Optimal Learning for Multi-Armed Bandits

Matthew Darlington

September 9, 2019

What is a multi-armed bandit?

What is a multi-armed bandit?

Definition: Multi-Armed Bandits

A multi-armed bandit is a set of probability distributions $B = \{B_1, B_2, \cdots, B_K\}$, with each distribution being associated with the rewards of one of the $K \in \mathbf{N}$ levers.

Definition: Bernoulli Multi-Armed Bandits

$B_i \sim Bernoulli(p_i)$ where $p_i \in [0, 1]$ is a fixed constant for each i.

More Maths

Definition: Horizon

H, the number of pulls we are allowed to make

Definition: Regret

$\rho(T) = \max_i \left(\sum_{t=1}^{T} r_i^t \right) - \sum_{t=1}^{T} \hat{r}_t$

More Maths

We use a Bayesian approach

How do we store information?

$\alpha_i =$ "The successes of i" and $\beta_i =$ "The failures of i"

Estimating p_i

$$p_i^t = \mathbf{E}[p_i|S^t] = \frac{\alpha_i^t + \alpha_i^{prior}}{\alpha_i^t + \alpha_i^{prior} + \beta_i^t + \beta_i^{prior}}$$

Dynamic Programming

- Treat the problem as a Markov decision process. A Markov chain but at each time step we can make a decision and there is a reward associated to each outcome.
- Use dynamic programming to calculate the optimal policy using Bellman's equation:

$$\mathbb{V}(S^t, t) = \max_{i \in B} \left[r(S^t, i) + a \sum_{S^{t+1}} \mathbb{P}(S^{t+1} | S^t, i) \mathbb{V}(S^{t+1}, t+1) \right]$$
$$\mathbb{V}(S^{\mathcal{H}}, \mathcal{H}) = 0$$

- Not a very practical solution due to time and memory required to compute, but is proven to be optimal.

Greedy Policy

- Always exploit and never explore.
- Problem: No exploration leads to poor results.
- For example, if we have two bandits with $p_1 = 0.8$ and $p_2 = 1$ then the greedy method could get 'stuck' pulling the first arm every time.

Epsilon Greedy Policy

- A simple improvement is to add in some random exploration.
- At each time step with probability $\epsilon \in [0, 1]$ instead of being greedy, choose a random arm.
- Still has obvious errors as we do not care as much about exploring as time goes on.
- Can replace ϵ by some decreasing sequence $\epsilon_t \in [0, 1]^H$.

Optimizing parameters

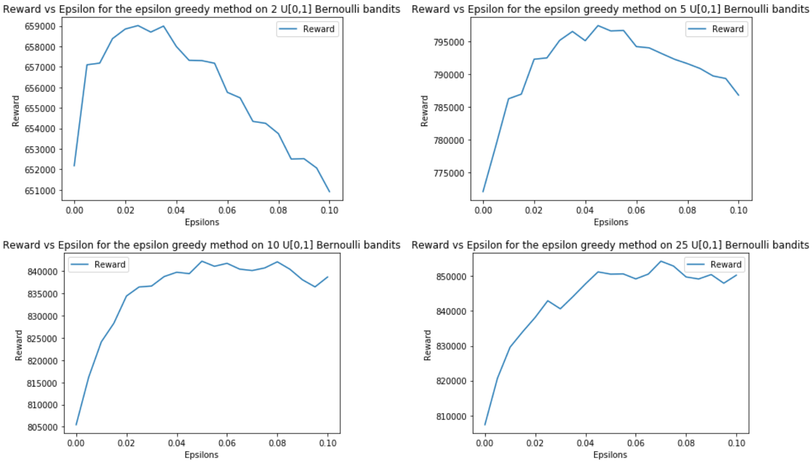


Figure: Looking for the values of epsilon for epsilon greedy as we increase the number of bandits

Thompson Sampling

- Another problem with the epsilon greedy methods is we do not think about how we choose where to explore.
- Each arm is given a $Beta(\alpha_i^t, \beta_i^t)$ prior distribution.
- We take a sample and pull the arm with the largest.
- The posterior distribution is then $\text{Beta}(\alpha_i^{t+1}, \beta_i^{t+1})$ due to conjugacy properties of the distributions.
- With an infinite horizon always learns about the best arm.

Thompson Sampling

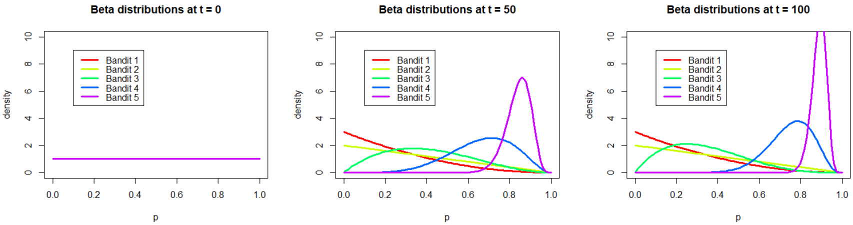


Figure: Looking how the prior distributions change for B $= (0.1, 0.3, 0.5, 0.7, 0.9)$

Knowledge Gradient

- We look at the value of the knowledge we gain by choosing an arm, if that is the last choice we get to make.
- The value of being in a state can be defined as,

$$V^t(S^t) = p_j^t = \max_i p_i^t$$

- We then define the knowledge gradient as,

$$v_i^{KG,t} = \mathbf{E}\left[V^{t+1}(S^{t+1}(i)) - V^t(S^t) \;\middle|\; S^t\right]$$

- Our choice is then made by picking,

$$\arg\max_i \left[p_i^t + (H-t)v_i^{KG,t}\right]$$

Comparing Methods

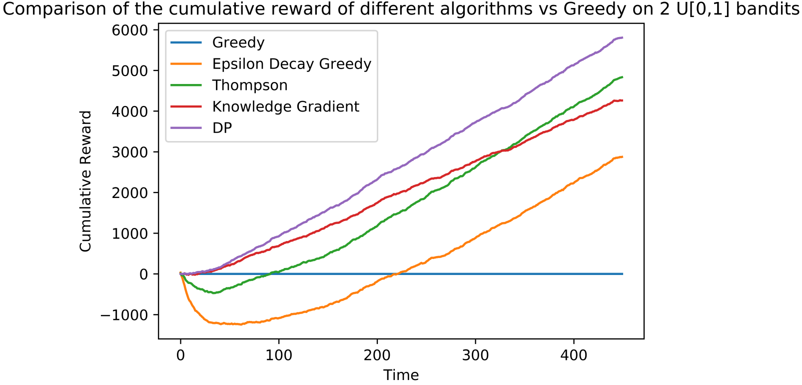
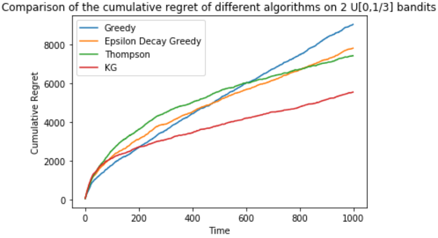
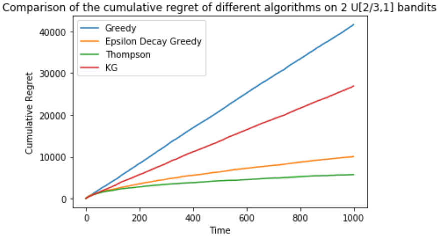


Figure: Comparing the reward of different policies

Comparing Methods



(a)



(b)

Figure: Comparing the methods in different scenarios

Thank you for listening
Any questions?