“Convergence”
&
Trust Region Policy Optimization

Lucas Janson and Sham Kakade

CS/Stat 184: Introduction to Reinforcement Learning
Fall 2022

Today

- **Next class: embedded ethics (from Jenna Donohue, a postdoc in the Philosophy dept)**
  - **Course Plan: consider different ethical implications of different possible utility functions for a (fictional) RL algorithm that was setting dynamic prices for rides.**
  - **Please come to the next class. (There will be a discussion.)**
  - **Please do the assigned reading (John Rawls) in advance.**

- **Recap++**

- Today:
  1. Convergence of Fitted Policy Iteration
  2. Trust Region Policy Optimization

Recap + Examples

Is there an iterative version of Policy Evaluation?

(that is faster, but approximate?)

Algorithm (Iterative PE):

1. Initialization: $V^0 : \|V^0\|_\infty \in \left[0, \frac{1}{1-\gamma}\right]$
2. Iterate until convergence: $V^{k+1} \leftarrow R + \gamma P V^k$

   Equivalently,

$$\forall s, V^{k+1}(s) = r(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim P(s,\pi(s))} V^k(s')$$

[Policy Eval Subroutine]: TD Learning for "tabular" case

[Iterative Policy Eval Subroutine/TD]

input: policy π, sample size N

1. Sample trajectories $\tau_1, \ldots \tau_N \sim \rho_\pi$ which gives us a dataset D

   (each trajectory is of the form $\tau_i = \{s_0, a_0, r_0, \ldots s_{H-1}, a_{H-1}, r_{H-1},\}$)
2. For k = $0, \ldots, K$:
   1. Sample a transition $(s_h, r_h, s_{h+1},) \in D$ and update:

   $$V_{k+1}(s_h) = V_k(s_h) - \eta_k \Big(V_k(s_h) - \big(r_h + V_k(s_{h+1})\big) \Big)$$
3. Return the function V_K as an estimate of V^π

Another [Policy Eval Subroutine]:
Fit $V^\pi(s)$ using the iterative policy evaluation alg.

[Iterative Policy Eval Subroutine/TD]

input: policy π, sample size N, init w_0

1. Sample trajectories $\tau_1, \ldots \tau_N \sim \rho_\pi$ which gives us a dataset D
2. For k = $0,\ldots,K$:
   1. Construct an *empirical loss function*:
      $$L_k(w) = \frac{1}{NH} \sum_{i=1}^{N} \sum_{(s_h,r_h,s_{h+1}) \in \tau_i} \Big(f_w(s_h) - \big(r_h + f_{w_k}(s_{h+1}) \big) \Big)^2$$
   2. Update with either:
      full minimization:
      $$w_{k+1} \approx \arg\min_w L_k(w)$$
      TD learning: (one step of SGD)
      $$w_{k+1} = w_k - \eta_k \widetilde{\nabla} L_k(w_k)$$
3. Return the function f_{w_K} as an estimate of V^π

Fitted Dynamic Programming Methods for learning $Q^\star$ and $\pi^\star$

Policy Iteration (PI)

- Initialization: choose a policy $\pi^0 : S \mapsto A$
- For $k = 0,1,\ldots$
  1. **Policy Evaluation**: compute $Q^{\pi^k}(s, a)$
  2. **Policy Improvement**: set

$$\pi^{k+1}(s) := \arg\max_a Q^{\pi^k}(s, a)$$

Fitted Policy Iteration:
(aka Approximate Policy Iteration API)

1. Initialize staring policy π_0, samples size M
2. For k = 0, … :
   1. [Q-Evaluation Subroutine]

      Using M sampled trajectories, $\tau_1, \ldots \tau_N \sim \rho_{\pi_k}$,

      $\widetilde{Q}_k(s,a) \approx Q_h^{\pi_k}(s,a)$
   2. Policy Update

      $\pi_{k+1}(s) := \arg\max_a \widetilde{Q}^{\pi_k}(s,a)$
3. Return $\widetilde{Q}_K$ and π_K as an estimate of $Q^\star$ and $\pi^\star$

Alternative Version: Bellman Operator $\mathcal{T}$ on Q

(HW2 Q2 is the Q-version of the Bellman Equations)

- Given a function $Q : S \times A \mapsto \mathbb{R}$, define $\mathcal{T}Q : S \times A \mapsto \mathbb{R}$ as
$$\big(\mathcal{T}Q\big)(s, a) := r(s, a) + \gamma \mathbb{E}_{s' \sim P(s,a)} \max_{a' \in A} Q(s', a')$$
- (Bellman equations for Q)
Q is equal to $Q^\star$ if and only if $\mathcal{T}Q = Q$.

