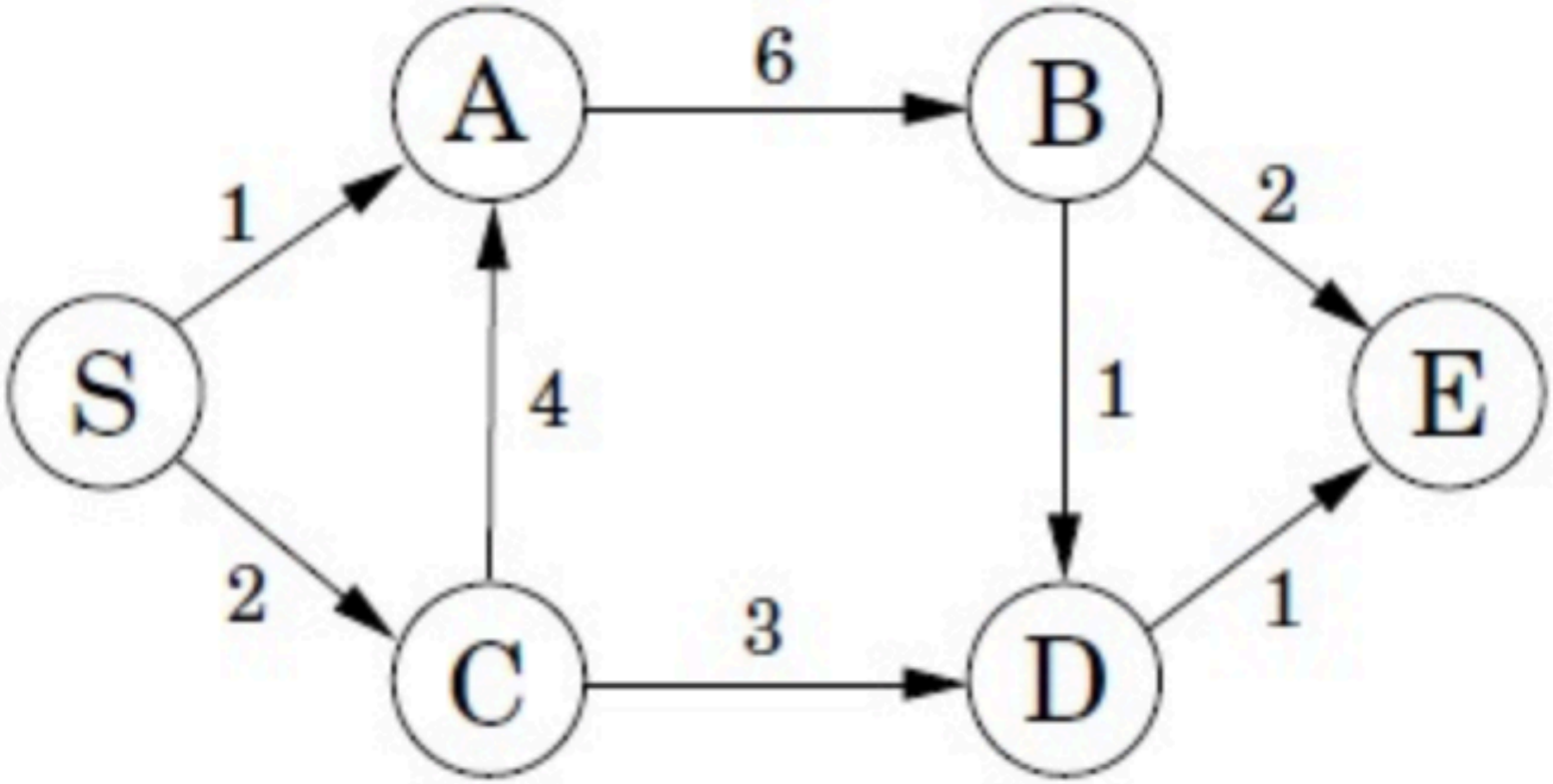
- The update:

  $J \leftarrow TJ$

  is called value iteration.
- We have shown: $T^K V$ gives the optimal values (e.g., cost to go) in a dynamic programming problem with K steps.
- We have discussed how, with a little bit of work, you can recover all the actions you need to take.
- **Claim:** the update $J \leftarrow TJ$ always converges to some J_{lim}. The limit $J_{\text{lim}}$is the vector of optimal values for the infinite-cost reinforcement learning problem.

