Policy Iteration and approximations

Rollout, Monte-Carlo Tree Search, and Approximate policy iteration

Daniel Russo

April 6, 2020

Columbia University

Table of Contents

Policy iteration basics

Policy improvement with real-time lookahead

Intro to rollout

Ideas to make rollout work in practice

When does policy iteration converge rapidly?

Approximate policy iteration

Policy iteration

Policy iteration solves the $\min_\mu J_\mu$ *by solving a sequence of single period problems* $\mu_{k+1} \in \operatorname{argmin}_\mu T_\mu J_{\mu_k}$

- For $k = 0, 1, 2 \cdots$
    1. Policy evaluation:

    $$J_{\mu_k} = g_{\mu_k} + \alpha P_{\mu_k} J_{\mu_k}$$

    2. Policy improvement: $\mu_{k+1} \in G(J_{\mu_k}) = \{\mu : T_\mu J_{\mu_k} = TJ_{\mu_k}\}$, i.e.

    $$\mu_{k+1}(s) \in \operatorname*{argmin}_{u \in U(s)} g_{\mu_k}(s, u) + \alpha \sum_{s' \in \mathcal{S}} P_{s,s'}(u) J_{\mu_{k+1}}(s') \qquad \forall s \in \mathcal{S}$$

Policy iteration with Q functions

Define the state-action cost-to-go function

$$Q_\mu(s,u) = g(s,u) + \alpha \sum_{s'} P_{ss'}(u) J_\mu(s')$$

This satisfies the Bellman equation:

$$Q_\mu(s,u) = g(s,u) + \alpha \sum_{s'} P_{ss'}(u)) Q_\mu(s', \mu(s'))$$

Approximate PI:

- For $k = 0, 1, 2 \cdots$
  1. Policy evaluation:

  $$Q_{\mu_k}(s,u) = g(s,u) + \alpha \sum_{s'} P_{ss'}(u) Q_{\mu_k}(s', \mu_k(s')) \qquad \forall s, u$$

  2. Policy improvement:

  $$\mu_{k+1}(s) \in \operatorname*{argmin}_{u \in U(s)} Q_{\mu_k}(s,u) \qquad \forall s \in \mathcal{S}$$

Policy improvement property

Each step of policy iteration produces and improved policy, and the improvement is strict until an optimal policy is reached:

$$J_{\mu_{k+1}} = J_{\mu_k} \iff J_{\mu_k} = TJ_{\mu_k} \iff J_{\mu_k} = J^*.$$

Lemma $J_{\mu_{k+1}} \preceq TJ_{\mu_k} \preceq J_{\mu_k}$

Proof.
$J_{\mu_k} = T_{\mu_k} J_{\mu_k} \succeq TJ_{\mu_k} = T_{\mu_{k+1}} J_{\mu_k} \succeq T^2_{\mu_{k+1}} J_{\mu_k} \succeq \cdots \succeq J_{\mu_{k+1}}.$ $\square$

Classic convergence analysis of policy iteration

- Assume the state space is finite.
- As long as the current policy is suboptimal, policy iteration produces a strictly better policy.
- Since there are only finitely many policies, *policy iteration will reach an optimal policy within a finite number of iterations*.

A simple convergence rate

Policy iteration is at least as fast as value iteration.

Lemma: $\|J_{\mu_k} - J^*\|_\infty \leq \alpha^k \|J_{\mu_0} - J^*\|_\infty$.

Proof.
Since $J_{\mu_k} \preceq TJ_{\mu_{k-1}}$.

$$J_{\mu_k} - J^* \preceq TJ_{\mu_{k-1}} - J^* = TJ_{\mu_{k-1}} - TJ^*$$

Since both sides are non-negative, taking the max-norm gives

$$\|J_{\mu_k} - J^*\|_\infty \leq \| TJ_{\mu_{k-1}} - TJ^*\|_\infty \leq \alpha \|J_{\mu_{k-1}} - J^*\|.$$

The result follows by induction. $\square$

Policy iteration iteration vs. value iteration

- Each iteration of policy iteration is more costly than value iteration
    - It requires evaluating the cost-to-go function J_{μ_k} (e.g. by solving a linear system, with cost $O(|\mathcal{S}|^3)$.
- Practical experience suggests policy iteration often converges in very few iterations.
    - But this is not necessarily true. Your homework includes a "bad example" for PI where it requires as many iterations as states.
    - We'll look at a bit of theory suggesting why it sometimes converges in few iterations.

