16-785: Integrated Intelligence in Robotics: Vision, Language, and Planning

Spring 2019

Reinforcement learning – part 1

Overview

- Probability basics
- Reinforcement learning
  - Markov Decision Process
  - Solving MDPs
- References
  - Reinforcement learning – Sutton & Barto, Second edition, 2018

    http://incompleteideas.net/book/the-book-2nd.html

How can we make [robots] to make [optimal] decisions?

Planning recap

- A*
  - Admissible (underestimate) heuristics
  - Consistent (monotonic) heuristics
- Dynamic programming – recursion

  $V(x_t)$ as a function of $V(x_{t+1})$ or $V(x_{t-1})$
- D* dynamic repair

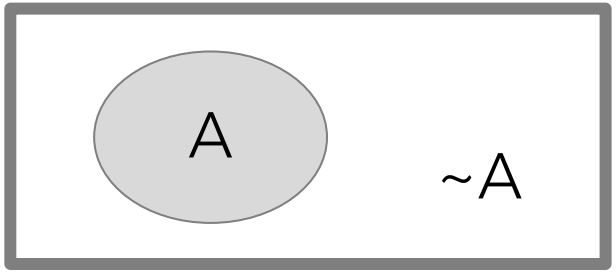
Exit from deterministic world
Enter stochastic world

Probability preliminaries

- Probability represents uncertainty
- $0 \leq P(A) \leq 1$
- $P(True) = 1$
- $P(False) = 0$

Probability preliminaries

- Probabilities over all events sum to 1
- P(A) + P(~A) = 1
- P(~A) = 1 – P(A)



p(A): Probability of A being true
p(~A): probability of A not being true

Probability preliminaries

- Probabilities over all events sum to 1

$$\sum_{v \in D(x)} P(x = v) = 1$$

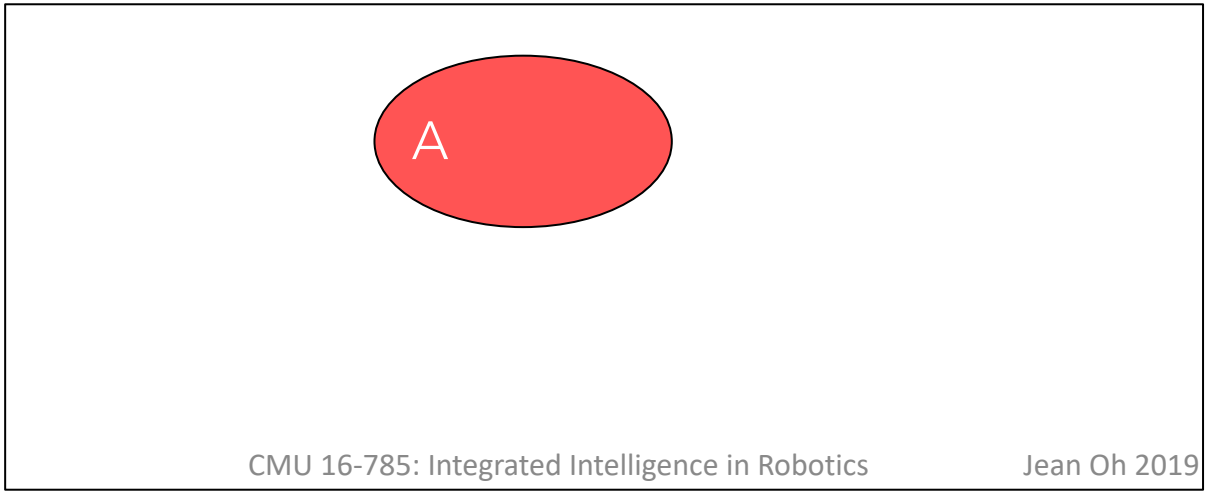
Where D(x) denotes the ***domain*** of valid values for variable x

P(A or B) = P(A) + P(B) – P(A and B)

Conditional probability

p(A): Probability of A (given the world)

A: You will be in Greece next month. P(A) = .1



Conditional probability

p(A): Probability of A (given the world)

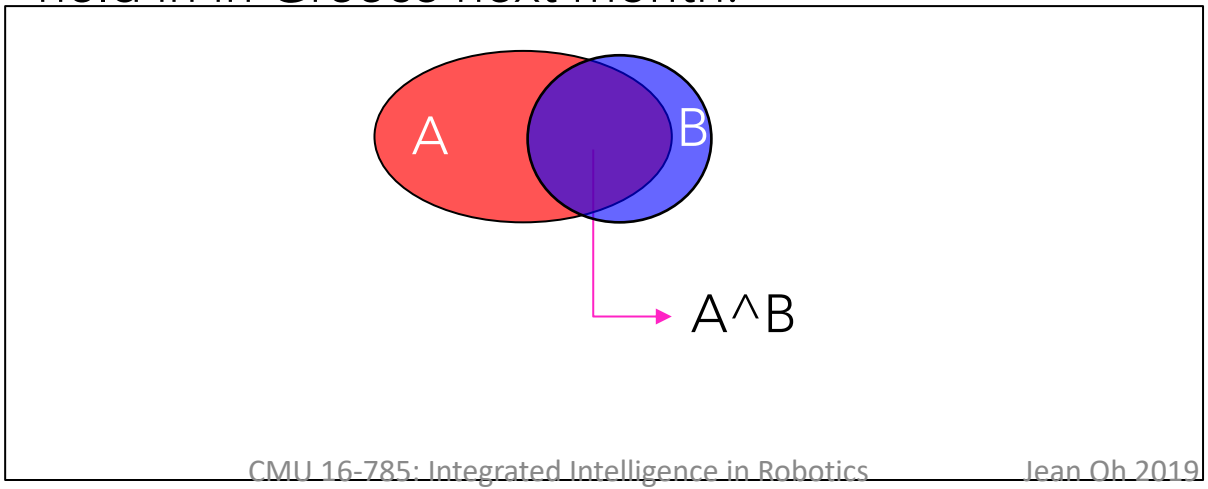
A: You will be in Greece next month.