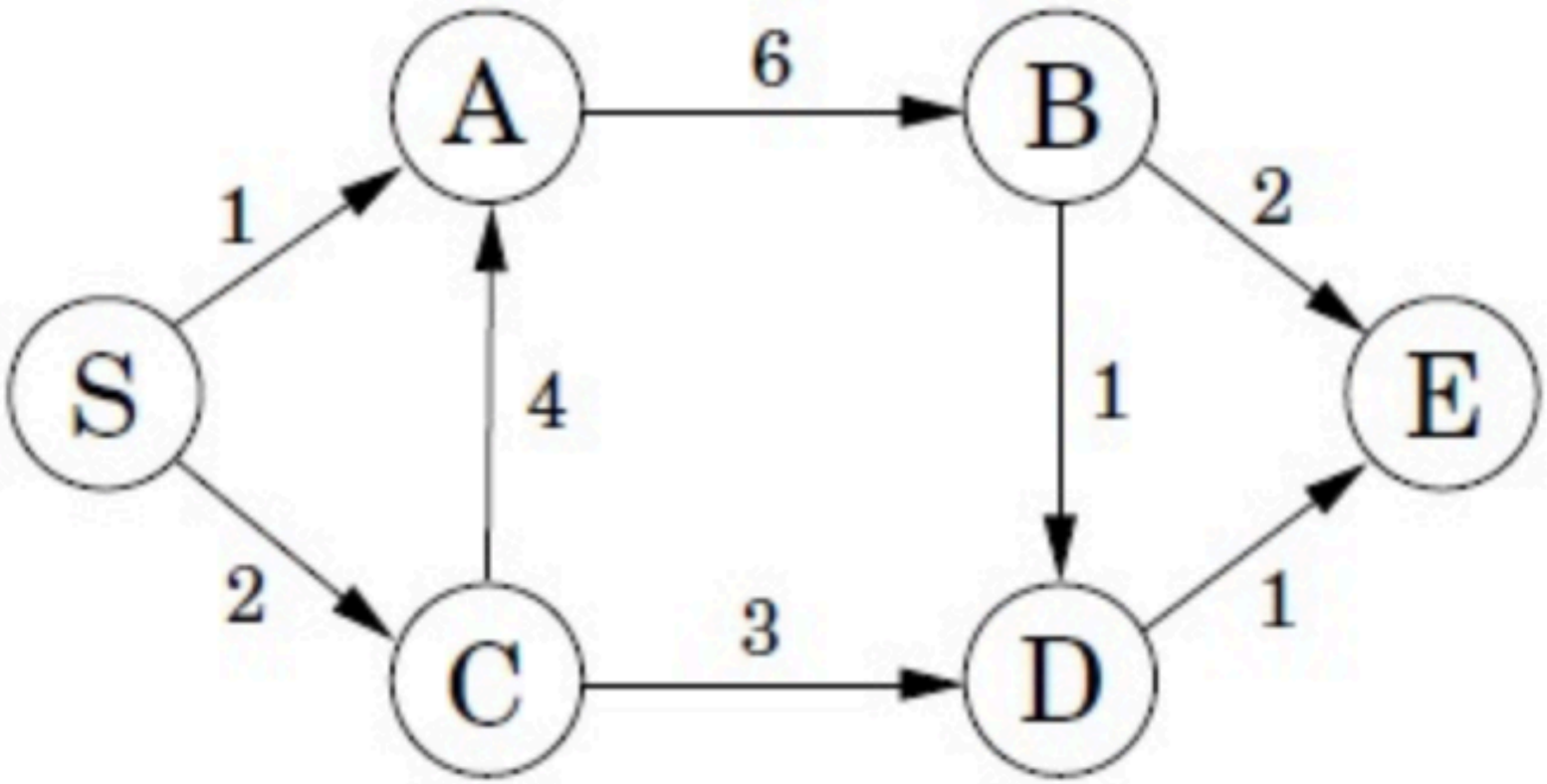
  The actions can be recovered by, for each state s, looking at which action achieves the maximum in the definition of $TJ(s)$.
- ....you will code this on your homework 1.



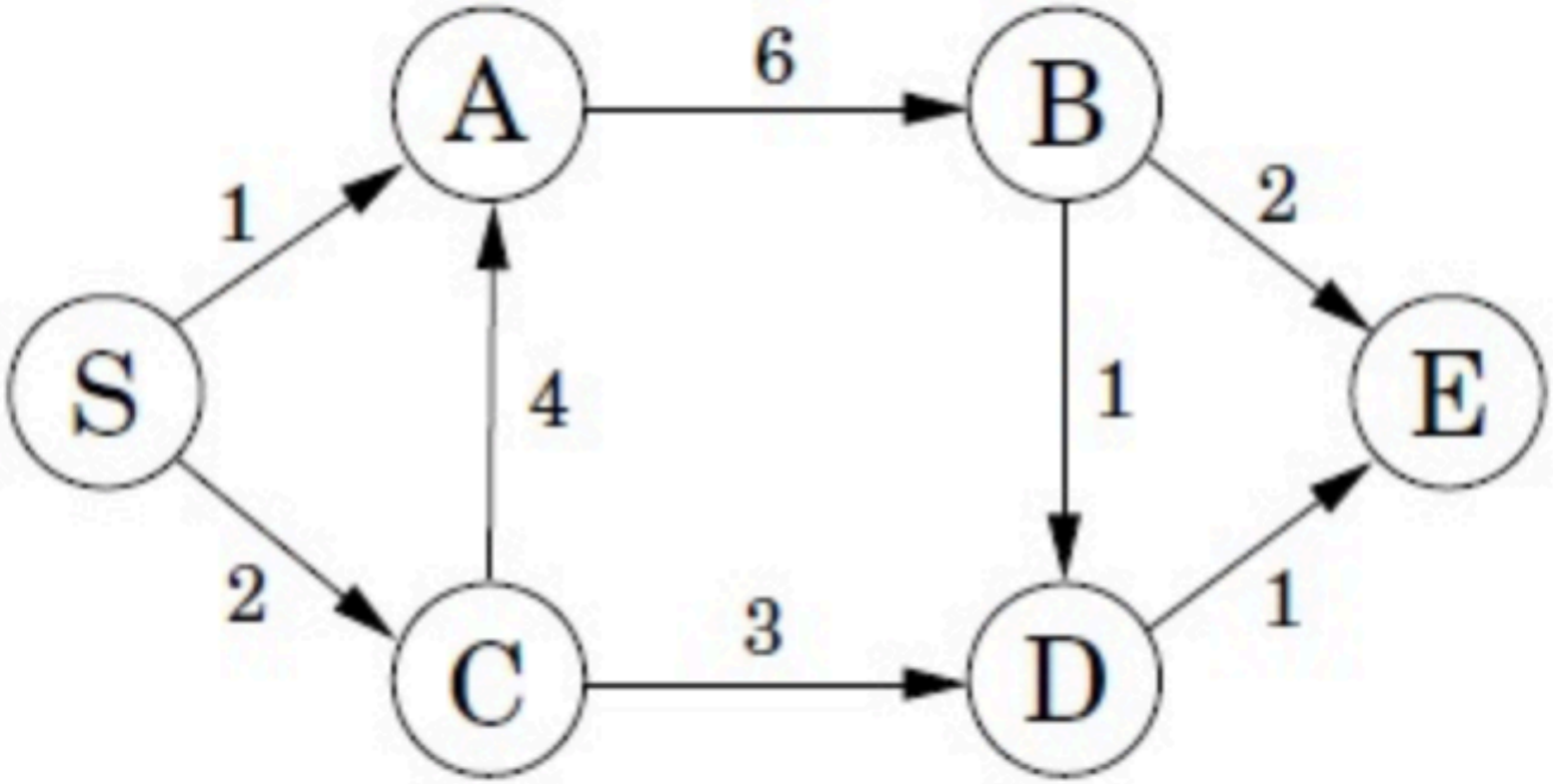
Let's go back here.

