Offline Reinforcement Learning: Part 1

CS 224R

Course reminders

- Homework 2 due **Wednesday**
- Project proposal feedback coming out soon.

The plan for today

Offline RL: Part 1

1. Why offline RL? Can we just run off-policy methods?
2. Data constraint methods
3. Conservative methods
4. Data stitching

Part of homework 3!

Key learning goals:

- the key challenges arising in offline reinforcement learning
- two approaches for offline RL (& why they work!)
- how offline RL can improve over imitation learning

Why offline RL?

Online RL process (on-policy or off-policy)

Collect data

Update policy on latest data or data so far

Offline RL process

- Given static dataset
- Train policy on provided dataset

Why, or when, might offline RL be more useful?

- leverage datasets collected by people, existing systems
- online policy collection may be risky, unsafe
- reuse previously collected data rather than recollecting
  (e.g. previous experiments, projects, robots, institutions)

Note: A blend of offline then online RL is also possible!

Why offline RL?

Offline RL process

- Given static dataset
- Train policy on provided dataset

More formally:

Offline dataset $\mathcal{D} : \{(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)\}$ sampled from some *unknown policy* π_β

"behavior policy"

$\mathbf{s} \sim d^{\pi_\beta}(\cdot)$

$\mathbf{a} \sim \pi_\beta(\cdot \mid \mathbf{s})$

$\mathbf{s}' \sim p(\cdot \mid \mathbf{s}, \mathbf{a})$

$r = r(\mathbf{s}, \mathbf{a})$

(**Note**: π_β may be a mixture of policies)

Objective: $\max_\theta \sum_t \mathbb{E}_{\mathbf{s}_t \sim d^{\pi_\theta}(\cdot), \mathbf{a}_t \sim \pi_\theta(\cdot \mid \mathbf{s}_t)} \left[r(\mathbf{s}_t, \mathbf{a}_t) \right]$

Why offline RL?

Offline RL process

- Given static dataset
- Train policy on provided dataset

Where does the data come from?

- human collected data
- data from a hand-designed system / controller
- data from previous RL run(s)
- a mixture of sources

Can we just use off-policy algorithms?

Recall: Q-learning objective $\displaystyle\sum_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}} \left\| Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') \right) \right\|^2$

What happens if you optimize this using a static dataset?

(e.g. say data collected by a mediocre policy)

Can we just use off-policy algorithms?