Q-Value Iteration Algorithm:

1. Initialization: $Q^0 : \|Q^0\|_\infty \in \left[0, \frac{1}{1-\gamma}\right]$
2. Iterate until convergence: $Q_{k+1} \leftarrow \mathcal{T} Q_k$

$$Q_{k+1}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} \max_{a' \in A} Q_k(s',a')$$

The Offline Learning Setting:

We don't know the MDP and our data collection is under some fixed distribution.

The Finite Horizon, Offline Learning Setting:

- We have N trajectories $\tau_1, \ldots \tau_N \sim \rho_{\pi_{data}}$
- π_{data} is often referred to as our data collection policy.

Q-Learning for "tabular" case

[Iterative Policy Eval Subroutine/TD]

input: **offline dataset** $\tau_1, \ldots \tau_N \sim \rho_{\pi_{data}}$

1. For k = $0, \ldots, K$:
   1. Sample a transition $(s_h, a_h, r_h, s_{h+1}) \in D$ and update:
   $$Q_{k+1}(s_h, a_h) = Q_k(s_h, a_h) - \eta_k\Big(Q_k(s_h, a_h) - \big(r_h + \max_{a'} Q_k(s_{h+1}, a')\big)\Big)$$
2. Return the function Q_K as an estimate of $Q^\star$

Fitted Q-Iteration

input: **offline dataset** $\tau_1, \ldots \tau_N \sim \rho_{\pi_{data}}$, init w_0

1. For $k = 0,1,\ldots K$:
   1. Construct an *empirical loss function*:

   $$L_k(w) = \frac{1}{NH} \sum_{i=1}^{N} \sum_{(s_h,a_h,r_h,s_{h+1}) \in \tau_i} \left(f_w(s_h, a_h) - \left(r_h + \max_a f_{w_k}(s_{h+1}, a) \right) \right)^2$$

   2. Update with either:
   full minimization:

   $$w_{k+1} \approx \arg\min_w L_k(w)$$

   Q-learning: (one step of SGD)

   $$w_{k+1} = w_k - \eta_k \widetilde{\nabla} L_k(w_k)$$

2. Return the function $f_{\widetilde{w}_K}$ as an estimate of $Q^\star$

Today:

“Convergence” & Trust Region Policy Optimization

Outline:

1. Convergence of Fitted Policy Iteration
   1. “Tabular” case
   2. Fitted case
2. Trust Region Policy Optimization
   1. Quick intro on KL-divergence
   2. TRPO formulation

Sample Based Policy Iteration in the Tabular Case:

(the easiest case to think about fitted Policy Iteration)

1. For k = 0, … :
   1. Q-Evaluation:

      For each (s, a, h), suppose that:
      1. we are able to draw M trajectories as follows:

         start at $(s_h = s, a_h = a)$, run π_k, and end trajectory at time H
      2. Set $\widetilde{Q}_k(s, a)$ as the empirical average of the cumulative reward on these trajectories.

         (i.e. $\widetilde{Q}_k(s, a)$ is unbiased sample of $Q_h^{\pi_k}(s, a)$, with M samples)
   2. Policy Update

      $\pi_{k+1}(s) := \arg\max_a \widetilde{Q}^{\pi_k}(s, a)$
2. Return $\widetilde{Q}_K$ and π_K as an estimate of $Q^\star$ and $\pi^\star$

[Theorem] Using polynomial many total samples and polynomial computation time (in $|S|, |A|, H, 1/\epsilon$), we have that $\|\widetilde{Q}_K - Q^\star\|_\infty \leq \epsilon$ and $\|Q^{\pi_K} - Q^\star\|_\infty \leq \epsilon$.