B: Your paper got accepted at a conference to be held in in Greece next month.

P(A) = .1

P(B) = .05

P(A^B) = .04



Conditional probability

p(A): Probability of A (given the world)

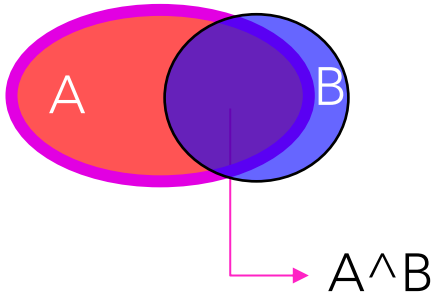
A: You will be in Greece next month.

B: Your paper got accepted at a conference to be held in in Greece next month.

P(A) = .1

P(B) = .05

P(A^B) = .04



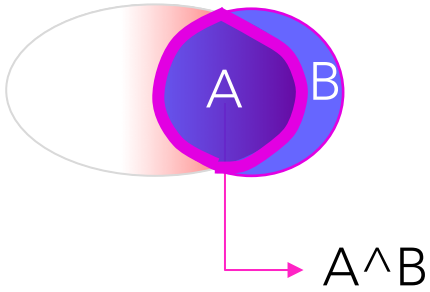
P(A) + P(~A) = 1

Conditional probability

p(A|B): Probability of A "given" B

A: You will be in Greece next month.
B: Your paper got accepted at a conference to be held in in Greece next month.

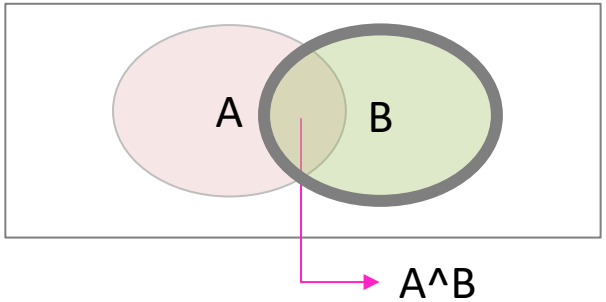
P(A) = .1
P(B) = .05
P(A^B) = .04
P(A|B) = .04/.05
= .8



$$p(A \mid B) = \frac{p(A \wedge B)}{p(B)}$$

P(A|B) + P(~A|B) = 1

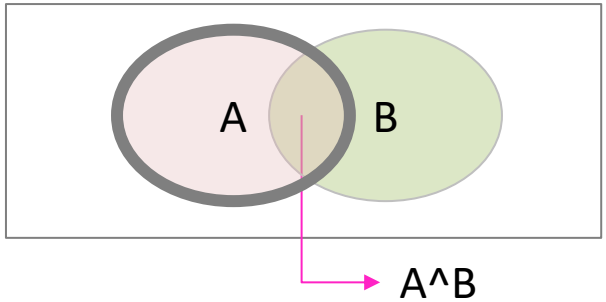
Conditional probability
p(A|B): Probability of A "given" B



$$p(A \mid B) = \frac{p(A \wedge B)}{p(B)}$$

$$p(A \wedge B) = \underline{p(A \mid B)p(B)}$$

Conditional probability
P(B|A): Probability of B "given" A

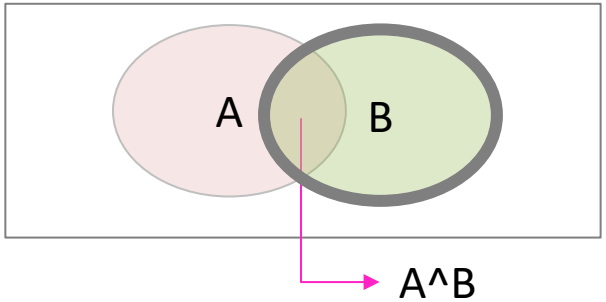


$$p(B \mid A) = \frac{p(A \wedge B)}{p(A)}$$

$$p(A \wedge B) = \underline{p(B \mid A)p(A)}$$

$$p(A \mid B)p(B) = p(B \mid A)p(A)$$

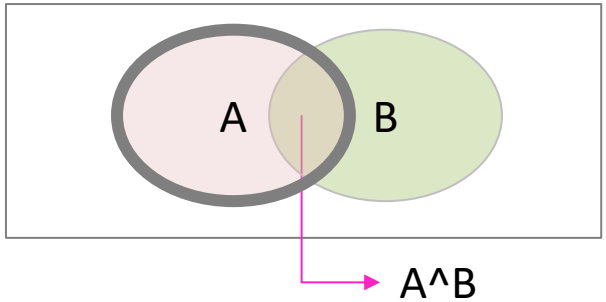
Conditional probability
p(A|B): Probability of A "given" B



