Bandit Algorithms

Zhifeng Wang

Department of Statistics
Florida State University

Outline

- Multi-Armed Bandits (MAB)
  - Exploration-First
  - Epsilon-Greedy
  - Softmax
  - UCB
  - Thompson Sampling
- Adversarial Bandits
  - Exp3
- Contextual Bandits
  - LinUCB

K Slot Machines

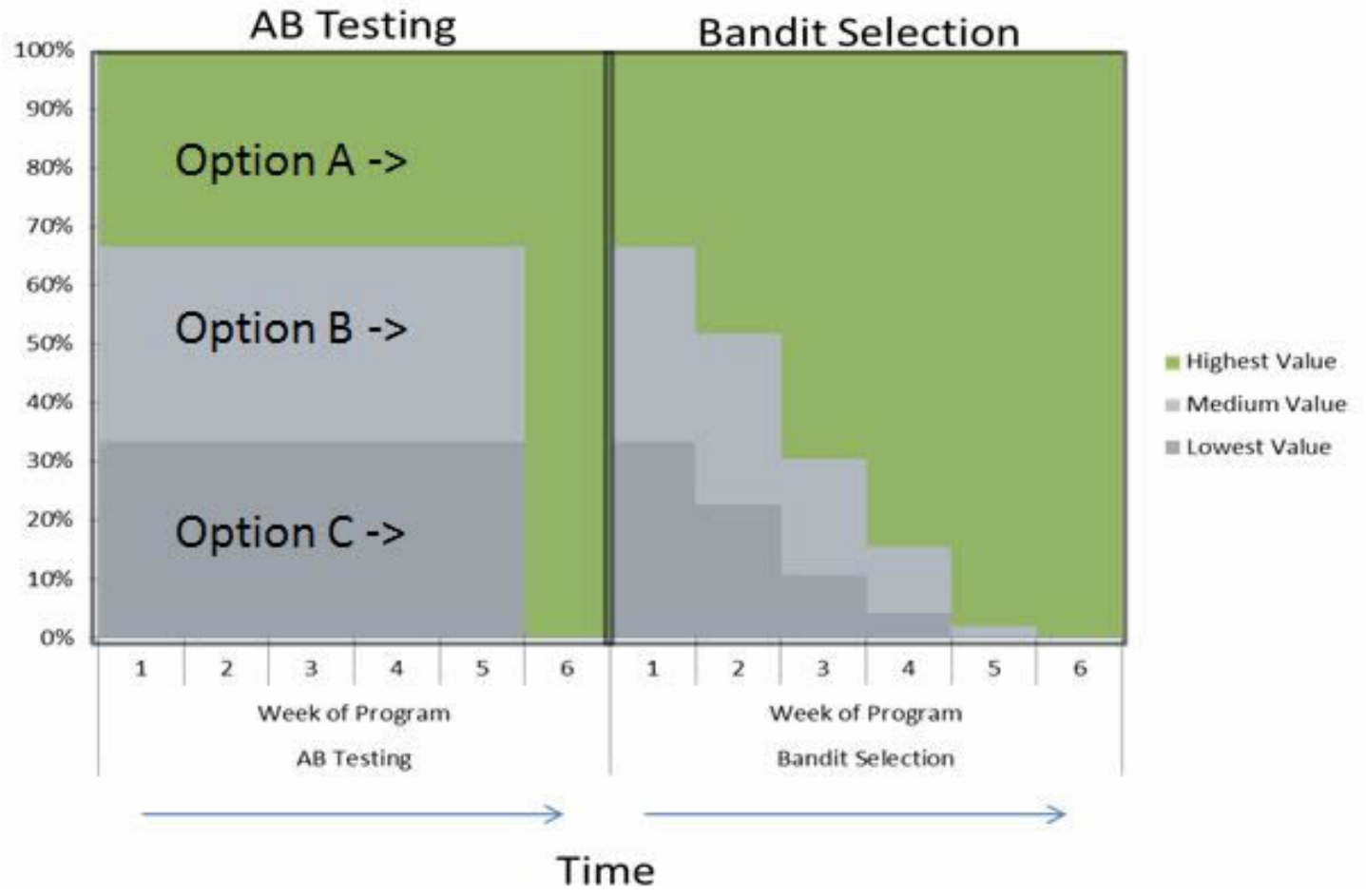
K Slot Machines

- Choose a machine and get a reward
- Only have T chances
- **Goal**: maximize the cumulative rewards
- How to choose the machines (arms)?

