Assignment 3 Stoch Opt

Helene Randi Behrens

December 2019

1 Calculation of Gittins index

The prior information we have is that the success probabilities of the arms is uniformly distributed. Furthermore, we know that a beta(k,l) distribution with $k = l = 1$ is the uniform distribution, and that the posterior distribution of beta is also the beta distribution. So, we use that given s successes and f failures, the success probabilities follows the distribution Beta$(1 + s, 1 + f)$. Thus, the expected success probability given s successes and f failures for one arm is

$$\mathbf{E}[\text{Beta}(1 + s, 1 + f)] = \frac{1 + s}{1 + s + 1 + f} = \frac{1 + s}{2 + s + f}.$$

Before doing any realisations of the system, we have no information about the success-probability of each arm. Thus, at this point, the expected revenue at all future times is 0.5 (excluded the discounting); one times the expected probability of success, which at the beginning is E[Beta(1,1)]. We then get the same Gittin's index m for both arms, which is calculated as:

$$\begin{aligned}
m &= \max_T \{\mathbf{E}[\sum_{t=1}^{T} \alpha^t r_t]/\mathbf{E}[\sum_{t=1}^{T} \alpha^t]\} \\
&= \max_T \{\sum_{t=1}^{T} \alpha^t \mathbf{E}[r_t]/\sum_{t=1}^{T} \alpha^t\} \\
&= \max_T \{\sum_{t=1}^{T} \alpha^t \frac{1}{2}/\sum_{t=1}^{T} \alpha^t\} \\
&= \max_T \{\frac{1}{2}\sum_{t=1}^{T} \alpha^t/\sum_{t=1}^{T} \alpha^t\} \\
&= \frac{1}{2}.
\end{aligned}$$

Here, α is the discount factor.

Note that in our case, the Gittins index is simply the success probability given the previous number of successes and failures for an arm.

2 Assumptions and choices of parameters

For all methods, we run the simulation from $t = 0$ to $t = 120$. This choice was made because it is the largest T that would allow us to implement the full-information method (due to memory issues). For better comparison, this T is then used for all methods. Note, however, that this choice of T might not be that unreasonable. As the reward is reduced with a factor 0.05 for each time step, the additional possible reward at each time step would be less than 0.001 already at time $t = 134$, as we have

$$0.95^t = 0.001 \Rightarrow t = \frac{lg(0.001)}{lg(0.95)} \approx 134.$$

We then run the simulations 200 times each, and find the mean of the revenues for each method. The choice of 200 was simply to have a reasonable amount of simulations, while still keeping the run time relatively low.

2.1 Q-learning

- For the greedy, optimistic Q-learning, I chose initial values for Q to be 10. This choice was mad because the Q-values appeared to roughly converge towards values of ≈ 5, and thus $Q_0 = 10$ seemed like a reasonable optimistic value. The algorithm was also tested for initial values much higher adn much lower, and neither gave higher average rewards.

- I chose to use $\gamma_t = 1/t$ as it seemed to produce just as good results as keeping γ at a low constant. Additionally, this choice of γ was preferred, since it fulfils the criterion for optimal convergence for the ϵ-greedy version;
$$\sum_t \gamma_t = \infty, \qquad \sum_t \gamma_t^2 < \infty$$

- I have used the same function to simulate both optimistic Q-learning and ϵ-greedy Q-learning. In the case of optimistic Q-learning, $= 0$ is given as parameter and in the ϵ-case, the initial values are set to 0.

- In the ϵ-greedy case, I have used $\epsilon = 0.01$, simply because that value seemed to work well.

2.2 Full information

- The value function V and the optimal policy π are represented as four dimensional arrays, with the four dimensions being number of successes and failures for each arm. In the code, s1 (success for arm 1), f1 (failures for arm 1), s2 and f2 represent these. Note that, as R is 1-indexed, I have added a 1 to the numbers in the indices that represent each state. This means that the initial value, with no successes or failures for any arm has the index (1,1,1,1).

- The full-information simulation is divided into two parts. The first, implemented in function Find_Pi, finds the optimal policy Π by solving the Optimality equation, in our case:
$$V(s_1, f_1, s_2, f_2) = \max_{arm}\{r(arm) + \alpha[\mathbf{P}(\text{success for arm}) * V[s_{arm} + 1, :] + (1 - \mathbf{P}(\text{success for arm}))V[f_{arm} + 1, :]\}.$$
The expected revenues for each arm, given a number of observations, is the success probability of that arm:
$$\mathbf{P}(\text{success for arm}) = \frac{1 + s_{arm}}{2 + f_{arm} + s_{arm}}.$$
Then, the simulation of the reward from this method is done implemented in function full_info_r, where we simulate the reward given the optimal policy.

2.3 Bayesian with Gittins index

- This method implies always choosing the arm with the largest Gittins index, which is calculated given the number of successes and failures for each arm as previously explained.

2.4 Thompson Sampling

- In the Thompson Sampling algorithm, the optimal arm at each time is decided by sampling from the beta distribution of the success probabilities for each arm, given observations of successes and failures for each arm. The algorithm chooses the arm with the highest sampled value.