$$p(A \mid B) = \frac{p(A \wedge B)}{p(B)}$$

$$p(A \wedge B) = p(A \mid B)p(B)$$

Conditional probability
P(B|A): Probability of B "given" A



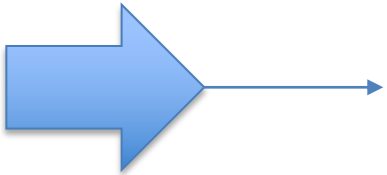
$$p(B \mid A) = \frac{p(A \wedge B)}{p(A)}$$

$$p(A \wedge B) = p(B \mid A)p(A)$$

$$p(A \mid B)p(B) = p(B \mid A)p(A)$$

Bayes' rule: $$p(A \mid B) = \frac{p(B \mid A)p(A)}{p(B)}$$

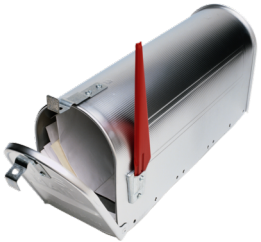
Expected value



$100

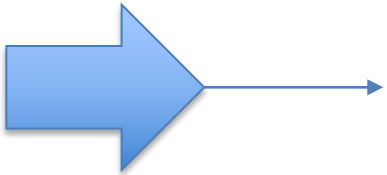
Expected value of mailing your stub = $100

$100 - $60 = $40



If you mail me your stub I will pay you $100.

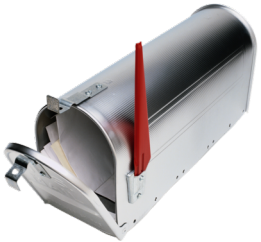
Expected value



$100

Expected value of mailing your stub = $100

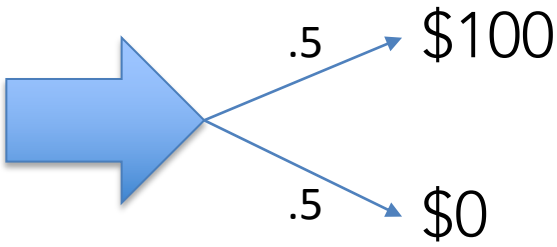
$100 - $60 = $40



If you mail me your stub I will pay you $100.

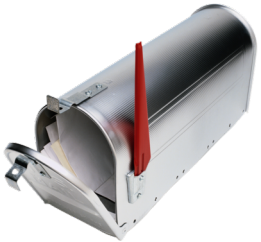
If I don't receive it I won't pay anything.
Probability of mail getting lost is 0.5.

Expected value



Expected value of mailing your stub
= .5x$100 + .5x$0
= $50

$50 - $60 = -$10



If you mail me your stub I will pay you $100.

If I don't receive it I won't pay anything.
Probability of mail getting lost is 0.5.

Expected value

- $X = x_1, \ldots, x_n$: set of n possible outcomes
- $v(x_i)$: value associated with outcome x_i
- $p(x_i)$: $p(X=x_i)$ probability of outcome x_i; $\sum_{i=1}^{n} p(x_i) = 1$
- Expected value for a single value $X=x_i$: $p(x_i)v(x_i)$

Expected value

- $X = x_1, \ldots, x_n$: set of n possible outcomes
- $v(x_i)$: value associated with outcome x_i
- $p(x_i)$: $p(x_i)$: $p(X=x_i)$ probability of outcome x_i; $\sum_{i=1}^{n} p(x_i) = 1$
- Expected value for a single value $X=x_i$: $p(x_i)v(x_i)$
- Expected value of X:

$$E(X) = \sum_{i=1}^{n} p(x_i)v(x_i)$$

Questions on preliminaries?

Learning what to do to solve a problem

- e.g. how to open a door

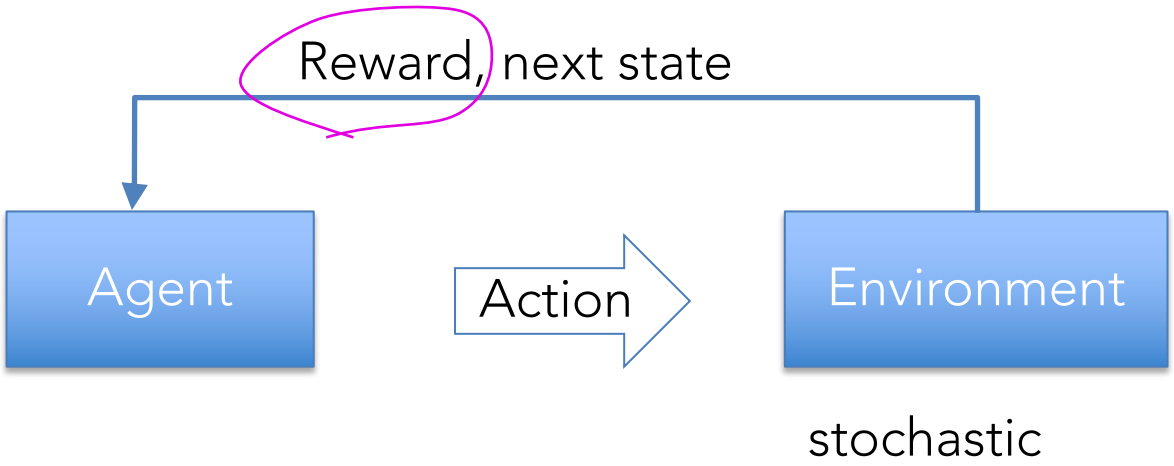
Supervised learning

- Told what to do from someone who knows how to solve the problem
- Learning from examples