Suppose you are at B.

Does it help to be able to toss a coin, and go to E if it comes up on heads and D if it comes up tails?



Now suppose the discount factor equals 1/2.
Suppose you are at B.
Does it help to be able to toss a coin, and go to E if it comes up on heads and D if it comes up tails?



Now suppose you are at C, under either of the two discount factors.
Does tossing a coin help?

- So

$$T^K J = \min_{\pi \in \Pi_{\text{det}}} E\left[r_0 + \gamma r_1 + \cdots + \gamma^{K-1} r_{K-1} + \gamma^K J(s_K)\right]$$

- But also $T^K J = \min_{\pi \in \Pi_{\text{rand}}} E\left[r_0 + \gamma r_1 + \cdots + \gamma^{K-1} r_{K-1} + \gamma^K J(s_K)\right]$
- Let's look again at the definition of T:

$$(TV)(s) = \max_a r(s,a) + \gamma \sum_{s'} P(s' \mid s, a) V(s')$$

- Recall that a linear map is of the form $V \to MV$ for some matrix M.
- An affine map is of the form $V \to V_0 + MV$ for some matrix M and some vector V_0.
- However, the operator T is neither linear nor affine, because of the max in the definition.

- It might have occurred to you that “Dynamic Programming” is a weird name for all this.
- The origins of all the material in this class date back to work done by Richard Bellman in the 1950s at the RAND corporation.
- “An interesting question is, ‘Where did the name, dynamic programming, come from?’
  The 1950s were not good years for mathematical research. We had a very interesting gentleman in Washington named Wilson. He was Secretary of Defense, and he actually had a pathological fear and hatred of the word, research. I’m not using the term lightly; I’m using it precisely. His face would suffuse, he would turn red, and he would get violent if people used the term, research, in his presence. You can imagine how he felt, then, about the term, mathematical.
  The RAND Corporation was employed by the Air Force, and the Air Force had Wilson as its boss, essentially. Hence, I felt I had to do something to shield Wilson and the Air Force from the fact that I was really doing mathematics inside the RAND Corporation. What title, what name, could I choose? In the first place I was interested in planning, in decision making, in thinking. But planning, is not a good word for various reasons. I decided therefore to use the word, ‘programming.’ I wanted to get across the idea that this was dynamic, this was multistage, this was time-varying—I thought, let’s kill two birds with one stone. Let’s take a word that has an absolutely precise meaning, namely dynamic, in the classical physical sense. It also has a very interesting property as an adjective, and that is it’s impossible to use the word, dynamic, in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. It’s impossible.
  Thus, I thought dynamic programming was a good name. It was something not even a Congressman could object to. So I used it as an umbrella for my activities.”
- https://mathshistory.st-andrews.ac.uk/Biographies/Bellman/

- Next, let's define a closely related operator.
- Given any deterministic policy π, we define

$$(T_\pi V)(s) = r(s, \pi(s)) + \gamma \sum_{s'} P(s' \mid s, \pi(s)))V(s')$$

- Interpretation: $(T_\pi V)(s)$ is the expected cost in a one-stage dynamic programming with terminal reward V problem **when you follow policy π starting from state s.**
- Given a randomized policy which choose action a in state s with probability $\pi(a \mid s)$, define

$$(T_\pi V)(s) = E\left[r(s, \pi(s)) + \gamma \sum_{s'} P(s' \mid s, \pi(s)))V(s') \right]$$

  where the expectation is taken with respect to the randomness of the policy π.
- Alternatively, can write this out explicitly as

$$(T_\pi V)(s) = \sum_a \pi(a \mid s) r(s, a) + \gamma \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a) V(s')$$

- Same interpretation: expected cost in a one-stage dynamic programming problem with terminal reward V when you follow randomized policy π starting from state s.
- Compare to

$$(TV)(s) = \max_a r(s, a) + \gamma \sum_{s'} P(s' \mid s, a) V(s')$$

- Somewhat confusingly, **both** T and T_π are called the Bellman operators.

- Next, let's define a closely related operator.
- Given any deterministic policy π, we define

$$(T_\pi V)(s) = r(s, \pi(s)) + \gamma \sum_{s'} P(s' \mid s, \pi(s)))V(s')$$

- Interpretation: $(T_\pi V)(s)$ is the expected cost in a one-stage dynamic programming with terminal reward V problem **when you follow policy π starting from state s.**
- Given a randomized policy which choose action a in state s with probability $\pi(a \mid s)$, define

$$(T_\pi V)(s) = E\left[r(s, \pi(s)) + \gamma \sum_{s'} P(s' \mid s, \pi(s)))V(s') \right]$$

  where the expectation is taken with respect to the randomness of the policy π.
- Alternatively, can write this out explicitly as

$$(T_\pi V)(s) = \sum_a \pi(a \mid s) r(s, a) + \gamma \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a) V(s')$$

- Of course, I could take the operator T_π and apply it to a vector multiple times.
- $(T_\pi^2 V)(s)$ is the total reward in a 2-stage dynamic programming problem with terminal reward V when you follow policy π starting from state s.
- $T_\pi^n V$ is the cost in an n-stage dynamic programming problem with terminal reward V :

  $$T_\pi^K J = E_\pi \left[r_0 + \gamma r_1 + \cdots + \gamma^{K-1} r_{K-1} + \gamma^K J(s_K) \right],$$

  where the expectation is with respect to the rewards/states obtained by following π starting from state s.
- The arguments go analogously to what we have seen before.

