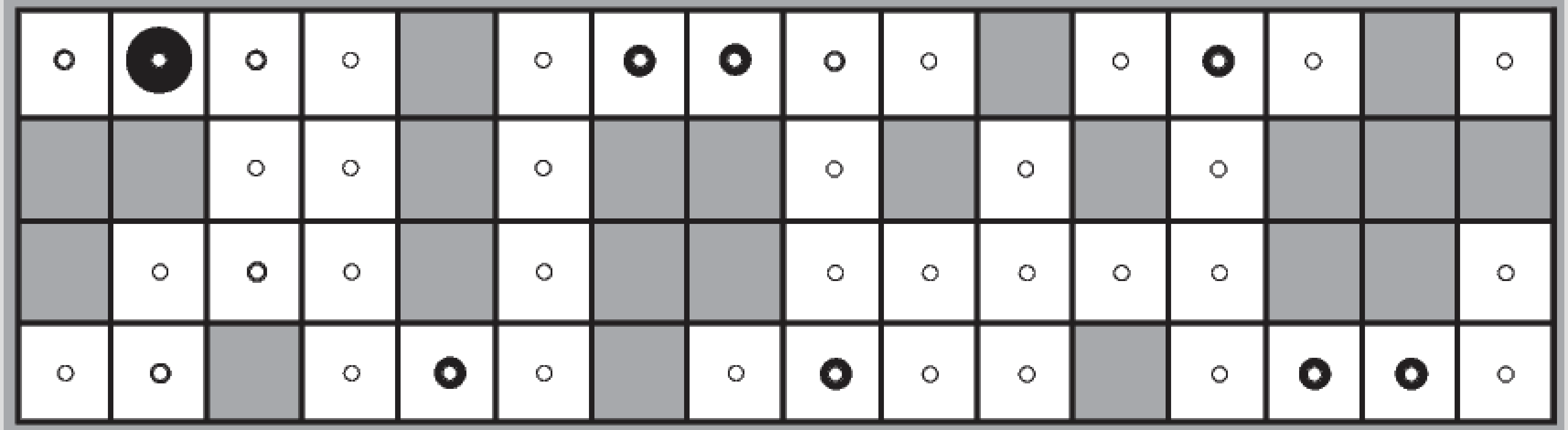


Filtering + Localization



(a) Posterior distribution over robot location after $E_1 = NSW$

where walls are

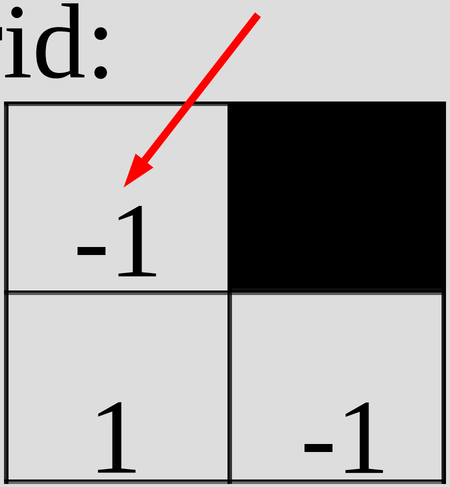


(b) Posterior distribution over robot location after $E_1 = NSW, E_2 = NS$

POMDPs

rewards, R(s)

Let's examine this much simpler grid:

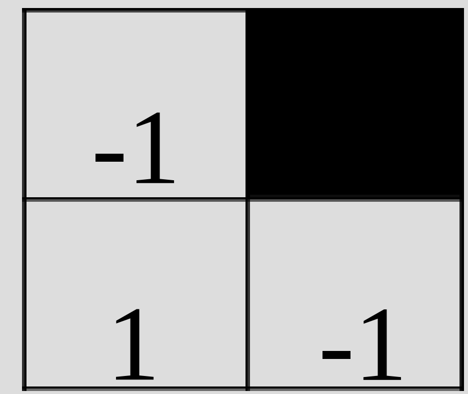


Instead of knowing our exact state, we have a <u>belief state</u>, which is a probability for being in an state

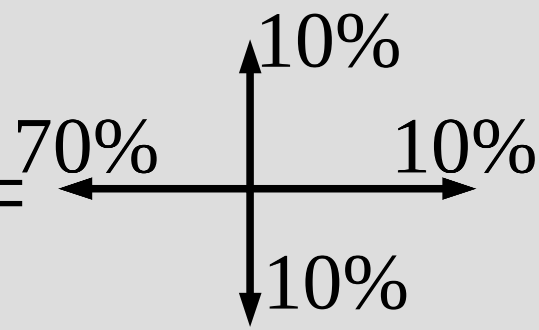
Additionally, we assume we cannot perfectly sense the state, instead we observe some evidence, e, and have P(e|s)