Reinforcement learning

- Interact with environment to learn what works and what doesn't

Reinforcement learning



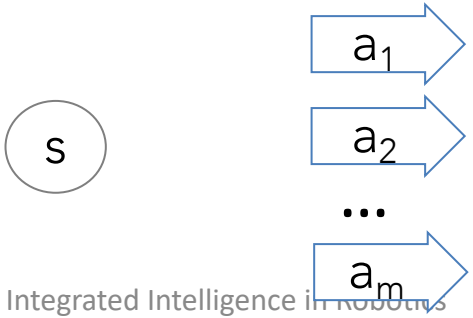
What is the goal of reinforcement learning?

Policy

Prepare what to do to maximize long-term, possibly discounted, expected reward wherever you are.

Mapping from state to action(s)

- Deterministic policy: $\pi(s) \rightarrow a$
- Stochastic policy: $\pi(s,a) = \pi(a|s) \rightarrow p(a|s)$ for each available action a



Reinforcement learning

- Reward function: immediate reward of state
- Value function: long-term reward of state
- Policy: a mapping from state to action(s)
- Find a policy that maximizes long-term reward

Markov property

It doesn't matter how you got here, but whatever happens from now on depends only on your current state.

$$p(s_{t+1}=s', r_{t+1}=r \mid s_t, a_t, s_{t-1}, a_{t-1}, \ldots, s_0, a_0) = p(s_{t+1}=s', r_{t+1}=r \mid s_t, a_t)$$

Reward vs. utility

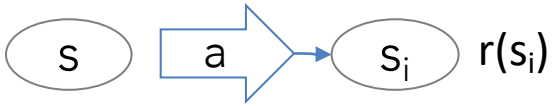
- Decision theory
- Reward: face value
- Utility: relative value based on which people make decisions
- e.g., the second piece of cheesecake or $100K today vs. $100K in 10 years
- Discounted rewards

Short-term vs. long-term benefits

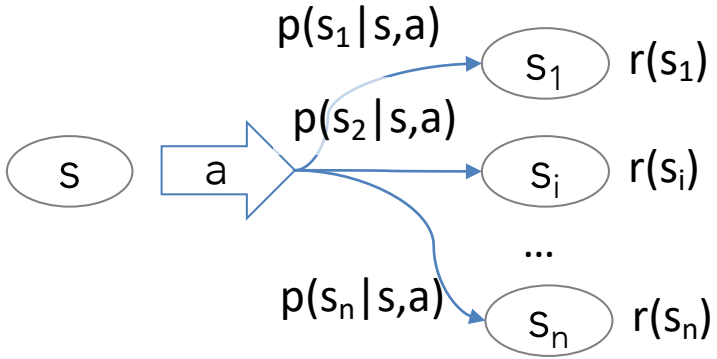
- Decision making based on:
  - Immediate rewards
  - Long-term rewards
  - Long-term expected rewards
  - Discounted long-term expected rewards

Stochastic state transition

- Deterministic



- Nondeterministic



$$\sum_{i=1}^{n} p(s_i \mid s,a) = 1$$

Markov Decision Process

- MDP = { S, A, T, r, γ }
- S: set of states
- A: set of actions
- T: state transition matrix SxAxS→$\mathcal{R}$
- r: reward function SxA→$\mathcal{R}$
- γ: discount factor (0,1]

Value function

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\}$$
$$= E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\}$$

State-value function for policy π: expected return of starting at state s and follow policy π thereafter.

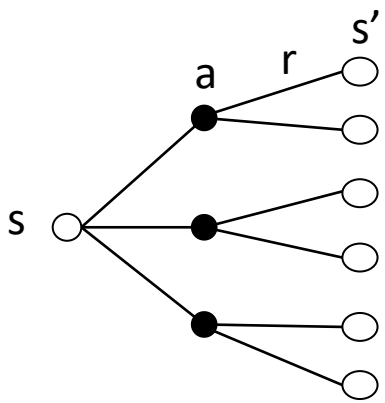
$$Q^{\pi}(s,a) = E_{\pi}\{R_t \mid s_t = s, a_t = a\}$$
$$= E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right\}$$

Action-value function for policy π

State-value function

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\}$$
$$= E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\}$$

State-value function for policy π: expected return of starting at state s and follow policy π thereafter.