Two Phases: Explore-Exploit

- **Explore**: Try different arms.
- **Exploit**: Play the most rewarding arm.
- **A/B Testing**:
  - Draw the arms uniformly during $T/2$ rounds
  - Then draw the empirical best until the end
  - Jump from pure exploration to pure exploitation.
- **Bandits**: Transit from *exploration* to *exploitation* more smoothly.

A/B Testing v.s. Bandits



Multi-Armed Bandits (MAB)

- A set of K arms, denoted by $\mathcal{A}$.
- The number of rounds T is fixed.
- At the round t, player picks arm at $a_{i_t} \in \mathcal{A}$.
- Player gets rewards $r_{i_t} \in [0, 1]$ for the chosen arm (no full information).
- Assume for each arm a, its rewards $\overset{iid}{\sim} \mathcal{D}_a$ (unknown)
- People usually call this **iid** setting "**stochastic bandits**".

Regret

- **Regret**:

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

  where $\mu(a_i) := \mathbb{E}(r_i)$ denotes the **mean reward** for the arm a_i and $\mu^* := \max_{a \in \mathcal{A}} \mu(a)$ is the **optimal** mean reward.

- Maximizing rewards $\iff$ Minimizing regret.

- MAB Algorithms

Exploration-First Algorithm

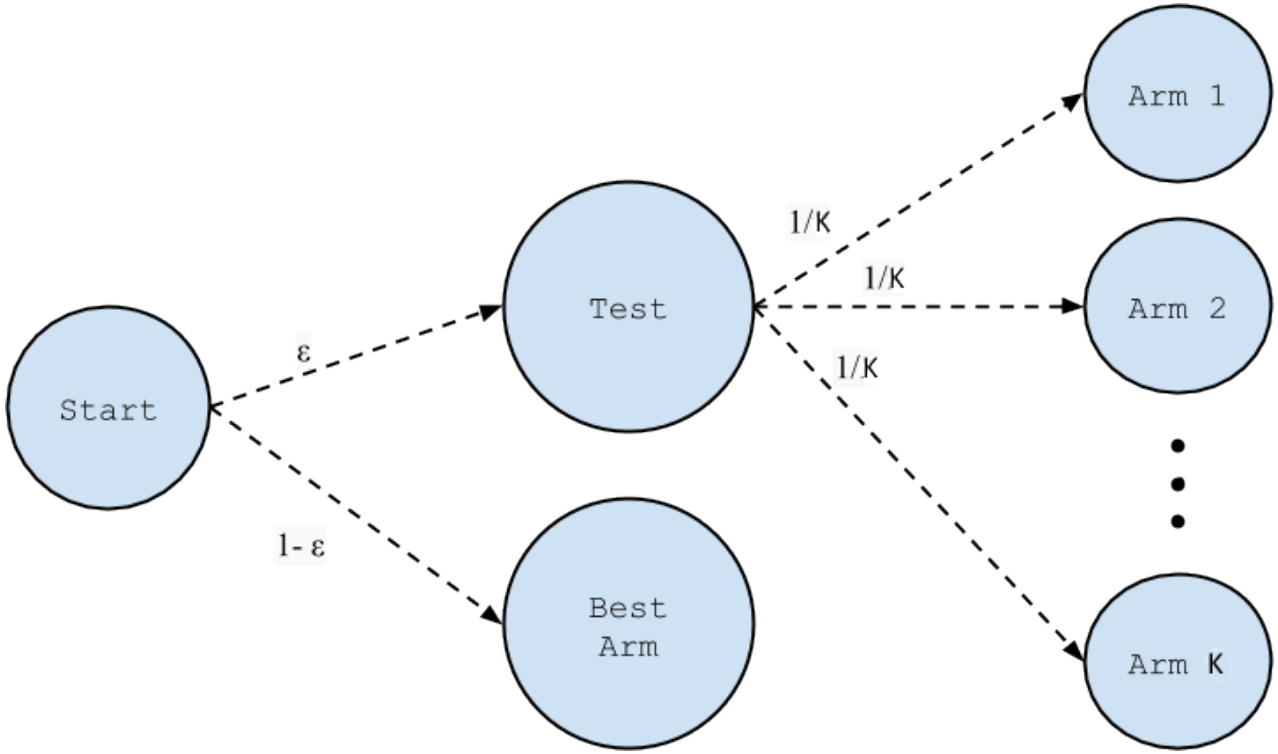
- A/B testing style
- **Exploration** phase: try each arm N times
- Select the arm a^{opt} with the highest average reward
- **Exploitation** phase: play arm a^{opt} in all remaining rounds
- Regret bound: with a proper value of N,

$$R(T) \leq 1 + \frac{2K \log(2KT)}{\Delta_{\min}^2},$$

where $\Delta_{\min} = \min_{a \in \mathcal{A} \setminus \{a^*\}} \big(\mu^* - \mu(a)\big)$.