POMDPs

Let's assume our movement is a bit more erratic: 70% in intended direction, 10% in any other direction



So move "left" =



Given our rewards, you want to reach the bottom left square and stay there as long as possible

POMDPs

Suppose our sensor could detect if we are in the bottom left square, but not perfectly

Suppose P(e|s) is:

20%	
90%	20%

... and P(¬e|s) is:

80%	
10%	80%

POMDPs

Assume our starting belief state is:

Obviously, we want to go either down or left as best action

Suppose we went “left” and saw evidence “e”

What is the resulting belief state?

POMDPs

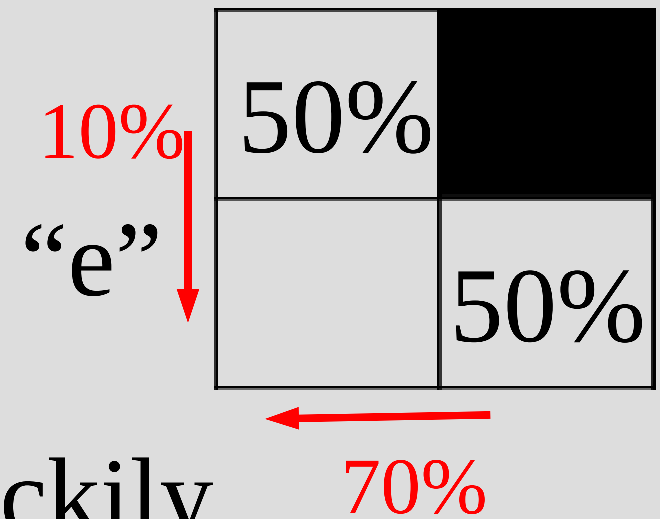
If we are in the top square,
we could see “e” by:

(1) Luckily moving down, see “e”

$$0.1 \cdot 0.9 = 0.09$$

(2) Saying in top, see “e” unluckily

$$0.9 \cdot 0.2 = 0.18$$



... or we could be in right square and:

(1) Move left and see “e”: $0.7 \cdot 0.9 = 0.63$

(2) Unluckily stay, see “e” unluckily

$$0.3 \cdot 0.2 = 0.06$$

POMDPs

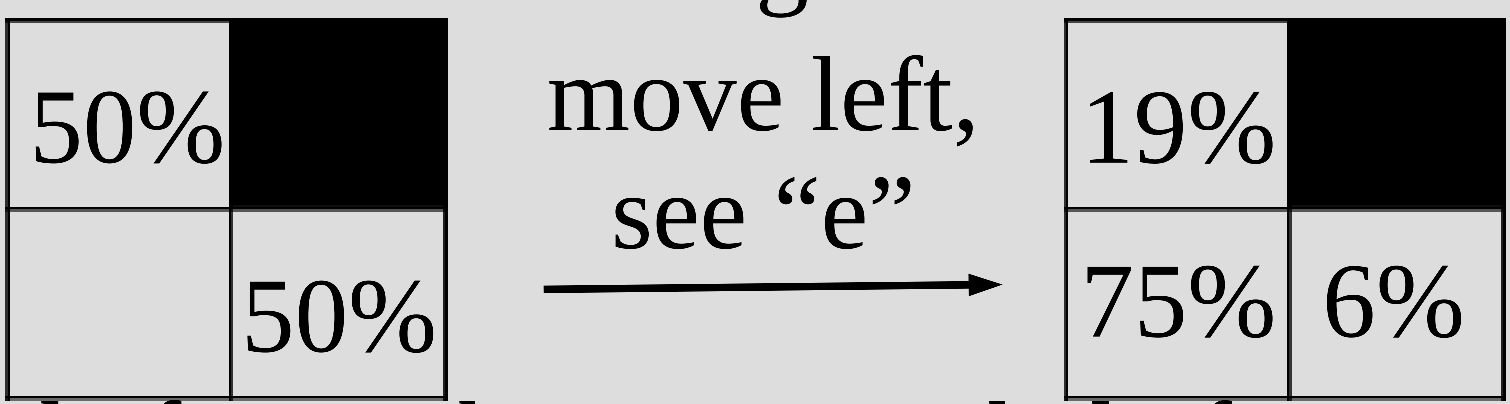
Since both top and right have a 50% chance of starting there, probability of bottom-left is:

$$0.5 \cdot (0.09) + 0.5 \cdot (0.63) = 0.36$$

Thus probability top-left: $0.5 \cdot (0.18) = 0.09$

... and bottom-right: $0.5 \cdot (0.06) = 0.03$

... then normalize so we get:



belief state: b belief state: b'

POMDPs

Formally, we can write how to get the next belief state (given “a” and “e”) as:

$$b'(s') = \alpha \cdot P(e|s') \cdot \sum_s P(s'|s,a) \cdot b(s)$$

What does this look like?

