Iterative epsilon greedy policy improvement

Is epsilon greedy a better policy?

Is epsilon greedy a better policy?

Theorem

For any MDP, $\epsilon - greedy(\pi) \geq \pi$.

Is epsilon greedy a better policy?

Theorem

For any MDP, $\epsilon - greedy(\pi) \geq \pi$ *, if* π *is* $\epsilon - soft.$

Definition ($\epsilon - \mathrm{soft}$ policy)

A policy π is $\epsilon - \mathrm{soft}$ if it takes random actions with a probability greater than or equal to ϵ for all states in the MDP.

Definition ($\epsilon - \mathrm{soft}$ policy)

A policy π is $\epsilon - \mathrm{soft}$ if it takes random actions with a probability greater than or equal to ϵ for all states in the MDP.

Example (Pole direction policy in `CartPole-v0`)

Definition ($\epsilon - \mathrm{soft}$ policy)

A policy π is $\epsilon - \mathrm{soft}$ if it takes random actions with a probability greater than or equal to ϵ for all states in the MDP.

Example (Pole direction policy in `CartPole-v0`)

- In $\pi_{\text{pole direction policy}}$, probability of taking random action is 0.

Definition ($\epsilon-\mathrm{soft}$ policy)

A policy π is $\epsilon-\mathrm{soft}$ if it takes random actions with a probability greater than or equal to ϵ for all states in the MDP.

Example (Pole direction policy in `CartPole-v0`)

- In $\pi_{\text{pole direction policy}}$, probability of taking random action is 0.
- For any $\epsilon > 0$, the probability of taking random actions is **not** $\geq \epsilon$. Therefore, it is **not** $\epsilon-\mathrm{soft}$.

Definition ($\epsilon - \mathrm{soft}$ policy)

A policy π is $\epsilon - \mathrm{soft}$ if it takes random actions with a probability greater than or equal to ϵ for all states in the MDP.

Example (Pole direction policy in `CartPole-v0`)

- In $\pi_{\text{pole direction policy}}$, probability of taking random action is 0.
- For any $\epsilon > 0$, the probability of taking random actions is **not** $\geq \epsilon$. Therefore, it is **not** $\epsilon - \mathrm{soft}$.
- $\epsilon - \mathrm{greedy}(\pi_{\text{pole direction policy}})$ is not guaranteed to give a policy improvement.

Definition ($\epsilon - \mathrm{soft}$ policy)

A policy π is $\epsilon - \mathrm{soft}$ if it takes random actions with a probability greater than or equal to ϵ for all states in the MDP.

Example (Pole direction policy in `CartPole-v0`)

- In $\pi_{\text{pole direction policy}}$, probability of taking random action is 0.
- For any $\epsilon > 0$, the probability of taking random actions is **not** $\geq \epsilon$. Therefore, it is **not** $\epsilon - \mathrm{soft}$.
- $\epsilon - \mathrm{greedy}(\pi_{\text{pole direction policy}})$ is not guaranteed to give a policy improvement.

Example (Random policy in `CartPole-v0`)

Definition ($\epsilon - \text{soft}$ policy)

A policy π is $\epsilon - \text{soft}$ if it takes random actions with a probability greater than or equal to ϵ for all states in the MDP.

Example (Pole direction policy in `CartPole-v0`)

- In $\pi_{\text{pole direction policy}}$, probability of taking random action is 0.
- For any $\epsilon > 0$, the probability of taking random actions is **not** $\geq \epsilon$. Therefore, it is **not** $\epsilon - \text{soft}$.
- $\epsilon - \text{greedy}(\pi_{\text{pole direction policy}})$ is not guaranteed to give a policy improvement.

Example (Random policy in `CartPole-v0`)

- π_{random} takes random actions with probability 1.

Definition ($\epsilon - \mathrm{soft}$ policy)

A policy π is $\epsilon - \mathrm{soft}$ if it takes random actions with a probability greater than or equal to ϵ for all states in the MDP.

Example (Pole direction policy in `CartPole-v0`)

- In $\pi_{\text{pole direction policy}}$, probability of taking random action is 0.
- For any $\epsilon > 0$, the probability of taking random actions is **not** $\geq \epsilon$. Therefore, it is **not** $\epsilon - \mathrm{soft}$.
- $\epsilon - \mathrm{greedy}(\pi_{\text{pole direction policy}})$ is not guaranteed to give a policy improvement.

Example (Random policy in `CartPole-v0`)

- π_{random} takes random actions with probability 1.
- For any $\epsilon \leq 1$, for example $\epsilon = 0.9$, the probability of taking random actions is greater than or equal to ϵ in this policy.

Definition ($\epsilon - \mathrm{soft}$ policy)

A policy π is $\epsilon - \mathrm{soft}$ if it takes random actions with a probability greater than or equal to ϵ for all states in the MDP.

