Introduction to bandits

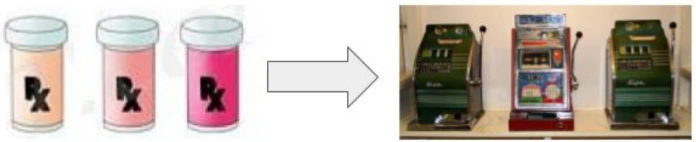
(some slides stolen from Csaba's AAAI tutorial)

Motivation

Do not have complete information about the effectiveness or side-effects of the drugs.
Aim: Infer the **best** drug by running a sequence of trials

Mapping to a bandits algorithm:

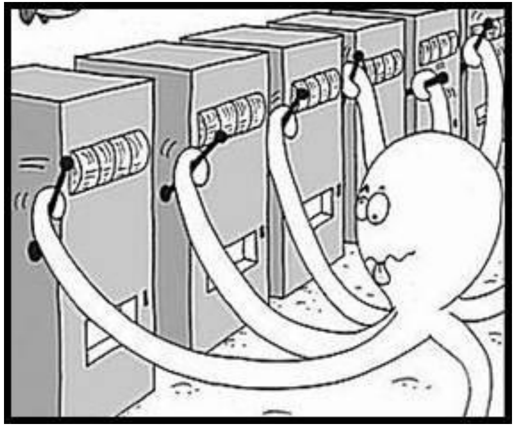
- Each drug choice is mapped to an **arm** and its **reward** is mapped to the drug's effectiveness.
- Administering a drug is an **action** and is equivalent to **pulling** the corresponding arm.
- The trial goes on for n **rounds**.



Other applications: Recommender Systems, Viral Marketing, Network Routing, Ad Placement

Introduction

How to tell if your problem is a bandit problem?



Three core properties:

1. Sequentially taking **actions** of **unknown** quality
2. The **feedback** provides information about quality of chosen action
3. There is no **state**

Assumptions:

1. **Stochasticity:** The reward for each arm is sampled from its *underlying distribution*. The
2. **Finiteness and Independence:** The number of arms is *finite* and the reward for each arm is *independent* of the others.
3. **Stationarity:** The reward distributions of the arms do not change over time.

Introduction

Algorithm 1 GENERIC BANDIT FRAMEWORK

```
for t = 1 to T do
    SELECT: Use the bandit algorithm to decide which arm(s) to pull.
    OBSERVE: Pull the selected arm(s) and observe the reward and associated feed-back.
    UPDATE: Update the estimated reward for the arms(s).
```

Is a special tractable case of RL

Performance Metric: Cumulative regret

$$R_n = n\mu^* - \mathbb{E}\left[\sum_{t=1}^{n} X_t\right]$$

Results in an **exploration-exploitation trade-off**:
Exploration: Pull an arm to learn more about it.
Exploitation: Pull the arm that we know has a higher reward.

Multi-armed bandits

OBSERVE: Can observe reward immediately on pulling the arm. Rewards are scalars bounded on the [0,1] interval.

UPDATE: Use the mean of rewards obtained on pulling arm *i* as the empirical estimated reward for that arm.

SELECT: Explore-Then-Commit, Epsilon-Greedy, Upper Confidence Bound, Thompson sampling

Explore-Then-Commit

1. Choose each action m times
2. Find the empirically best action $I \in \{1, 2, \ldots, K\}$
3. Choose $A_t = I$ for all remaining rounds

Explore-Then-Commit

When to commit: $m = \left\lceil \frac{4}{\Delta^2} \log\left(\frac{n\Delta^2}{4}\right) \right\rceil$

$$R_n \leq \min\left\{ n\Delta,\ \Delta + \frac{4}{\Delta}\log\left(\frac{n\Delta^2}{4}\right) + \frac{4}{\Delta} \right\}$$ (Gap-dependent Bound)

Worst case is when $\Delta \approx \sqrt{1/n}$ with $R_n \approx \sqrt{n}$ (Gap-free Bound)

- Need advance knowledge of the horizon n
- Optimal tuning depends on Δ
- Does not behave well with $K > 2$

Epsilon-Greedy

$A_t = \text{Uniform}\{1, 2, \ldots K\}$ (With probability ε)

Find the empirically best action $I \in \{1, 2, \ldots, K\}$

Choose $A_t = I$ (With probability $1 - \varepsilon$)

+ Interleaves exploration and exploitation.
+ Doesn’t require knowledge of the gap or the horizon.
+ Popularly used and works well in practice.

- Performance is sensitive to the choice of epsilon.
- Results in suboptimal n^{⅔} regret.

