IEOR 8100-001: Learning and Optimization for Sequential Decision Making 03/30/16

Lecture 19: Bayesian bandit problem, and Introduction to MDP

Instructor: Shipra Agrawal Scribed by: Xu Sun

1 Bayesian bandit problem

Recall Bernoulli arms where reward is generated from a Bernoulli random variable with parameter μ_i each time the algorithm pulls arm i,, i.e. $\mathbb{P}(r_t = 1|I_t = i) = \mu_i$. The regret at time T is defined by

$$R(T, \{\mu_i\}) = T\mu^* - \sum_{t=1}^{T} \mu_{I_t},$$

with $\mu^* = \max_i\{\mu_i\}$.

In a Bayesian setup, the algorithm knows for each arm i, a prior distribution $D_i = beta(\alpha_i, \beta_i)$ from which the parameter μ_i is sampled. The goal is either to minimize the expected regret

$$\mathbb{E}_{\mu_i \sim D_i}[R(T, \{\mu_i\})] = \mathbb{E}_{\mu_i \sim D_i}[T\mu^* - \sum_{t=1}^{T} \mu_{I_t}].$$

(here μ^* denotes the maximum for *sampled* $\{\mu_i\}$) or to maximize the expected reward

$$\mathbb{E}_{\mu_i \sim D_i}\left[\sum_{t=1}^{T} r_t \gamma^{t-1}\right].$$

We formulate Bayesian bandit problem with known priors as a Markovian bandit problem (MBP), so that the optimal *online* algorithm is same as the optimal policy for this Markovian bandit problem.

At time $t = 1$, each μ_i follows a Beta distribution with parameter (α_i, β_i). That $\mathbb{P}(\mu_i = \theta) \propto \theta^{\alpha_i - 1}(1 - \theta)^{\beta_i - 1}$ implies

$$\mathbb{E}[r_1|I_1 = i] = \mathbb{E}[\mu_i] = \frac{\alpha_i}{\alpha_i + \beta_i}.$$

Note that any algorithm for this problem can use history $\mathcal{H}_{t-1}$ to made decisions at time t. For Bernoulli arms, the history includes the number of successes and failures of each arm. Let $S_{i,t-1}$ and $F_{i,t-1}$ denote the number of successes and failures respectively for arm i. Using Bayes formula, we have

$$\begin{aligned}\mathbb{P}(\mu_i = \theta|\mathcal{H}_{t-1}) &= \mathbb{P}(\mu_i = \theta|S_{i,t-1}, F_{i,t-1}) \\ &\propto \mathbb{P}(S_{i,t-1}, F_{i,t-1}|\mu_i = \theta)Be(\theta) \\ &\propto \theta^{S_{i,t-1}}(1 - \theta)^{F_{i,t-1}}\theta^{\alpha_i - 1}(1 - \theta)^{\beta_i - 1}\end{aligned}$$

which implies $\sim Beta(S_{i,t-1} + \alpha_i, F_{i,t-1} + \beta_i)$. As a consequence, the expected reward at time t

from pulling arm i given history is

$$\mathbb{E}_{\mu_i \sim D_{i,t}}[\mu_i] = \frac{S_{i,t-1} + \alpha_i}{S_{i,t-1} + \alpha_i + F_{i,t-1} + \beta_i}.$$

It follows from the above analysis that pulling an arm generates different expected reward every time it is pulled. Indeed, the expected reward depends on $(S_{i,t-1}, F_{i,t-1})$. If we view the pair (S_i, F_i) as the state of arm i, then what will happen at the current step *only* depends on the previous state. We are now ready to show that finding the optimal online algorithm for Bayesian bandit problem with known prior distributions is equivalent to solving a corresponding Markovian bandit problem with known (r, S, p) for all arms. Indeed, the Markovian bandit problem can be described as follows:

- The state space consists of all pairs (s, f) such that $s = 1, \ldots, T$, $f = 1, \ldots, T$ and $s + f \leq T$. At time t, the state of arm i is $(s_{i,t-1}, f_{i,t-1})$.
- The expected reward from pulling arm i is $r_i((s, f)) = \frac{\alpha_i + s}{\alpha_i + \beta_i + s + f}$.
- Transition function: $(s, f) \to (s + 1, f)$ with probability $\mathbb{E}[Beta(s + \alpha_i, f + \beta_i)] = \frac{s + \alpha_i}{s + \alpha_i + f + \beta_i}$ and $(s, f) \to (s, f + 1)$ with probability $1 - \mathbb{E}[Beta(s + \alpha_i, f + \beta_i)] = \frac{f + \beta_i}{s + \alpha_i + f + \beta_i}$.

Clearly the more the algorithm pulls an arm i, the more it knows about the sampled parameter μ_i. Here algorithm is facing trade-offs between exploration (getting more accurate posterior, better future states) and exploitation (getting better immediate reward).

Remark 1. *If the discounted reward is used, one can then apply Gittins index policy to achieve "optimal" Bayesian online algorithm. However no closed form solution is known for the case where* $\gamma = 1$.

2 Markov Decision processes

We can think of bandit problems as the simplest example of sequential decision problems, which involve an exploitation/exploration trade-off. In Markov Decision processes (MDPs), the strategy's actions also influence the state, in a probabilistic way. More formally, we have the following definition for MDPs.