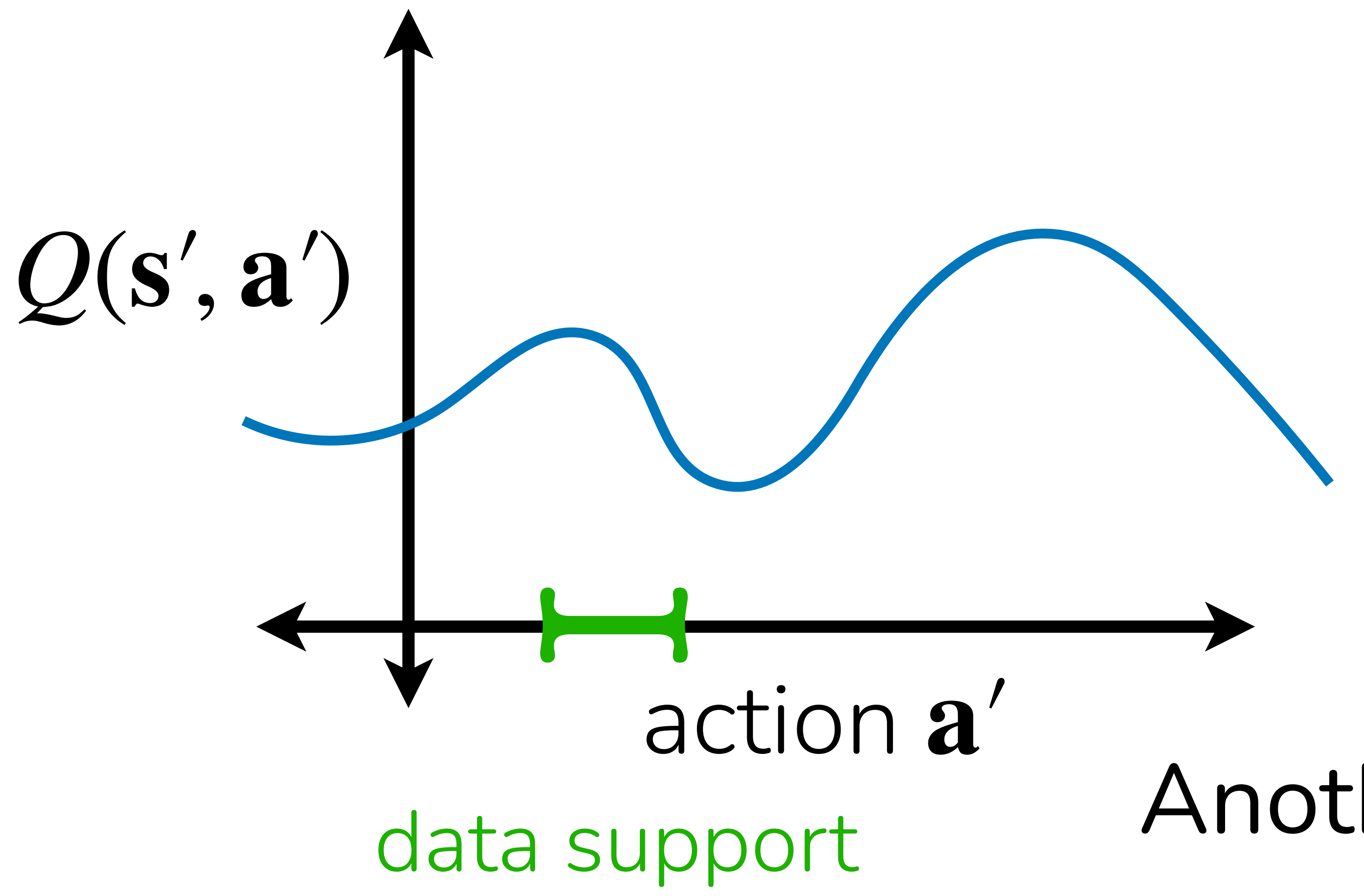
Recall: Q-learning objective $\sum_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}} \left\| Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') \right) \right\|^2$

What happens if you optimize this using a static dataset?

(e.g. say data collected by a mediocre policy)

What happens when evaluating Q on actions $\mathbf{a}'$ not in the dataset?

Randomly init. Q-function for state $\mathbf{s}'$



- Q-function will be unreliable on OOD actions
- $\max$ will seek out actions where Q-function is over-optimistic
- After values propagate, Q-values will become substantially overestimated.

Another perspective: learned policy deviates too much from behavior policy.

How to mitigate overestimation in offline RL?

This is the **core goal** of offline RL methods!

The plan for today

Offline RL: Part 1

1. Why offline RL? Can we just run off-policy methods?
2. **Data constraint methods**
3. Conservative methods
4. Data stitching

Part of homework 3!

How to mitigate overestimation in offline RL?

Recall: Q-learning objective $\sum_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}} \left\| Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') \right) \right\|^2$

Can we constrain $\mathbf{a}'$ to stay close to behavior policy?

If so: we could avoid querying Q on OOD actions!

New objective:

$$\sum_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}} \left\| Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma E_{\mathbf{a}'\sim\pi_{\text{new}}(\cdot|\mathbf{s}')} Q(\mathbf{s}',\mathbf{a}') \right) \right\|^2$$

$$\pi_{\text{new}} = \arg\max_{\pi} E_{\mathbf{a}'\sim\pi(\cdot|\mathbf{s}')} Q(\mathbf{s}',\mathbf{a}') \text{ s.t. } \pi \text{ close to } \pi_\beta$$

How to mitigate overestimation in offline RL?

Can we constrain $\mathbf{a}'$ to stay close to behavior policy?

If so: we could avoid querying Q on OOD actions!

New objective:

$$\sum_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}} \left\| Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma E_{\mathbf{a}'\sim\pi_{\text{new}}(\cdot|\mathbf{s}')} Q(\mathbf{s}',\mathbf{a}')\right)\right\|^2$$

$$\pi_{\text{new}} = \arg\max_{\pi} E_{\mathbf{a}'\sim\pi(\cdot|\mathbf{s}')} Q(\mathbf{s}',\mathbf{a}') \text{ s.t. } \pi \text{ close to } \pi_\beta$$

Issue: We don't know what π_β is!

Many "data constraint" methods will *fit a policy* to the data.