Table of Contents

Policy iteration basics

Policy improvement with real-time lookahead

- Intro to rollout
- Ideas to make rollout work in practice

When does policy iteration converge rapidly?

Approximate policy iteration

Table of Contents

Policy iteration basics

Policy improvement with real-time lookahead

- Intro to rollout
- Ideas to make rollout work in practice

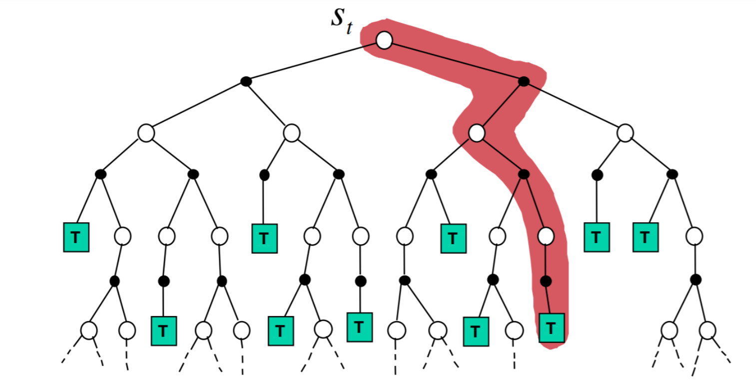
When does policy iteration converge rapidly?

Approximate policy iteration

A success story

Rollout

Choose which action to take at the current state by lookahead.



Rollout (one-period lookahead)

- We have a base policy $\bar{\mu}$.
- At time k, in state s_k, we select

$$u_k \in \operatorname*{argmin}_u Q_{\bar{\mu}}(s_k, u)$$

- But we do this without storing the function Q_μ. How?
    - We can evaluate each control u by simulating many trajectories in which we first apply u and apply $\bar{\mu}$ thereafter:

$$\begin{aligned} Q_\mu(s,u) &= g(s,u) + \alpha \sum_{s' \in \mathcal{S}} P_{ss'}(u) J_\mu(s') \\ &= \mathbb{E}\left[\sum_{k=0}^{\infty} \alpha^k g(s_k, u_k) : u_0 = u, s_0 = s,\ u_k = \bar{\mu}(s_k)\ k > 0\right] \end{aligned}$$

- This is done at decision-time, once s_k is observed.
    - No need to compute a full policy iteration update. Just lookahead in the current subproblem.

Policy improvement property

Rollout can only improve the base policy.

- A single period Rollout is a policy iteration update
  - If at decision time, we apply single period rollout to the base policy $\bar{\mu}$, then our decision policy is the policy iteration update $\mu^+ \in G(J_{\bar{\mu}})$.
- It follows that $J_{\mu^+} \preceq TJ_{\bar{\mu}} \preceq J_{\bar{\mu}}$.

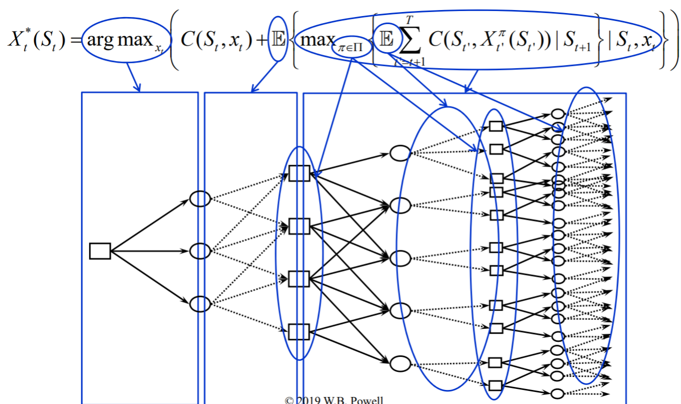
Multi-period rollout

At time k apply the control that is optimal under m-step lookahead:

$$u_k = \operatorname*{argmin}_{u \in U(s_k)} g(s_k, u) + \alpha \sum_{s'} P_{s_k, s'}(u) T^m J_{\bar{\mu}}(s')$$

$$= \operatorname*{argmin}_{u \in U(s_k)} g(s_k, u) + \alpha \sum_{s'} P_{s_k, s'}(u) \min_{\mu_1, \cdots \mu_m} T_{\mu_1} \cdots T_{\mu_m} J_{\bar{\mu}}(s')$$

Picture from Warren Powell. . . uses different notation. . .



Multi-period rollout

Multi-period rollout involves nested single period rollouts.
It's feasible when the ("branching factor") number of actions and successor states is not too large.

View 1-period rollout as an approximation to $\min_u Q_{\bar{\mu}}(s, u)$
We could in principle querry this algorithm many times.

Single Period Rollout

Input: $\bar{\mu}$, current state s
Approximate $\hat{Q}(s, u) \approx Q_{\mu}(s, u)$ for all $u \in U(S)$ by simulation.
Return: $\operatorname{argmin}_u \hat{Q}(s, u)$, $\min_u \hat{Q}(s, u)$

Computing a two-period rollout

At time k we want to apply the control

$$\begin{aligned} u_k &= \operatorname*{argmin}_{u \in U(s_k)} g(s_k, u) + \alpha \sum_{s'} P_{s_k, s'}(u) \min_{\mu} T_\mu J_{\bar{\mu}}(s') \\ &= \operatorname*{argmin}_{u \in U(s_k)} g(s_k, u) + \alpha \sum_{s'} P_{s_k, s'}(u) \min_{u; \in U(s')} Q_{\bar{\mu}}(s', u) \end{aligned}$$

Simulation based approximation

1. For each $u \in U(s_k)$. Draw N successor states $s_u^1, \cdots s_u^N$
2. For each *unique* successor state sampled s' approximate $\min_{u'} \hat{Q}_{\bar{\mu}}(s', u')$ by single-period rollout.
3. Find $\operatorname{argmin}_{u \in U(s_k)} N^{-1} \sum_{i=1}^{N} \left[g(s_k, u) + \alpha \min_{u'} \hat{Q}_{\bar{\mu}}(s_u^i, u') \right]$

Multi-period policy improvement property

At time k apply the control that is optimal under $m+1$-step lookahead:

$$\mu_R(s_k) = \operatorname*{argmin}_{u \in U(s_k)} g(s_k, u) + \alpha \sum_{s'} P_{s_k, s'}(u) T^m J_{\bar{\mu}}(s')$$

The rollout policy μ_R satisfies $\mu_R \in G(T^m J_{\bar{\mu}})$, i.e

$$T_{\mu_R}(T^m J_{\bar{\mu}}) = T(T^m J_{\bar{\mu}})$$

Akin to the analysis of policy iteration, one can show:

$$J_{\mu_R} \preceq T^{m+1} J^{\bar{\mu}} \preceq J^{\bar{\mu}}$$

and

$$\|J_{\mu_R} - J^*\|_\infty \leq \alpha^{m+1} \|J_{\bar{\mu}} - J^*\|_\infty.$$

Table of Contents

Policy iteration basics

Policy improvement with real-time lookahead

- Intro to rollout
- Ideas to make rollout work in practice

When does policy iteration converge rapidly?

Approximate policy iteration

Making this work in practice (1): Truncated rollouts with cost-to-go approximations

It is common to use K-period simulation with terminal values given by a cost-to-go approximation.