- Given a policy π, consider the problem of computing

  $$E_\pi \left[r_0 + \gamma r_1 + \cdots + \gamma^{K-1} r_{K-1} + \gamma^K J(s_K) \right]$$

  for each initial state s .
- Solution: $T_\pi^K J$.
- Now consider the problem of computing the **infinite** sum

  $$E_\pi \left[r_0 + \gamma r_1 + \cdots + \gamma^{K-1} r_{K-1} + \cdots \right]$$

  This is called **policy evaluation problem**.
- Claim: solution is to update $J \leftarrow T_\pi J$ until convergence. This is called **implementing policy evaluation.**
- In the next couple of lectures, we'll put this on firmer ground.

- Let's pause and discuss where we are.
- We want to solve the fundamental RL problem: we want to figure out the optimal policy in a continuing MDP.
- This is hard. So we simplify the problem by assuming there are K stages.
- Now the problem has a solution, and we figured it out!
- You just take the operator T and apply it K times to the vector of terminal costs.
  If there's no terminal cost, this is the same as the terminal cost vector being a vector of zeros.
- OK, so might guess what's coming!
- To solve the RL problem, approximate it with an K stage problem, and then let K go to $+\infty$.
- This means we need to apply T to the zero vector K times, and then let $K \to \infty$ to obtain the optimal rewards. **Need to argue this process converges to the right solution.**
- Next: we study some fundamental properties of the operator T and develop some aspects of a general theory of operators like T. Once this is done, we'll revisit this argument.
- Punchline: the next bit feels like its a foray into math. But we need it.

- OK, let's discuss this more deeply.
- We need to make an assumption which we'll make throughout this course.

  Assumption: there exists some $M > 0$ such that all rewards lie in $[-M, M]$ with probability one.
- Referred to as the bounded rewards assumption.
- Usually holds, unless we break things by introducing $+\infty$ as a reward.
- Now let's discuss the consequences of this assumption.

- Consider the RL problem (i.e., infinite time horizon).
- Fix a policy π.
- Quite reasonably, let us define

  $J_\pi(s) = E\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots\right]$

  Expectation here is taken with respect to the distribution of the rewards given that you start at node s and follow policy π.

  We refer to this as "the value of node s under π."
- Sometimes, people will write $J_\pi(s) = E_{s,\pi}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots\right]$.

  Other times, the subscripts should be inferred from contexts.
- Clearly, $J_\pi(s) \leq M + \gamma M + \gamma^2 M + \cdots = \dfrac{M}{1-\gamma}$.
- Similarly, $J_\pi(s) \geq -\dfrac{M}{1-\gamma}$.

- Now let's consider the difference between the RL and dynamic programming problems under the bounded reward assumption.
- [BTW, we make the bounded rewards assumption by default from now on...won't even mention it.]
- Same MDP. Only difference is that at one point we stop the game after K steps. At the other point, we let it go on forever.
- So $J_\pi^{\mathrm{RL}}(s) = E\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots\right]$ while

  $J_\pi^{\mathrm{DP(K)}}(s) = E\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots + \gamma^K r_K\right]$
- On any sample run, the difference between them is just the discounted reward from step $K+1$ onward.
- So $|J_\pi^{RL} - J_\pi^{DP(n)}| = E\left|\gamma^{K+1} r_{K+1} + \gamma^{K+2} r_{K+2} + \cdots\right| \leq \gamma^{K+1}\dfrac{M}{1-\gamma}.$
- Conclusion:

  $\lim_{K \to +\infty} |J_\pi^{RL} - J_\pi^{\mathrm{DP(K)}}| = 0$ for all policies π and K .

  Looks good!

- Now let's talk about the optimal value functions.
- Define $J^{\mathrm{RL}}(s) = \max_{\pi \in \Pi_{\mathrm{det}}} E\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots\right]$ while

  $J^{\mathrm{DP(K)}}(s) = \max_{\pi \in \Pi_{\mathrm{det}}} E\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots + \gamma^K r_K\right]$
- For any fixed policy π, the difference between the value functions corresponding to π is at most $\gamma^{K+1}\frac{M}{1-\gamma}$.
- It follows that $|J^{\mathrm{RL}}(s) - J^{\mathrm{DP(K)}}(s)| \leq \gamma^{K+1}\frac{M}{1-\gamma}$
- Conclusion: $\lim_{K \to \infty} |J^{\mathrm{RL}}(s) - J^{\mathrm{DP(K)}}(s)| = 0.$
- Punchline: as expected, dynamic programming approximates RL as $K \to +\infty$.

- Where we are: we want to solve RL.
- We know how to solve dynamic programming.
- But dynamic programming approximates RL as the time horizon goes to infinity!
- So suppose you have a K stage problem with zero terminal cost. We know what the solution is:

  $$J^{*,\mathrm{DP(k)}} = T^K \mathbf{0},$$

  where $\mathbf{0}$ refers to the all-zero vector.
- Conclusion: the optimal rewards in the RL problem can be obtained as

  $$J^* = \lim_{K \to \infty} T^K \mathbf{0}$$

- Also, if you have a concrete policy π and you want to evaluate it, you can then do so by computing $J_\pi = \lim_{t \to \infty} T_\pi^K \mathbf{0}$.
- This is nice, but we should be aware of the things:
  — do you really have to apply the operators an infinite number of times?
  — arguably, what we want is the optimal policy and not the optimal rewards. But clearly the two are connected?