(i.e. learn a proxy for π_β through imitation)

How to mitigate overestimation in offline RL?

Can we constrain $\mathbf{a}'$ to stay close to behavior policy?

If so: we could avoid querying Q on OOD actions!

New objective:

$$\sum_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}} \left\| Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma E_{\mathbf{a}'\sim\pi_{\text{new}}(\cdot|\mathbf{s}')} Q(\mathbf{s}',\mathbf{a}') \right) \right\|^2$$

$$\pi_{\text{new}} = \arg\max_{\pi} E_{\mathbf{a}'\sim\pi(\cdot|\mathbf{s}')} Q(\mathbf{s}',\mathbf{a}') \text{ s.t. } \pi \text{ close to } \pi_\beta$$

Forms of policy constraints?

1. support constraint: $\pi(\mathbf{a}\,|\,\mathbf{s}) > 0$ only if $\pi_\beta(\mathbf{a}\,|\,\mathbf{s}) \geq \epsilon$

   \+ close to what we want - challenging to implement in practice

2. KL divergence: $D_{KL}(\pi\|\pi_\beta)$

   \+ easy to implement - not necessarily what we want

How to implement data constraint methods?

1. Change actor update:

Lagrange multiplier

$$\theta \leftarrow \arg\max_{\theta} E_{\mathbf{s}\sim D, \mathbf{a}\sim\pi_\theta(\cdot|\mathbf{s})}\left[Q(\mathbf{s},\mathbf{a})\right] - \lambda D_{KL}(\pi_\theta \| \pi_\beta)$$

$$\theta \leftarrow \arg\max_{\theta} E_{\mathbf{s}\sim D, \mathbf{a}\sim\pi_\theta(\cdot|\mathbf{s})}\left[Q(\mathbf{s},\mathbf{a}) + \lambda \log \pi_\beta(\mathbf{a}|\mathbf{s}) + \lambda \mathcal{H}\left(\pi_\theta(\cdot|\mathbf{s})\right)\right]$$

Easy to compute for
Gaussian, categorical policies

2. Modify the reward function:

$$\bar{r}(\mathbf{s},\mathbf{a}) = r(\mathbf{s},\mathbf{a}) - D_{KL}(\pi_\theta \| \pi_\beta)$$