POMDPs

Formally, we can write how to get the next belief state (given “a” and “e”) as:

$$b'(s') = \alpha \cdot P(e|s') \cdot \sum_s P(s'|s,a) \cdot b(s)$$

What does this look like?

This is basically the “forward” message in filtering for HMMs

POMDPs

This equation is nice if we choose an action and see some evidence, but we want to find which action is best without knowing evidence

In other words, we want to start with some belief state (on below) and determine what the best action is (move down)

19%	
75%	6%

How can you do this?

POMDPs

Well, you can think of this as a transition from b to b' given action a... so we sum over e

$$
\begin{aligned}
P(b'|b,a) &= \sum_e P(b',e|b,a) \\
&= \sum_e P(b'|b,a,e) \cdot P(e|b,a) \\
&= \sum_e P(b'|b,a,e) \sum_{s'} \cdot P(e,s'|b,a) \\
&= \sum_e P(b'|b,a,e) \sum_{s'} \cdot P(e|s',b,a) \cdot P(s'|b,a) \\
&= \sum_e P(b'|b,a,e) \sum_{s'} \cdot P(e|s') \sum_s P(s,s'|b,a) \\
&= \sum_e P(b'|b,a,e) \sum_{s'} \cdot P(e|s') \sum_s P(s'|b,a,s)P(s|b,a) \\
&= \sum_e P(b'|b,a,e) \sum_{s'} \cdot P(e|s') \sum_s P(s'|a,s)b(s)
\end{aligned}
$$

P(b'|b,a,e) = 1
if b' is the forward filtering message...
0 otherwise

POMDPs

Thus, we can define transitions between belief states: P(b' | b, a)

50%	
	50%

belief state: b

move left →

do not assume see "e"

19%	
75%	6%

48% chance of this b'

belief state: b'

$$\sum_e P(b'|b,a,e) \sum_{s'} \cdot \boxed{P(e|s') \sum_s P(s'|a,s)b(s)}$$ ← calc as before

$$= 1 \cdot \left(0.36 + 0.09 + 0.03\right) = 0.48$$ ← 52% chance b' with ¬e

And we can find the expected reward of b' as: $\sum_s b(s) \cdot R(s)$, so for our b': $0.19 \cdot (-1) + 0.75 \cdot (1) + 0.06 \cdot (-1) = 0.5$

POMDPs

Essentially, we have reduce a POMDP
to a simple MDP, except we have transitions
and rewards of belief states (not normal states)

This is slightly problematic as belief states
involve probabilities, so there are an infinite
amount of them (and probability numbers)

This makes them harder to reason on,
but not impossible...

Value Iteration in POMDPs

Let's consider an even more simplified problem to run a modified value iteration:

0	1

We will only have two states: s_0, s_1, with $R(s_0)=0$, $R(s_1)=1$

Thus we can use the Bellman equation, except with belief states (let $\gamma=1$)

$$U(s) = R(s) + \gamma \cdot \max_a \sum_{s'} P(s'|s,a) \cdot U(s')$$

Value Iteration in POMDPs

Assume there are only two actions: “go” and “stay” (with 0.9 chance of result you want)

