Finite Markov Decision Processes (MDPs)

Tuesday, June 16, 2020 2:50 PM

In Markov Decision Processes (MDP's) actions influence not just immediate rewards, but also subsequent situations and through those, future rewards
In MDP's we estimate the value $q_*(s,a)$ of each action a in each state s, or we estimate the value $v_*(s)$ of each state given optimal action selections

Agent:

The learner and decision maker in a MDP

Environment:

The thing the agent interacts with in an MDP (everything outside the agent)

This can also be considered as everything that cannot be changed arbitrarily changed/controlled by the agent

Markov Decision Process:

The agent and environment interact at each of a sequence of discrete time steps. At each time step the agent receives a representation of the environment's state and selects an action. One time step later the agent receives a numerical reward and finds itself in a new state

Trajectory:

A sequence detailing all states, rewards, and actions taken in a MDP

Form:

$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \cdots$

Where S_t corresponds to the state seen by the agent at time step t, A_t corresponds to the action taken by the agent at time step t and R_t corresponds to the reward received by the agent time step t

Finite MDP:

A MDP where the sets of states, actions, and rewards all have a finite number of elements

Dynamics Function of a Finite MDP:

$p(s', r|s, a) \equiv \Pr\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\}$

This is the probability of transition to state s' with reward r from state s taking action a

This is a statement that the variables R_t and S_t have well defined discrete probability distributions dependent only on the preceding state and action

Probability Summation:

$$\sum_{s' \in S} \sum_{r \in R} p(s', r|s, a) = 1$$

State Transition Probability:

$$p(s'|\, s, a) \equiv \Pr\{S_t = s'|S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in R} p(s', r|s, a)$$

Expected Rewards:

$$r(s, a) \equiv E[R_t|S_{t-1} = s, A_{t-1} = a] = \sum_{r \in R} r \sum_{s' \in S} p(s', r|s, a)$$

Expected Rewards for State-Action-Next State Triples:

$$r(s, a, s') \equiv E[R_t|S_{t-1} = s, A_{t-1} = a, S_t = s'] = \sum_{r \in R} r \frac{p(s', r|s, a)}{p(s'|s, a)}$$

Markov Property:

The state including all aspects of the past agent-environment interaction that make a difference for the future

Note that this is a restriction on the state

Reward Hypothesis:

All of what we mean by goals and purposes can be well thought of as the maximization of the

expected value of the cumulative sum of a received scalar signal

Return:

$G_t \equiv f(R_{t+1}, R_{t+2}, \cdots R_T)$

Where R_x is the reward received at time step x and T is the final state which may be infinity for continuous tasks

In general a reinforcement learning agent seeks to maximize the expected return

Episodic Tasks Conventional Return:

$$G_t \equiv \sum_{k=0}^{\infty} R_{t+k}$$

Episodic tasks are tasks that break easily into episodes

This is not the only way return can be formulated for episodic tasks, but is the most common

Discounted Returns:

$$G_t \equiv \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = \gamma^0 R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

Where $0 \leq \gamma \leq 1$ is the discount rate

If the discount rate is less than 1, then the infinite sum has a finite value

Incremental Return Computation:

$G_t \equiv R_{t+1} + \gamma G_{t+1}$

Note that this works for all time steps $t < T$, even the step before termination with the definition $G_T = 0$

Episodic tasks can be made to fit the more general continuous framework by calling the final state of the episode the terminal state and setting it to have 0 reward and only map to itself

Policy:

$\pi(a \mid s)$

This is the probability of selecting action a given state s

In general a policy is a mapping from states to probabilities of selecting each possible action

Alternate Definition:

A stochastic rule by which the agent selects actions as a function of states

State-Value Function:

$$v_\pi(s) \equiv E_\pi[G_t | S_t = s] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|S_t = s\right]$$

This is the expected return when starting in state s and following policy π thereafter

Note that the value of the terminal state (if there is a terminal state) is always 0

Bellman Equation:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')]$$

Note that this is a consistency condition between the value of s and the value of its possible successor states

Action-Value Function:

$$q_\pi(s, a) \equiv E[G_t | S_t = s, A_t = a] = E\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]$$

This is the expected return of starting in state s taking action a and following policy π thereafter

Monte Carlo Methods:

Methods to estimate the value of each action by keeping an average of the reward received for taking each action in the corresponding state

Optimal Policy:

The policy that has an expected return greater than or equal to all other policies (maximizes expected return)

Optimal State-Value Function:

$v_*(s) \equiv \max_{\pi} v_\pi(s)$

Optimal Action-Value Function:

$q_*(s,a) \equiv \max_{\pi} q_\pi(s,a) = E\left[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a\right]$

Note that the optimality in the second equation comes from the optimal state-value function

Bellman Optimality Equation:

$$v_*(s) \equiv \max_a E\left[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a\right] = \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_*(s')]$$

This is a statement that the value of a state under an optimal policy must equal the expected return for the best action from that state

For finite MDP's this has a unique solution

Note that since this exists for each state it is really a system of equations (this system will be nonlinear)

This system will have the same number of equations as possible states

If this function v_* is known then a greedy policy for this function is optimal, a one step search just must be done over all possible actions

Specifically an optimal policy will always take the action that obtains a maximum in this equation

Note that this is equivalent to an exhaustive search

Assumptions for Practical Application of this Equation:

The dynamics of the system are accurately known

Computational resources are sufficient to complete the calculation

The states have the Markov property

Many decision making methods can be viewed as approximately solving this equation

The continuous analog to this equation is the Hamilton-Jacobi-Bellman equation

Bellman Optimality Equation for q_*:

$$q_*(s,a) = E\left[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') | S_t = s, A_t = a\right] = \sum_{s',r} p(s', r|s, a)\left[r + \gamma \max_{a'} q_*(s', a')\right]$$

If this is known, then the action that maximizes this is the optimal action

Hamilton's Equation can be thought of as an action-value function and Newtonian dynamics can be thought of as the policy greedy with respect to this function