Trust-Region Optimization & Covariant Policy Optimization
Recap

Natural Policy Gradient:

$$\theta = \theta + \eta F_\theta^\dagger \nabla V^{\pi_\theta}$$
Recap

Natural Policy Gradient:

$$\theta = \theta + \eta F_\theta^\top \nabla V^{\pi_\theta}$$

$$F_\theta = \mathbb{E}_{s, a \sim d_\theta^\pi} \nabla \ln \pi_\theta(a \mid s) \nabla \ln \pi_\theta(a \mid s)^\top$$

$$\nabla V^{\pi_\theta} = \mathbb{E}_{s, a \sim d_\theta^\pi} \left[ \nabla \ln \pi_\theta(a \mid s) A^{\pi_\theta}(s, a) \right]$$
Recap

Natural Policy Gradient:

\[ \theta = \theta + \eta F_\theta^\dagger \nabla V^{\pi_\theta} \]

\[ F_\theta = \mathbb{E}_{s,a \sim d^{\pi_\theta}} \nabla \ln \pi_\theta(a \mid s) \nabla \ln \pi_\theta(a \mid s)^\top \]

\[ \nabla V^{\pi_\theta} = \mathbb{E}_{s,a \sim d^{\pi_\theta}} \left[ \nabla \ln \pi_\theta(a \mid s) A^{\pi_\theta}(s, a) \right] \]

\( F_\theta^\dagger \nabla V^{\pi_\theta} \) is the solution of the following least square:
Recap

Natural Policy Gradient:

\[ \theta = \theta + \eta F_{\theta}^{\dagger} \nabla V^{\pi_{\theta}} \]

\[ F_{\theta} = \mathbb{E}_{s,a \sim d_{\pi_{\theta}}} \nabla \ln \pi_{\theta}(a | s) \nabla \ln \pi_{\theta}(a | s)^{\top} \]

\[ \nabla V^{\pi_{\theta}} = \mathbb{E}_{s,a \sim d_{\pi_{\theta}}} \left[ \nabla \ln \pi_{\theta}(a | s) A_{\pi_{\theta}}(s, a) \right] \]

\[ F_{\theta}^{\dagger} \nabla V^{\pi_{\theta}} \text{ is the solution of the following least square:} \]

\[ \hat{w} \in \arg \min_{w} \mathbb{E}_{s,a \sim d_{\pi_{\theta}}} \left[ (w^{\top} \nabla \theta \ln \pi_{\theta}(a | s) - A_{\pi_{\theta}}(a | s))^{2} \right] \]
Recap

Natural Policy Gradient:

\[ \theta = \theta + \eta F_\theta^\dagger \nabla V^{\pi_\theta} \]

\[ F_\theta = \mathbb{E}_{s,a \sim d^{\pi_\theta}} \nabla \ln \pi_\theta(a \mid s) \nabla \ln \pi_\theta(a \mid s)^\top \]

\[ \nabla V^{\pi_\theta} = \mathbb{E}_{s,a \sim d^{\pi_\theta}} \left[ \nabla \ln \pi_\theta(a \mid s) A^{\pi_\theta}(s, a) \right] \]

\( F_\theta^\dagger \nabla V^{\pi_\theta} \) is the solution of the following least square:

\[ \hat{w} \in \arg\min_w \mathbb{E}_{s,a \sim d^{\pi_\theta}} \left[ (w^\top \nabla \ln \pi_\theta(a \mid s) - A^{\pi_\theta}(a \mid s))^2 \right] \]

\[ \theta' = \theta + \eta \hat{w} \]
Recap

Softmax-linear policy: \( \pi_\theta(a \mid s) = \frac{\exp(\theta^T \phi(s, a))}{\sum_{a'} \exp(\theta^T \phi(s, a'))} \)
Recap

Softmax-linear policy: \( \pi_\theta(a \mid s) = \frac{\exp(\theta^T \phi(s, a))}{\sum_{a'} \exp(\theta^T \phi(s, a'))} \)

\[ \nabla_\theta \ln \pi_\theta(a \mid s) = \phi(s, a) - \mathbb{E}_{a' \sim \pi_\theta(\cdot \mid s)} \phi(s, a') := \bar{\phi}^\theta(s, a) \]
Recap

Softmax-linear policy: \( \pi_\theta(a \mid s) = \frac{\exp(\theta^T \phi(s, a))}{\sum_{a'} \exp(\theta^T \phi(s, a'))} \)

\[ \nabla_\theta \ln \pi_\theta(a \mid s) = \phi(s, a) - \mathbb{E}_{a' \sim \pi_\theta(\cdot \mid s)} \phi(s, a') := \bar{\phi}^\theta(s, a) \]

\[ w_* \in \arg \min_{w: \|w\|_2 \leq W} \mathbb{E}_{s, a \sim d_\pi^\theta} \left[ (w^T \bar{\phi}^\theta(s, a) - A_{\pi_\theta}(a, s))^2 \right] \]
Recap

Softmax-linear policy: $\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s, a))}{\sum_{a'} \exp(\theta^\top \phi(s, a'))}$

\[
\nabla_\theta \ln \pi_\theta(a \mid s) = \phi(s, a) - \mathbb{E}_{a' \sim \pi_\theta(\cdot \mid s)} \phi(s, a') := \bar{\phi}^\theta(s, a)
\]

\[
w_* \in \arg \min_{w: \|w\|_2 \leq W} \mathbb{E}_{s, a \sim d^\pi_\theta} \left[ (w^\top \bar{\phi}^\theta(s, a) - A^\pi_\theta(a, s))^2 \right]
\]

NPG-Update: $\theta' = \theta + \eta w_*$
Recap

Softmax-linear policy: $\pi_\theta(a \mid s) = \frac{\exp(\theta^T \phi(s, a))}{\sum_{a'} \exp(\theta^T \phi(s, a'))}$

$$\nabla_{\theta} \ln \pi_\theta(a \mid s) = \phi(s, a) - \mathbb{E}_{a' \sim \pi_\theta(\cdot \mid s)} \phi(s, a') := \bar{\phi}^\theta(s, a)$$

$$w_* \in \arg \min_{w: \|w\|_2 \leq W} \mathbb{E}_{s, a \sim d_{\pi_\theta}} \left[ (w^T \bar{\phi}^\theta(s, a) - A_{\pi_\theta}(a, s))^2 \right]$$