Given approximate cost-to-go function $\hat{J}_\theta \approx J_{\bar{\mu}}$,

$$
\begin{aligned}
Q_\mu(s,u) &= g(s,u) + \alpha \sum_{s' \in \mathcal{S}} P_{ss'}(u) J_\mu(s') \\
&= \mathbb{E}\left[\sum_{k=0}^{\infty} \alpha^k g(s_k,u_k) : u_0 = u, s_0 = s,\ u_k = \bar{\mu}(s_k)\ k > 0\right] \\
&\approx \mathbb{E}\left[\sum_{k=0}^{K-1} \alpha^k g(s_k,u_k) + \alpha^K \hat{J}_\theta(s_K) \mid u_0 = u, s_0 = s,\ u_k = \bar{\mu}(s_k)\right]
\end{aligned}
$$

We can approximate the final expectation by simulation.

Making this work in practice (2:)
Planning with an approximate model

As an example, consider the nolinear continuous control problem:

$$\begin{aligned} \min \quad & \mathbb{E} \sum_{k=0}^{\infty} \alpha^k \left[s_k^\top Q s_K + u_k^\top R u_k \right] \\ \text{subject to} \quad & s_{k+1} = f(x_k, u_k, w_k) \\ & u_k \in U(s_k) \end{aligned}$$

<u>**Model predictive control:**</u> To compute u_k given current state s_k

- Create a locally linear, certainty equivalent (i.e. no w_k) model: $s_{t+1} \approx A s_t + B u_t \quad \text{for } s_t \approx s_k$.
- Return the u_k by solving for a sequence of controls:

$$\begin{aligned} \min_{u_k, \cdots u_M} \quad & \mathbb{E} \sum_{t=k}^{N+k} \alpha^t \left[s_t^\top Q s_t + u_t^\top R u_t \right] \\ \text{subject to} \quad & s_{t+1} = A s_t + B u_t,\ \ s_t = s_k,\ \ s_{k+N} = 0,\ \ u_t \in U(s_t) \end{aligned}$$

Making this work in practice (3:) Selective search depth

Monte Carlo Tree Search

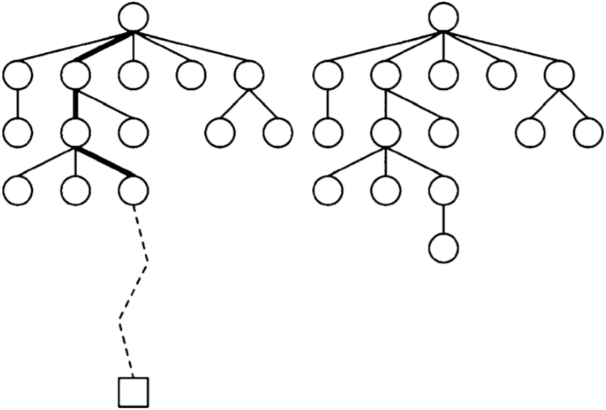


Table of Contents

Policy iteration basics

Policy improvement with real-time lookahead

Intro to rollout

Ideas to make rollout work in practice

When does policy iteration converge rapidly?

Approximate policy iteration

Policy iteration is faster than value iteration? Connection with Newton's method

PI is Newton's method applied to solving Bellman's equation.

Define the gap in Bellman's equation:

$$B(J) = J - TJ$$

Assuming differentiability, Newton's method is

$$J_{k+1} = J_k - [\nabla B(J_k)]^{-1} B(J_k)$$

Newton's method converges quadratically. That is, for J_k sufficiently close to J^*, we should have

$$\|J_{k+1} - J^*\| \leq C\|J_k - J^*\|^2.$$

Asymptotically faster than the linear rate under value iteration.

Connection with Newton's iteration (a reminder)

Recall this variational form of Bellman's equation:

Lemma: For any $J \in \mathbb{R}^n$ and policy μ,

$$J - J_\mu = (I - \alpha P_\mu)^{-1}(J - T_\mu J).$$

Connection with Newton's iteration (continued)

Fix J and suppose there is a unique greedy policy $G(J) = \{\mu\}$.