Example (Pole direction policy in `CartPole-v0`)

- In $\pi_{\text{pole direction policy}}$, probability of taking random action is 0.
- For any $\epsilon > 0$, the probability of taking random actions is **not** $\geq \epsilon$. Therefore, it is **not** $\epsilon - \mathrm{soft}$.
- $\epsilon - \mathrm{greedy}(\pi_{\text{pole direction policy}})$ is not guaranteed to give a policy improvement.

Example (Random policy in `CartPole-v0`)

- π_{random} takes random actions with probability 1.
- For any $\epsilon \leq 1$, for example $\epsilon = 0.9$, the probability of taking random actions is greater than or equal to ϵ in this policy.
- π_{random} is $\epsilon - \mathrm{soft}$ for any $\epsilon \leq 1$

Definition ($\epsilon - \mathrm{soft}$ policy)

A policy π is $\epsilon - \mathrm{soft}$ if it takes random actions with a probability greater than or equal to ϵ for all states in the MDP.

Example (Pole direction policy in `CartPole-v0`)

- In $\pi_{\text{pole direction policy}}$, probability of taking random action is 0.
- For any $\epsilon > 0$, the probability of taking random actions is **not** $\geq \epsilon$. Therefore, it is **not** $\epsilon - \mathrm{soft}$.
- $\epsilon - \mathrm{greedy}(\pi_{\text{pole direction policy}})$ is not guaranteed to give a policy improvement.

Example (Random policy in `CartPole-v0`)

- π_{random} takes random actions with probability 1.
- For any $\epsilon \leq 1$, for example $\epsilon = 0.9$, the probability of taking random actions is greater than or equal to ϵ in this policy.
- π_{random} is $\epsilon - \mathrm{soft}$ for any $\epsilon \leq 1$
- In particular, π_{random} is $0.9 - \mathrm{soft}$

$$\pi_{\text{random}} \leq \epsilon - \text{greedy}(\pi_{\text{random}})\big|_{\epsilon=0.9}$$

$$\pi_{\text{random}} \leq \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} \coloneqq \pi_1$$

$$\pi_{\text{random}} \leq \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} := \pi_1$$

Example (π_1 in `CartPole-v0`)

$$\pi_{\text{random}} \leq \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} := \pi_1$$

Example (π_1 in `CartPole-v0`)

- π_1 takes random actions with probability 0.9.

$$\pi_{\text{random}} \leq \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} := \pi_1$$

Example (π_1 in `CartPole-v0`)

- π_1 takes random actions with probability 0.9.
- For any $\epsilon \leq 0.9$, for example $\epsilon = 0.8$, the probability of taking random actions is greater than or equal to ϵ in this policy.

$$\pi_{\text{random}} \le \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} := \pi_1$$

Example (π_1 in `CartPole-v0`)

- π_1 takes random actions with probability 0.9.
- For any $\epsilon \le 0.9$, for example $\epsilon = 0.8$, the probability of taking random actions is greater than or equal to ϵ in this policy.
- π_1 is $\epsilon - \text{soft}$ for any $\epsilon \le 0.9$.

$$\pi_{\text{random}} \leq \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} := \pi_1$$

Example (π_1 in `CartPole-v0`)

- π_1 takes random actions with probability 0.9.
- For any $\epsilon \leq 0.9$, for example $\epsilon = 0.8$, the probability of taking random actions is greater than or equal to ϵ in this policy.
- π_1 is $\epsilon - \text{soft}$ for any $\epsilon \leq 0.9$.
- In particular, π_1 is $0.8 - \text{soft}$

$$\begin{aligned}\pi_{\text{random}} &\leq \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} \coloneqq \pi_1 \\ &\leq \epsilon - \text{greedy}(\pi_1)|_{\epsilon=0.8} \coloneqq \pi_2\end{aligned}$$

$$
\begin{aligned}
\pi_{\text{random}} &\leq \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} := \pi_1 \\
&\leq \epsilon - \text{greedy}(\pi_1)|_{\epsilon=0.8} := \pi_2 \\
&\leq \epsilon - \text{greedy}(\pi_2)|_{\epsilon=0.7} := \pi_3 \\
&\ldots
\end{aligned}
$$

Under which condition does iterative greedy policy improvement converge to the optimal policy?

$$
\begin{aligned}
\pi_{\text{random}} &\leq \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} := \pi_1 \\
&\leq \epsilon - \text{greedy}(\pi_1)|_{\epsilon=0.8} := \pi_2 \\
&\leq \epsilon - \text{greedy}(\pi_2)|_{\epsilon=0.7} := \pi_3 \\
&\dots \\
&= \pi_*
\end{aligned}
$$