NPG-Update: $\theta' = \theta + \eta w_*$

Another Way of Writing the Update Procedure (i.e., soft policy iteration):

$$\pi'(a \mid s) = \frac{\pi(a \mid s) \exp(\eta w_*^T \phi(s, a))}{Z_s}$$
Recap

Softmax-linear policy: \( \pi_\theta(a \mid s) = \frac{\exp(\theta^T \phi(s, a))}{\sum_{a'} \exp(\theta^T \phi(s, a'))} \)

\[ \kappa = \frac{1}{\sigma_{\min}} \left( \mathbb{E}_{s_0, a_0 \sim \nu} \phi(s_0, a_0) \phi(s_0, a_0)^T \right) < \infty \]

Then for any MDP whose \( Q^\pi(\cdot, \cdot, \cdot) \) is linear in feature \( \phi \) for any \( \pi \) (i.e., linear MDPs), NPG learns a policy \( \hat{\pi} \) with \( \hat{V}^\pi(\rho) \geq V^*(\rho) - \epsilon \), with # of samples

\[ \widetilde{O} \left( \text{poly} \left( d, A, \kappa, \frac{1}{\epsilon}, \frac{1}{1 - \gamma}, W \right) \right) \]
Today:
A trust region optimization perspective of NPG (also recovers the TRPO algorithm)

History:

Covariant Policy Search
J. Andrew Bagnell and Jeff Schneider
Robotics Institute
Carnegie-Mellon University
Pittsburgh, PA 15213
{dbagnell,schneider}@ri.cmu.edu

IJCAI 2003

NeurIPS 2002

Trust Region Policy Optimization

John Schulman
Sergey Levine
Philipp Moritz
Michael Jordan
Pieter Abbeel

University of California, Berkeley, Department of Electrical Engineering and Computer Sciences

ICML 2015
Notations and Settings:

Finite horizon setting: $\mathcal{M} = \{S, A, H, r, P, \rho\}$
Notations and Settings:

Finite horizon setting: $\mathcal{M} = \{S, A, H, r, P, \rho\}$

Average state-action distribution:

$$d^\pi(s, a) = \frac{1}{H} \sum_{h=0}^{H-1} \mathbb{P}_h^\pi(s, a)$$
Notations and Settings:

Finite horizon setting: \( \mathcal{M} = \{S, A, H, r, P, \rho\} \)

Average state-action distribution:

\[
d^\pi(s, a) = \frac{1}{H} \sum_{h=0}^{H-1} P^\pi_h(s, a)
\]

Policy class:

\( \Pi = \{\pi : S \mapsto A\} \subset S \mapsto A \)

\( \pi^* = \arg \max_{\pi \in \Pi} V^\pi(\rho) \)
Notations and Settings:

Finite horizon setting: $\mathcal{M} = \{S, A, H, r, P, \rho\}$

Average state-action distribution:

$$d^{\pi}(s, a) = \frac{1}{H} \sum_{h=0}^{H-1} P^{\pi}_h(s, a)$$

Policy class:

$$\Pi = \{\pi : S \mapsto A\} \subset S \mapsto A$$

$$\pi^* = \arg \max_{\pi \in \Pi} V^{\pi}(\rho)$$

Trajectory distribution:

$$\Pr^{\pi}(\tau) = \rho(s_0)\pi(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\pi(a_1 \mid s_1)\ldots P(s_{H-1} \mid s_{H-2}, a_{H-2})\pi(a_{H-1} \mid s_{H-1})$$
Revisit Gradient Descent:

\[ \theta = \theta_0 - \eta \nabla \theta \ell(\theta_0) \]
Revisit Gradient Descent:

\[ \theta = \theta_0 - \eta \nabla_\theta \ell(\theta_0) \]

In other words:

\[
\min_{\theta} \nabla \ell(\theta_0)^T(\theta - \theta_0), \text{ subject to } \|\theta - \theta_0\|_2^2 \leq \delta,
\]
Revisit Gradient Descent:

\[ \theta = \theta_0 - \eta \nabla_{\theta} \ell(\theta_0) \]

In other words:

\[ \min_{\theta} \nabla \ell(\theta_0)^{\top}(\theta - \theta_0), \text{ subject to } \|\theta - \theta_0\|_2^2 \leq \delta, \]

We in default are using Euclidean distance in the parameter $\theta$ space.
Revisit Gradient Descent:

\[ \theta = \theta_0 - \eta \nabla_\theta \mathcal{L}(\theta_0) \]

In other words:

\[ \min_{\theta} \nabla \mathcal{L}(\theta_0)^\top(\theta - \theta_0), \text{ subject to } \|\theta - \theta_0\|_2^2 \leq \delta, \]

We in default are using Euclidean distance in the parameter \( \theta \) space.

Different re-parameterization (scaling & translation) can lead to a quite different GD path.
\[ \ell : \Theta \mapsto \mathbb{R} \]
\( \Omega \)

\( \Theta \quad \ell : \Theta \mapsto \mathbb{R} \)
\[ \Omega \quad A \quad A^{-1} \quad w_0 \quad \Theta \quad \ell : \Theta \mapsto \mathbb{R} \quad \theta_0 \]
Map the loss function to $\Omega$ space:

$$g(w) := \ell(Aw)$$
Map the loss function to $\Omega$ space:

$g(w) := \ell(Aw)$

$w = w_0 - \eta \nabla g(w_0)$

$\ell(\theta)$

$\theta = \theta_0 - \eta \nabla_\theta \ell(\theta_0)$
Map the loss function to $\Omega$ space:

\[
g(w) := \ell(Aw)
\]

\[
w = w_0 - \eta \nabla g(w_0)
\]

\[
= A^{-1} \theta_0 - \eta \nabla g(w_0)
\]
Map the loss function to $\Omega$ space:

$$g(w) := \ell(Aw)$$

$$w = w_0 - \eta \nabla g(w_0)$$

$$= A^{-1} \theta_0 - \eta \nabla g(w_0)$$

$$\left( \nabla_w g(w) = \nabla_w \ell(Aw) = A \nabla_\theta \ell(\theta) \big|_{\theta = Aw} \right)$$
Map the loss function to $\Omega$ space:

$$g(w) := \ell(Aw)$$

$$w = w_0 - \eta \nabla g(w_0) = A^{-1} \theta_0 - \eta \nabla g(w_0)$$

$$\left( \nabla_w g(w) = \nabla_w \ell(Aw) = A \nabla_\theta \ell(\theta) \big|_{\theta = Aw} \right)$$

$$= A^{-1} \theta_0 - \eta A \nabla_\theta \ell(\theta_0)$$
Map the loss function to $\Omega$ space:

$$g(w) := \ell(Aw)$$

$$w = w_0 - \eta \nabla g(w_0)$$

$$= A^{-1} \theta_0 - \eta \nabla g(w_0)$$

$$= A^{-1} \theta_0 - \eta \nabla \ell(\theta) |_{\theta = Aw}$$

$$\nabla_w g(w) = \nabla_w \ell(Aw) = A \nabla_\theta \ell(\theta) |_{\theta = Aw}$$

$$= A^{-1} \theta_0 - \eta A \nabla_\theta \ell(\theta_0)$$

$$\theta' = \theta_0 - \eta A^2 \nabla_\theta \ell(\theta_0)$$
Map the loss function to $\Omega$ space:

$$g(w) := \ell(Aw)$$

$$w = w_0 - \eta \nabla g(w_0)$$

$$= A^{-1} \theta_0 - \eta \nabla g(w_0)$$

$$(\nabla_w g(w) = \nabla_w \ell(Aw) = A \nabla_\theta \ell(\theta) |_{\theta = Aw})$$

$$= A^{-1} \theta_0 - \eta A \nabla_\theta \ell(\theta_0)$$

$$\theta' = \theta_0 - \eta A^2 \nabla_\theta \ell(\theta_0)$$

Linear transformation $A$ makes the GD path different!
i.e., not invariant wrt linear transformation (scaling, rotations, etc)
What would happen if we use a different distance metric...

\[
\begin{align*}
\min_w \nabla_w g(w_0)^T (w - w_0) \\
\text{s.t., } (w - w_0)^T (AA)(w - w_0) \leq \delta
\end{align*}
\]
What would happen if we use a different distance metric...

\[
\min_w \nabla_w g(w_0)^T (w - w_0)
\]

s.t., \((w - w_0)^T (AA)(w - w_0) \leq \delta\)

This gives us:

\[
w = w_0 - \eta A^{-2} \nabla g(w_0)
\]
What would happen if we use a different distance metric...

$$\min_w \nabla_w g(w_0)^T (w - w_0)$$

s.t., $$(w - w_0)^T (AA)(w - w_0) \leq \delta$$

This gives us:

$$w = w_0 - \eta A^{-2} \nabla g(w_0)$$

$$\left( \nabla_w g(w_0) = \nabla_w \ell(Aw_0) = A \nabla_{\theta} \ell(\theta)|_{\theta=Aw_0} \right)$$
What would happen if we use a different distance metric...

\[
\min_w \nabla_w g(w_0)^T (w - w_0)
\]

\[
\text{s.t., } (w - w_0)^T (AA)(w - w_0) \leq \delta
\]

This gives us:

\[
w = w_0 - \eta A^{-2} \nabla g(w_0) \quad (\nabla_w g(w_0) = \nabla_w \ell(Aw_0) = A \nabla_{\theta} \ell(\theta)|_{\theta=Aw_0})
\]

\[
w = A^{-1} \theta_0 - \eta A^{-1} \nabla_{\theta} \ell(\theta_0)
\]
What would happen if we use a different distance metric...

\[
\begin{align*}
\min_w \nabla_w g(w_0)^T (w - w_0) \\
\text{s.t., } (w - w_0)^T (AA)(w - w_0) \leq \delta
\end{align*}
\]

This gives us:

\[
\begin{align*}
w &= w_0 - \eta A^{-2} \nabla g(w_0) \\
\nabla_w g(w_0) &= \nabla_w \ell(Aw_0) = A \nabla_\theta \ell(\theta) \big|_{\theta = Aw_0} \\
w &= A^{-1} \theta_0 - \eta A^{-1} \nabla_\theta \ell(\theta_0) \\
\Rightarrow \theta &= \theta_0 - \eta \nabla \ell(\theta_0)
\end{align*}
\]
Back to Policy Optimization:

\[
\max_{\pi_0} V_{\pi_0}(\rho)
\]

s.t., \( KL(\Pr_{\pi_0} \| \Pr_{\tilde{\pi}}) \leq \delta \)
Back to Policy Optimization:

\[
\max_{\pi_{\theta}} V_{\pi_{\theta}}(\rho)
\]

s.t., \(KL \left( \Pr\pi_{\theta} \, \middle| \middle| \Pr\pi_{\theta} \right) \leq \delta\)

Sequential convex programming:
We linearize the objective function & quadratize the KL constraint
Back to Policy Optimization:

\[
\max_{\pi_\theta} V^{\pi_\theta}(\rho)
\]

s.t., \( KL (Pr^{\pi_0} \parallel Pr^{\pi_\theta}) \leq \delta \)

Sequential convex programming:
We linearize the objective function & quadratize the KL constraint

We know the first order Taylor expansion of \( V^{\pi_\theta}(\rho) \)

\[
V^{\pi_0}(\rho) + \nabla V^{\pi_0}(\rho)^\top (\theta - \theta_0)
\]
Back to Policy Optimization:

$$\max_{\pi_\theta} V^{\pi_\theta}(\rho)$$

s.t., $KL(\text{Pr}^{\pi_0} \mid \mid \text{Pr}^{\pi_\theta}) \leq \delta$

Sequential convex programming:
We linearize the objective function & quadratize the KL constraint

We know the first order Taylor expansion of $V^{\pi_\theta}(\rho)$

$$V^{\pi_0}(\rho) + \nabla V^{\pi_0}(\rho)^\top (\theta - \theta_0)$$

Q: How to do second-order Taylor expansion on the KL constraint?
Let's do second order Taylor Expansion on the KL-divergence
Let's do second order Taylor Expansion on the KL-divergence

\[
\frac{1}{H} KL \left( \Pr^{\pi_0} \big| \big| \Pr^{\pi} \right) = \frac{1}{H} \sum_{\tau} \Pr^{\theta_0}(\tau) \ln \frac{\Pr^{\theta_0}(\tau)}{\Pr^{\theta}(\tau)} = \frac{1}{H} \sum_{\tau} \Pr^{\theta_0}(\tau) \sum_{h=0}^{H-1} \ln \frac{\pi^{\theta_0}(a_h \mid s_h)}{\pi^{\theta}(a_h \mid s_h)}
\]
Let’s do second order Taylor Expansion on the KL-divergence

\[
\frac{1}{H} KL \left( Pr^{\pi_{\theta_0}} | Pr^{\pi_\theta} \right) = \frac{1}{H} \sum_{\tau} Pr^{\theta_0}(\tau) \ln \frac{Pr^{\theta_0}(\tau)}{Pr^{\theta}(\tau)} = \frac{1}{H} \sum_{\tau} Pr^{\theta_0}(\tau) \sum_{h=0}^{H-1} \ln \frac{\pi^{\theta_0}(a_h | s_h)}{\pi^{\theta}(a_h | s_h)}
\]