- Summary:

$$(TV)(s) = \max_a \left(r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V(s') \right)$$

$$(T_\pi V)(s) = \sum_a \pi(a \mid s) r(s,a) + \gamma \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s,a) V(s')$$

- So suppose you have a K stage problem with zero terminal cost. We know what the solution is:

  $J^{*,\mathrm{DP(k)}} = T^K \mathbf{0},$

  where $\mathbf{0}$ refers to the all-zero vector. These are the rewards to go.
- The optimal rewards in the RL problem can be obtained as

  $J^* = \lim_{K \to \infty} T^K \mathbf{0}$
- Also, if you have a concrete policy π and you want to evaluate it with termination cost of $\mathbf{0}$, you can then do so by computing

  $J_\pi^{*,\mathrm{DP(k)}} = T_\pi^K \mathbf{0}$

  $J_\pi = \lim_{t \to \infty} T_\pi^K \mathbf{0}.$

- If V_1, V_2 are vectors, will say that $V_1 \leq V_2$ if the inequality holds elementwise. For example, $\begin{pmatrix} 1 \\ 3 \end{pmatrix} \leq \begin{pmatrix} 2 \\ 4 \end{pmatrix}$
- Note that for any two numbers x, y we have that either $x \leq y$ or $y \leq x$ (or both). But the same is not true for vectors.
- For example, $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 3 \end{pmatrix}$ are incomparable.
- **Claim:** If $V_1 \leq V_2$, then $TV_1 \leq TV_2$.
- **Claim:** If $V_1 \leq V_2$, then $T_\pi V_1 \leq T_\pi V_2$ for any policy π.
- Why is this true?
- This is called the **monotonicity lemma.**

- Let us denote by $\mathbf{e}$ the all-ones vector in $\mathbb{R}^n$, e.g., $\mathbf{e} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

- What is $T(V + \mathbf{e})$?
- Let us go back to the definition:

$$(TV)(s) = \max_a r(s,a) + \gamma \sum_{s'} P(s' \mid s, a) V(s')$$

- So

$$(T(V + \mathbf{e}))(s) = \max_a r(s,a) + \gamma \sum_{s'} P(s' \mid s, a)(V(s') + 1)$$

$$= \max_a r(s,a) + \gamma \sum_{s'} P(s' \mid s, a) V(s') + \gamma$$

- Or: $T(V + \mathbf{e}) = TV + \gamma \mathbf{e}$.
- Similarly, we have that $T(V + \alpha \mathbf{e}) = TV + \gamma \alpha \mathbf{e}$
- This is a fundamental property of the Bellman operator.

- Let's play with this some more.
- What is $T^2(V + \mathbf{e})$?
- Well, it is $T(T(V + \mathbf{e}))$
- But using the argument we just made, this is $T(TV + \gamma\mathbf{e})$.
- OK, but this is $T(TV) + \gamma^2\mathbf{e}$.
- Conclusion: $T^2(V + \mathbf{e}) = T^2V + \gamma^2\mathbf{e}$.
- Let's go deeper:

  $T^3(V + e) = T(T^2V + \gamma^2\mathbf{e}) = T^3V + \gamma^3\mathbf{e}$.
- More generally, $T^n(V + \mathbf{e}) = T^nV + \gamma^n\mathbf{e}$.

- The same properties hold for the operator T_π.
- Indeed, from the definition:

$$(T_\pi V)(s) = \sum_a \pi(a \mid s) r(s,a) + \gamma \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a) V(s')$$

- So

$$(T_\pi(V + \mathbf{e}))(s) = \sum_a \pi(a \mid s) r(s,a) + \gamma \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a)(V(s') + 1)$$

$$= \sum_a \pi(a \mid s) r(s,a) + \gamma \sum_a \pi(a \mid s) \left(\sum_{s'} P(s' \mid s, a) V(s') + 1 \right)$$

$$= \sum_a \pi(a \mid s) r(s,a) + \gamma + \gamma \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a) V(s')$$

- Thus $T_\pi(V + \mathbf{e}) = T_\pi V + \gamma \mathbf{e}$.
- From this it follows that $T_\pi(V + \alpha\mathbf{e}) = T_\pi V + \gamma\alpha\mathbf{e}$ as before.
- Similarly, $T_\pi^k(V + \mathbf{e}) = T_\pi^k V + \gamma^k \mathbf{e}$.
- This property is called sub-homogeneity.
- Maybe you are thinking: this is all very nice, but where is this going?
- Well, we've established two properties of the operators T, T_π: monotonicity and subhomogeneity.
- ...not done yet. Next, we discuss a further property of these maps: **contraction**.

- The infinity norm of a vector, denoted by $||x||_\infty$:

$$||x||_\infty = \max_i |x_i|$$

- Example: $\left|\left|\begin{pmatrix}1\\2\\3\end{pmatrix}\right|\right|_\infty = 3$ and $\left|\left|\begin{pmatrix}-4\\2\\3\end{pmatrix}\right|\right|_\infty = 4.$
- We have $||x||_\infty = 0$ if and only if $x = 0$.
- We have that if α is a scalar, then $||\alpha x||_\infty = |\alpha| ||x||_\infty$.
- We have that $||x + y||_\infty \leq ||x||_\infty + ||y||_\infty$.
- These properties are usually taken to be the definition of being a norm.

- Definition: the mapping T is a strict contraction in the infinity norm if for any two vectors x, y we have that

  $$||Tx - Ty||_\infty < ||x - y||_\infty$$

- Interpretation: T brings vectors closer together.
- Definition: if $\alpha \in (0,1)$, then the map T is an α-contraction in the infinity norm if

  $$||Tx - Ty||_\infty \leq \alpha ||x - y||_\infty .$$

- Note: we can talk about contractions in any norm, of course. But for RL, we will mostly focus on the infinity norm.