Epsilon-Greedy Algorithm

- Exploit: with probability $1 - \epsilon$, play the best arm so far
- Explore: with probability ϵ, choose an arm **randomly**



Epsilon-Greedy Algorithm

- If ϵ is held constant, only a linear bound on $R(T)$ can be achieved.
- When ϵ decreases with time, it could obtain

$$R(T) \leq \mathcal{O}\left(\frac{K \log T}{\Delta_{\min}^2}\right)$$

with $\epsilon_t = \mathcal{O}(1/t)$.

Adaptive Exploration

- A big **flaw**: the exploration is completely **random**, does not depend on the **history** of the observed rewards.
- Adaptive v.s. non-adaptive exploration.

Softmax Algorithm

- **Idea**: pick each arm with a probability that is proportional to its average reward.
- Given initial empirical means $\hat{\mu}_i$ for $i = 1, \ldots, K$
- $\mathbb{P}(\texttt{Play } k\texttt{th Arm}) = \frac{\hat{\mu}_k}{\sum_{i=1}^{K} \hat{\mu}_i}$
- **Softmax Algorithm** (Boltzmann Exploration):

$$\mathbb{P}(\texttt{Play } k\texttt{th Arm}) = \frac{\exp(\hat{\mu}_k/\tau)}{\sum_{i=1}^{K} \exp(\hat{\mu}_i/\tau)}$$

- τ: a temperature parameter, controlling the randomness of the choice.
- **Annealing**: dynamically decrease τ over time.

Epsilon-Greedy and Softmax

- Tend to select the best arm currently.
- Sometimes decide to explore and choose an option that is not currently best.
- **Drawbacks**:
  - ignore the **randomness** (noise) of rewards
  - could be easily misled by negative experiences: **non-robust**

UCB1 Algorithm

- Given initial empirical means $\hat{\mu}_i$ for $i = 1, \ldots, K$
- Play the jth arm with

$$j = \arg\max_{i=1,\ldots,K} \texttt{UCB}(a_i) := \hat{\mu}_i + \sqrt{\frac{\alpha \log t}{n_i}},$$

  where n_i is the number of times a_i was played so far.
- Usually, $\alpha = 2$.

UCB

- UCB: Upper Confidence Bounds
- $\texttt{UCB}(a_i) := \hat{\mu}_i + \sqrt{\frac{\alpha \log t}{n_i}}$
- By Hoeffding's inequality

$$\mathbb{P}\left(\hat{\mu}_i + \sqrt{\frac{\alpha \log t}{n_i}} \leq \mu_i\right) \leq \exp\left(-2n_i\left(\frac{\alpha \log t}{n_i}\right)\right) = \frac{1}{t^{2\alpha}}$$

Regret Bound for UCB1

- Regret Bound for UCB1 (Auer et al., 2002)

$$\begin{aligned}\mathbb{E}(R(T)) &\leq 8 \sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + \left(1 + \frac{\pi^2}{3}\right) \sum_{i=1}^{K} \Delta_i, \\ &\leq \mathcal{O}\left(\frac{K \log T}{\Delta_{\min}}\right),\end{aligned}$$

  where $\Delta_i = \mu^* - \mu_i$.

- No need to set any parameter!

UCB1-Tuned

- A variant which takes into account the **variance** of each arm.
- Play the jth arm with

$$j = \arg\max_{i=1,\ldots,K} \ \hat{\mu}_i + \sqrt{\frac{\log t}{n_i} \min\left(\frac{1}{4}, V_i\right)},$$

where

$$V_i = \hat{\sigma}_i^2 + \sqrt{\frac{2 \log t}{n_i}}$$

- $\hat{\sigma}_i^2$ can be computed from the historical rewards.

Thompson Sampling

- **Bayes** point of view: quantify the reward in terms of a **distribution** rather than a point estimate.
- Prior: assume rewards follow $\texttt{Beta}(S, F)$. S: wins, F: fails.
- Set $S_i = F_i = 0$ for $i = 1, \ldots, K$.
- At the t-th round,
  - Sample $\theta_i \sim \texttt{Beta}(S_i + 1, F_i + 1)$, $\forall i$.
  - Play arm $j_t := \arg\max_i \theta_i$ and receive reward r_t
  - Generate $w_t \sim \texttt{Ber}(r_t)$.
  - Update $S_{j_t} = S_{j_t} + 1$ if $w_t = 1$, otherwise $F_{j_t} = F_{j_t} + 1$.

