Exploration and Exploitation

$Q[s, a]$ does not specify what an agent should do.
At each step, the agent could

- exploit what it has found to get higher rewards.
  In state s, it can do an action a that maximizes $Q[s, a]$.
- explore to build a better estimate of the Q-function
  It could select an action at random at each time.

The theoretical properties of the exploration-exploitation tradeoff are often studied in for bandits.
(A one-armed bandit is slot-machine / poker-machine.)
Each machine has its own distribution of payouts.
The action is to choose which machine to play;
— the agent repeatedly chooses an action from the same state.

Exploration Strategies

- optimism in the face of uncertainty: initialize Q to values that encourage exploration, meaning use an overestimate of Q-function.
    - Takes a long time to converge.
    - If actions are stochastic, a good action could get a bad outcome at random, and then it is never selected again.
- ϵ-greedy strategy: choose random action with probability ϵ choose a best action with probability $1 - \epsilon$.
  Problem:
    - Very bad actions get selected as much as promising actions that are not maximal.

Softmax Exploration

- Actions with a higher Q-value are more likely to be selected.
  Softmax action selection: in state s, choose a with probability

$$\frac{e^{Q[s,a]/\tau}}{\sum_a e^{Q[s,a]/\tau}}$$

  where $\tau > 0$ is a *temperature*.
- How much more likely is a to be chosen than a'?

$$\frac{P(a \text{ is selected})}{P(a' \text{ is selected})} = \frac{e^{Q[s,a]/\tau}}{e^{Q[s,a']/\tau}}$$

$$= e^{(Q[s,a]-Q[s,a'])/\tau}$$

$$= (e^{1/\tau})^{(Q[s,a]-Q[s,a'])}$$

τ	$e^{1/\tau}$
10	1.105
1	2.718
0.1	22026.5

Upper Confidence Bound

- Softmax selection doesn't take into account how many times an action has been tried, which affects how good the Q estimate is.
- The upper confidence bound is an estimate of the expected value such that it is be very unlikely that the actual value is greater than this.
- The upper confidence bound UCB1 is:

$$UCB1(s,a) = Q[s,a] + C * \sqrt{\frac{\log N(s)}{N[s,a]}}$$

  where
  - $N[s,a]$ is how many times action a has been selected in state s
  - $N(s) = \sum_a N[s,a]$ is how many times state s has been visited.
  - C is a constant that depends on the magnitude of the Q-values. If the values are all in range [0,1], then $C = \sqrt{2}$ has good theoretical properties
- A agent chooses action a with the highest $UCB1(s,a)$ value.

Thompson sampling

- In Thompson sampling, the agent selects a value from the posterior distribution of the Q-values.
- If the values are all 0 or 1 (e.g., win/loss), sample from the beta-distribution.
- If the return is a real number, you could assume the distribution is a Gaussian, parameterized by the mean and the variance. For each state, choose the action a that maximizes

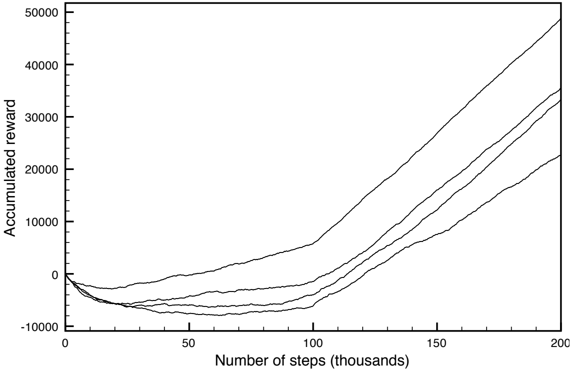
  $$Q[s,a] + C * \frac{randn()}{\sqrt{N[s,a]}}$$

  where $randn()$ returns a random number using the standard normal distribution (mean is 0, variance is 1).
  C is chosen to reflect the scale of the Q-values.

Stochastic Policy

- Use a stochastic policy $\pi(a \mid s)$
- $V^\pi(s) = \sum_a \pi(a \mid s) Q^\pi(s, a)$
- For an MDP, a stochastic policy is optimal if and only if all of the actions with a non-zero probability for a state have the same Q-value for that state, and that value is higher than the Q-value for any other action.
- How to update distribution given feedback? See Chapter 14.

Evaluating Reinforcement Learning Algorithms



Each algorithm stops exploring at 100,000 steps.

- Alternative #1: plot mean reward received, but it is noisy
- Alternative #2: plot discounted reward for each time step, but it can only be evaluated in retrospect.