\[
= \mathbb{E}_{s_h, a_h \sim d^{\pi_{\theta_0}}} \left[ \ln \frac{\pi^{\theta_0}(a_h | s_h)}{\pi^{\theta}(a_h | s_h)} \right] := \ell(\theta)
\]
Let’s do second order Taylor Expansion on the KL-divergence

\[
\frac{1}{H} KL \left( \Pr_{\theta_0} \mid \mid \Pr_{\theta} \right) = \frac{1}{H} \sum_{\tau} \Pr_{\theta_0}(\tau) \ln \frac{\Pr_{\theta_0}(\tau)}{\Pr_{\theta}(\tau)} = \frac{1}{H} \sum_{\tau} \Pr_{\theta_0}(\tau) \sum_{h=0}^{H-1} \ln \frac{\pi_{\theta_0}(a_h \mid s_h)}{\pi_{\theta}(a_h \mid s_h)} \\
= \mathbb{E}_{s_h, a_h \sim d_{\theta_0}} \left[ \ln \frac{\pi_{\theta_0}(a_h \mid s_h)}{\pi_{\theta}(a_h \mid s_h)} \right] := \ell(\theta) \quad \ell(\theta_0) = 0
\]
Let's do second order Taylor Expansion on the KL-divergence

\[
\frac{1}{H} \text{KL} \left( \text{Pr}^{\pi_{\theta_0}} \mid \text{Pr}^{\pi_{\theta}} \right) = \frac{1}{H} \sum_{\tau} \text{Pr}^{\theta_0}(\tau) \ln \frac{\text{Pr}^{\theta_0}(\tau)}{\text{Pr}^{\theta}(\tau)} = \frac{1}{H} \sum_{\tau} \text{Pr}^{\theta_0}(\tau) \sum_{h=0}^{H-1} \ln \frac{\pi^{\theta_0}(a_h \mid s_h)}{\pi^{\theta}(a_h \mid s_h)}
\]

\[
= \mathbb{E}_{s_h, a_h \sim d^{\pi_{\theta_0}}} \left[ \ln \frac{\pi^{\theta_0}(a_h \mid s_h)}{\pi^{\theta}(a_h \mid s_h)} \right] := \ell(\theta) \quad \ell(\theta_0) = 0
\]

\[
\nabla_{\theta} \ell(\theta) \mid_{\theta = \theta_0} = \mathbb{E}_{s \sim d^{\pi_{\theta_0}}} \sum_{a} \pi^{\theta_0}(a \mid s) \left( - \nabla_{\theta} \ln \pi^{\theta}(a_h \mid s_h) \mid_{\theta = \theta_0} \right)
\]
Let's do second order Taylor Expansion on the KL-divergence

\[
\frac{1}{H} KL \left( Pr^{\pi_{\theta_0}} \mid Pr^{\pi_{\theta}} \right) = \frac{1}{H} \sum_{\tau} Pr^{\theta_0}(\tau) \ln \frac{Pr^{\theta_0}(\tau)}{Pr^{\theta}(\tau)} = \frac{1}{H} \sum_{\tau} Pr^{\theta_0}(\tau) \sum_{h=0}^{H-1} \ln \frac{\pi^{\theta_0}(a_h \mid s_h)}{\pi^{\theta}(a_h \mid s_h)}
\]

\[
= \mathbb{E}_{s_h, a_h \sim d^{\pi_{\theta_0}}} \left[ \ln \frac{\pi^{\theta_0}(a_h \mid s_h)}{\pi^{\theta}(a_h \mid s_h)} \right] := \ell(\theta) \quad \ell(\theta_0) = 0
\]

\[
\nabla_{\theta} \ell(\theta) \mid_{\theta=\theta_0} = \mathbb{E}_{s \sim d^{\pi_{\theta_0}}} \sum_{a} \pi^{\theta_0}(a \mid s) \left( - \nabla_{\theta} \ln \pi^{\theta}(a_h \mid s_h) \mid_{\theta=\theta_0} \right)
\]

\[
= - \mathbb{E}_{s \sim d^{\pi_{\theta_0}}} \sum_{a} \pi^{\theta_0}(a \mid s) \nabla_{\theta} \pi^{\theta_0}(a \mid s) \frac{\pi^{\theta_0}(a \mid s)}{\pi^{\theta}(a \mid s)}
\]
Let’s compute the Hessian of the KL-divergence
Let’s compute the Hessian of the KL-divergence

\[ \mathbb{E}_{s,a \sim d^{\pi_0}} \left[ \ln \frac{\pi_{\theta_0}(a_h \mid s_h)}{\pi_0(a_h \mid s_h)} \right] := \ell(\theta) \]
Let’s compute the Hessian of the KL-divergence

\[
\mathbb{E}_{s,a \sim d^{\pi_{\theta_0}}} \left[ \ln \frac{\pi_{\theta_0}(a_h \mid s_h)}{\pi_{\theta}(a_h \mid s_h)} \right] := \ell(\theta)
\]

\[
\nabla^2_{\theta} \ell(\theta) \big|_{\theta=\theta_0} = \mathbb{E}_{s \sim d^{\pi_{\theta_0}}} \sum_a \pi_{\theta_0}(a \mid s) \left( -\nabla^2_{\theta} \ln \pi_{\theta}(a \mid s) \big|_{\theta=\theta_0} \right)
\]
Let's compute the Hessian of the KL-divergence

\[ \mathbb{E}_{s,a \sim d^\pi_0} \left[ \ln \frac{\pi_{\theta_0}(a_h | s_h)}{\pi_\theta(a_h | s_h)} \right] := \ell(\theta) \]

\[ \nabla_\theta^2 \ell(\theta) |_{\theta = \theta_0} = \mathbb{E}_{s \sim d^\pi_0} \sum_a \pi_{\theta_0}(a | s) \left( - \nabla_\theta^2 \ln \pi_\theta(a | s) |_{\theta = \theta_0} \right) \]

\[ = - \mathbb{E}_{s \sim d^\pi_0} \sum_a \pi_{\theta_0}(a | s) \left( \frac{\nabla_\theta^2 \pi_{\theta_0}(a | s)}{\pi_{\theta_0}(a | s)} - \frac{\nabla_\theta \pi_{\theta_0}(a | s) \nabla_\theta \pi_{\theta_0}(a | s)^T}{\pi_{\theta_0}^2(a | s)} \right) \]
Let’s compute the Hessian of the KL-divergence