$$U(s) = R(s) + \gamma \cdot \max_a \sum_{s'} P(s'|s,a) \cdot U(s')$$

A=“go” at s_0: $0 + \gamma(0.9 \cdot 1 + 0.1 \cdot 0) = 0.9$

A=“go” at s_1: $1 + \gamma(0.9 \cdot 0 + 0.1 \cdot 1) = 1.1$

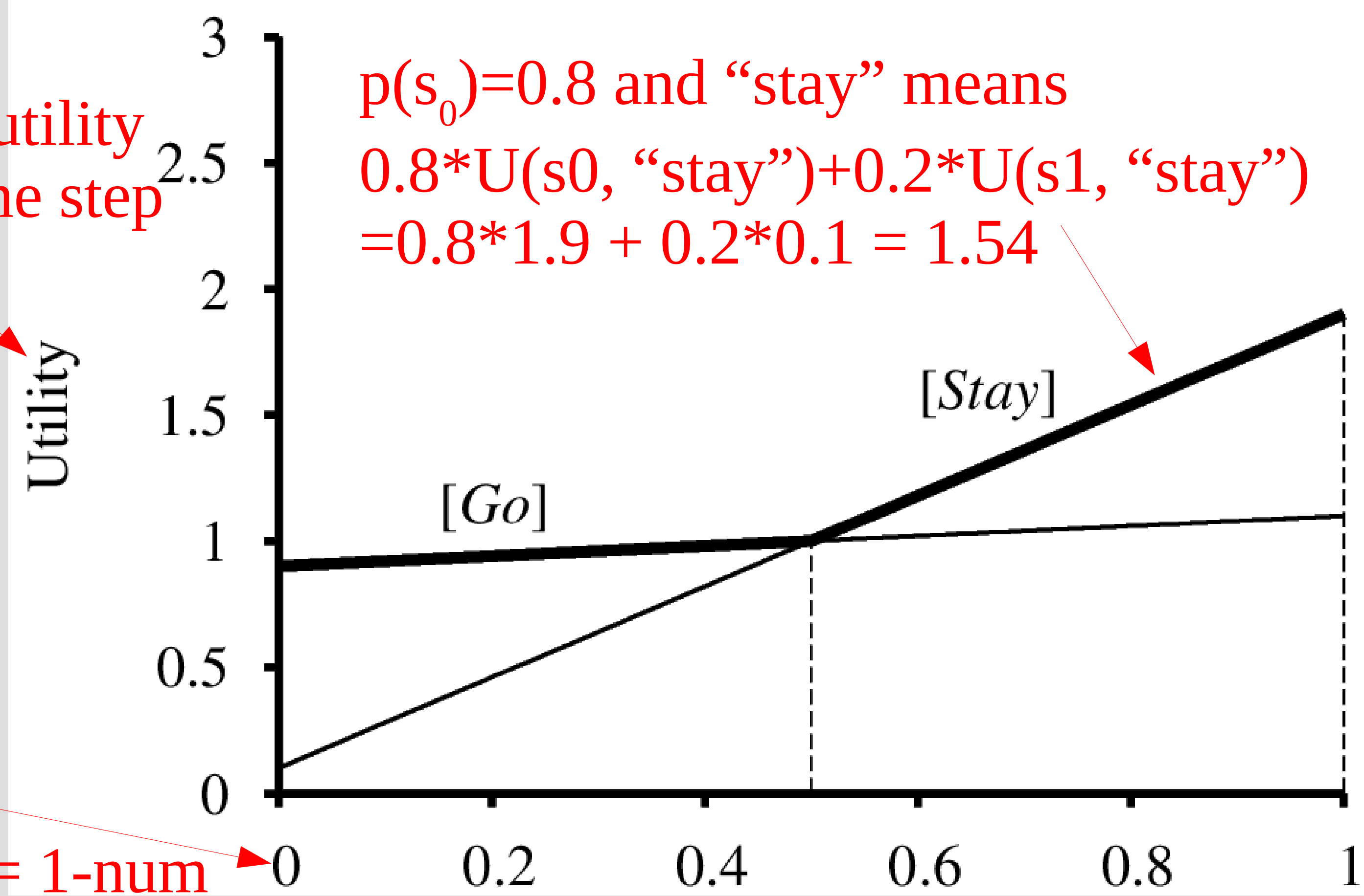
A=“stay” at s_0: $0 + \gamma(0.9 \cdot 0 + 0.1 \cdot 1) = 0.1$

A=“stay” at s_1: $1 + \gamma(0.9 \cdot 1 + 0.1 \cdot 0) = 1.9$

... thus we can graph the actions as lines on belief probability vs utility graph

Value Iteration in POMDPs

Just like with the Bellman equations, we want max action, so pick “Go” if prob<0.5



Value Iteration in POMDPs

In fact, as we compute the overall utility of a belief state as: $U(b) = \sum_s U(s) \cdot b(s)$

... this will always be linear in terms of b(s)

So in our 2-D example, we will always get a number of lines that we want to find max of

For larger problems, these would be hyper-planes (i.e. if we had 3 states, planes)