Regret Bound for Thompson Sampling

- Regret Bound for Thompson sampling (Agrawal and Goyal, 2012)

$$\mathbb{E}(R(T)) \leq \mathcal{O}\left(\left(\sum_{i:\Delta_i > 0} \frac{1}{\Delta_i^2} \right)^2 \log T \right).$$

- This rate is the same as that in `UCB1`, but is inferior in terms of constant factors and dependence on Δ.
- **Cheap** in computation and have competitive performance to `UCB1`.

Comparison

- We generated **bernoulli** rewards with $K = 5, T = 10000$.

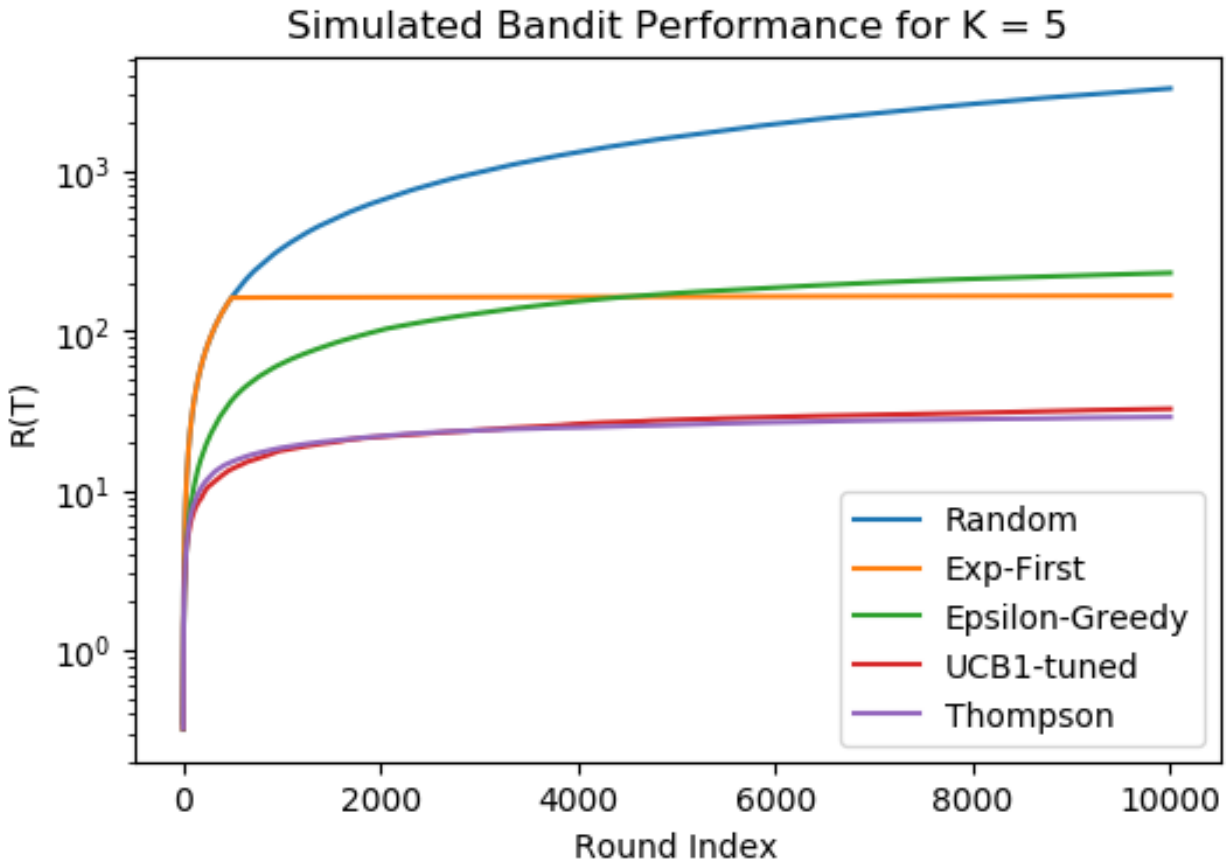


Figure: Average of 100 runs

Adversarial Bandits

- **Non-stochastic**: sometimes the reward distribution cannot be modeled by a *stationary* distribution.
- Adversarial bandit game: at time t
  - The player chooses arm a_{i_t}
  - Simultaneously, an adversary chooses a vector of rewards

$$[r_1^t, r_2^t, \ldots, r_K^t]$$

  - The player only receives the reward $r_{i_t}^t$.
- We still assume $r \in [0, 1]$ for simplicity.