Optimism in the face of uncertainty

Let $\hat{\mu}_i(t) = \frac{1}{T_i(t)} \sum_{s=1}^{t} \mathbb{1}(A_s = i) X_s$

optimistic estimate $= \hat{\mu}_i(t-1) + \sqrt{\dfrac{2\log(1/\delta)}{T_i(t-1)}}$

1 Choose each action once

2 Choose the action maximising

$$A_t = \text{argmax}_i \, \hat{\mu}_i(t-1) + \sqrt{\frac{2\log(t^3)}{T_i(t-1)}}$$

3 **Goto** 2

Optimism in the face of uncertainty

$$R_n = O\left(\sum_{i:\Delta_i>0}\left(\Delta_i + \frac{\log(n)}{\Delta_i}\right)\right)$$

$$R_n = O\left(\sqrt{Kn\log(n)}\right)$$

+ Doesn’t require knowledge of the gap or the horizon.
+ Results in near-optimal regret.

Thompson sampling

P_i is the posterior distribution (conditioned on the observed rewards) for arm *i*

$$\tilde{\mu}_i \sim P_i$$

$$A_t = argmax \ \tilde{\mu}_i$$

Update P_{A_t}

- \+ Simple to implement. Only requires a sampling procedure
- \+ Theoretically, it results in near-optimal regret.
- \+ Often works better than UCB in practice.

- \- In some variants, it tends to over-explore.

Structured Bandits

- Arms (choices) can be related by a structural assumption on the action space or according to their corresponding features. Eg: Items in a Rec-sys.
- In problems with large number of arms, learning about each arm separately is inefficient.
- **Contextual Bandits:** Each arm *j* has a feature vector x_j and there exists θ^*

$$\mathbb{E}[\text{reward for arm } j] = h(x_j, \theta^*)$$

- **Linear Bandits:** $h(x, \theta) = \langle x, \theta \rangle$
- **Combinatorial Bandits:** The space of arms are related according to a combinatorial constraint.

Contextual Bandits

UPDATE: $\mathcal{L}_t(\theta) = \sum_{i \in \mathcal{D}_t} \log \left[\mathcal{P}(y_i | x_i, \theta) \right]$

$$\widehat{\theta}_t \in \arg\max_\theta \mathcal{L}_t(\theta)$$

Linear Bandits:

$$R_n = \mathbb{E}\left[\sum_{t=1}^{n} \max_{a \in \mathcal{A}_t} \langle a, \theta_* \rangle - X_t \right]$$

(Non)-Linear Bandits

Epsilon-Greedy

$$j_t \sim \text{Uniform}\{1, 2, \ldots K\} \qquad \text{(With probability } \varepsilon\text{)}$$

$$j_t = \arg\max_j \langle \mathbf{x}_j, \widehat{\theta}_t \rangle \qquad \text{(With probability } 1 - \varepsilon\text{)}$$

- O(n^{2/3}) regret

+ Easy to extend for non-linear bandits

LinUCB

$$j_t = \arg\max_j \left[\langle \mathbf{x}_j, \widehat{\theta}_t \rangle + c \cdot \sqrt{\mathbf{x}_j^\top M_t^{-1} \mathbf{x}_j} \right]$$

$$\sqrt{8dn\beta_n \log\left(\frac{\text{trace}(V_0) + nL^2}{d \det^{\frac{1}{d}}(V_0)}\right)}$$

- Don't know how to construct confidence intervals for complex functions

(Non)-Linear Bandits

Thompson sampling

$$\widetilde{\theta} \sim \mathcal{P}(\theta|\mathcal{D}_t)$$

$$j_t = \arg\max_j \langle \mathbf{x}_j, \widetilde{\theta} \rangle$$

+ O(d n^{½}) regret
+ Can use approximate sampling procedures for complex functions

Bootstrapping

$$\widetilde{\mathcal{L}}(\theta) = \sum_{i \in \widetilde{\mathcal{D}}_j} \log \left[\mathcal{P}(y_i | x_i, \theta) \right]$$

$$\widetilde{\theta}_j \; \in \; \arg\max_\theta \widetilde{\mathcal{L}}(\theta)$$

- Not well developed theory.
+ Need to compute only point estimates.

Bandits everywhere!

- Adversarial Bandits (relaxing assumption 1)
- Gaussian process Bandits (relaxing assumption 2)
- Restless Bandits (relaxing assumption 3)
- Rotting Bandits
- Duelling Bandits
- Firing Bandits
- ………….

Difference objective functions:
Best-arm identification
Bayesian bandits