Outline:

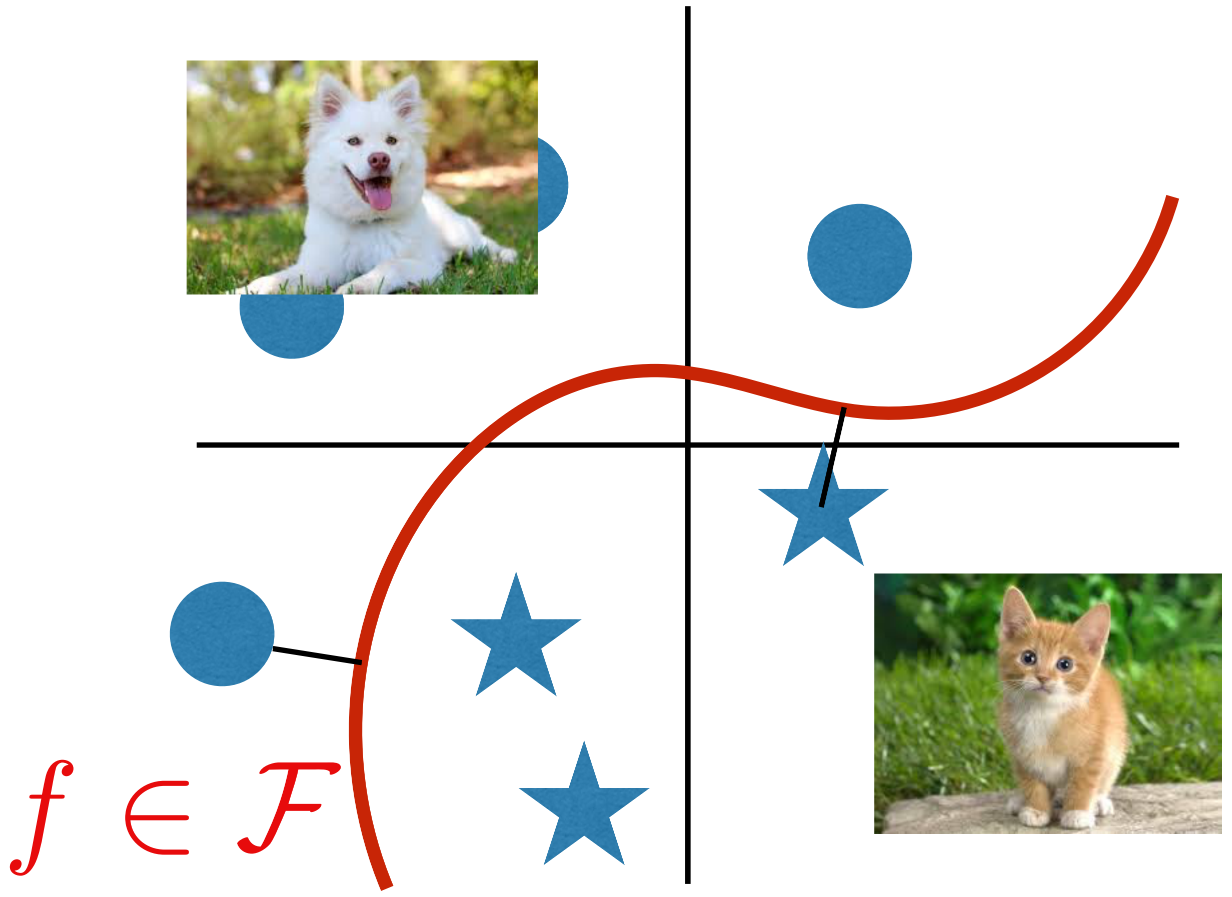
1. Convergence of Fitted Policy Iteration
    1. “Tabular” case
    2. Fitted case
2. Trust Region Policy Optimization
    1. Quick intro on KL-divergence
    2. TRPO formulation