\[ \mathbb{E}_{s,a \sim \mathcal{d}^{\pi_{\theta_0}}} \left[ \ln \frac{\pi_{\theta_0}(a_h | s_h)}{\pi_{\theta}(a_h | s_h)} \right] := \ell(\theta) \]

\[ \nabla_{\theta}^2 \ell(\theta) \bigg|_{\theta = \theta_0} = \mathbb{E}_{s \sim \mathcal{d}^{\pi_{\theta_0}}} \sum_a \pi_{\theta_0}(a | s) \left( -\nabla_{\theta}^2 \ln \pi_{\theta}(a | s) \bigg|_{\theta = \theta_0} \right) \]

\[ = -\mathbb{E}_{s \sim \mathcal{d}^{\pi_{\theta_0}}} \sum_a \pi_{\theta_0}(a | s) \left( \frac{\nabla_{\theta}^2 \pi_{\theta_0}(a | s)}{\pi_{\theta_0}(a | s)} - \frac{\nabla_{\theta} \pi_{\theta_0}(a | s) \nabla_{\theta} \pi_{\theta_0}(a | s)^T}{\pi_{\theta_0}(a | s)^2} \right) \]

\[ = \mathbb{E}_{s,a \sim \mathcal{d}^{\pi_{\theta_0}}} \left[ \nabla_{\theta} \ln \pi_{\theta_0}(a | s) \left( \nabla_{\theta} \ln \pi_{\theta_0}(a | s) \right)^T \right] \]
Let's compute the Hessian of the KL-divergence

\[ \mathbb{E}_{s,a \sim d^{\theta_0}} \left[ \ln \frac{\pi_{\theta_0}(a_h | s_h)}{\pi(\theta_0)(a_h | s_h)} \right] := \ell(\theta) \]

\[ \nabla_\theta^2 \ell(\theta) |_{\theta = \theta_0} = \mathbb{E}_{s \sim d^{\theta_0}} \sum_a \pi_{\theta_0}(a | s) \left( - \nabla_\theta^2 \ln \pi_\theta(a | s) |_{\theta = \theta_0} \right) \]

\[ = - \mathbb{E}_{s \sim d^{\theta_0}} \sum_a \pi_{\theta_0}(a | s) \left( \frac{\nabla_\theta^2 \pi_{\theta_0}(a | s)}{\pi_{\theta_0}(a | s)} - \frac{\nabla_\theta \pi_{\theta_0}(a | s) \nabla_\theta \pi_{\theta_0}(a | s)^\top}{\pi_{\theta_0}^2(a | s)} \right) \]

\[ = \mathbb{E}_{s,a \sim d^{\theta_0}} \left[ \nabla_\theta \ln \pi_{\theta_0}(a | s) \left( \nabla_\theta \ln \pi_{\theta_0}(a | s) \right)^\top \right] \]

Fisher Information Matrix!
Second-order Taylor Expansion of KL at $\theta_0$

$$\frac{1}{H} KL (\Pr_{\theta_0} || \Pr_{\theta}) \leq \delta \Rightarrow \frac{1}{2} (\theta - \theta_0)^\top F_{\theta_0}(\theta - \theta_0) \leq \delta$$
Second-order Taylor Expansion of KL at $\theta_0$

$$\frac{1}{H} KL(\Pr_{\theta_0} || \Pr_{\theta}) \leq \delta \Rightarrow \frac{1}{2} (\theta - \theta_0)^\top F_{\theta_0}(\theta - \theta_0) \leq \delta$$
Second-order Taylor Expansion of KL at $\theta_0$

$$
\frac{1}{H} KL \left( \Pr_{\theta_0} \parallel \Pr_{\theta} \right) \leq \delta \Rightarrow \frac{1}{2} (\theta - \theta_0)^\top F_{\theta_0} (\theta - \theta_0) \leq \delta
$$
Second-order Taylor Expansion of KL at $\theta_0$

$$\frac{1}{H} KL (\Pr_{\pi_{\theta_0}} \, || \, \Pr_{\pi_0}) \leq \delta \Rightarrow \frac{1}{2} (\theta - \theta_0)^\top F_{\theta_0}(\theta - \theta_0) \leq \delta$$

$$ (p_1, p_2) = \left( \frac{\exp(\theta)}{1 + \exp(\theta)}, \frac{1}{1 + \exp(\theta)} \right)$$
Second-order Taylor Expansion of KL at $\theta_0$

$$\frac{1}{H} KL \left( Pr_{\theta_0}^{\pi} || Pr_{\theta}^{\pi} \right) \leq \delta \Rightarrow \frac{1}{2} (\theta - \theta_0)^{\top} F_{\theta_0}(\theta - \theta_0) \leq \delta$$

$$(p_1, p_2) = \left( \frac{\exp(\theta)}{1 + \exp(\theta)}, \frac{1}{1 + \exp(\theta)} \right)$$

$F_\theta \to 0^+, \text{ as } \theta \to \infty$$
Second-order Taylor Expansion of KL at $\theta_0$

$$\frac{1}{H} KL(Pr_{\theta_0} || Pr_{\theta}) \leq \delta \Rightarrow \frac{1}{2}(\theta - \theta_0)^\top F_{\theta_0}(\theta - \theta_0) \leq \delta$$

$$F_\theta \to 0^+, \text{ as } \theta \to \infty$$

$$F_{\theta_0}(\theta - \theta_0)^2 \leq \delta \Rightarrow (\theta - \theta_0)^2 \leq \frac{\delta}{F_{\theta_0}} \to \infty, \text{ as } \theta_0 \to \infty$$
Second-order Taylor Expansion of KL at $\theta_0$

\[
\frac{1}{H} KL(\Pr^{\pi_{\theta_0}} \| \Pr^{\pi_\theta}) \leq \delta \Rightarrow \frac{1}{2} (\theta - \theta_0)^T F_{\theta_0}(\theta - \theta_0) \leq \delta
\]

