ORF 418 : OPTIMAL LEARNING

LECTURE 2 : September 8, 2025

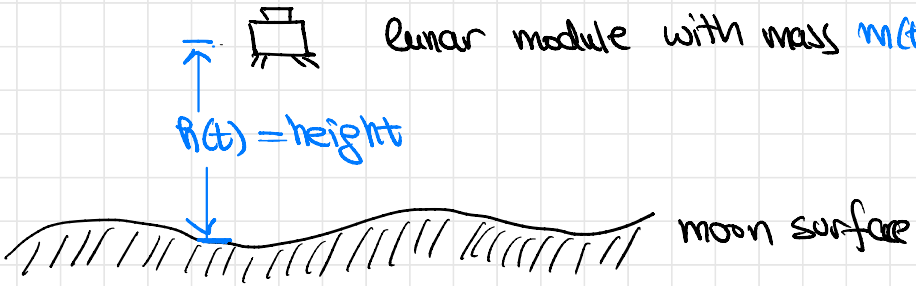
(slightly revised after lecture).

Examples

1) Moon Landing
2) Armed Bandits
   a) Problem
   b) Discussion
   c) Explore-Exploit

I. Moon Landing:

From last lecture



Newton law: $m(t)h(t)'' = -g_{moon}m(t) + \alpha(t)$, where g_{moon} is the gravitational constant of moon and $\alpha(t)$ is the control.

Goal is to minimize total fuel $\int_0^\tau \alpha(t)dt$ and we want soft-landing: $v(\tau) = h'(\tau) = 0$ where τ is the landing time.

state: $x(t) = (h(t), v(t), m(t)) =$ position, velocity, mass

control: $\alpha(t) =$ fuel pushed down to slow down the lunar module

admissible controls $\mathcal{U}$: all $\alpha(t)$ so that we have a soft-landing, i.e., $v(\tau) = 0$ where τ is the landing time.

state equation (or dynamics) (simplified)

$$\frac{d}{dt}x(t) = \begin{pmatrix} h'(t) \\ v'(t) \\ m'(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ -g_{moon} + \frac{\alpha(t)}{m(t)} \\ -k\alpha(t) \end{pmatrix}, \quad t>0$$

where k is a known constant.

Goal

$$\text{minimize} \quad \int_0^\tau \alpha(t)\,dt,$$

where τ is defined as:

$$\tau := \inf\{t>0 : h(t)=0\}.$$

In this problem there is no learning as all constants are known from physics and no randomness.

See notes for the discussion.

Remark This problem is naturally set in continuous time and not discrete. So it is not in the scope of this course. But the solution technique is very similar to the ones we will develop. It is included because of its historical importance.

II. MULTI-ARMED BANDITS.

One armed bandit is generally taught of as a slot machine in a casino. They are called bandits as people believe that they are rigged to take money from players. The important part is that when played we do not know the odds. This makes the problem a learning one.

We start with an example taken from the book "Introduction to Multi-Armed Bandits" by A. Slivkins.

a) Example: News Website

A news website offers personalized news headers for any user that visits their site. The goal is to maximize the number clicks of the users. To achieve this goal the website should learn the characteristics of a particular user and present headers that would be interesting to that user whereby increasing the probability of a click.

This problem has

(i) learning : site does not know the interests of a user and learns it by observing their choices;

(ii) randomness : site can only learn statistics and can never be sure of the outcome

b) Mathematical Formulation : News Website

Suppose that site has K headers to offer. Each one is regarded as a "slot-machine". When the header labelled $a \in \{1,..,K\}$ is offered, probability of a click is $q(a)$. This value $q(a) \in [0,1]$ is not initially known and has to be learned.

Suppose that a user visits the site T-times. Let $(a_1,..,a_T)$ be the (random) sequence of headers offered to this user, and let $r_t \in \{0,1\}$ be the random reward, 1 indicating the user read the offered news. The goal is to maximize.

$$J(\alpha) := \mathbb{E}\left[\sum_{t=1}^{T} r_t\right].$$

If we make the unrealistic assumption that $q(a)$'s are known to the site, then the solution is trivial:

$$J(\alpha) = \sum_{t=1}^{T} \mathbb{E}[r_t] = \sum_{t=1}^{T} q(a_t)$$

if $\alpha = (a_1, \ldots, a_T)$.

$$\max_{\alpha=(a_1,\ldots,a_T)} J(\alpha) = \max_{\alpha=(a_1,\ldots,a_T)} \sum_{t=1}^{T} q(a_t)$$

$$= \sum_{t=1}^{T} \max_{a_t} q(a_t) = T\, q(a^*),$$

where

$$a^* = \arg\max_a q(a).$$

So the trivial decision is to choose a^* all the time. Here the important structural consideration is that prior actions do not influence $\mathbb{E}[q(a_t)]$. This is not the case when we introduce learning into the model.

c) Solution with learning:

We provide an "intuitive" approach and formalize it later in the course.

First step is estimation.

Let $\hat{q}_t(a)$ be the best estimate of the unknown probability $q(a)$ calculated by the site at time t using the outcomes of the user's reaction up to this time.

In fact the vector $(\hat{q}_t(1), \dots, \hat{q}_t(K))$ is the state of the problem. The initial estimates $(\hat{q}_0(1) \dots \hat{q}_0(K))$ is part of the description of the model and is assumed to be known.

Based on Bayesian updating (recalled in detail later in the course), we assume that the updates are given as follows,

$$\hat{q}_{t+1}(a) = \begin{cases} \hat{q}_0(a) & \text{if } N_t(a) = 0, \\ \frac{1}{N_t(a)} \sum_{k=1}^{t} r_k \chi_{\{a_k = a\}}, \end{cases}$$

where $N_t(a)$ is the total number of times header a is offered up to time t, i.e.,

$$N_t(a) = \sum_{k=1}^{t} \chi_{\{a_k = a\}} \qquad a \in \{1 \dots K\},\ t = 1, \dots T.$$

Note that, by law of large numbers,

$$\lim_{t \to \infty} \hat{q}_t(a) = q(a) \quad \text{if} \quad \lim_{t \to \infty} N_t(a) = +\infty .$$

In words, the site will eventually learn $q(a)$ provided that it offers the header a to this user infinitely many times.

Intuitive Solution: Exploit & Explore

At time t, given the estimates $(\hat{q}_t(1) .. \hat{q}_t(K))$ the site has two options:

exploit: choose the header a_t^* that maximizes the probability estimates $\hat{q}(\cdot)$.

However, if the site only exploits it may not learn and will be stuck.

explore: occasionally, the site should offer a random header to get a better estimate of the click probability.

This suggests the following ε-Greedy

algorithm :

$$a_t = \begin{cases} \underset{a}{\operatorname{argmax}}\, \hat{q}_t(a) & : \text{with probability } 1-\varepsilon, \\ \tilde{a}_t & : \text{with probability } \varepsilon, \end{cases}$$

where $\tilde{a}_t$ is chosen randomly from $\{1, \ldots, k\}$.

One can show that by letting $\varepsilon \downarrow 0$ slowly, we can achieve the optimality as $T \uparrow \infty$.