Under which condition does iterative greedy policy improvement converge to the optimal policy?

$$
\begin{aligned}
\pi_{\text{random}} &\leq \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} := \pi_1 \\
&\leq \epsilon - \text{greedy}(\pi_1)|_{\epsilon=0.8} := \pi_2 \\
&\leq \epsilon - \text{greedy}(\pi_2)|_{\epsilon=0.7} := \pi_3 \\
&\dots \\
&= \pi_*
\end{aligned}
$$

Theorem (Greedy in the Limit of Infinite Exploration (GLIE) guarantees convergence)

Under which condition does iterative greedy policy improvement converge to the optimal policy?

$$
\begin{aligned}
\pi_{\text{random}} &\leq \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} := \pi_1 \\
&\leq \epsilon - \text{greedy}(\pi_1)|_{\epsilon=0.8} := \pi_2 \\
&\leq \epsilon - \text{greedy}(\pi_2)|_{\epsilon=0.7} := \pi_3 \\
&\dots \\
&= \pi_*
\end{aligned}
$$

Theorem (Greedy in the Limit of Infinite Exploration (GLIE) guarantees convergence)

$$
\lim_{t \to \infty} \texttt{visit\_number}[(s, a)] \to \infty
$$

Under which condition does iterative greedy policy improvement converge to the optimal policy?

$$
\begin{aligned}
\pi_{\text{random}} &\le \epsilon - \text{greedy}(\pi_{\text{random}})|_{\epsilon=0.9} := \pi_1 \\
&\le \epsilon - \text{greedy}(\pi_1)|_{\epsilon=0.8} := \pi_2 \\
&\le \epsilon - \text{greedy}(\pi_2)|_{\epsilon=0.7} := \pi_3 \\
&\dots \\
&= \pi_*
\end{aligned}
$$

Theorem (Greedy in the Limit of Infinite Exploration (GLIE) guarantees convergence)

$$\lim_{t\to\infty} \texttt{visit\_number}[(s, a)] \to \infty$$

$$\lim_{t\to\infty} \epsilon \to 0$$

In practice, it is not possible to satisfy the GLIE condition

In practice, it is not possible to satisfy the GLIE condition

- Computer programs cannot run for infinite time.

In practice, it is not possible to satisfy the GLIE condition

- Computer programs cannot run for infinite time.
- Even human life is limited!

In practice, it is not possible to satisfy the GLIE condition

- Computer programs cannot run for infinite time.
- Even human life is limited!

In practice, we do the following

In practice, it is not possible to satisfy the GLIE condition

- Computer programs cannot run for infinite time.
- Even human life is limited!

In practice, we do the following

- Choose a finite number of policy improvement steps e.g. 10000 policy improvement steps.

In practice, it is not possible to satisfy the GLIE condition

- Computer programs cannot run for infinite time.
- Even human life is limited!

In practice, we do the following

- Choose a finite number of policy improvement steps e.g. 10000 policy improvement steps.
- At each policy improvement step, slightly reduce ϵ until it is nearly 0 at the 10000^{th} step.

In practice, it is not possible to satisfy the GLIE condition

- Computer programs cannot run for infinite time.
- Even human life is limited!

In practice, we do the following

- Choose a finite number of policy improvement steps e.g. 10000 policy improvement steps.
- At each policy improvement step, slightly reduce ϵ until it is nearly 0 at the 10000^{th} step.
- Stop at the 10000^{th} policy improvement step and hope that we have converged to the optimal policy.

Epsilon schedule

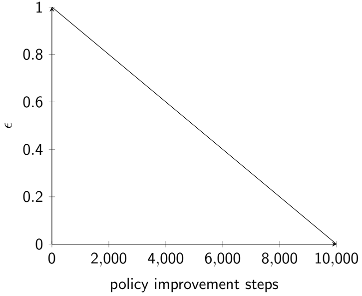

Figure: Linear ϵ schedule

Epsilon schedule

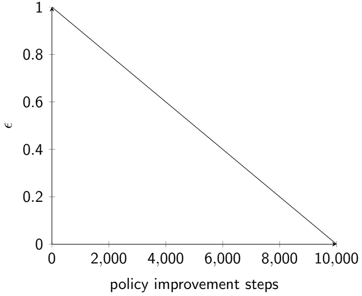

Figure: Linear ϵ schedule

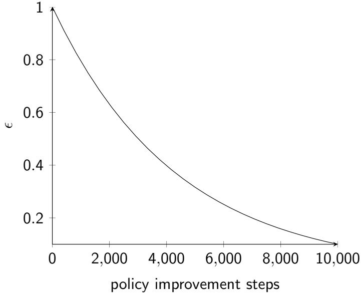

Figure: Exponential ϵ schedule