Policy will also account for *future* divergence

See: Wu, Tucker, Nachum. *Behavior Regularized Offline RL*. ‘19

The plan for today

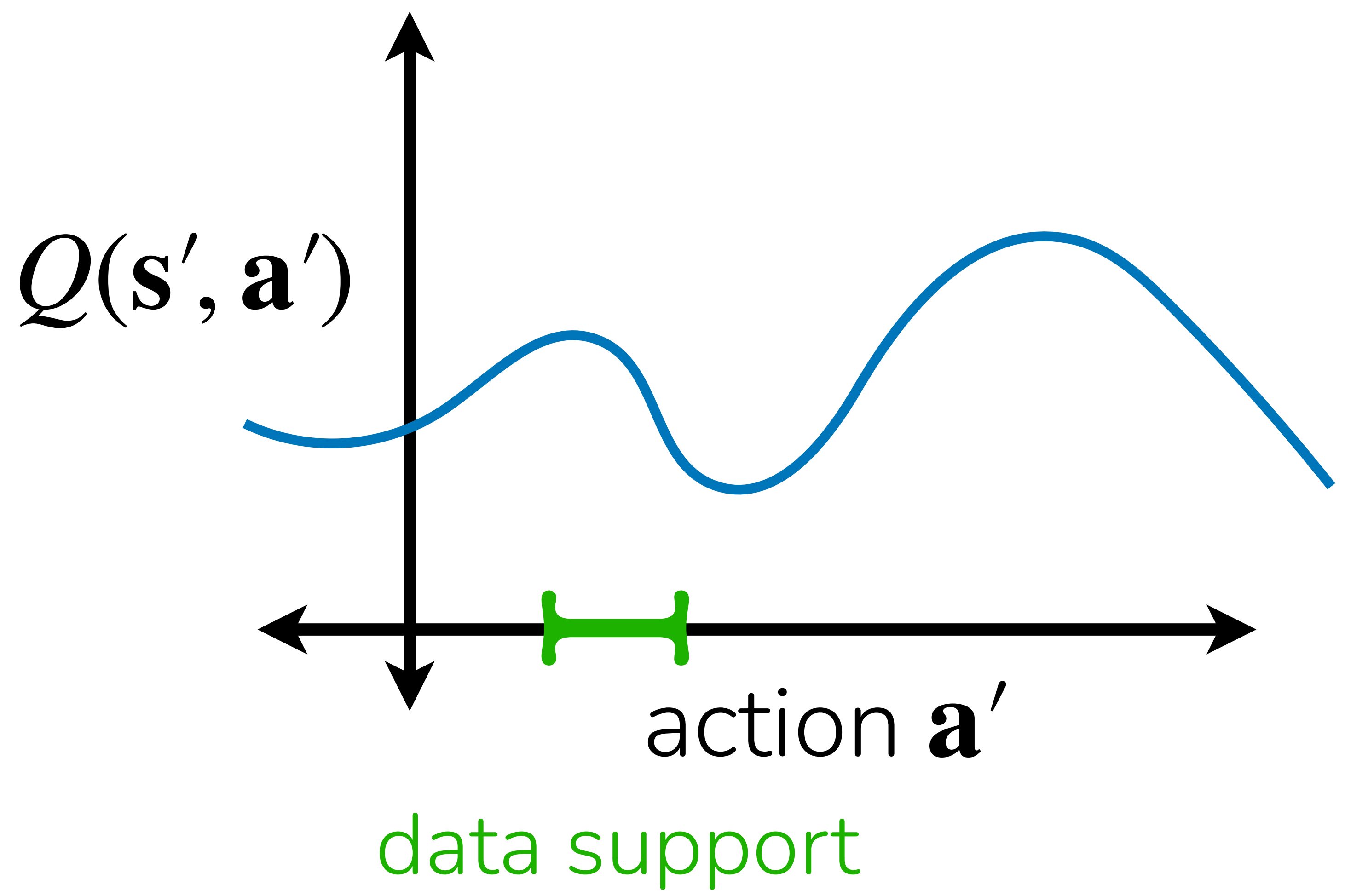
Offline RL: Part 1

1. Why offline RL? Can we just run off-policy methods?
2. Data constraint methods
3. **Conservative methods**
4. Data stitching

Part of homework 3!

How to mitigate overestimation in offline RL?

Recall: Randomly init. Q-function for state $\mathbf{s}'$



Can we discourage overestimation?
without explicitly modeling the behavior policy

What if we just push down on large Q-values?

$$\hat{Q}^\pi = \arg\min_Q \max_\mu E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[\underbrace{\left(Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma E_\pi[Q(\mathbf{s}',\mathbf{a}')]\right)\right)^2}_{\text{standard critic update}}\right] + \underbrace{\alpha E_{\mathbf{s}\sim D, \mathbf{a}\sim\mu(\cdot|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})]}_{\text{push down on large Q-values}}$$

Can show that $\hat{Q}^\pi \leq Q^\pi$ for large enough α

Slide adapted from Sergey Levine

How to mitigate overestimation in offline RL?

Can we discourage overestimation?

without explicitly modeling the behavior policy

$$\hat{Q}^{\pi} = \arg\min_{Q} \max_{\mu} E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\Big[\Big(\underbrace{Q(\mathbf{s},\mathbf{a}) - \big(r(\mathbf{s},\mathbf{a}) + \gamma E_{\pi}[Q(\mathbf{s}',\mathbf{a}')]\big)}_{\text{standard critic update}}\Big)^2\Big] + \underbrace{\alpha E_{\mathbf{s}\sim D,\mathbf{a}\sim\mu(\cdot|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})]}_{\text{push down on large Q-values}}$$

$$- \underbrace{\alpha E_{(\mathbf{s},\mathbf{a})\sim D}[Q(\mathbf{s},\mathbf{a})]}_{\text{push up on Q-values for } (\mathbf{s},\mathbf{a}) \text{ in the data}}$$

No longer guaranteed that $\hat{Q}^{\pi} \leq Q^{\pi}$ for all $(\mathbf{s},\mathbf{a})$.

BUT, guaranteed that $E_{\pi(\mathbf{a}|\mathbf{s})}[\hat{Q}^{\pi}(\mathbf{s},\mathbf{a})] \leq E_{\pi(\mathbf{a}|\mathbf{s})}[Q^{\pi}(\mathbf{s},\mathbf{a})]$ for all $\mathbf{s} \in D$.

Conservative Q-learning (CQL)

Slide adapted from Sergey Levine

How to mitigate overestimation in offline RL?

