Multi-armed Bandits

Tuesday, June 16, 2020 10:44 AM

Reinforcement learning uses training information that evaluates actions rather than instructing by giving correct actions

Nonassociative setting:

One in which the agent learns to act in only one situation

K-armed Bandit Problem:

You are faced repeatedly with a choice among k different options/actions. After each choice you receive a numerical reward chosen from a stationary probability distribution that depends on the action you selected. Your objective is to maximize the expected total reward over some time period

The name comes from analogy to a slot machine (one-armed bandit) except with k-levers

Actions are meant to become concentrated on the best levers

Analysis:

Value of an Action:

$q_*(a) \equiv E\left[R_t \middle| A_t = a\right]$

Where A_t is the action selected at time step t, R_t is the corresponding reward, and E denotes that this is an expected quantity

This is best stated as the expected reward of taking that action

Estimated Value:

$Q_t(a)$

This is an internal estimation of the quantity above

Greedy actions are those which correspond to the greatest expected reward

Exploration/nongreedy actions are those which do not correspond to the greatest expected reward

Taking greedy actions corresponds to exploitation and is the right thing to do to maximize the expected reward on that step, but exploration may be necessary to reach greater total rewards in the long run

Action-Value Methods:

Methods that estimate the values of actions and use these estimates to make action selection decisions

Sample-Average Method:

$$Q_t(a) \equiv \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{I}_{\mathrm{A_i=a}}}{\sum_{i=1}^{t-1} \mathbb{I}_{\mathrm{A_i=a}}} = \frac{\text{sum of rewards when a taken prior to t}}{\text{number of times a taken prior to t}}$$

Where $\mathbb{I}_{\mathrm{x}}$ denotes a variable that is 1 if x is true and 0 if it is not and if the denominator is 0, a default value is chosen (such as 0)

This is the natural approach of estimating values of actions as the mean of the rewards received when those actions have been taken in the past

Note that as the denominator goes to 0 $Q_t(a) \to q_*(a)$ (the estimate approaches the true value)

Incremental Average Update:

$$Q_{n+1} = Q_n + \frac{1}{n}\left[R_n - Q_n\right]$$

Where Q_x here denotes the value after the action has been tried x times

Note the general form here

$NewEstimate \leftarrow OldEstimate + StepSize\left[\,Target\, - OldEstimate\,\right]$

This is obviously comparable to gradient descent

The step-size here decreases over time, but may be held constant in other systems to allow the system to effectively deal with nonstationary reward

probabilities, in yet other systems the step-size may vary over time in other ways

Conditions Required for Step-Size to Allow System Convergence:

$$\sum_{n=1}^{\infty} \alpha_n(a) = \infty$$

$$\sum_{n=1}^{\infty} \alpha_n^2(a) < \infty$$

Where $\alpha_n(a)$ is the step-size used to process the reward after the n-th selection of action a

If these conditions are met, the value estimates are guaranteed to converge to the true values as time goes to infinity

The first condition guarantees that the steps are large enough to eventually overcome the initial conditions or random fluctuations and the second condition guarantees that the steps eventually become small enough to assure convergence

Note that both of these conditions are met for the incremental average update, but are not met for constant step-size (but this may be desirable when working with nonstationary reward distributions)

Incremental Update with Constant Step-Size:

$$Q_{n+1} = Q_n + \alpha\left[R_n - Q_n\right] = (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

Where $\alpha \in (0,1]$ is the step-size

Note that this is a weighted average since the sum of the weights is:

$$(1-\alpha)^n + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} = 1$$

This is sometimes called an exponential recency-weighted average

Constant Step-Size Without Initial Bias:

$$\beta_n \equiv \frac{\alpha}{\bar{o}_n}$$

$$\bar{o}_n \equiv \bar{o}_{n-1} + \alpha(1 - \bar{o}_{n-1})$$

Where β_n is here the step-size, adjusted from constant step size α, and $\bar{o}_0$ is set to 0

Greedy Action Selection Method:

$$A_t \equiv \underset{a}{\operatorname{argmax}}\, Q_t(a)$$

Note that if the argmax is the same for two or more actions (there are two or more greedy actions), one is chosen in an arbitrary way (e.g. randomly)

This is often modified by adding a small probability ε that an action is selected from all possible actions with equal probability

These methods are called ε-greedy methods

These methods have the advantage that as the number of steps increases to infinity, all actions will be tried an infinite number of time so all value estimates will converge to their actual values and the probability of selecting the optimal action converges to $1-\varepsilon$

Some methods also reduce ε over time so that more optimal actions are selected after the environment has been explored

Initial values in a reinforcement learning system can be set realistically to allow prior knowledge to help the system or high (optimistic) to encourage exploration

Using optimistic initial values is generally unsuitable for nonstationary problems since the drive to explore it incurs is temporary

Upper Confidence Bound Action Selection:

$$A_t \equiv \underset{a}{\operatorname{argmax}} \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$$

Where $N_t(a)$ denotes the number of times action a has been selected prior to time t, and $c > 0$ controls the degree of exploration

The idea behind this is that it is better to select from actions according to their potential for being optimal, taking into account both their current estimates and the uncertainty in the estimates

Specifically, the square root term is a measure of the uncertainty or variance in the estimate of a's value

Soft-max Distribution for Action Preference:

$$\pi_t(a) \equiv \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}}$$

Where $\pi_t(a)$ denotes the probability of taking action a at time t, and $H_t(a)$ denotes a numerical preference for each action that will be learned

The initial preferences are the same for all actions

Note that only the relative reference of one action over another is relevant

Preference Learning Algorithm (Stochastic Gradient Ascent):

$$H_{t+1}(A_t) \equiv H_t(A_t) + \alpha(R_t - \bar{R}_t)\left(1 - \pi_t(A_t)\right)$$

$$H_{t+1}(a) \equiv H_t(a) - \alpha(R_t - \bar{R}_t)\pi_t(a)$$

Where $\bar{R}_t$ is the average of the rewards up to but not including time t

Note that the first applies to the preference for the action that was actually taken and the second applies to the preferences for all actions that weren't actually taken

The average is taken as a baseline, an action that gives a reward higher than the average has its probability increased while an action that gives a reward below the average has its probability decreased and the non selected actions move in the direction opposite the selected action

Contextual Bandits:

Associative search tasks where the correct action to take must be associated with the current situation