First: let’s summarize a few things about Supervised Learning

Recap on Supervised Learning: Classification

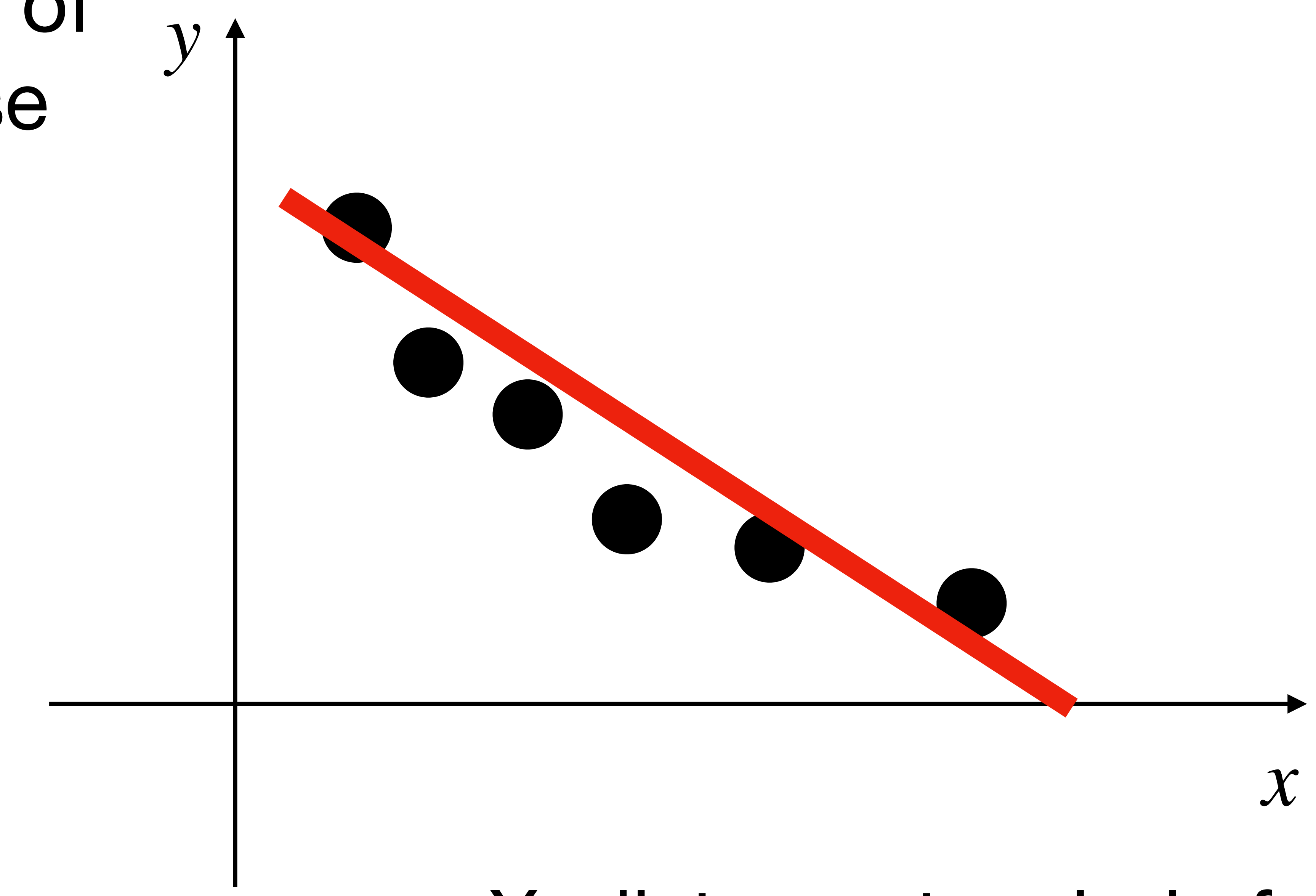
Given i.i.d examples at training:

( ,cat) ( ,cat) ( ,dog)



Using function approximator, we are able to predict on cats/dogs that we **never see before** (i.e., we **generalize**)

Recap on Supervised Learning: Regression



Using function approximation, we are able to predict on the value of some house not from the training data

Recap on Supervised Learning: regression

We have a data distribution $\mathscr{D}$, $x_i \sim \mathscr{D}, y_i = f^\star(x_i) + \epsilon_i$, where noise $\mathbb{E}[\epsilon_i] = 0, |\epsilon_i| \leq c$

We want to approximate $f^\star$ using finite training samples;

Let us introduce an abstract function class $\mathscr{F} = \{f : \mathscr{X} \mapsto \mathbb{R}\}$, and do least squares:

Empirical Risk Minimizer (ERM) $$\hat{f} = \arg\min_{f\in\mathscr{F}} \sum_{i=1}^{N} \left(f(x_i) - y_i\right)^2$$

Q: quality of ERM $\hat{f}$?