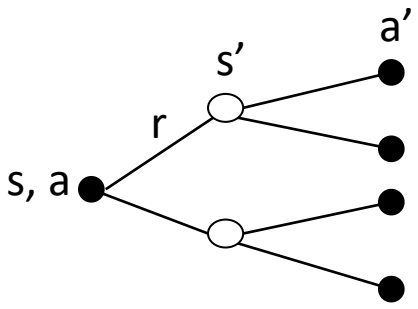


Backup diagram

Action-value function

$$Q^{\pi}(s,a) = E_{\pi}\{R_t \mid s_t = s, a_t = a\}$$
$$= E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right\}$$

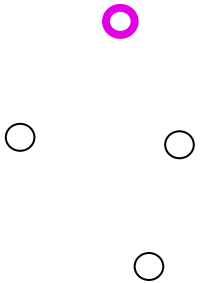
Action-value function for policy π: expected return of starting at state s, taking action a, and follow policy π thereafter.



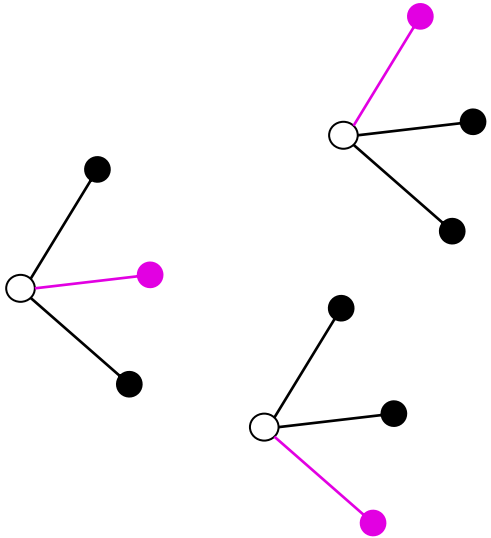
Backup diagram

Which action should we take?

V(s)



Q(s,a)

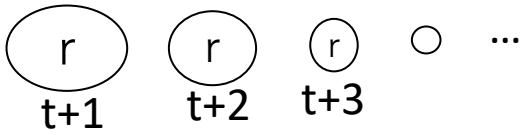


Bellman equation of value function

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\}$$

$$= E_{\pi}\left\{\boxed{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}} \middle| s_t = s\right\}$$

$$=$$



$$\gamma^0 r_{t+1} + \gamma^1 r_{t+2} + \gamma^2 r_{t+3} + \ldots$$

$$= r_{t+1} + \gamma(\gamma^0 r_{t+2} + \gamma^1 r_{t+3} + \ldots)$$

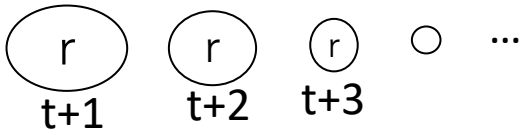
$$=$$

Bellman equation of value function

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\}$$

$$= E_{\pi}\left\{\boxed{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}} \middle| s_t = s\right\}$$

$$= E_{\pi}\left\{\boxed{r_{t+1} + \gamma\sum_{k=0}^{\infty}\gamma^k r_{t+k+2}} \middle| s_t = s\right\}$$



$$\gamma^0 r_{t+1} + \gamma^1 r_{t+2} + \gamma^2 r_{t+3} + \ldots$$

$$= r_{t+1} + \gamma(\gamma^0 r_{t+2} + \gamma^1 r_{t+3} + \ldots)$$

$$= \boxed{r_{t+1} + \gamma\sum_{k=0}^{\infty}\gamma^k r_{(t+1)+k+1}}$$

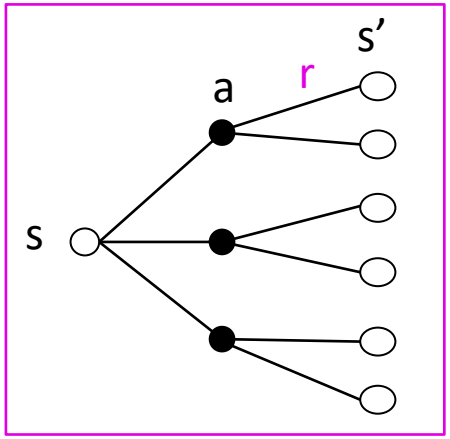
Bellman equation of value function

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\}$$

$$= E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1} \middle| s_t = s\right\}$$

$$= \boxed{E_{\pi}\left\{r_{t+1}\right.} + \gamma\sum_{k=0}^{\infty}\gamma^k r_{t+k+2} \middle| s_t = s\Big\}$$

$$=$$



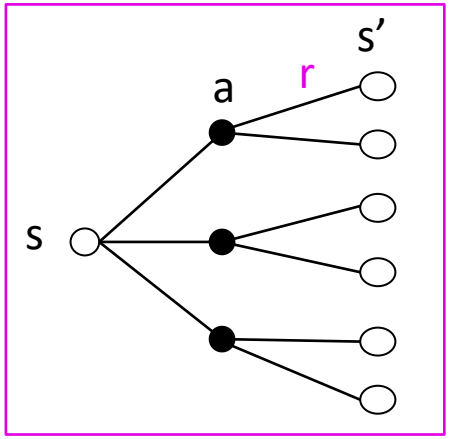
Bellman equation of value function

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\}$$

$$= E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1} \middle| s_t = s\right\}$$

$$= \boxed{E_{\pi}\left\{r_{t+1}\right.} + \gamma\sum_{k=0}^{\infty}\gamma^k r_{t+k+2} \middle| s_t = s\Big\}$$

$$= \boxed{\sum_{a}\pi(a \mid s)\sum_{s'}p(s' \mid s,a)\Big[E(r_{t+1} \mid s,a,s')} + \gamma E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+2} \middle| s_{t+1} = s'\right\}\Big]$$