For some sufficiently small ϵ

$$\begin{aligned} \|J' - J\| \leq \epsilon &\implies G(J') = \{\mu\} \\ &\implies B(J') = J' - T_\mu J' = (I - \alpha P_\mu) J' + g_\mu. \end{aligned}$$

We find

$$\nabla B(J) = (I - \alpha P_\mu)$$

A Newton step to J produces J_μ:

$$\begin{aligned} J - [\nabla B(J)]^{-1} B(J) &= J - (I - \alpha P_\mu)^{-1} (J - T_\mu J) \\ &= J - (J - J_\mu) \\ &= J_\mu \end{aligned}$$

Hence $\{J_{\mu_k}\}$ is the sequence produced by Newton's method with initial iterate J_{μ_0}.

When is policy iteration fast? ...an attempt

Each policy iteration step makes enormous progress if that policy visits states with frequency similar to an optimal policy.

Let $u = (1/|\mathcal{S}|, \cdots, 1/|\mathcal{S}|)$ be the uniform distribution.
Recall the discounted state occupancy measure

$$d_\infty^\mu = (1-\alpha)u\left(I - \alpha P_\mu\right)^{-1} = (1-\alpha)\sum_{t=0}^{\infty} \alpha^t u P_\mu^t$$

Define the distribution shift constant

$$c_k = \left\| \frac{d_\infty^{\mu^*}}{d_\infty^{\mu_{k+1}}} \right\|_\infty = \max_s \frac{d_\infty^{\mu^*}(s)}{d_\infty^{\mu_{k+1}}(s)}$$

Proposition: $c_k \leq |\mathcal{S}|/(1-\alpha)$ and

$$\left\|J_{\mu_{k+1}} - J^*\right\|_1 \preceq \left(1 - c_k^{-1}\right) \left\|J_{\mu_k} - J^*\right\|_1$$

When is policy iteration fast?...an attempt (2)

A more precisely quantification of policy improvement.

Lemma: $J_{\mu_k} - J_{\mu_{k+1}} = (I - \alpha P_{\mu_{k+1}})^{-1} (J_{\mu_k} - TJ_{\mu_k})$

Proof.
Apply the variational Bellman eq w/ $J \equiv J_{\mu_k}$ and $\mu \equiv \mu_{k+1}$:

$$\begin{aligned} J_{\mu_k} - J_{\mu_{k+1}} &= (I - \alpha P_{\mu_{k+1}})^{-1} (J_{\mu_k} - T_{\mu_{k+1}} J_{\mu_k}) \\ &= (I - \alpha P_{\mu_{k+1}})^{-1} (J_{\mu_k} - TJ_{\mu_k}) \end{aligned}$$

$\square$

Heuristically at least, this is suggestive of much faster convergence than value iteration:

- $J_{\mu_k} - TJ_{\mu_k}$ is on the order of $\alpha(J_{\mu_k} - J^*)$.
- $(I - \alpha P_{\mu_k})^{-1}$ is on the order of $(1-\alpha)^{-1}$ if $(I - \alpha P_{\mu_k})^{-1}$ is fairly uniform.

When is policy iteration fast?...an attempt (3)

Lemma: $J_{\mu_k} - J_{\mu_{k+1}} = (I - \alpha P_{\mu_{k+1}})^{-1}(J_{\mu_k} - TJ_{\mu_k})$

Lemma: $J_{\mu_k} - J^* \preceq (I - \alpha P_{\mu^*})^{-1}(J_{\mu_k} - TJ_{\mu_k})$

Proof.

$$\begin{aligned} J_{\mu_k} - J_{\mu^*} &= (I - \alpha P_{\mu^*})^{-1}(J_{\mu_k} - T_{\mu^*}J_{\mu_k}) \\ &\preceq (I - \alpha P_{\mu^*})^{-1}(J_{\mu_k} - TJ_{\mu_k}). \end{aligned}$$

$\square$

When is policy iteration fast?...an attempt (4)

Lemma: $J_{\mu_k} - J_{\mu_{k+1}} = (I - \alpha P_{\mu_{k+1}})^{-1} (J_{\mu_k} - TJ_{\mu_k})$

Lemma: $J_{\mu_k} - J^* \preceq (I - \alpha P_{\mu^*})^{-1} (J_{\mu_k} - TJ_{\mu_k})$