Conservative Q-learning (CQL)

Full algorithm

1. Update $\hat{Q}^{\pi}$ using L_{CQL} using D

2. Update policy π

If actions are discrete: $\pi(\mathbf{a} \mid \mathbf{s}) = \begin{cases} 1 \text{ if } \mathbf{a} = \arg\max_{\bar{\mathbf{a}}} \hat{Q}(\mathbf{s}, \bar{\mathbf{a}}) \\ 0 \text{ otherwise} \end{cases}$

If actions are continuous: $\theta \leftarrow \theta + \eta \nabla_{\theta} E_{\mathbf{s} \sim D, \mathbf{a} \sim \pi_{\theta}(\cdot \mid \mathbf{s})} \left[\hat{Q}(\mathbf{s}, \mathbf{a}) \right]$

Slide adapted from Sergey Levine

How to mitigate overestimation in offline RL?

Conservative Q-learning (CQL)

1. Update $\hat{Q}^{\pi}$ using L_{CQL} using D
2. Update policy π

How compute objective L_{CQL}?

$$\hat{Q}^{\pi} = \arg\min_{Q} \max_{\mu} E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[\left(Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma E_{\pi}[Q(\mathbf{s}',\mathbf{a}')]\right)\right)^2\right] + \underline{\alpha E_{\mathbf{s}\sim D,\mathbf{a}\sim\mu(\cdot|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})]}$$

$$- \alpha E_{(\mathbf{s},\mathbf{a})\sim D}[Q(\mathbf{s},\mathbf{a})] + \underbrace{R(\mu)}_{\text{regularizer}}$$

Common choice: $R(\mu) = E_{\mathbf{s}\sim D}[\mathcal{H}(\mu(\cdot|\mathbf{s}))]$

With max entropy regularizer R, optimal $\mu(\mathbf{a}\,|\,\mathbf{s}) \propto \exp\left(Q(\mathbf{s},\mathbf{a})\right)$

Then: $\underline{E_{\mathbf{s}\sim D,\mathbf{a}\sim\mu(\cdot|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})]} = \log \sum_{\mathbf{a}} \exp\left(Q(\mathbf{s},\mathbf{a})\right)$

Don't need to construct μ directly.

You will implement CQL in homework 3!

Slide adapted from Sergey Levine

Aside: Model-based offline RL

Key idea: Instead of minimizing Q-values of policy actions,
minimize Q-values of model-generated (s, a)

CQL objective:

state action tuples from model

$$\hat{Q}^{\pi} = \arg\min_{Q}\max_{\mu} E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[\left(Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma E_{\pi}[Q(\mathbf{s}',\mathbf{a}')]\right)\right)^2\right] + \alpha E_{\overset{\rho(\mathbf{s},\mathbf{a})}{\cancel{\mathbf{s}\sim D,\mathbf{a}\sim\mu(\cdot|\mathbf{s})}}}[Q(\mathbf{s},\mathbf{a})] - \alpha E_{(\mathbf{s},\mathbf{a})\sim D}[Q(\mathbf{s},\mathbf{a})]$$

Add data from model to D

Intuition: If model produces data that look clearly different from the real data, it's easy for the Q-function to make it look bad.

The plan for today

Offline RL: Part 1

1. Why offline RL? Can we just run off-policy methods?
2. Data constraint methods
3. Conservative methods
4. **Data stitching**

$\left.\begin{matrix} \\ \\ \\ \end{matrix}\right\}$ Part of homework 3!

