Reinforcement learning: k-armed bandit
Written by Olga Bane (May 2020)

k-armed bandit: **agent** chooses between "k" **actions** and receives a **reward** for chosen action.

Each action has expected reward $q_*(a)$ (*value* of action). We do *not* know the reward of each action (if we did, we would always select the action with the highest value), although we may have estimates $Q_t(a)$:

$$\textit{Ideally } Q_t(a) \text{ is close to } q_*(a)$$

A_t = Action selected on time step t
T_t = Corresponding reward.
$q_*(a)$ = The value of an arbitrary action a (*action-value*) = Expected reward given that a was selected.
$Q_t(a)$ = estimated value of action a at timestep t.

$$q_*(a) = E[R_t|A_t]$$

Greedy actions *exploit* knowledge: If we keep track of the estimated reward values of each action at each time step, then we can choose the action with the higher estimated reward. **Non-greedy actions** enable you to improve the estimates of the action's value by *exploring.*

Choice between *exploration* and *exploitation* is challenging, and depends on estimated reward values, uncertainties, and the number of remaining steps.

Estimating $Q_t(a)$, using the sample-average method:

One simple method: average the reward among the samples we've already seen (*sample average of observed reward*).

$$Q_t(a) = \frac{\text{sum of reward when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i||A_{i=a}}{\sum_{i=1}^{t-1} ||A_{i=a}}$$

If the reward variance is 0, then the bandit would know the true reward value of an action after trying it only once. However, reward variance is often not 0, *and* the value of an action can also vary over time.

Given that reward variance != 0, let's re-define the action-value:

$$q_*(a) = E[R_t|A_t]$$

$$= \sum_r p(r|a)r, \text{where } p \text{ is the probability of the reward.}$$

Action selection rules:

1. Simplest: Always choose the greedy action (action with the highest estimated value).
2. ϵ**-greedy**: Behave greedily most of the time, but select randomly among all actions with equal probability, some fraction ϵ of the time.

To update $Q_t(a)$ (the average of all previous rewards for this action) **efficiently**, we need to only keep the previous $Q_t(a)$ value, the number of steps n at which this action has been selected. (R_n can be easily computed from these two values).

Update rule for a single action:

$$Q_n = \frac{R_1 + R_2 + ... + R_n}{n-1}$$
$$Q_{n+1} = \frac{Q_n * (n-1) + R_n}{n} = Q_n + \frac{1}{n}(R_n - Q_n)$$

Generalized update rule: fill in

Update rule for stationary vs. non-stationary problems

In a **stationary problem**, reward of action does not change. The averaging method described above is appropriate here.

In a **non-stationary problem**, reward of action changes. In this case, we might choose to give more weight to more recent rewards. One way to do this is to *use a constant step-size parameter,* α, where $\alpha \in (0,\ 1]$, resulting in a weighted average of the initial estimate and past rewards:

$$\begin{aligned} Q_{n+1} &= Q_n + a(R_n - Q_n) \\ &= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i \end{aligned}$$

(Note that sum of weights = 1, and that weight decays exponentially.)

How to determine initial action-value estimate, $Q_1(a)$:

This initial parameter must be set to *something*. It can be advantageous, as it allows the user to provide some prior knowledge. Setting **optimistic** action-value estimates encourages exploration. However, there are limitations. For example, optimistic initial values encourage only early exploration.

An improvement on the ϵ-greedy method to balance exploration and exploitation:

We must explore because greedy actions look best at present, but there may be other actions that are higher value. In ϵ-greedy, we try all actions with equal probability. It would be better to select actions based on their potential for being optimal. To do this, we must take into account *value estimates* and *uncertainties*. In the equation below, the square root term is a measure of the variance (uncertainty) in the estimate:

$$A_t = argmax_a[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}]$$

Evaluating different methods:

We will need to average many (noisy) runs of the bandit (with each bandit taking the same amount of steps) to compare between algorithms.

To do next:
1. Thompson sampling and Gittinis index.