Model Free Control

SARSA and Q-Learning

Marius Lindauer

Leibniz Universität Hannover

Automated Machine Learning Hannover

Model-free Policy Iteration with TD Methods

- Use temporal difference methods for policy evaluation step
- Initialize policy π
- Repeat:
  - Policy evaluation: compute Q^π using temporal difference updating with Q-greedy policy
  - Policy improvement: Same as Monte Carlo policy improvement, set π to ϵ-greedy (Q^π)
- First consider SARSA, which is an on-policy algorithm

General Form of SARSA Algorithm

- Initialization:
  - ϵ-greedy policy
  - $t = 0$
  - initial state $s_t = s_0$
- Loop
  - Take action $a_{t+1} \sim \pi(s_{t+1})$
  - Observe (r_{t+1}, s_{t+2})
  - Update Q given $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$$

  - $\pi(s_t) \in \arg\max_{a \in A} Q(s_t, a)$ with probability $1 - \epsilon$, else random
  - $t = t + 1$

Convergence Properties of SARSA

- Theorem:
- SARSA for finite-state and finite-action MDPs converges to the optimal action-value, $Q(s,a) \to Q^*(s,a)$, under the following conditions:
  1. The policy sequence $\pi_t(a \mid s)$ satisfies the condition of GLIE
  2. The step-sizes α_t satisfy the Robbins-Munro sequence such that

$$\sum_{t=1}^{\infty} \alpha_t = \infty$$

$$\sum_{t=1}^{\infty} \alpha_t^2 < \infty$$

- For example $\alpha_t = \frac{1}{T}$ satisfies the above condition
- Would one want to use a step size choice that satisfies the above in practice? Likely not.

Q-Learning: Learning the Optimal State-Action Value

- SARSA is an on-policy learning algorithm
- SARSA estimates the value of the current behavior policy (policy using to take actions in the world)
- And then updates the policy trying to estimate
- Alternatively, can we directly estimate the value of π^* while acting with another behavior policy π_b?
- Yes! Q-learning, an off-policy RL algorithm
- Key idea: Maintain state-action Q estimates and use to bootstrap– use the value of the best future action
- Recall SARSA

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha((r_t + \gamma Q(s_{t+1}, a_{t+1})) - Q(s_t, a_t))$$

- Q-Learning:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha((r_t + \gamma \max_{a' \in A} Q(s_{t+1}, a')) - Q(s_t, a_t))$$

Q-Learning with ϵ-greedy Exploration

- Initialization:
  - $Q(s, a) . \forall s \in S, a \in A$
  - initial state $s_t = s_0$
- Loop
  - Take action $a_t \sim \pi_b(s_t)$
  - Observe (r_t, s_{t+1})
  - Update Q

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t))$$

  - $\pi(s_t) \in \arg\max_{a \in A} Q(s_t, a)$ with probability $1 - \epsilon$, else random
  - $t = t + 1$

Automated Machine Learning Hannover

Q-Learning with ϵ-greedy Exploration

- Conditions for convergence to Q^*?
  - Visit all (s, a) pairs infinitely often
  - the step-sizes α_t satisfy the Robbins-Munro sequence
  - Note: the algorithm does not have to be greedy in the limit of infinite exploration (GLIE) to satisfy this

- Conditions for convergence to optimal π^*
  - The above requirements to converge to optimal Q^*
  - The algorithm is GLIE