Policy Evaluation

The Big Picture

Marius Lindauer

Leibniz Universität Hannover

Automated Machine Learning Hannover

AutoML.org

Recap I : Markov Decision Process (MDP)

- Markov Decision Process is Markov Reward Process + actions
- Definition of MDP
  - S is a (finite) set of Markov states $s \in S$
  - A is a (finite) set of actions $a \in A$
  - P is dynamics/transition model for each action, that specifies $P(s_{t+1} = s' \mid s_t = s, a_t = a)$
  - R is a reward function $R(s_t = s, a_t = a) = \mathbb{E}[r_r \mid s_t = s, a_t = a]$
    - ★ Sometimes R is also defined based on (s) or on (s, a, s')
  - Discount factor $\gamma \in [0, 1]$
- MDP is tuple (S, A, P, R, γ)

Automated Machine Learning Hannover

AutoML.org

Recap I : Markov Decision Process (MDP)

- Markov Decision Process is Markov Reward Process + actions
- Definition of MDP
  - S is a (finite) set of Markov states $s \in S$
  - A is a (finite) set of actions $a \in A$
  - P is dynamics/transition model for each action, that specifies $P(s_{t+1} = s' \mid s_t = s, a_t = a)$
  - R is a reward function $R(s_t = s, a_t = a) = \mathbb{E}[r_r \mid s_t = s, a_t = a]$
    - ★ Sometimes R is also defined based on (s) or on (s, a, s')
  - Discount factor $\gamma \in [0, 1]$
- MDP is tuple (S, A, P, R, γ)

$\rightsquigarrow$ Unfortunately, we often do not have access to true MDP models

AutoML.org Automated Machine Learning Hannover

Recap II

- Definition of Return G_t (for a MRP)
  - Discounted sum of rewards from time step t to horizon

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots$$

Automated Machine Learning Hannover

Recap II

- Definition of Return G_t (for a MRP)
  - Discounted sum of rewards from time step t to horizon

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots$$

- Definition of State Value Function $V^\pi(s)$
  - Expected return from starting in state s under policy π

$$V^\pi(s) = \mathbb{E}[G_t \mid s_t = s] = \mathbb{E}_\pi[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots \mid s_t = s]$$

Automated Machine Learning Hannover

Recap II

- Definition of Return G_t (for a MRP)
  - Discounted sum of rewards from time step t to horizon

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots$$

- Definition of State Value Function $V^\pi(s)$
  - Expected return from starting in state s under policy π

$$V^\pi(s) = \mathbb{E}[G_t \mid s_t = s] = \mathbb{E}_\pi[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots \mid s_t = s]$$

- Definition of State-Action Value Function $Q^\pi(s, a)$
  - Expected return from starting in state s, taking action a and then following policy π

$$\begin{aligned} Q^\pi(s, a) &= \mathbb{E}_\pi[G_t \mid s_t = s, a_t = a] \\ &= \mathbb{E}_\pi[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots \mid s_t = s, a_t = a] \end{aligned}$$

AutoML.org Automated Machine Learning Hannover

Goal for this week

- Assumption: We don't have the exact model of the environment (i.e., model-free), but we can query the environment ("playing roll-outs")
  - state space and action space are in principle known
  - we don't know the transition probabilities beforehand
  - we don't know the reward distribution beforehand

Goal for this week

- Assumption: We don't have the exact model of the environment (i.e., model-free), but we can query the environment ("playing roll-outs")
  - state space and action space are in principle known
  - we don't know the transition probabilities beforehand
  - we don't know the reward distribution beforehand

- Remarks:
  - If we would know the MDP, we only have to do "planning" to find the optimal policy
  - If we first learn the MDP and then apply planning to the learned MDP, we do "model-based" RL (not today!)

Goal for this week

- Assumption: We don't have the exact model of the environment (i.e., model-free), but we can query the environment ("playing roll-outs")
  - state space and action space are in principle known
  - we don't know the transition probabilities beforehand
  - we don't know the reward distribution beforehand
- Remarks:
  - If we would know the MDP, we only have to do "planning" to find the optimal policy
  - If we first learn the MDP and then apply planning to the learned MDP, we do "model-based" RL (not today!)
- Goal for this week: We want to learn $V^\pi(s)$ or $Q^\pi(s,a)$ (depending on the RL algorithm we want to use) by only querying the unknown MDP