Value Iteration in POMDPs

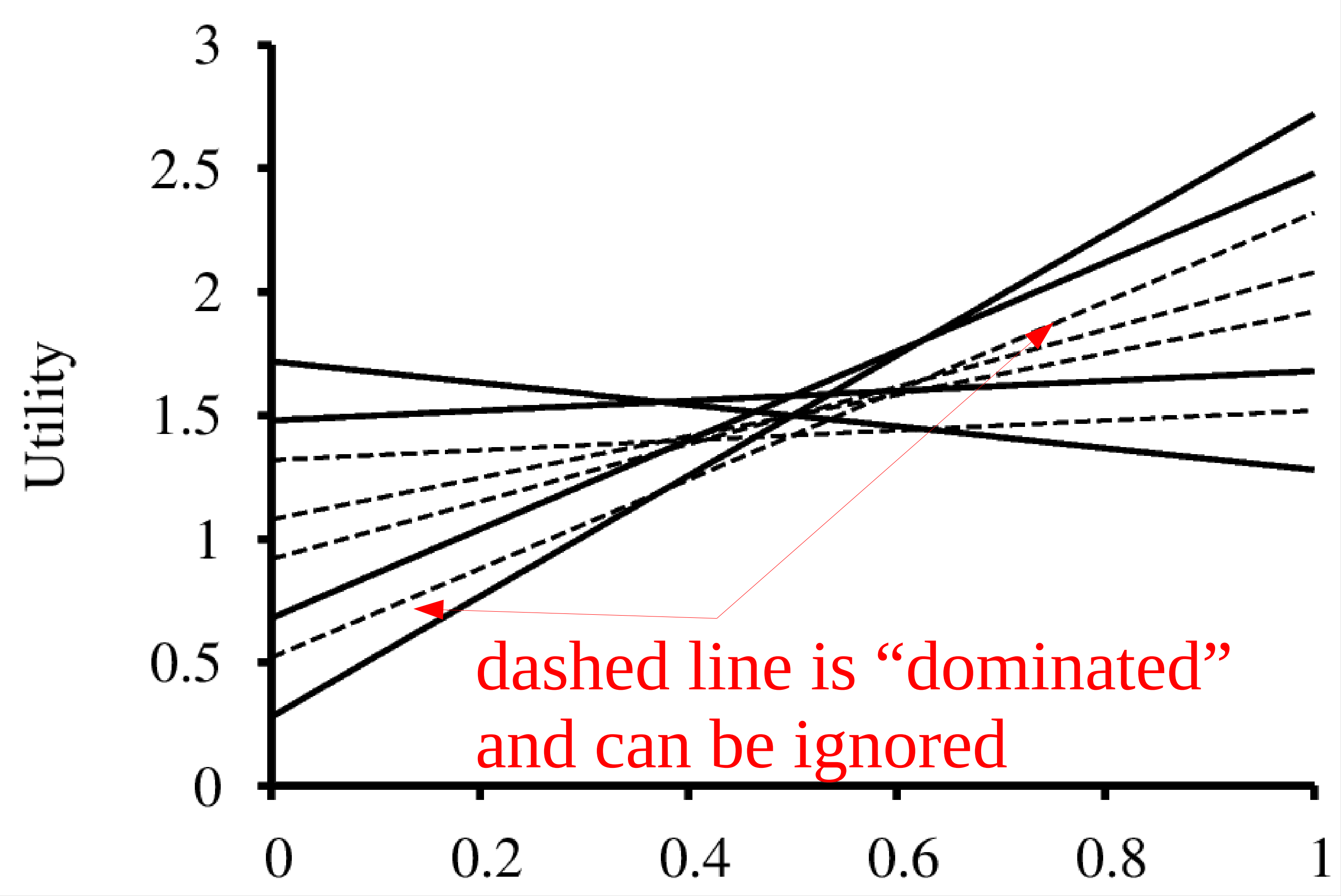
However, to find the best second action we need to account for the seen evidence

Assume our evidence has 2 options(true/false), then we'll need to generate eight lines for potential next best actions

For each of the initial two lines, we have four have to consider four combinations of next actions (based on evidence)