Adversarial Bandits

- For any deterministic algorithm there exists a sequence of rewards such that $R(T) \geq T/2$.
- The idea is to add **randomization** to the selection of the arm.

Exp3 Algorithm

- Set $p_i^0 = 1/K, i = 1, \ldots, K$ and $G_i^0 = 0$ (estimated cumulative rewards)
- At the round t ($t \geq 1$):
  - Sample arm i based on the probability $\boldsymbol{p}^{t-1}$, observe $r_{i_t}^t$
  - Estimating rewards for all actions:

$$\boldsymbol{g}^t = [0, \ldots, 0, \frac{r_{i_t}^t}{p_i^{t-1}}, 0, \ldots, 0]$$

  - Update cumulative rewards:

$$\boldsymbol{G}^t = \boldsymbol{G}^{t-1} + \boldsymbol{g}^t$$

  - Update sampling probability

$$\boldsymbol{p}^t = \frac{\exp(-\eta \boldsymbol{G}^t)}{\langle 1, \exp(-\eta \boldsymbol{G}^t) \rangle}$$

Exp3 Algorithm

- **Unbiased** estimate of *unseen* rewards

$$\mathbb{E}[g_j^t] = \mathbb{E}\Big[\frac{r_j^t}{p_j^{t-1}} 1_{j=i_t}\Big] = \frac{r_j^t}{p_j^{t-1}} \mathbb{E}[1_{j=i_t}] = \frac{r_j^t}{p_j^{t-1}} p_j^{t-1} = r_j^t$$

- A variant:

$$\boldsymbol{p}^t = (1-\gamma)\frac{\exp(-\eta \boldsymbol{G}^t)}{\langle 1, \exp(-\eta \boldsymbol{G}^t)\rangle} + \gamma \frac{1}{K}, \quad \gamma \in (0, 1]$$

- A mixture of the uniform distribution (*exploration*) and a exponential weights (*exploitation*).

Contextual Bandits

- Some additional information available at each round.
- This information (**context**) could help with the arm choices.
- *Web article recommendation system.*
- Contextual information about visitors:
  - demographic
  - browsing history
  - location

Contextual Bandits

- Each round t proceeds
  - The player observes a "context" x_t
  - The player chooses an arm a_{i_t}
  - Reward $r_t \in [0, 1]$ is realized
- Reward distribution could be stochastic (iid) or adversarial.

Example: Web Article Recommendation

- 3 articles but only one space: $K = 3$
- **2 user features**: if they had clicked on a *sports* article or a *politics* article in the past.
- Find which articles are best for people given their past click behaviors.

arm	clk_sports	clk_politics	reward
1	1	0	0.58
1	1	1	0.69
2	0	0	0.19
3	0	1	0.51
...	...	...	...

LinUCB

- A combination of supervised learning and UCB
- At each round, fit a ridge regression for **each** arm

$$\hat{\boldsymbol{\theta}}_a = (\boldsymbol{X}_a^\top \boldsymbol{X}_a + \boldsymbol{I})^{-1} \boldsymbol{X}_a^\top \boldsymbol{r}_a$$

- $\boldsymbol{X}_a$ is the feature matrix for the arm a **so far**.
- $\boldsymbol{r}_a$ is the reward vector from choosing the arm a **so far**.

$$\boldsymbol{X}_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \ \boldsymbol{X}_2 = \begin{bmatrix} 0 & 0 \end{bmatrix}, \ \boldsymbol{X}_3 = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$\boldsymbol{r}_1 = \begin{bmatrix} 0.58 \\ 0.69 \end{bmatrix}, \quad \boldsymbol{r}_2 = \begin{bmatrix} 0.19 \end{bmatrix}, \quad \boldsymbol{r}_3 = \begin{bmatrix} 0.51 \end{bmatrix}$$

LinUCB

- Choose the arm which gives the largest UCB for a **new** observed context $\boldsymbol{x}_t^\top$

$$a^+ = \arg\max_{a \in \mathcal{A}} \left(\underbrace{\boldsymbol{x}_t^\top \hat{\boldsymbol{\theta}}_a}_{prediction} + \alpha \underbrace{\sqrt{\boldsymbol{x}_t^\top (\boldsymbol{X}_a^\top \boldsymbol{X}_a + \boldsymbol{I})^{-1} \boldsymbol{x}_t}}_{standard\ deviation} \right)$$

LinUCB

- **Feature engineering** is extremely important.
- Could use both user features and arm features.
- `Hybrid-LinUCB` allows arms to **share** contextual variables.
- `GLM-UCB` for rewards following distributions from exponential family.