Recap on Supervised Learning: regression

We have a data distribution $\mathscr{D}$, $x_i \sim \mathscr{D}, y_i = f^\star(x_i) + \epsilon_i$, where noise $\mathbb{E}[\epsilon_i] = 0, |\epsilon_i| \leq c$

$$\hat{f} = \arg\min_{f \in \mathscr{F}} \sum_{i=1}^{N} \left(f(x_i) - y_i\right)^2$$

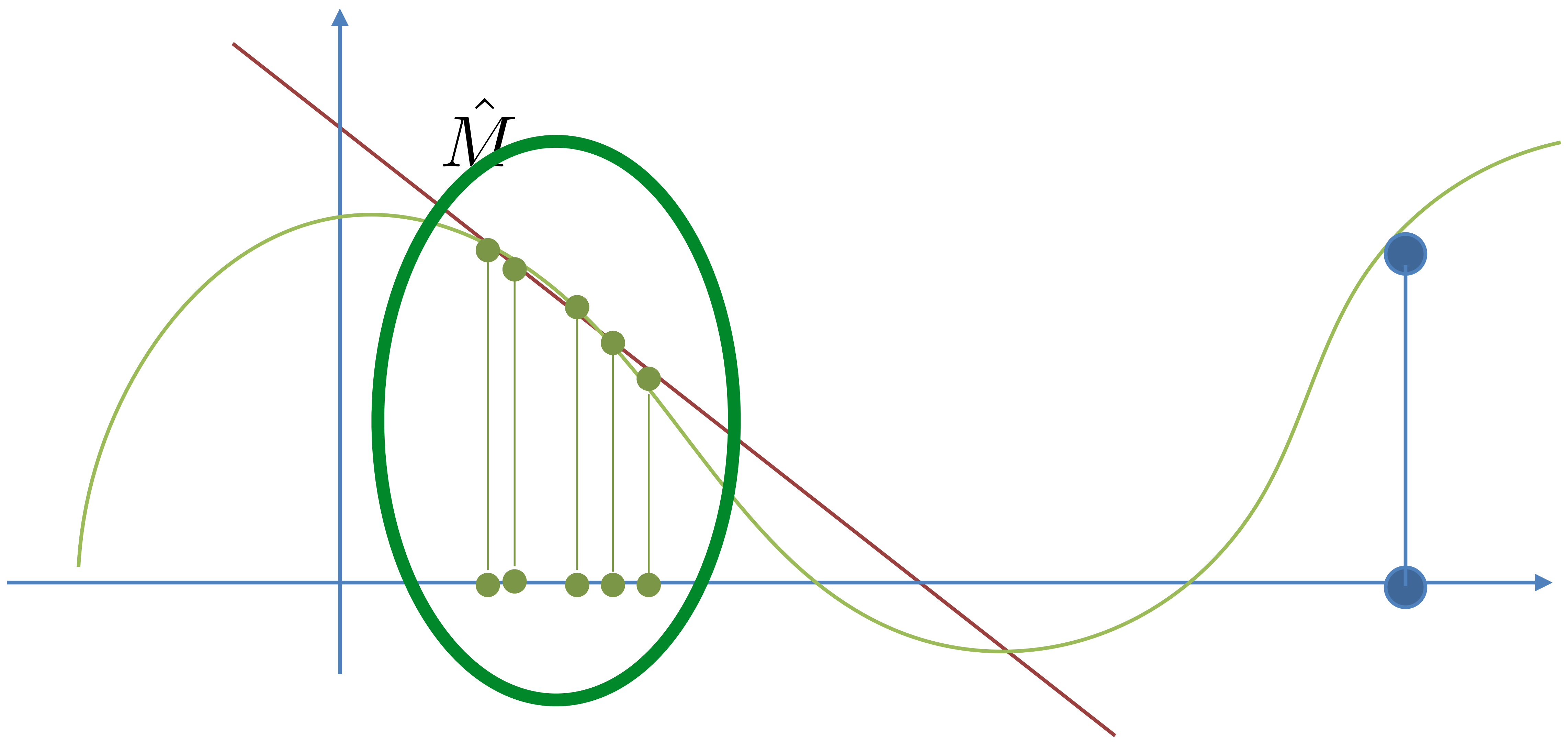
Supervised learning theory (e.g., VC theory) says that we can indeed **generalize**,
i.e., we can predict well **under the same distribution:**