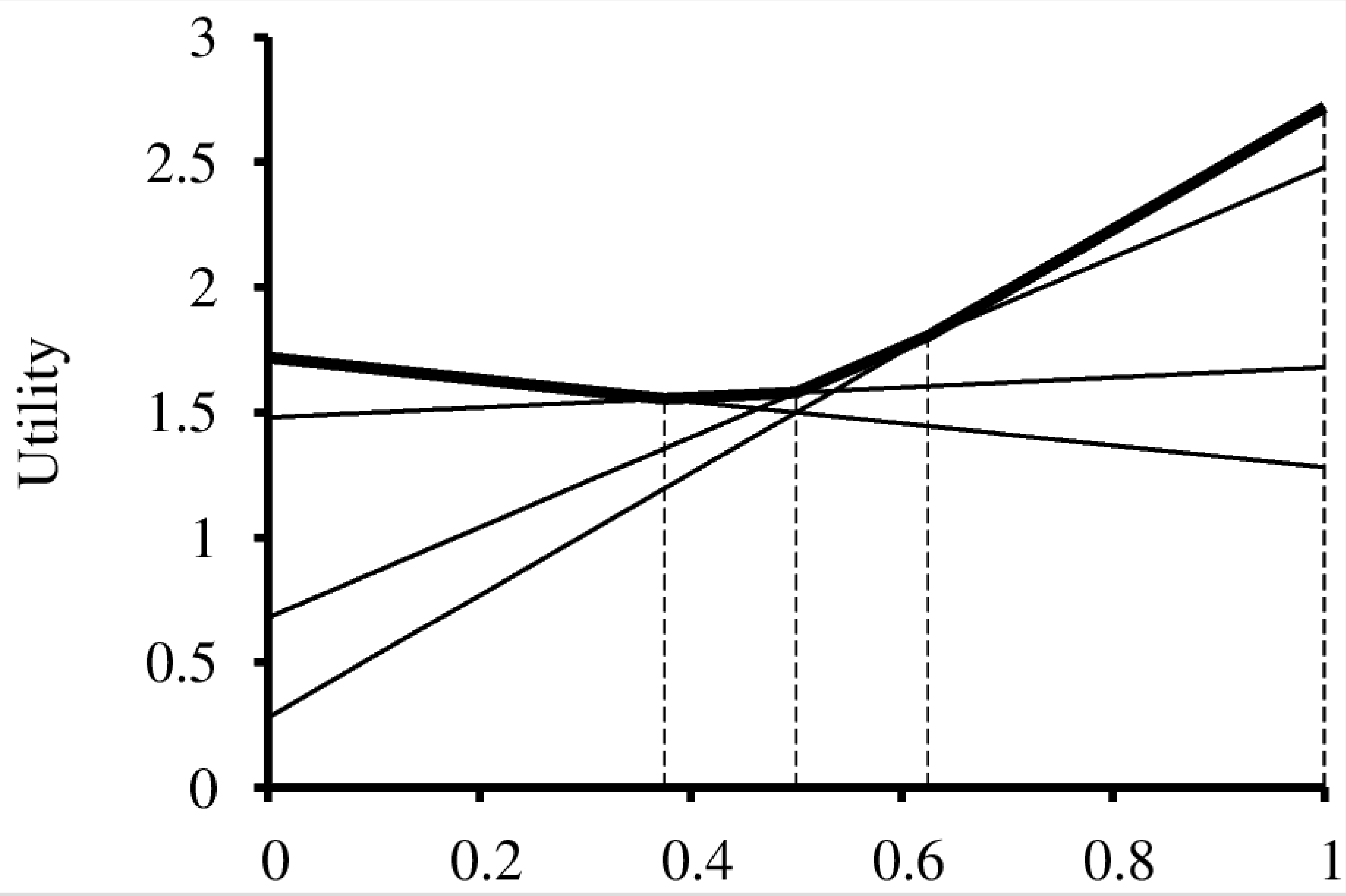
Value Iteration in POMDPs

All 8 possibilities of two sequence actions:



Value Iteration in POMDPs

4 options after dropping terrible choices:



Value Iteration in POMDPs

These non-dominated actions make a utility function: (1) linear (piece-wise) (2) convex

Unfortunately, the worst-case is approximately $|A|^{|E|^d}$, so even in our simple 2-evidence & 2-action POMDP at depth 8 it has 2^{256} lines

Thankfully, if you remove dominated lines, at depth 8 there are only 144 lines that form the utility function estimate

Value Iteration in POMDPs

This website gives a bit better visualizations than the book:
https://pomdp.org/tutorial/pomdp-vi-example.html

It shows how you progressively update the values you will get depending on the initial distribution of probabilities

(Though it skips all the formulas)

Online Algorithm in POMDPs

You could also break down the actions/evidence to build a tree to search

Requires leaf as estimate, but is: $|A|^d \cdot |E|^d$