Bellman equation of value function

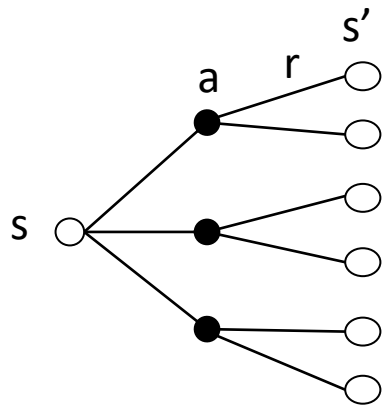
$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\}$$

$$= E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1} \,\middle|\, s_t = s\right\}$$

$$= E_{\pi}\left\{r_{t+1} + \gamma\sum_{k=0}^{\infty}\gamma^k r_{t+k+2} \,\middle|\, s_t = s\right\}$$

$$= \sum_a \pi(a \mid s)\sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k r_{(t+1)+k+1} \,\middle|\, s_{t+1} = s'\right\}\right]$$

$$= \sum_a \pi(a \mid s)\sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V^{\pi}(s')\right]$$



Optimal policy

- $\pi \geq \pi'$ iff. $V^{\pi}(s) \geq V^{\pi'}(s)$ for every state s
- Optimal policies π^* share optimal state-value function V^* (or action-value function Q^*)

$$V^*(s) = \max_{\pi} V^{\pi}(s), \forall s \in S$$

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a), \forall s \in S, a \in A$$

$$Q^*(s,a) = E\left\{ r_{t+1} + \gamma V^*(s_{t+1}) \middle| s_t = s, a_t = a \right\}$$

Bellman optimality equation for V*

$$\boxed{V^*(s) = \max_{a \in A(s)} Q^{\pi^*}(s,a)}$$

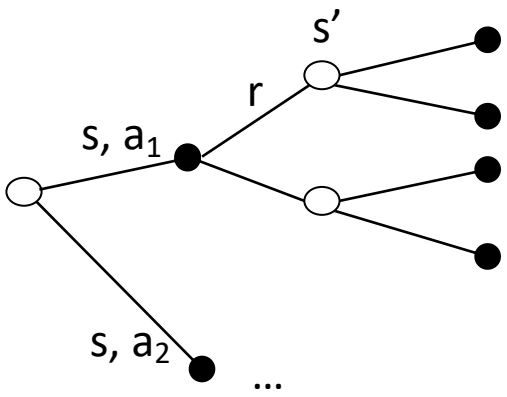
$$= \max_a E_{\pi^*}\left\{R_t \middle| s_t = s, a_t = a\right\}$$

$$= \max_a E_{\pi^*}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s, a_t = a\right\}$$

$$= \max_a E_{\pi^*}\left\{r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \middle| s_t = s, a_t = a\right\}$$

$$= \max_a E\left\{r_{t+1} + \gamma V^*(s_{t+1}) \middle| s_t = s, a_t = a\right\}$$

$$= \max_a \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \middle| s,a,s') + \gamma V^*(s')\right]$$



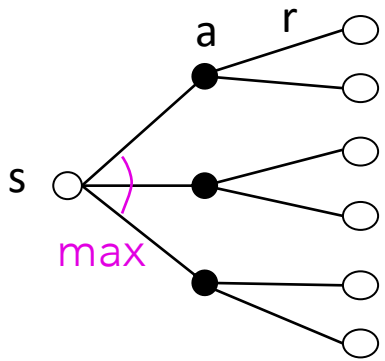
Bellman optimality equation for Q*

$$Q^*(s,a) = E\left\{ r_{t+1} + \gamma \max_a Q^*(s_{t+1}, a') \middle| s_t = s, a_t = a \right\}$$

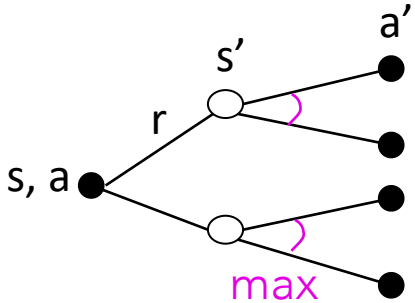
$$= \sum_{s'} p(s' \mid s,a) \left[E(r_{t+1} \middle| s,a,s') + \gamma \max_a Q^*(s',a') \right]$$

Bellman optimality equations

$$V^*(s) = \max_a E_{\pi^*}\left\{r_{t+1} + \gamma V^*(s_{t+1}) \middle| s_t = s, a_t = a\right\}$$
$$= \max_a \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \middle| s,a,s') + \gamma V^*(s')\right]$$



$$Q^*(s,a) = E\left\{r_{t+1} + \gamma \max_a Q^*(s_{t+1}, a') \middle| s_t = s, a_t = a\right\}$$
$$= \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \middle| s,a,s') + \gamma \max_a Q^*(s',a')\right]$$



Solving MDP

- Have a probability model
- Dynamic programming
  - Policy iteration
  - Value iteration

Backward value iteration

$$G_k^*(x_k) = \min_{u_k,\ldots,u_K} \left\{ \sum_{i=k}^{K} l(x_i, u_i) + l_F(x_F) \right\}$$