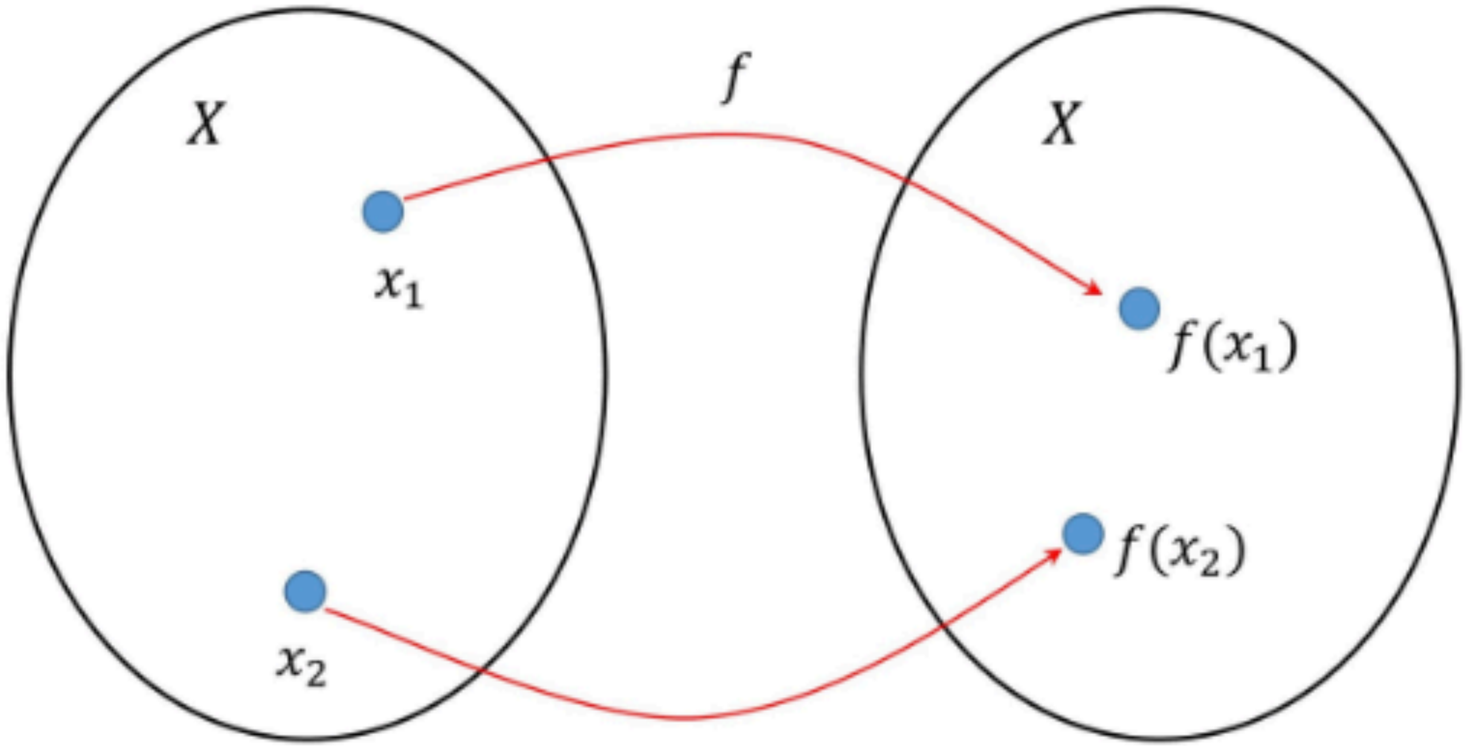


Figure 5.2.1: Contraction mapping $f : X \to X$ shrinks distances between arbitrary two points in a normed space X.

Good image to have in mind.
For us, the space X will always be R^n
And we will measure distances using the infinity norm

- **Claim**: the map T is a γ-contraction in the infinity norm.
- Proof: consider two vectors V_1, V_2.
- Suppose $c = ||V_1 - V_2||_\infty$. Want to argue that

  $$||TV_1 - TV_2||_\infty \leq \gamma c.$$
- We have that

  $$V_1 - c\mathbf{e} \leq V_2 \leq V_1 + c\mathbf{e}$$
- Apply monotonicity: $T(V_1 - c\mathbf{e}) \leq T(V_2) \leq T(V_1 + c\mathbf{e})$
- Apply subhomogeneity: $T(V_1) - c\gamma\mathbf{e} \leq T(V_2) \leq T(V_1) + \gamma c\mathbf{e}$
- Conclude: $||T(V_2) - T(V_1)||_\infty \leq \gamma c$. Done!
- **This also implies T_π is a γ-contraction**. Why?

- OK, but this gives us a new perspective on the claims that $J_\pi = \lim_{K\to\infty} T_\pi^K \mathbf{0}$ and $J^* = \lim_{K\to\infty} T^K \mathbf{0}$.
- For any two vectors $V_1, V_2,$ we have that $||TV_1 - TV_2|| \leq \gamma ||V_1 - V_2||_\infty$.
- OK, but this means $||T^2V_1 - T^2V_2||_\infty = ||T(TV_1) - T(TV_2)||_\infty \leq \gamma ||TV_1 - TV_2||_\infty \leq \gamma^2 ||V_1 - V_2||_\infty$
- Likewise, $||T^KV_1 - T^KV_2||_\infty \leq \gamma^K ||V_1 - V_2||_\infty$.
- In particular, $\lim_{K\to\infty} ||T^KV_1 - T^KV_2||_\infty = 0.$
- Conclusion: $J^* = \lim_{K\to\infty} T^K J$ for any vector J.
- Similarly: $J_\pi = \lim_{K\to\infty} T^K J$ for any vector J.

- Recall: $TJ^* = J^*$ and $T_\pi J_\pi = J_\pi$. These are called the Bellman equations.
- This is good! We now have equations satisfied by the things we are looking for.
- Let's look deeper into this.
- Consider the equation $J = T_\pi J$. Could it have more than one solution?
- More broadly, consider the equation $x = Tx$ where T is an α-contraction where $\alpha \in (0,1)$. Could this equation have more than one solution?
- **Claim:** no.

- Suppose $x_1 = Tx_1$ and $x_2 = Tx_2$.
- $Tx_1 - Tx_2 = x_1 - x_2$
- So $||Tx_1 - Tx_2||_\infty = ||x_1 - x_2||_\infty$.
- But also $||Tx_1 - Tx_2||_\infty \leq \alpha ||x_1 - x_2||_\infty$ with $\alpha \in (0,1)$.
- So $||x_1 - x_2||_\infty \leq \alpha ||x_1 - x_2||_\infty$.
- Conclusion: $||x_1 - x_2||_\infty = 0$ or $x_1 = x_2$.
- Consequences:

  — J_π is the **unique** solution of $T_\pi J = J$.