Proof of Proposition: The definition of c_k gives

$$c_k u(I - \alpha P_{\mu_{k+1}})^{-1} \succeq u(I - \alpha P_{\mu^*})^{-1}.$$

Let $e = (1, 1, \cdots, 1)$ be a column vector of 1's.

$$\begin{aligned}
\|J_{\mu_{k+1}} - J^*\|_1 &= e^\top (J_{\mu_{k+1}} - J^*) \\
&= e^\top \left[J_{\mu_k} - J^* - (I - \alpha P_{\mu_{k+1}})^{-1} (J_{\mu_k} - TJ_{\mu_k}) \right] \\
&\preceq e^\top [J_{\mu_k} - J^*] - c_k^{-1} e^\top \left[(I - \alpha P_{\mu^*})^{-1} (J_{\mu_k} - TJ_{\mu_k}) \right] \\
&\preceq \left(1 - c_k^{-1}\right) e^\top (J_{\mu_k} - J^*) \\
&= \left(1 - c_k^{-1}\right) \|J_{\mu_k} - J^*\|_1
\end{aligned}$$

Table of Contents

Policy iteration basics

Policy improvement with real-time lookahead

- Intro to rollout
- Ideas to make rollout work in practice

When does policy iteration converge rapidly?

Approximate policy iteration

Policy iteration reminder

- For $k = 0, 1, 2 \cdots$
  1. Evaluate the current policy:

$$J_{\mu_k} = g_{\mu_k} + \alpha P_{\mu_k} J_{\mu_k}$$

  2. Policy improvement: $\mu_{k+1} \in G(J_{\mu_k}) = \{\mu : T_\mu J_{\mu_k} = T J_{\mu_k}\}$:

$$\mu_{k+1}(s) \in \operatorname*{argmin}_{u \in U(s)} g_{\mu_k}(s, u) + \alpha \sum_{s' \in \mathcal{S}} P_{s,s'}(u) J_{\mu_{k+1}}(s') \qquad \forall s \in \mathcal{S}$$

Approximate policy iteration

- For $k = 0, 1, 2 \cdots$
  1. Approximate the cost-to-go under the current policy $J_{\theta_k} \approx J_{\mu_k}$.
  2. Policy improvement: $\mu_{k+1} \in G(J_{\theta_k}) = \{\mu : T_\mu J_{\mu_k} = T J_{\mu_k}\}$:

$$\mu_{k+1}(s) \in \operatorname*{argmin}_{u \in U(s)} g_{\mu_k}(s, u) + \alpha \sum_{s' \in \mathcal{S}} P_{s,s'}(u) J_{\theta_k}(s') \qquad \forall s \in \mathcal{S}$$

Approximate Policy iteration with Q functions

Q functions are typically used instead, because it is easy to apply the one-step policy improvement with respect to a Q estimate.

Define the state-action cost-to-go function

$$Q_\mu(s,u) = g_\mu(u) + \alpha \sum_{s'} P_{ss'}(u) Q_\mu(s', \mu(s'))$$

The policy $\operatorname{argmin}_{u \in U(s)} Q_\mu(s,a)$ is a policy iteration update to μ.

Approximate PI:

- For $k = 0, 1, 2 \cdots$
  1. Approximate the cost-to-go under μ_k: $Q_{\theta_k} \approx Q_{\mu_k}$
  2. Solve for an improved policy

$$\mu_{k+1}(s) \in \operatorname*{argmin}_{u \in U(s)} Q_{\theta_k}(s,u) \qquad \forall s \in \mathcal{S}$$

Q_{μ_k} can be approximated by either TD or Monte Carlo methods.