Plain GD in $\theta$ will move to $\theta = \infty$ at a constant speed, while Natural GD can traverse faster and faster when $\theta$ gets bigger (Infinitely fast when $\theta \to \infty$)
Now we can solve the following quadratic programming:

$$\max_\theta \nabla V^{\pi_{0}}(\rho)^T(\theta - \theta_0)$$

s.t. $$(\theta - \theta_0)^T F_{\theta_0}(\theta - \theta_0) \leq \delta$$
Now we can solve the following quadratic programming:

$$\max_{\theta} \nabla V^{\pi_{\theta_0}}(\rho)^T(\theta - \theta_0)$$

s.t. $$(\theta - \theta_0)^T F_{\theta_0}(\theta - \theta_0) \leq \delta$$

We have a closed form solution:

$$\theta = \theta_0 + \sqrt{\frac{\delta}{(\nabla V^{\pi_{\theta_0}})^T F^{-1}_{\theta_0} \nabla V^{\pi_{\theta_0}}}} \cdot F^{-1}_{\theta_0} \nabla V^{\pi_{\theta_0}}$$
Now we can solve the following quadratic programming:

$$\max_{\theta} \nabla V_{\pi_{\theta_0}}(\rho)^T(\theta - \theta_0)$$

s.t. $$(\theta - \theta_0)^T F_{\theta_0}(\theta - \theta_0) \leq \delta$$

We have a closed form solution:

$$\theta = \theta_0 + \sqrt{\frac{\delta}{(\nabla V_{\pi_{\theta_0}})^T F_{\theta_0}^{-1} \nabla V_{\pi_{\theta_0}} \cdot F_{\theta_0}^{-1} \nabla V_{\pi_{\theta_0}}}}$$

Self-normalized step-size
(Learning rate is adaptive)
Natural Policy Gradient invariant to linear transformation
(Trust region constraint in terms KL on trajectory distributions)

Second order Taylor expansion of $\mathcal{L}(\theta) := KL \left( \Pr^{\pi_{\theta_0}} \mid \mid \Pr^{\pi_{\theta}} \right)$ at $\theta_0$ is $(\theta - \theta_0)^{\top} F_{\theta_0}(\theta - \theta_0)$
Approximate Policy Iteration
& Conservative Policy Iteration
Recall Policy Iteration (PI):

Assume we know $A^\pi(s, a)$ for all $s, a$, PI updates policy as:

$$\pi'(s) = \arg \max_a A^\pi(s, a)$$
Recall Policy Iteration (PI):

Assume we know $A^\pi(s, a)$ for all $s, a$, PI updates policy as:

$$\pi'(s) = \arg\max_a A^\pi(s, a)$$

i.e., be greedy with respect to $\pi$ at every state $s$, 

Recap

Recall Policy Iteration (PI):

Assume we know $A^\pi(s, a)$ for all $s, a$, PI updates policy as:

$$\pi'(s) = \arg \max_a A^\pi(s, a)$$

i.e., be greedy with respect to $\pi$ at every state $s$,

However, there is no way we will be able to know $A^\pi(s, a)$ at all $s, a$, so how can we do policy update?
Setting and Notation

Discounted infinite horizon MDP:

\[ \mathcal{M} = \{ S, A, \gamma, r, P, \mu \} \]

State visitation:

\[ d^\pi_{\mu}(s) = (1 - \gamma) \sum_{h=0}^{\infty} \gamma^h P^\pi_h(s; \mu) \]

Unbiased estimate of \( A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s) \)

As we will consider large scale MDP here, we start with a (restricted) function class \( \Pi \):

\[ \Pi = \{ \pi : S \mapsto \Delta(A) \} \]
Attempt One: *Approximate* Policy Iteration (API)
Attempt One: **Approximate Policy Iteration (API)**

Given the current policy $\pi^t$, let’s act greedily wrt $\pi$ under $d^t_\mu$
Attempt One: **Approximate Policy Iteration (API)**

Given the current policy $\pi^t$, let’s act greedily wrt $\pi$ under $d^\pi_{\mu}$

i.e., let’s aim to (approximately) solve the following program:

$$\arg \max_{\pi \in \Pi} \mathbb{E}_{s \sim d^\pi_{\mu}} \left[ A^{\pi^t}(s, \pi(s)) \right]$$
Attempt One: **Approximate Policy Iteration (API)**

Given the current policy $\pi^t$, let’s act greedily wrt $\pi$ under $d^\pi$.

i.e., let’s aim to (approximately) solve the following program:

$$\arg\max_{\pi \in \Pi} \mathbb{E}_{s \sim d^\pi} \left[ A^{\pi^t}(s, \pi(s)) \right]$$

**Greedy Policy Selector**
Attempt One: **Approximate Policy Iteration (API)**

Given the current policy $\pi^t$, let’s act greedily wrt $\pi$ under $\mu^{\pi^t}$

i.e., let’s aim to (approximately) solve the following program:

\[
\arg \max_{\pi \in \Pi} \mathbb{E}_{s \sim d^{\mu^{\pi^t}}} \left[ A^{\pi^t}(s, \pi(s)) \right]
\]

**Greedy Policy Selector**

But we can only sample from $d^{\pi^t}$, and we can only get an approximation of $A^{\pi^t}(s, a)$
Attempt One: **Approximate Policy Iteration (API)**

Given the current policy $\pi^t$, let’s act greedily wrt $\pi$ under $d^\pi_\mu$

i.e., let’s aim to (approximately) solve the following program:

$$\arg \max_{\pi \in \Pi} \mathbb{E}_{s \sim d^\pi_\mu} \left[ A^{\pi^i}(s, \pi(s)) \right] \quad \text{Greedy Policy Selector}$$

But we can only sample from $d^\pi_\mu$, and we can only get an approximation of $A^{\pi^i}(s, a)$

We can hope for an Approximate Greedy Policy Selector via (1) a Classification Oracle and (2) Regression Oracle
Implementing Approximate Greedy Policy Selector via Classification

Think about $\pi$ as a classifier, and recall the classic weighted classification oracle:
Implementing Approximate Greedy Policy Selector via Classification

Think about $\pi$ as a classifier, and recall the classic weighted classification oracle:

Dataset $\mathcal{D} = \{s_i, r_i\}$, where $r_i \in \mathbb{R}^d$

$$\text{CO}(\mathcal{D}, \Pi) = \arg \max_{\pi \in \Pi} \sum_i r_i[\pi(s_i)]$$
Implementing Approximate Greedy Policy Selector via Classification

Think about $\pi$ as a classifier, and recall the classic weighted classification oracle:

Dataset $\mathcal{D} = \{s_i, r_i\}$, where $r_i \in \mathbb{R}^d$

$$\text{CO}(\mathcal{D}, \Pi) = \arg\max_{\pi \in \Pi} \sum_i r_i[\pi(s_i)]$$

$$\arg\max_{\pi \in \Pi} \mathbb{E}_{s \sim \mu}[A^\pi(s, \pi(s))]$$
Implementing Approximate Greedy Policy Selector via Classification

Think about $\pi$ as a classifier, and recall the classic weighted classification oracle:

Dataset $\mathcal{D} = \{s_i, r_i\}$, where $r_i \in \mathbb{R}^d$

$$\text{CO} (\mathcal{D}, \Pi) = \arg \max_{\pi \in \Pi} \sum_i r_i[\pi(s_i)]$$

$$\arg \max_{\pi \in \Pi} \mathbb{E}_{s \sim d^{s_i}} \left[ A^{\pi'}(s, \pi(s)) \right]$$

1. Collect samples

$\{s_i, a_i, \tilde{A}^i\}$,

$s_i \sim d^{s_i}, a_i \sim U(A), \mathbb{E} \left[ \tilde{A}^i \right] = A^{\pi'}(s_i, a_i)$
Implementing Approximate Greedy Policy Selector via Classification

Think about $\pi$ as a classifier, and recall the classic weighted classification oracle:

Dataset $\mathcal{D} = \{s_i, r_i\}$, where $r_i \in \mathbb{R}^d$

$$\text{CO}(\mathcal{D}, \Pi) = \arg \max_{\pi \in \Pi} \sum_i r_i[\pi(s_i)]$$

2. Form weighted classification dataset:

$$\arg \max_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^\pi} \left[A^\pi(s, \pi(s)) \right]$$

1. Collect samples

$$\{s_i, a_i, \tilde{A}^i\},$$

$$s_i \sim d_\mu^\pi, a_i \sim U(A), \mathbb{E} \left[\tilde{A}^i\right] = A^\pi(s_i, a_i)$$
Implementing Approximate Greedy Policy Selector via Classification

Think about $\pi$ as a classifier, and recall the classic weighted classification oracle:

Dataset $\mathcal{D} = \{s_i, r_i\}$, where $r_i \in \mathbb{R}^d$

$$\text{CO}(\mathcal{D}, \Pi) = \arg \max_{\pi \in \Pi} \sum_i r_i[\pi(s_i)]$$

1. Collect samples

$$\{s_i, a_i, \overline{A}^i\},$$

$s_i \sim d^\pi_\mu, a_i \sim U(A), \mathbb{E} \left[ \overline{A}^i \right] = A^{\pi'}(s_i, a_i)$

2. Form weighted classification dataset:

$$\mathcal{D} = \{s_i, r_i\}$$

where $r_i[a] = \begin{cases} 0, & a \neq a_i \\ \frac{\overline{A}^i}{1/A}, & a = a_i \end{cases}$
Implementing Approximate Greedy Policy Selector via Classification

Think about $\pi$ as a classifier, and recall the classic weighted classification oracle:

Dataset $\mathcal{D} = \{s_i, r_i\}$, where $r_i \in \mathbb{R}^d$

$$\text{CO}(\mathcal{D}, \Pi) = \arg\max_{\pi \in \Pi} \sum_i r_i[\pi(s_i)]$$

1. Collect samples

$$\{s_i, a_i, \overline{A}_i\},$$

$s_i \sim d^\pi_\mu$, $a_i \sim U(A)$, $\mathbb{E} \left[ \overline{A}_i \right] = A^\pi(s_i, a_i)$

2. Form weighted classification dataset:

$$\mathcal{D} = \{s_i, r_i\}$$

Claim: $\mathbb{E} \left[ r_i(a) | s_i \right] = A^\pi(s_i, a), \forall a$

(Proof: importance weighting)

where $r_i[a] = \begin{cases} 0, & a \neq a_i \\ \frac{\overline{A}_i}{1/A}, & a = a_i \end{cases}$
Implementing Approximate Greedy Policy Selector via Classification

Think about $\pi$ as a classifier, and recall the classic weighted classification oracle:

Dataset $\mathcal{D} = \{s_i, r_i\}$, where $r_i \in \mathbb{R}^d$

$$\text{CO}(\mathcal{D}, \Pi) = \arg\max_{\pi \in \Pi} \sum_i r_i[\pi(s_i)]$$

1. Collect samples

$$\{s_i, a_i, \tilde{A}^i\},$$

$s_i \sim d^r_\mu, a_i \sim U(A), \mathbb{E} \left[\tilde{A}^i\right] = A^{\pi'}(s_i, a_i)$

2. Form weighted classification dataset:

$$\mathcal{D} = \{s_i, r_i\}$$

where $r_i[a] = \begin{cases} 0, & a \neq a_i \\ \frac{\tilde{A}^i}{1/A}, & a = a_i \end{cases}$

3. Set $\hat{\pi} = \text{CO}(\mathcal{D}, \Pi)$

Claim: $\mathbb{E} \left[ r_i(a) \mid s_i \right] = A^{\pi'}(s_i, a), \forall a$

(Proof: importance weighting)
Implementing Approximate Greedy Policy Selector via Classification

Claim [Approximate Greedy Policy Selector]: with N data points, with probability at least $1 - \delta$, the classification oracle returns $\hat{\pi}$, such that

$$\mathbb{E}_{s \sim d_{\mu}^{\pi_t}} \left[ A^{\pi_t}(s, \hat{\pi}(s)) \right] \geq \max_{\pi \in \Pi} \mathbb{E}_{s \sim d_{\mu}^{\pi_t}} \left[ A^{\pi}(s, \pi(s)) \right] - \frac{A}{1 - \gamma} \sqrt{\frac{\ln(\|\Pi\|/\delta)}{N}}$$
Implementing Approximate Greedy Policy Selector via Classification

Claim [Approximate Greedy Policy Selector]: with N data points, with probability at least \(1 - \delta\), the classification oracle returns \(\hat{\pi}\), such that

\[
\mathbb{E}_{s \sim d^\pi_t} \left[ A^{\pi_t}(s, \hat{\pi}(s)) \right] \geq \max_{\pi \in \Pi} \mathbb{E}_{s \sim d^\pi_t} \left[ A^{\pi_t}(s, \pi(s)) \right] - \frac{A}{1 - \gamma} \sqrt{\frac{\ln(|\Pi|/\delta)}{N}}
\]

In other words, we can get an \(\epsilon\) approximate greedy policy selector w/ # of samples

\[
O \left( \ln(|\Pi|/\delta) \frac{A^2}{(1 - \gamma)^2} \frac{1}{\epsilon^2} \right)
\]
Implementing Approximate Greedy Policy Selector via Regression

We can also do a **reduction to Regression** via Advantage function approximation

\[ \mathcal{F} = \{ f : S \times A \mapsto \mathbb{R} \} \quad (\approx A^\pi) \]
Implementing Approximate Greedy Policy Selector via Regression

We can also do a **reduction to Regression** via Advantage function approximation

\[
\mathcal{F} = \{f : S \times A \mapsto \mathbb{R} \} \quad ( \approx A^\pi )
\]

\[
\Pi = \{ \pi(s) = \arg \max_a f(s, a) : f \in \mathcal{F} \}
\]
Implementing Approximate Greedy Policy Selector via Regression