  — J^* is the unique solution of $TJ = J$

- Let's look at the Bellman equations, starting with the Bellman equation for a policy: $T_\pi J = J$.
- OK, but let's go back to the definition of T_π:

  $$(T_\pi V)(s) = \sum_a \pi(a \mid s) r(s,a) + \gamma \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a) V(s')$$

- Let's write this in matrix form! Let $r_\pi(s) = \sum_a \pi(a \mid s) r(s,a)$ be the expected reward in the next transition when you follow policy π.
- So $(T_\pi V)(s) = r_\pi(s) + \gamma \sum_{s'} \sum_a \pi(a \mid s) P(s' \mid s, a) V(s')$
- Further, let us set $P_\pi(s' \mid s) = \sum_a \pi(a \mid s) P(s' \mid s, a)$ to be the probability of transitioning from s to s' if you follow policy π.
- So $(T_\pi V)(s) = r_\pi(s) + \gamma \sum_{s'} P_\pi(s' \mid s) V(s')$. What does this look like using linear algebra?

- We write the definition

  $$(T_\pi V)(s) = r_\pi(s) + \gamma \sum_{s'} P(s' \mid s) V(s')$$

  in vector form as

  $$T_\pi V = r_\pi + \gamma P_\pi V$$

  where P_π is the matrix whose entry in row s and column s' is $P_\pi(s' \mid s)$.
- OK so the Bellman equation $T_\pi J = J$ can be written as

  $$r + \gamma P_\pi J = J.$$
- This is a linear system of equations! We can solve it:

  $$r = (I - \gamma P_\pi) J$$

  and therefore

  $$J_\pi = (I - \gamma P_\pi)^{-1} r.$$
- Punchline: we can find the value of any fixed policy by solving a linear system of equations.

- Let’s consider a few simple equations.
- Consider an MDP with one state.

  You have a single action to take, get a reward of 1, and transition to the same/only state.

  Discount factor equals 1/2
- What is the value of this policy? First, let’s compute it directly…
- 1 + (1/2) + (1/2)^2 + (1/2)^3 + … = 2.
- Using the method on the previous slide, the value of the single state is the solution of

  $$V(1) = r(1) + \gamma \cdot V(1)$$

  [here the matrix P_π is just the scalar 1]

  or

  $$V(1) = 1 + \frac{1}{2}V(1)$$

- This does indeed reduce to $V(1) = 2$ as we expect.

- How about an MDP with two states. Given the policy we pursue:
  — at state 1, you get a reward of 2 on each transition, and transition to state 2 with probability 1/2
  — at state 2, you get a reward of 1 on each transition, and transition to state 2 with probability 1.
- The matrix P_π is $P_\pi = \begin{pmatrix} 1/2 & 1/2 \\ 0 & 1 \end{pmatrix}$.
- Bellman eq:

  $T_\pi V = V$

  $T_\pi V = r_\pi + \gamma P_\pi V$
- $$\begin{pmatrix} V(1) \\ V(2) \end{pmatrix} = \begin{pmatrix} r_\pi(1) \\ r_\pi(2) \end{pmatrix} + \gamma \begin{pmatrix} 1/2 & 1/2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} V(1) \\ V(2) \end{pmatrix}$$
- $V(1) = 2 + \gamma \left(\frac{1}{2} V(1) + \frac{1}{2} V(2) \right)$

  $V(2) = 1 + \gamma V(2)$
- Supposing $\gamma = 1/2$, we can get $V(1) = 10/3, V(2) = 2$

- How about an MDP with two states. Given the policy we pursue:
  — at state 1, you get a reward of 2 on each transition, and transition to state 2 with probability 1/2
  — at state 2, you get a reward of 1 on each transition, and transition to state 2 with probability 3/4, and to state 1 with probability 1/4.
- The matrix P_π is $P_\pi = \begin{pmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{pmatrix}$.
- $V(1) = 2 + \gamma \left(\frac{1}{2} V(1) + \frac{1}{2} V(2) \right)$

  $V(2) = 1 + \gamma \left(\frac{1}{4} V(1) + \frac{3}{4} V(2) \right)$
- Supposing $\gamma = 1/2$, we have

$$V = \left(I - \frac{1}{2} \begin{pmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{pmatrix} \right)^{-1} \begin{pmatrix} 2 \\ 1 \end{pmatrix} \approx \begin{pmatrix} 3.43 \\ 2.26 \end{pmatrix}$$

- OK, but that is just the value of following a policy π.

  Would be amazing if we use a similar method to solve for J^*.
- Unfortunately, there the situation is more complicated. As we have already discussed, J^* is the unique solution of the equation $TJ = J$.

  But recall the definition of the operator T:

  $$(TV)(s) = \max_a r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V(s')$$

- It is not linear: it has a max in it. Trying to solve $TJ = J$ directly means solving a nonlinear system of equations, and it is not entirely clear how to do that.
- In fact, a good method is to just pick any J (the zero vector works fine), and construct the sequence $TJ, T^2J, T^3J, \ldots$
- This is called **value iteration**. As we discussed, the limit is J^*. How fast does it converge?

- Summary:

  $$(TV)(s) = \max_a r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V(s')$$

- Value iteration is the update

  $$V_{t+1} = TV_t$$

  It converges to the optimal reward-to-go of an infinite horizon RL problem.

- Sometimes I'll write this as $J_{t+1} = TJ_t$ or $J \leftarrow TJ$ or $V \leftarrow TV$
- That optimal reward to go is V^* and it satisfies the equation

  $$V^* = TV^*.$$

- Let's write it out:

  $$V^*(s) = \max_a r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V^*(s')$$

- Let's think about this equation:

  $$V^*(s) = \max_a r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V^*(s')$$

- It should make sense on a gut level.
- $V^*(s) = E[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots]$ starting from state s under the **best** policy.
- But this is the same as $E[r_0] + \gamma E[r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots]$ (again starting from state s under the best possible policy)
- If you choose action a, $r(s,a)$ is the expectation of the first reward.
- If you choose action a and then follow the best possible policy, $\sum_{s'} P(s' \mid s,a) V^*(s')$ is $E[r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots]$
- So the sum $r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V^*(s')$ is your expected reward, over the entire infinite time horizon, if you choose action a and then follow the optimal policy.
- So the best possible choice of action is the one that maximizes $r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V^*(s')$.
- Think about this logic carefully, because your next homework will test it.