$G^*_{k+1}(x_{k+1})$

$$G_k^*(x_k) = \min_{u_k} \left\{ l(x_k, u_k) + \boxed{\min_{u_{k+1},\ldots,u_K} \left\{ \sum_{i=k+1}^{K} l(x_i, u_i) + l_F(x_F) \right\}} \right\}$$

$$G_k^*(x_k) = \min_{u_k} \left\{ l(x_k, u_k) + \boxed{G_{k+1}^*(x_{k+1})} \right\}$$

Any length optimal planning

$$G^*(x) = \min_u \left\{ l(x,u) + G^*(f(x,u)) \right\}$$

Optimal cost to go

$$u^* = \underset{u \in U(x)}{\operatorname{argmin}} \left\{ l(x,u) + G^*(f(x,u)) \right\}$$

Optimal action

Policy iteration

- Policy evaluation
- Policy improvement

Policy iteration

Policy evaluation

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \ldots \xrightarrow{I} \pi^* \xrightarrow{E} V^*$$

Policy improvement
"greedification"

Iterative policy evaluation

Estimate values of states following a certain policy

- Initialize $V_0(s) = 0$ for all states s in S
- Update rule based on Bellman equation:

$$V_{k+1}(s) = E_\pi\left\{r_{t+1} + \gamma V_k(s_{t+1}) \middle| s_t = s\right\}$$

$$= \sum_a \pi(a \mid s) \sum_{s'} p(s' \mid s, a)\left[E(r_{t+1} \mid s, a, s') + \gamma V_k(s')\right]$$

$$V^\pi(s) = E_\pi\left\{r_{t+1} + \gamma V^\pi(s_{t+1}) \middle| s_t = s\right\}$$

$$= \sum_a \pi(a \mid s) \sum_{s'} p(s' \mid s, a)\left[E(r_{t+1} \mid s, a, s') + \gamma V^\pi(s')\right]$$

- Repeat until values converge for all states
- Output $V \approx V^\pi$

Policy improvement

Given two deterministic policies π & π′:

If $Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s)$, for all s in S

Then $V^{\pi'}(s) \geq V^{\pi}(s)$, for all s in S

While following policy π, if changing one action according to π′ yields higher or equal return then π′ must be better or as good as π.

Policy improvement

Find new greedy policy π′ given policy π:

$$\pi'(s) = \underset{a}{\operatorname{argmax}}\, Q^{\pi}(s,a)$$

$$= \underset{a}{\operatorname{argmax}}\, E\left\{ r_{t+1} + \gamma V^{\pi}(s_{t+1}) \middle| s_t = s, a_t = a \right\}$$

$$= \underset{a}{\operatorname{argmax}} \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V^{\pi}(s') \right]$$

If $V^{\pi} = V^{\pi'}$

$$V^{\pi'}(s) = \max_{a} E_{\pi^*}\left\{ r_{t+1} + \gamma V^{\pi'}(s_{t+1}) \middle| s_t = s, a_t = a \right\}$$

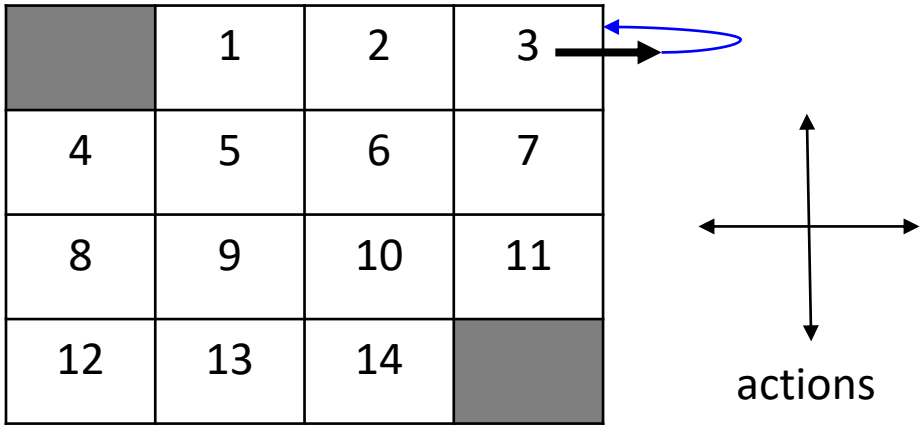
$$= \max_{a} \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V^{\pi'}(s') \right]$$

Bellman optimality equation

Policy improvement must result in a strictly better policy unless the original policy is already optimal.

Example: gridworld

[Sutton & Barto 2nd edition, Example 4.1]



- R_t = -1 on all transitions
- γ=1 (no discount)
- State transition: deterministic (no fail).
- Actions that would result in outside the grid put the agent back to the original state.

