Chapter 1

n-Armed bandits

1.1 Notes

1.1.1 n-Armed Bandit Problem

We have n different options (actions) representing n different slot machines. Each action has a given reward, sampled from a stationary probability $q(a)$ only dependent on the chosen action a. We want to maximize the (expected) total reward over a given (large) time T: $\sum t = 1^T R_t$. To do that, we estimate the value $Q_t(a)$ of each action given what we have seen so far. Let R_t the reward at time t and $N_t(a)$ the number of times the action a has been chosen so far.

1.1.2 Estimating value

We estimate the value with:

$$Q_t(a) = \frac{R_1 + ... + R_{N_t(a)}}{N_t(a)}$$

with $Q_t(a) = Q_1(a)$ a default value. With $N_t(a) \to \infty$ we have $Q_t(a) \to q(a)$.

Step-by-step, this can be calculated using incremental implementation to save computation time:

$$Q_{k+1} = \frac{1}{k} \sum_{i=1} k R_t$$

$$...$$

$$Q_{k+1} = Q_k + \frac{1}{k} (R_k - Q_k)$$

This looks like $NewEstimate \leftarrow OldEstimate + StepSize\,(Target - OldEstimate)$, with $StepSize = \frac{1}{k}$ here.

For tasks that never stop this estimation diverges, plus we may be interested in tracking a nonstationary problem. To achieve this, we can introduce a constant step size, that effectively weights recent rewards more heavily:

$$Q_{k+1} = Q_k + \alpha\,(R_k - Q_k)$$
$$\ldots$$
$$Q_{k+1} = (1-\alpha)^k Q_1 + \alpha \sum_{i=1}^{k} (1-\alpha)^{k-i} R_t$$

As it turns out, this defines a weighted average with weights $(1-\alpha)^k, \alpha(1-\alpha)^k \ldots, \alpha(1-\alpha)^{k-i}$ (they sum to 1).

By denoting $\alpha_k(a)$ the weight (step-size) used for the k-th selection of action a, we need to have two conditions:

1. $\sum_{k=1}^{\infty} \alpha_k(a) = \infty$, to guarantee that we overcome initial estimate, and
2. $\sum_{k=1}^{\infty} \alpha_k^2(a) < \infty$, to guarantee convergence.

Chosing actions

To chose the action, the *greedy* way is to select the one with the highest value: $A_t = \operatorname{argmax}_a Q_t(a)$. Problem: this does not spend any time to sample other actions to refine the estimates $Q_t(a)$.

First solution: ϵ-greedy algorithms, where $A_t = \operatorname{argmax}_a Q_t(a)$ $1-\epsilon$ of the times and $A_t = uniform(a)$ the other ϵ of the times.

Second solution: optimistic initial values, to preferentially select unsampled actions.