Dynamic Programming

Author: Wenxiao Yang

Institute: Department of Mathematics, University of Illinois at Urbana-Champaign

Date: 2022

All models are wrong, but some are useful.

Contents

Chapter 1 Multi-Armed Bandits

1.1 Multi-Armed Bandit Problem

Definition 1.1 (Multi-Armed Bandit Problem)

A decision-maker ("gambler") chooses one of n actions ("arms") in each time step. Chosen arm produces random payoff from unknown distribution. Goal: Maximize expected total payoff. ♣

Every time, the DM pulls an arm $i \in \{1, ..., n\}$, then he observes a reward r_t which follows a distribution.

- Each arm has a **type** that determines its payoff distribution.
- Gambler has a prior distribution over **types** for each arm.
- **Types** are independent random variables.
- Objective: maximize expected discounted reward $\sum_{t=0}^{\infty} \beta^t r_t$

Definition 1.2 (State)

*After pulling arm i some number of times, gambler has a posterior distribution over types. Call this the **state** of arm i (i.e., the posterior distribution over types after getting information).*

State can be expected reward when pulled is governed by state. ♣

The player has two controls: either play the process or not. If the player chooses to play the bandit process i, $i = 1, ..., n$, the state of the bandit process i evolves in a Markovian manner while the state of all other processes remains frozen (i.e., it does not change). Such a bandit process is called a Markovian bandit process.

Let $\{X_t^i\}$ denote the bandit process. The state $X_t^i \in \mathcal{X}^i$ of the bandit process i at time t, where $\mathcal{X}^i$ is a finite (or countable) space.

Let $\vec{u}_t = (u_t^1, ..., u_t^n)$ denote the decision made by the player at time t. The component $u_t^i \in \{0, 1\}$ is binary valued and denotes whether the player chooses to play the bandit process i ($u_t^i = 1$) or not ($u_t^i = 0$). Since the player may only choose to play one bandit process at each time, $u_t^i = 1$ must have only one nonzero component:

$$\sum_{i=1}^{n} u_t^i = 1, \quad \forall t$$

Let $\mathcal{U} \subset \{0, 1\}^n$ denotes all vectors with this property, $\mathcal{U} = \{\vec{u} : u^i \in \{0, 1\}, i = 1, ..., n; \sum_{i=1}^{n} u^i = 1\}$.

The collection $\{\vec{X}_t = (X_t^1, ..., X_t^n)\}_{t=0}^{\infty}$ evolves as follows: $\forall i = 1, ..., n$

$$X_{t+1}^i = \begin{cases} f^i(X_t^i, W_t^i), & \text{if } u_t^i = 1 \\ X_t^i & \text{if } u_t^i = 0 \end{cases}$$

When bandit process i is played, it yields a reward r_t^i and all other processes yield no reward. The objective of

a player is to choose a decision strategy $g = \{g_t\}_t = 0^\infty$, where $g_t : \prod_{i=1}^n \mathcal{X}^i \to \mathcal{U}$, to maximize the expected total discounted reward

$$\mathbb{E}_g \left[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{n} r_t^i u_t^i \mid \vec{X}_0 = \vec{x}_0 \right]$$

where $\vec{x}_0 = (x_0^1, ..., x_0^n)$ is the initial starting state of all bandit processes.

1.2 Gittins Index

One possible solution concept for the MAB problem is to set it up as a Markov decision process (MDP) and use Markov decision theory to solve it. However, such an approach does not scale well with the number of bandit processes because of the *curse of dimensionality*.

Instead of solving the n-dimensional MDP with state-space $\prod_{i=1}^n \mathcal{X}^i$, an optimal solution is obtained by solving n 1-dimensional optimization problems:

Definition 1.3 (Gittins Index)

For each bandit i, $i = 1, ..., n$, and for each state $x^i \in \mathcal{X}^i$, compute

$$\mathcal{V}^i(x^i) = \max_{\tau > 0} \frac{\mathbb{E}\left[\sum_{t=0}^{\tau} \beta^t r_t^i \mid X_0^i = x^i\right]}{\mathbb{E}\left[\sum_{t=0}^{\tau} \beta^t \mid X_0^i = x^i\right]} \tag{1}$$

*where τ is a measurable stopping time. The function $\mathcal{V}^i(x^i)$ is called **Gittins index** of bandit process i at state x^i. The optimal τ is called the optimal stopping time at x^i.*

The optimal strategy is "At each time, play the arm with the highest Gittins index."

♣

To implement the optimal strategy, we need compute and store the Gittins index $\mathcal{V}^i(x^i)$ of all states $x^i \in \mathcal{X}^i$ of all bandit processes i, $i = 1, ..., n$.

Claim 1.1 (Equivalent Interpretation)

An equivalent interpretation of the Gittins index strategy is "Pick the arm with the highest Gittins index and play it until its optimal stopping time (or equivalently, until it enters the corresponding stopping set) and repeat."

♠

1.3 Off-Line Algorithms for Computing Gittins Index

Since the Gittins index of a bandit process depends just on that process, we drop the label i and denote the bandit process by $\{X_t\}_{t=0}^{\infty}$.

Off-line algorithms use the following property of the Gittins index. The Gittins index $\mathcal{V} : \mathcal{X} \to \mathbb{R}$ induces a

total order $\succeq$ on $\mathcal{X}$ that is given by

$$\forall a, b \in \mathcal{X}, a \succeq b \Leftrightarrow \mathcal{V}(a) \geq \mathcal{V}(b)$$

Using this total order, for any state $a \in \mathcal{X}$, the state space $\mathcal{X}$ may be split into two sets

$$\text{Continuation set:} \quad C(a) = \{b \in \mathcal{X} : b \succeq a\}$$

$$\text{Stopping set:} \quad S(a) = \{b \in \mathcal{X} : a \succeq b\}$$

These sets are, respectively, called the *continuation set* and the *stopping set*. The rationale behind the terminology is that if we start playing a bandit process in state a, then it is optimal to continue playing the bandit process in all states $C(a)$ because for any $b \in C(a)$, $\mathcal{V}(b) > \mathcal{V}(a)$. Thus, the stopping time that corresponds to starting the bandit process in state a is the hitting time $T(S(a))$ of set $S(a)$, that is, the first time the bandit process enters set $S(a)$.

$$T(S(a)) = \inf\{t > 0 : X_t \in S(a)\}$$

Using this characterization, the expression for Gittins index (1) simplifies to

$$\mathcal{V}(a) = \max_{S(a) \subseteq \mathcal{X}} \frac{\mathbb{E}\left[\sum_{t=0}^{T(S(a))} \beta^t r_t \mid X_0 = a\right]}{\mathbb{E}\left[\sum_{t=0}^{T(S(a))} \beta^t \mid X_0 = a\right]} \tag{2}$$