Model Free Control

Exploration

Marius Lindauer

Leibniz Universität Hannover

Automated Machine Learning Hannover

Recap Model-free Policy Iteration

- Initialize policy π
- Repeat:
  - Policy evaluation: compute Q^π
  - Policy improvement: update π given Q^π

Recap Model-free Policy Iteration

- Initialize policy π
- Repeat:
  - Policy evaluation: compute Q^π
  - Policy improvement: update π given Q^π

- May need to policy evaluation
  - If π is deterministic, we may not observe all possible actions A in a state s
  - So, we cannot compute $Q(s, a)$ for any $a \neq \pi(s)$

Recap Model-free Policy Iteration

- Initialize policy π
- Repeat:
  - Policy evaluation: compute Q^π
  - Policy improvement: update π given Q^π

- May need to policy evaluation
  - If π is deterministic, we may not observe all possible actions A in a state s
  - So, we cannot compute $Q(s, a)$ for any $a \neq \pi(s)$

$\rightsquigarrow$ How to interleave policy evaluation and improvement?

Policy Evaluation with Exploration

- Want to compute a model-free estimate of Q^{π}
- In general seems subtle
  - Need to try all (s, a) pairs but then follow π
  - Want to ensure resulting estimate Q^{π} is good enough so that policy improvement is a monotonic operator
- For certain classes of policies can ensure all (s, a) pairs are tried such that asymptotically Q^{π} converges to the true value

ϵ-greedy Policies

- Simple idea to balance exploration and exploitation
- Let $|A|$ be the number of actions
- Then a ϵ-greedy policy wrt a state-action value $Q(s, a)$ is $\pi(a \mid s) \in$
  - $\arg\max_{a \in A} Q(s, a)$ with probability $1 - \epsilon$
  - a random action with probability ϵ

Monotonic ϵ-greedy Policy Improvement

- Theorem: For any ϵ-greedy policy π_i, the ϵ-greedy policy wrt Q_i^π is a monotonic improvement $V^{\pi_{i+1}} \geq V^{\pi_i}$

$$
\begin{aligned}
Q^\pi(s, \pi_{i+1}(s)) &= \sum_{a \in A} \pi_{i+1}(a \mid s) Q^{\pi_i}(s,a) \\
&= (\epsilon/|A|) \left[\sum_{a \in A} Q^{\pi_i}(s,a) \right] + (1-\epsilon) \max_{a \in A} Q^{\pi_i}(s,a) \\
&= (\epsilon/|A|) \left[\sum_{a \in A} Q^{\pi_i}(s,a) \right] + (1-\epsilon) \max_{a \in A} Q^{\pi_i}(s,a) \frac{1-\epsilon}{1-\epsilon} \\
&= (\epsilon/|A|) \left[\sum_{a \in A} Q^{\pi_i}(s,a) \right] + (1-\epsilon) \max_{a \in A} Q^{\pi_i}(s,a) \sum_{a \in A} \frac{\pi_i(a \mid s) - \frac{\epsilon}{|A|}}{1-\epsilon} \\
&\geq (\epsilon/|A|) \left[\sum_{a \in A} Q^{\pi_i}(s,a) \right] + (1-\epsilon) \sum_{a \in A} \frac{\pi_i(a \mid s) - \frac{\epsilon}{|A|}}{1-\epsilon} Q^{\pi_i}(s,a) \\
&= \sum_{a \in A} \pi_i(a \mid s) Q^{\pi_i}(s,a) = V^{\pi_i}(s)
\end{aligned}
$$

Automated Machine Learning Hannover

AutoML.org

Greedy in the Limit of Infinite Exploration (GLIE)

- Definition of GLIE:
  - All state-action pairs are visited an infinite number of times

$$\lim_{i \to \infty} N_i(s, a) \to \infty$$

  - Behavior policy (policy used to act in the world) converges to greedy policy

$$\lim_{i \to \infty} \pi(a \mid s) \to \arg\max_{a \in A} Q(s, a)$$

    with probability 1

Greedy in the Limit of Infinite Exploration (GLIE)

- Definition of GLIE:
  - All state-action pairs are visited an infinite number of times

$$\lim_{i \to \infty} N_i(s, a) \to \infty$$

  - Behavior policy (policy used to act in the world) converges to greedy policy

$$\lim_{i \to \infty} \pi(a \mid s) \to \arg\max_{a \in A} Q(s, a)$$

    with probability 1

- Simple Strategy:
  - ϵ-greedy where ϵ is annealed to 0 with $\epsilon_i = 1/i$

Automated Machine Learning Hannover

Greedy in the Limit of Infinite Exploration (GLIE)

- Definition of GLIE:
  - All state-action pairs are visited an infinite number of times

$$\lim_{i \to \infty} N_i(s, a) \to \infty$$

  - Behavior policy (policy used to act in the world) converges to greedy policy

$$\lim_{i \to \infty} \pi(a \mid s) \to \arg\max_{a \in A} Q(s, a)$$

    with probability 1
- Simple Strategy:
  - ϵ-greedy where ϵ is annealed to 0 with $\epsilon_i = 1/i$
- Theorem:
  - GLIE Monte-Carlo control converges to the optimal state-action value function $Q(s, a) \to Q^*(s, a)$

Automated Machine Learning Hannover