Monte-carlo Q-function approximation

Suppose $\mu_i(s) = \text{argmin}_{u \in U(s)} Q_{\theta_i}(s, u)$. We want to solve

$$\theta_{i+1} = \min_{\theta} \| Q_\theta - Q_{\mu_i} \|_{2,\nu}^2 = \min_{\theta} \mathbb{E}_{(s,u)\sim\nu} \left[\left(Q_\theta(s,u) - Q_{\mu_i}(s,u) \right)^2 \right].$$

Regression based approximation to Q_μ by simulation μ

For simulation replications $m = 1, 2, \cdots, M$

- Sample $(s_0^m, u_0^m) \sim \nu$ and horizon $H_m \sim \mathrm{Geom}(1-\alpha)$.
- Apply u_0 and observe the next state s_1^m.
- For $k = 1, 2, \cdots, H_m$
  - Apply control $u_k^m = \text{argmin}_u Q_{\theta_i}(s_k^m, u)$ and observe s_{k+1}^m.
- Set $G^m = \sum_{k=0}^{H_m} \alpha^k g_\mu(s_k^m)$

Solve $\min_\theta \frac{1}{M} \sum_{m=1}^{M} \left(Q_\theta(s_0^m, u_0^m) - G^m \right)^2$.

Analysis of approximate PI (J_{θ_i} version)

Lemma: Let $\epsilon = \max_{i \leq k} \|J_{\theta_i} - J_{\mu_i}\|_\infty$. Then

$$\|J_{\mu_k} - J^*\|_\infty \leq \alpha^k \|J_{\mu_0} - J^*\|_\infty + \frac{\alpha\epsilon}{(1-\alpha)^2}$$

Proof $J_{\theta_i} \preceq J_{\mu_i} + \epsilon e$. By the definition of μ_{i+1} and the monotonicity/constant shift properties

$$T_{\mu_{i+1}} J_{\theta_i} = T J_{\theta_i} \preceq T\left(J_{\mu_i} + \epsilon e\right) = T J_{\mu_i} + \alpha\epsilon e.$$

Applying $T_{\mu_{i+1}}$ repeatedly to each side gives

$$J_{\mu_{i+1}} \preceq T J_{\mu_i} + \frac{\alpha\epsilon}{1-\alpha}$$

Subtracting J^* and the LHS and TJ^* on the RHS gives

$$\|J_{\mu_{i+1}} - J^*\|_\infty \leq \|T J_{\mu_i} - T J^*\|_\infty + \frac{\epsilon\alpha}{1-\alpha} \leq \alpha \|J_{\mu_i} - J^*\|_\infty + \frac{\epsilon\alpha}{1-\alpha}$$

We now apply this recursively.

Policy chattering in approximate policy iteration

Approximate policy iteration often does not converge. Instead, it cycles endlessly between a set of policies.

I will illustrate the issue in a simple case with on policy sampling

Example: Suppose we have 1 state with 2 actions. So we write $Q(u) = Q^*(s,u)$ for $u = 1, 2$,
We approximate $Q^*(u) \approx Q_\theta(u) = \Phi(u)\theta$, $\theta \in \mathbb{R}$.

Suppose, $\alpha = 0$, $Q^* = (g(1), g(2)) = (1, -1)$ and $\Phi = (2, 1)$.

- If $\theta > 0$, we think we should play action 1.
- If $\theta < 0$, we think we should play action 2.

Policy chattering continued

Here we consider on-policy sampling, i.e. ν is determined by simulating the current policy.

API with on-policy sampling

Input: $\theta_0,\ u_0 = \arg\max_u(\Phi\theta_0)(a)$

For $k = 0, 1, 2, \ldots$

- Evaluate policy: $\theta_{k+1} = \arg\min_\theta \left((\Phi\theta)(u_k) - Q(u_k)\right)^2$
- Policy improvement: $u_{k+1} = \arg\min_u(\Phi\theta_k)(u)$

So what happens with $(\theta_0 = -\frac{1}{2})$?

$$\theta_0 = \frac{1}{2} \Rightarrow Q_{\theta_0} = (1, \frac{1}{2}) \Rightarrow u_0 = 2$$

$$\Phi(2)\theta_1 = Q(2) \Rightarrow \theta_1 = -1 \Rightarrow u_1 = \arg\min_u Q_{\theta_1}(u) = 1$$

$$\Phi(1)\theta_2 = Q(1) \Rightarrow \theta_2 = \frac{1}{2} \Rightarrow u_2 = \arg\min_u Q_{\theta_2}(u) = 2$$

Policies cycle endlessly.