Exercise: Convergence of iterative policy evaluation

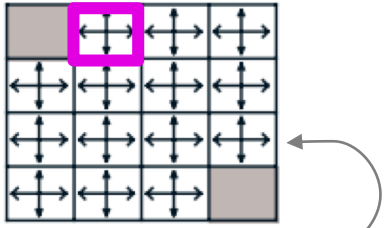
V_k for random policy

Greedy policy w.r.t. V_k

[Sutton & Barto 2nd edition, Fig. 4.1]

k=0

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0



Random policy

k=1

$$V_{k+1}(s) = E_\pi\left\{r_{t+1} + \gamma V_k(s_{t+1}) \middle| s_t = s\right\}$$

$$= \sum_a \pi(a \mid s) \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V_k(s')\right]$$

$$\pi'(s) = \operatorname*{argmax}_a Q^\pi(s,a)$$

$$= \operatorname*{argmax}_a E\left\{r_{t+1} + \gamma V^\pi(s_{t+1}) \middle| s_t = s, a_t = a\right\}$$

$$= \operatorname*{argmax}_a \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V^\pi(s')\right]$$

$V_1(s_1)$ = up(.25*(-1+0)) + down( ) + left( ) + right( ) = _____

$\pi(s_1)$ = argmax_a[up (-1 + (-1)), down ( ), left ( ), right ( )] =_____

Convergence of iterative policy evaluation

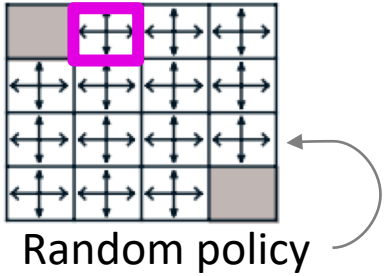
V_k for random policy

k=0

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

k=1

Greedy policy w.r.t. V_k



[Sutton & Barto 2nd edition, Fig. 4.1]

$$V_{k+1}(s) = E_\pi\left\{r_{t+1} + \gamma V_k(s_{t+1}) \middle| s_t = s\right\}$$

$$= \sum_a \pi(a \mid s) \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V_k(s')\right]$$

$$\pi'(s) = \operatorname*{argmax}_a Q^\pi(s,a)$$

$$= \operatorname*{argmax}_a E\left\{r_{t+1} + \gamma V^\pi(s_{t+1}) \middle| s_t = s, a_t = a\right\}$$

$$= \operatorname*{argmax}_a \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V^\pi(s')\right]$$

$V_1(s_1)$ = up(.25*(-1+0)) + down(.25*(-1+0)) + left(.25*(-1+0)) + right(.25*(-1+0))=-1

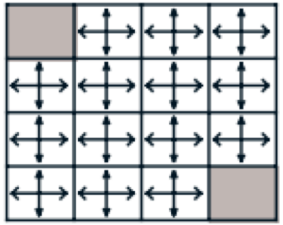
$\pi(s_1)$ = argmax_a[up(-1-1), down(-1-1), left(-1+0),right(-1,-1)] = left

Convergence of iterative policy evaluation

V_k for random policy | Greedy policy w.r.t. V_k | [Sutton & Barto 2nd edition, Fig. 4.1]

k=0

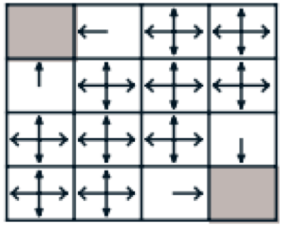
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0



Random policy

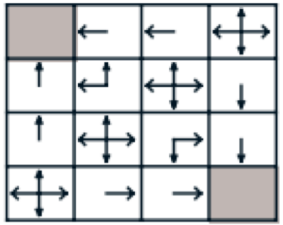
k=1

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0



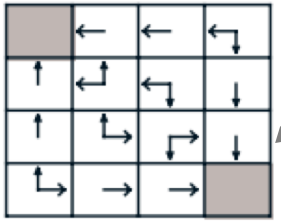
k=2

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0



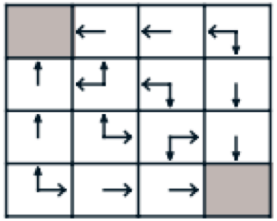
k=3

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0



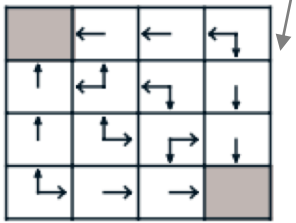
k=10

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0



k=∞

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0



Optimal policy

Policy iteration

Policy evaluation

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \ldots \xrightarrow{I} \pi^* \xrightarrow{E} V^*$$

Policy improvement
"greedification"

Policy iteration

1. Initialize for all s in S: $V(s) \in \Re, \pi(s) = A(s)$
2. Policy evaluation
   Repeat until values converge:

$$V_{k+1}(s) = \sum_a \pi(a \mid s) \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V_k(s')\right], \forall s \in S$$

3. Policy improvement:

$$\pi'(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V^{\pi}(s')\right], \forall s \in S$$

   – If policy stays stable stop; else go to 2

Value iteration

- Use Bellman optimality equation as update (or backup) function

$$V_{k+1}(s) = \max_a \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V_k(s')\right], \forall s \in S$$

Value iteration

- Initialize V(s)=0 for all s in S
- Repeat until values converge:

$$V_{k+1}(s) = \max_a \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V_k(s')\right], \forall s \in S$$

- Output policy

$$\pi(s) = \operatorname*{argmax}_a \sum_{s'} p(s' \mid s,a)\left[E(r_{t+1} \mid s,a,s') + \gamma V_k(s')\right], \forall s \in S$$

Summary

- Reinforcement learning and MDP formulation
- Dynamic programming for solving MDP
- Curse of dimensionality: number of states grow exponentially in the number of state variables
- Next
  - Exploration vs. exploitation
  - Reward shaping