Definition 2. *A Markov Decision Process (MDP) consists of*

1. *A state space* $\mathcal{S}$;
2. *An action space* $\mathcal{A}$;
3. *A set of Markov chains,* $\mathcal{M} = (\mathcal{S}, \mathbb{P}_a)$, *one for each* $a \in \mathcal{A}$;
4. *A reward distribution* $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ $r(s, a) = r_a(s)$.

Some of the commonly used objectives are summarized below:

a Maximize discounted reward over finite time horizon,

$$\max \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right]$$

b Maximize discounted reward over infinite time horizon,

$$\max \mathbb{E}\left[\lim_{T\to\infty} \sum_{t=1}^{T} \gamma^{t-1} r_t\right]$$

c Maximize reward over infinite time horizon *with a terminal state*,

$$\max \mathbb{E}\left[\lim_{T\to\infty} \sum_{t=1}^{T} r_t\right]$$

d Maximize average reward,

$$\max \mathbb{E}\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r_t\right]$$

The following example illustrates how a number of one-player games can be modeled as a Markov decision process.

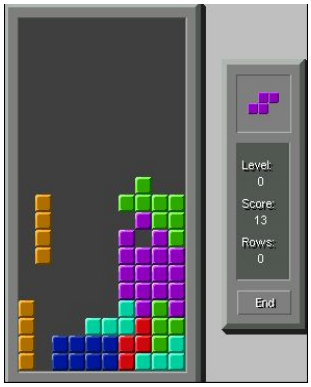

Figure 1: Illustration of the two-dimensional grid of the tetris game

Example 3. *(Tetris)*
Tetris is popular video game played on a two-dimensional grid. Each square in the grid can be either full or empty, making up a "wall of bricks" with holes. The squares fill up as objects of different shapes fall from the top of the grid and are added to the top of the wall, giving rise to a "jagged top". Each falling object can be moved horizontally and can be rotated by the player in all possible ways, subject to the constraints imposed by the sides of the grid. The game starts with an empty grid and ends when a square in the top row becomes full (the top of the wall touches the top of the grid). However whenever a row of full squares is created, this row is removed and the bricks lying above this row move one row downward (the player score one point).

The game can be essentially modeled as a MDP where the state consists of two components: (1) the board position (a binary description of the full/empty status of each square); (2) the shape of the current falling object. Possible actions to be taken include the horizontal positioning and rotation applied to the falling object. The reward is jointly determined by the current board position and how the falling object is placed. The transition from one board position to next, given the action is deterministic, but the shape of next falling action is random. Therefore, transition probability is given by the distribution of shape of falling objects (assuming the shape of next falling object is iid).

Example 4. *(Inventory Model)*
Each month the manager of a warehouse determines current inventory (stock on hand) of a single product. Based on this information, she decides whether or not to order additional stock from a supplier. In doing so, he is faced with a trade-off between holding costs and the lost sales or penalties associated with being unable to satisfy customer demand for the product. The objective is to maximize some measure of profit over decision-making horizon. Demand is a random viable with a probability distribution known to the manager.

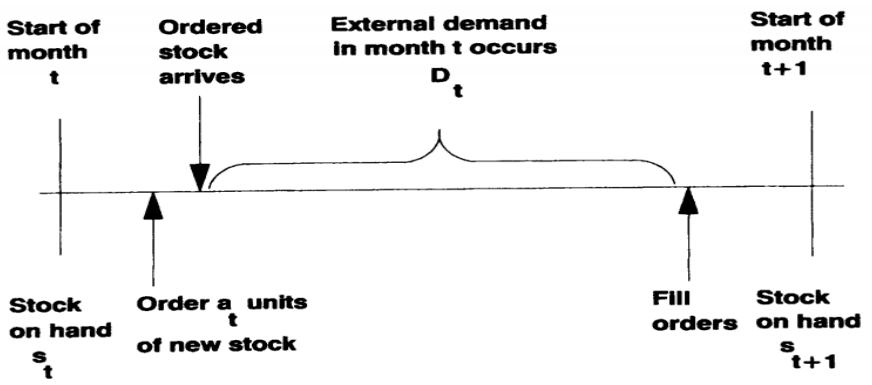

Figure 2: Timing of events in an inventory model

Let s_t denote the inventory on hand at the beginning of the tth time period, a_t the number of units ordered by the inventory manager period and D_t the random demand during this time period. We assume that the demand has a known time-homogeneous probability distribution $p_j = \mathbb{P}(D_t = j)$, $j = 0, 1, \ldots$. The inventory at decision epoch $t+1$ referred to as s_{t+1}, is related to the inventory at decision epoch t, s_t, through the system equation

$$s_{t+1} = \max\{s_t + a_t - D_t, 0\} \equiv [s_t + a_t - D_t]^+.$$

That backlogging is not allowed implies the non-negativity of the inventory level. Denote by $O(u)$ the cost of ordering u units in any time period. Assuming a fixed cost K for placing orders and a variable cost $c(u)$ that increases with quantity ordered, we have

$$O(u) = [K + c(u)]1_{\{u>0\}}.$$

The cost of maintaining an inventory of u units for a time period is represented by a nondecreasing function $h(u)$. Finally, if the demand is j units and sufficient inventory is available to meet demand,

the manager receives revenue with present value $f(j)$. In this model, the reward depends on the state of the system at the subsequent decision epoch, that is

$$r_t(s_t, a_t, s_{t+1}) = -O(a_t) - h(s_t + a_t) + f(s_t + a_t - s_{t+1}).$$