- Suppose value iteration generates the sequence $J_0, J_1, J_2, \ldots$

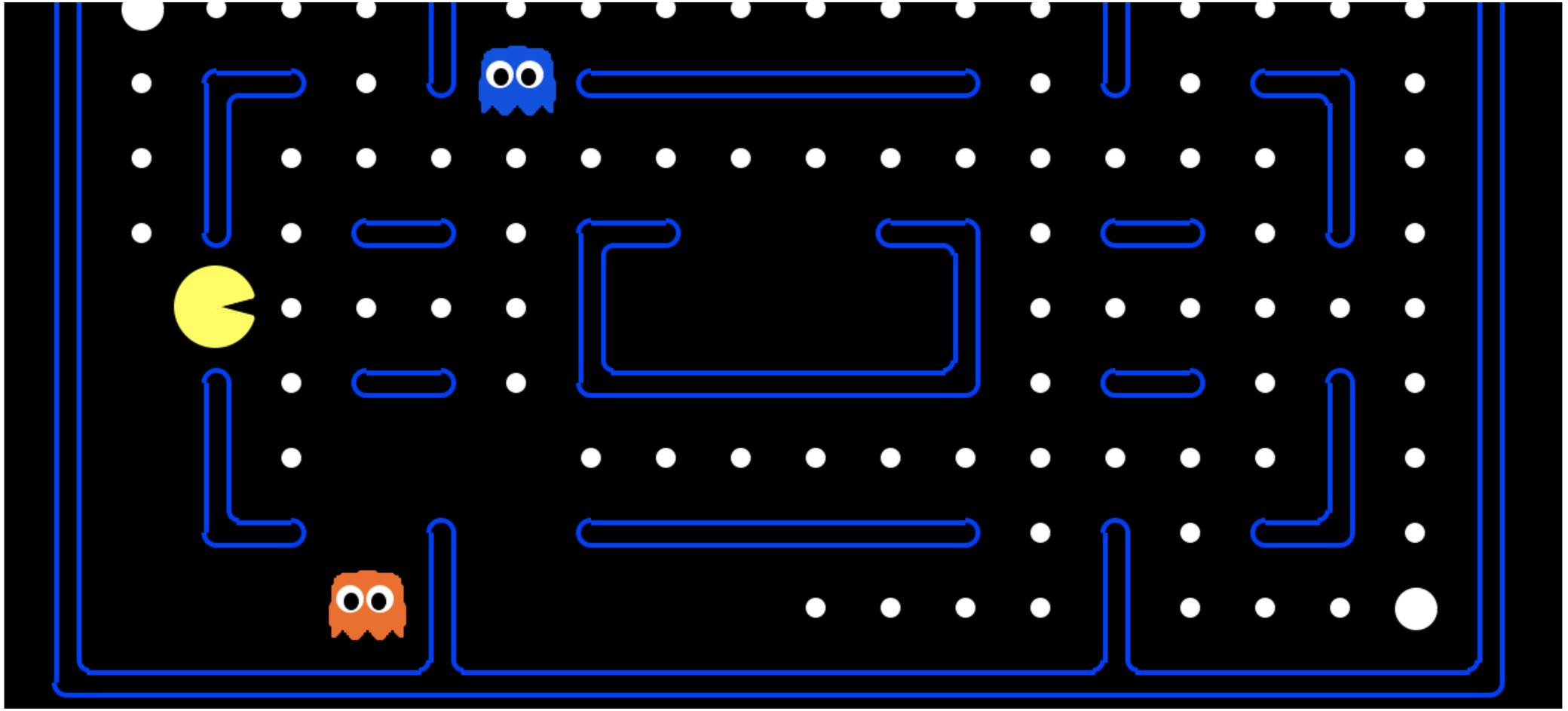
  We have that:

  $J^* = TJ^*$

  $J_{t+1} = TJ_t$
- $J_{t+1} - J^* = TJ_t - TJ^*$
- $||J_{t+1} - J^*||_\infty = ||TJ_t - TJ^*||_\infty \leq \gamma ||J_t - J^*||_\infty$
- So $||J_{t+2} - J^*||_\infty \leq \gamma^2 ||J_t - J^*||$
- So $||J_{t+3} - J^*||_\infty \leq \gamma^3 ||J_t - J^*||$
- $||J_t - J^*||_\infty \leq \gamma^t ||J_0 - J^*||_\infty$.
- Punchline: value iteration converges geometrically as γ^t.
- This is good in general but bad if you choose γ close to one.

- Why aren’t we done with the class?
- Reason number 1: computing J_k or updating $J \leftarrow TJ$ means maintaining a vector with one entry per every state.
- Methods that do this, i.e., maintain a single entry per state, are called *tabular methods*.

  Value iteration and policy iteration are tabular methods.
- But in the real world, the state space is huge!



A state of Pacman is a configuration of:

— location of player

— location of ghosts

— direction of ghosts

— which food is eaten

How many states does Pacman have?

I don't even know how to begin counting but a lot.

Number of states: number of valid input images.

- So the first problem is that we can't use tabular methods. We need methods based on approximation.
- That is, instead of updating

  $J_0, TJ_0, T^2J_0, \ldots$

  we need to deal with **low-dimensional approximations** of these quantities.
- But there's another problem: in the real world, it's a problem to even write down the MDP.
- Think of the self-driving car example: there are so many things to model... (other cars, pedestrians, birds, streets, etc).
- Instead, we need methods that learn the MDP as they go along.
- Next, we will discuss how to solve the *second* problem. After we do that, we'll go back and talk about the first problem.