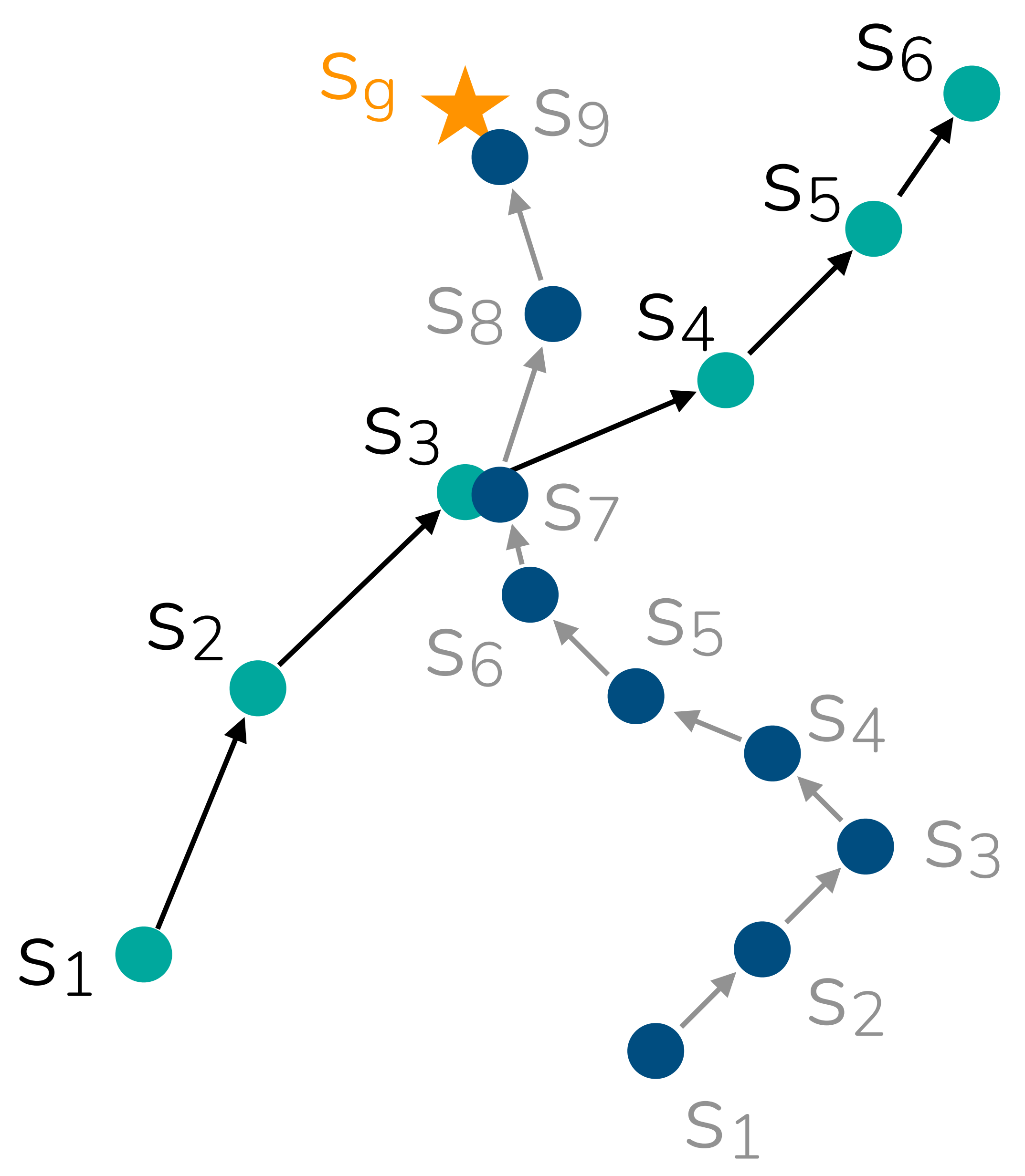
Why offline RL versus imitation learning?

(**Recall**: Imitation methods can't outperform the expert.)

Offline data may not be optimal!

—> Offline RL can leverage reward information to outperform behavior policy.

—> Good offline RL methods can ***stitch*** together good behaviors.



s_1 -> s_3 is good behavior

s_7 -> s_9 is good behavior

Offline RL methods can learn a policy that goes from s_1 to s_9!

The plan for today

Offline RL: Part 1

1. Why offline RL? Can we just run off-policy methods?
2. Data constraint methods
3. Conservative methods
4. Data stitching

} Part of homework 3!

Key learning goals:

- the key challenges arising in offline reinforcement learning
- two approaches for offline RL (& why they work!)
- how offline RL can improve over imitation learning

Summary

Why offline RL? Online data is expensive. *Reusing offline data* is good!

Key challenge: Overestimating Q-values because of shift between π_β and π_θ

- can **explicitly constrain** to the data by modeling π_β

  + fairly intuitive - often too conservative in practice

- implicitly constrain to data by **penalizing Q-values**

  + simple + can work well in practice - need to tune alpha

Trajectory stitching allows offline RL methods to improve over imitation.

Next time: other offline RL approaches & hyperparameter tuning

Course reminders

- Homework 2 due **Wednesday**
- Project proposal feedback coming out soon.