Model Free Control

Bias Maximization and Double Q-Learning

Marius Lindauer

Winter Term 2021

Maximization Bias

- Consider single-state MDP $(|S| = 1)$ with 2 actions, and both actions have 0-mean random rewards: $(r \mid a = a_1) = (r \mid a = a_2) = 0$
  - assume that reward is stochastic (e.g, $\mathcal{N}(0, 1)$)
- Then $Q(s, a_1) = Q(s, a_2) = 0 = V(s)$
- Assume there are prior samples of taking action a_1 and a_2

Maximization Bias

- Consider single-state MDP ($|S| = 1$) with 2 actions, and both actions have 0-mean random rewards: $(r \mid a = a_1) = (r \mid a = a_2) = 0$
  - assume that reward is stochastic (e.g, $\mathcal{N}(0, 1)$)
- Then $Q(s, a_1) = Q(s, a_2) = 0 = V(s)$
- Assume there are prior samples of taking action a_1 and a_2
- Let $\hat{Q}(s, a_1)$, $\hat{Q}(s, a_2)$ be the finite sample estimate of Q
- Use an unbiased estimator for Q, e.g., $\hat{Q}(s, a_1) = \frac{1}{N(s,a_1)} \sum_{i=1}^{N(s,a_1)} r_i(s, a_1)$

Maximization Bias

- Consider single-state MDP ($|S| = 1$) with 2 actions, and both actions have 0-mean random rewards: $(r \mid a = a_1) = (r \mid a = a_2) = 0$
  - assume that reward is stochastic (e.g, $\mathcal{N}(0, 1)$)
- Then $Q(s, a_1) = Q(s, a_2) = 0 = V(s)$
- Assume there are prior samples of taking action a_1 and a_2
- Let $\hat{Q}(s, a_1)$, $\hat{Q}(s, a_2)$ be the finite sample estimate of Q
- Use an unbiased estimator for Q, e.g., $\hat{Q}(s, a_1) = \frac{1}{N(s,a_1)} \sum_{i=1}^{N(s,a_1)} r_i(s, a_1)$
- Let $\hat{\pi} \in \arg\max_a \hat{Q}(s, a)$ be the greedy policy wrt the estimated $\hat{Q}$

Maximization Bias

- Consider single-state MDP ($|S| = 1$) with 2 actions, and both actions have 0-mean random rewards: $(r \mid a = a_1) = (r \mid a = a_2) = 0$
  - assume that reward is stochastic (e.g, $\mathcal{N}(0,1)$)
- Then $Q(s, a_1) = Q(s, a_2) = 0 = V(s)$
- Assume there are prior samples of taking action a_1 and a_2
- Let $\hat{Q}(s, a_1)$, $\hat{Q}(s, a_2)$ be the finite sample estimate of Q
- Use an unbiased estimator for Q, e.g., $\hat{Q}(s, a_1) = \frac{1}{N(s,a_1)} \sum_{i=1}^{N(s,a_1)} r_i(s, a_1)$
- Let $\hat{\pi} \in \arg\max_a \hat{Q}(s, a)$ be the greedy policy wrt the estimated $\hat{Q}$
- Even though each estimate of the state-action values is unbiased, the estimate of $\hat{\pi}$'s value $\hat{V}^{\hat{\pi}}$ can be biased:

$$
\begin{aligned}
\hat{V}^{\hat{\pi}}(s) &= \mathbb{E}[\max \hat{Q}(s, a_1), \hat{Q}(s, a_2)] \\
&\geq \max[\mathbb{E}[\hat{Q}(s, a_1)], \mathbb{E}[\hat{Q}(s, a_2)]] \\
&= \max[0, 0] = V^{\pi}
\end{aligned}
$$

Double Q-Learning

- The greedy policy w.r.t. estimated Q values can yield a maximization bias during finite-sample learning
- Avoid using max of estimates as estimate of max of true values
- Instead split samples and use to create two independent unbiased estimates of $Q_1(s_1, a_i)$ and $Q_2(s_1, a_i). \forall a \in A$
  - Use one estimate to select max action: $a^* \in \arg\max_{a \in A} Q_1(s_1, a)$
  - Use other estimate to estimate value of a^*: $Q_2(s, a^*)$
  - Yields unbiased estimate: $\mathbb{E}(Q_2(s, a^*)) = Q(s, a^*)$

$\leadsto$ Unbiased estimate of the max state-action value because of independent samples to estimate the value

Double Q-Learning for Full MDP

- Initialization:
  - $Q_1(s, a)$ and $Q_2(s, a)$ $\forall s \in S, a \in A$
  - $t = 0$
  - initial state $s_t = s_0$
- Loop
  - Select a_t using ϵ-greedy $\pi(s) \in \arg\max_{a \in A} Q_1(s_t, a) + Q_2(s_t, a)$
  - Observe (r_t, s_{t+1})
  - With 50-50 probability either
    1. $Q_1(s_t, a_t) \leftarrow Q_1(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in A} Q_2(s_{t+1}, a) - Q_1(s_t, a_t))$
       or
    2. $Q_2(s_t, a_t) \leftarrow Q_2(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in A} Q_1(s_{t+1}, a) - Q_2(s_t, a_t))$
  - $t = t + 1$

Double Q-Learning for Full MDP

- Initialization:
  - $Q_1(s, a)$ and $Q_2(s, a)$ $\forall s \in S, a \in A$
  - $t = 0$
  - initial state $s_t = s_0$
- Loop
  - Select a_t using ϵ-greedy $\pi(s) \in \arg\max_{a \in A} Q_1(s_t, a) + Q_2(s_t, a)$
  - Observe (r_t, s_{t+1})
  - With 50-50 probability either
    1. $Q_1(s_t, a_t) \leftarrow Q_1(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in A} Q_2(s_{t+1}, a) - Q_1(s_t, a_t))$
       or
    2. $Q_2(s_t, a_t) \leftarrow Q_2(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in A} Q_1(s_{t+1}, a) - Q_2(s_t, a_t))$
  - $t = t + 1$

$\rightsquigarrow$ Doubles the memory, same computation requirements, data requirements are subtle – might reduce amount of exploration needed due to lower bias

Double Q-Learning [Sutton & Barto 2018]