We can also do a reduction to Regression via Advantage function approximation

\[ \mathcal{F} = \{ f : S \times A \mapsto \mathbb{R} \} \quad (\approx A^\pi) \]
\[ \Pi = \{ \pi(s) = \arg \max_a f(s, a) : f \in \mathcal{F} \} \]
\[ \{ s_i, a_i, \tilde{A}_i \}, s_i \sim d^\pi, a \sim U(A), \mathbb{E} \left[ \tilde{A}_i \right] = A^\pi(s_i, a_i) \]
Implementing Approximate Greedy Policy Selector via Regression

We can also do a reduction to Regression via Advantage function approximation

\[ \mathcal{F} = \{ f : S \times A \mapsto \mathbb{R} \} \quad (\approx A^\pi) \]

\[ \Pi = \{ \pi(s) = \arg\max_a f(s, a) : f \in \mathcal{F} \} \]

\[ \{ s_i, a_i, \tilde{A}_i \}, s_i \sim d^\pi, a \sim U(A), \mathbb{E}\left[ \tilde{A}_i \right] = A^\pi(s_i, a_i) \]

Regression oracle:

\[ \hat{f} = \arg\min_{f \in \mathcal{F}} \sum_i \left( f(s_i, a_i) - \tilde{A}_i \right)^2 \]
Implementing Approximate Greedy Policy Selector via Regression

We can also do a **reduction to Regression** via Advantage function approximation

\[
\mathcal{F} = \{ f : S \times A \mapsto \mathbb{R} \} \quad (\approx A^\pi)
\]

\[
\Pi = \{ \pi(s) = \arg \max_a f(s, a) : f \in \mathcal{F} \}
\]

\[
\{ s_i, a_i, \tilde{A}_i \}, s_i \sim d^\pi_\mu, a_i \sim U(A), \mathbb{E} \left[ \tilde{A}_i \right] = A^\pi(s_i, a_i)
\]

Regression oracle:

\[
\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_i \left( f(s_i, a_i) - \tilde{A}_i \right)^2
\]

Act greedily wrt the estimator \( \hat{f} \) (as we hope \( \hat{f} \approx A^\pi \)): 
Implementing Approximate Greedy Policy Selector via Regression

We can also do a reduction to Regression via Advantage function approximation

\[ \mathcal{F} = \{ f : S \times A \mapsto \mathbb{R} \} \quad (\approx A^\pi) \]

\[ \Pi = \{ \pi(s) = \underset{a}{\arg\max} f(s, a) : f \in \mathcal{F} \} \]

\[ \{ s_i, a_i, \tilde{A}_i \}, s_i \sim d_\mu, a \sim U(A), \mathbb{E} \left[ \tilde{A}_i \right] = A^\pi(s_i, a_i) \]

Regression oracle:

\[ \hat{f} = \arg\min_{f \in \mathcal{F}} \sum_i \left( f(s_i, a_i) - \tilde{A}_i \right)^2 \]

Act greedily wrt the estimator \( \hat{f} \) (as we hope \( \hat{f} \approx A^\pi \)):

\[ \hat{\pi}(s) = \underset{a}{\arg\max} \hat{f}(s, a), \forall s \]
Implementing Approximate Greedy Policy Selector via Regression

We can also do a **reduction to Regression** via Advantage function approximation

$$\mathcal{F} = \{f : S \times A \mapsto \mathbb{R} \} \quad (\approx A^\pi)$$

$$\Pi = \{\pi(s) = \arg \max_a f(s, a) : f \in \mathcal{F} \}$$

$$\{s_i, a_i, \tilde{A}_i\}, s_i \sim d_\mu, a_i \sim U(A), \mathbb{E} \left[ \tilde{A}_i \right] = A^\pi(s_i, a_i)$$

**Regression oracle:**

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_{i} \left( f(s_i, a_i) - \tilde{A}_i \right)^2$$

Act greedily wrt the estimator $$\hat{f}$$ (as we hope $$\hat{f} \approx A^\pi$$):

$$\hat{\pi}(s) = \arg \max_a \hat{f}(s, a), \forall s$$

Do **finite sample analysis for Regression** first, and then transfer the guarantee to greedy policy selection
Summary So Far:

By reduction to Supervised Learning (i.e., classification using $\Pi$ or Regression using $\mathcal{F}$), with high probability, we get:
Summary So Far:

By reduction to Supervised Learning (i.e., classification using $\Pi$ or Regression using $\mathcal{F}$), with high probability, we get:

\[
\mathbb{E}_{s \sim d^\pi} \left[ A^{\pi}(s, \hat{\pi}(s)) \right] \geq \max_{\pi \in \Pi} \mathbb{E}_{s \sim d^\pi} \left[ A^{\pi}(s, \pi(s)) \right] - \frac{A}{1 - \gamma} \sqrt{\frac{\ln(|\Pi|/\delta)}{N}}
\]

statistical error: $\epsilon$
Summary So Far:

By reduction to Supervised Learning (i.e., classification using $\Pi$ or Regression using $\mathcal{F}$), with high probability, we get:

$$\mathbb{E}_{s \sim d_\mu^t} \left[ A^\pi(s, \hat{\pi}(s)) \right] \geq \max_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^t} \left[ A^\pi(s, \pi(s)) \right] - \frac{A}{1 - \gamma} \sqrt{\frac{\ln(|\Pi|/\delta)}{N}}$$

statistical error: $\epsilon$

In the rest of the lecture, as we will focus on convergence rather than sample complexity, we ignore the statistical error (goes to zero as $N$ increases),
Summary So Far:

By reduction to Supervised Learning (i.e., classification using $\Pi$ or Regression using $\mathcal{F}$), with high probability, we get:

$$\mathbb{E}_{s \sim d_{\mu}} \left[ A^\pi(s, \hat{\pi}(s)) \right] \geq \max_{\pi \in \Pi} \mathbb{E}_{s \sim d_{\mu}} \left[ A^\pi(s, \pi(s)) \right] - \frac{A}{1 - \gamma} \sqrt{\frac{\ln(|\Pi|/\delta)}{N}}$$

statistical error: $\epsilon$

In the rest of the lecture, as we will focus on convergence rather than sample complexity, we ignore the statistical error (goes to zero as $N$ increases),

i.e., we assume we can do the exact greedy policy selector: $\arg \max_{\pi \in \Pi} \mathbb{E}_{s \sim d_{\mu}} \left[ A^\pi(s, \pi(s)) \right]$