Assume $f^\star \in \mathscr{F}$ (this is called realizability), we can expect:

$$\mathbb{E}_{x \sim \mathscr{D}} \left(\hat{f}(x) - f^\star(x)\right)^2 \leq \delta$$

Supervise Learning can fail if there is train-test distribution mismatch

However, for some $\mathscr{D}' \neq \mathscr{D}$, $\mathbb{E}_{x \sim \mathscr{D}'} (f(x) - f^\star(x))^2$ might be arbitrarily large



Deeper neural nets and larger datasets are typically not enough to address "distribution shift"

Back to RL

Fitted Policy Improvement Guarantees

- For all k, suppose that:

$$E_{\tau \sim \rho_{\pi_k}}\Big[\sum_{h=1}^{H}\big(\widetilde{Q}_k(s_h,a_h) - Q_h^{\pi_k}(s_h,a_h)\big)^2\Big] \le \delta, \text{ and } \max_{s,a}|\widetilde{Q}_k(s_h,a_h) - Q_h^{\pi_k}(s_h,a_h)| \le \delta_\infty$$

- δ: the average case supervised learning error (reasonable to expect this can be made small)

  δ_∞: the worse case error (often unreasonable to expect to be small)

[Theorem:] We have that:

- One step performance degradation is bounded by the worst case error:

$$\widetilde{Q}_{k+1}(s,a) \ge \widetilde{Q}_k(s,a) - 2H\delta_\infty \quad \text{(and equality possible in some examples).}$$

- For large enough K, final performance also governed by the worst case error:

$$Q^{\pi_K}(s,a) \ge Q^\star(s,a) - 2H^2\delta_\infty$$

- (Intuition) If it somehow turns out that, for all iterations k, the density under the next policy, uniformly does not differ from that of previous policy, i.e. that

$$\max_{s,a,h}\left(\frac{\Pr(s_h = s, a_h = a \,|\, \pi_{k+1})}{\Pr(s_h = s, a_h = a \,|\, \pi_k)}\right) \le C_\infty$$

  then we can bound our sub-optimality by the average case error:

$$Q^{\pi_K}(s,a) \ge Q^\star(s,a) - 2H^2 \cdot C_\infty \cdot \delta$$

Outline:

1. Convergence of Fitted Policy Iteration
    1. “Tabular” case
    2. Fitted case
2. Trust Region Policy Optimization
    1. Quick intro on KL-divergence
    2. TRPO formulation

KL-divergence: measures the distance between two distributions

Given two distributions P & Q, where $P \in \Delta(X), Q \in \Delta(X)$,
KL Divergence is defined as:

$$KL(P \mid Q) = \mathbb{E}_{x \sim P}\left[\ln \frac{P(x)}{Q(x)}\right]$$

Examples:

If $Q = P$, then $KL(P \mid Q) = KL(Q \mid P) = 0$

If $P = \mathcal{N}(\mu_1, \sigma^2 I), Q = \mathcal{N}(\mu_2, \sigma^2 I)$, then $KL(P \mid Q) = \|\mu_1 - \mu_2\|_2^2 / \sigma^2$

Fact:

$KL(P \mid Q) \geq 0$, and being 0 if and only if $P = Q$

Outline:

1. Convergence of Fitted Policy Iteration
2. Trust Region Policy Optimization
    1. Quick intro on KL-divergence
    2. TRPO formulation

An “idealized” trust region formulation for policy update: (back to direct policy optimization)

At iteration t, with π_{θ_t} at hand, we compute θ_{t+1} as follows:

$$\max_{\pi_\theta} J(\theta) - J(\theta_t)$$

$$\text{s.t., } KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \leq \delta$$

We want to maximize performance improvement starting at π_{θ_t},

but we want the new policy to be close to π_{θ_t} (in the KL sense)

Summary:

1. Convergence of Fitted Policy Iteration
2. Trust Region Policy Optimization
    1. Quick intro on KL-divergence
    2. TRPO formulation

1-minute feedback form: https://bit.ly/3RHtlxy