Multi-Armed Bandits and Applications

Sangwoo Mo

KAIST

swmo@kaist.ac.kr

December 23, 2016

Papers

Theory

- Auer et al. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 2002.

Application

- Kveton et al. Cascading Bandits: Learning to Rank in the Cascade Model. *ICML*, 2015.
- Caron & Bhagat. Mixing Bandits: A recipe for Improved Cold-Start Recommendations in a Social Network. *SNA-KDD*, 2013.

Overview

1. Multi-Armed Bandit
2. UCB: The Optimal Algorithm
3. Application 1: Ranking
4. Application 2: Recommendation

Multi-Armed Bandit

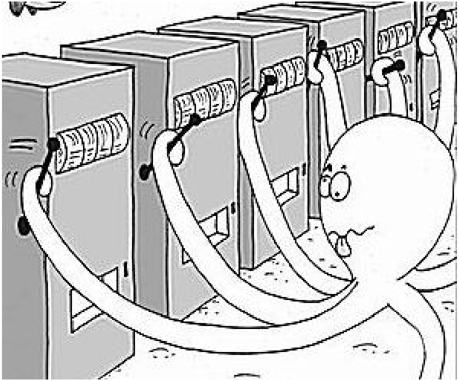
What is Multi-Armed Bandit?

- One-Armed Bandit = Slot Machine (English slang)

source: infoslotmachine.com

What is Multi-Armed Bandit?

- Multi-Armed Bandit = Multiple Slot Machine
- Objective: maximize reward in a casino



source: Microsoft Research

Real Motivation

- A/B Test, Online Advertisement, etc.
- Objective: maximize conversion rate, etc.

source: VWO

Real Motivation

CONTROL

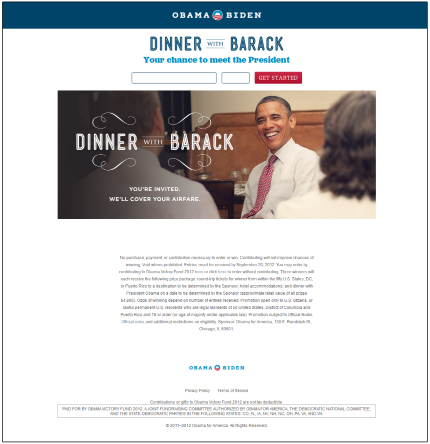
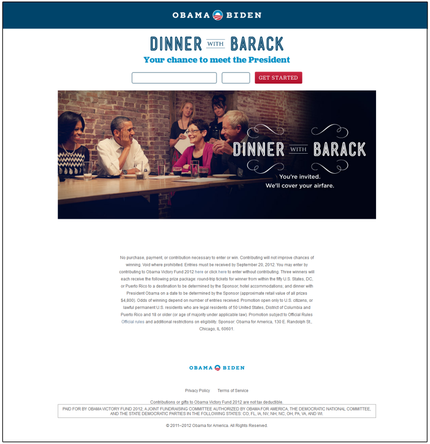


IMAGE VARIATION



⬆ +19%

http://kylerush.net

Problem Setting

- # of arms K, # of rounds T
- For each round $t = 1, ..., T$
  1. the reward vector $r_t = (r_{1,t}, ..., r_{K,t})$ is generated
  2. the agent chooses an arm $i_t \in \{1, ..., K\}$
  3. the agent recieves the reward $r_{i_t,t}$
- Remark: rewards of unchosen arms $r_{i \neq i_t,t}$ are not revealed
- We call this partially observable property as the bandit setting

Problem Setting (Stochastic Bandit)

- The reward $r_{i,t}$ follows the probability distribution $\mathcal{P}_i$, with mean μ_i
- Here, the agent should find the arm with the highest μ_i

source: Pandey et al.'s slide

- Today, we will only consider the stochastic bandit

Objective

- Objective: minimize the (expected cumulative) regret

$$R_T = \mathbb{E}[\sum_{t=1}^{T}(r_{i^*,t} - r_{i_t,t})] = \sum_{t=1}^{T}(\mu^* - \mu_{i_t}) = \sum_{i=1}^{K}\Delta_i n_i$$

where $i^* = \arg\max[\mu_i]$, $\Delta_i = \mu^* - \mu_i$, and $n_i = \sum_{t=1}^{T} \mathbb{1}[i_t = i]$

- It is shown that the asymptotic lower bound [LR 85] of the regret is

$$\lim_{T\to\infty} \frac{R_T}{\log T} \geq \sum_{\Delta_i > 0} \frac{\Delta_i}{KL(\mathcal{P}_i || \mathcal{P}_{i^*})}$$

- We call a bandit algorithm is optimal if its regret is $O(\log T)$

Exploration-Exploitation Dilema

- Exploration vs Exploitation

  exploration gather more information

  exploitation make the best decision with given information

- Two Naïve Algorithms

  Random (= full exploration): choose arm randomly

  Greedy (= full exploitation): choose the empirical best arm

- Both algorithm occurs the linear regret (why?)

Exploration-Exploitation Dilema

- Fundamental question of bandit:
  How to balance between exploration and exploitation?

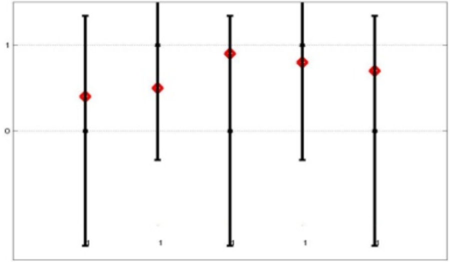
source: RSM Discovery

UCB: The Optimal Algorithm

Motivation of UCB

- Recall: Greedy algorithm occurs the linear regret
- Reason: It chooses the wrong answer with overconfidence

- Idea: Add confidence bonus to the estimated mean!
  (If the estimator is reliable, choose less; if not, choose more)



source: Garivier & Cappé's slide

UCB1 [ACF 02]

- UCB1: choose the arm s.t.

$$i_t = \arg\max \left[\hat{\mu}_i + \underbrace{\sqrt{\frac{c \log t}{n_i}}}_{\text{ucb}_i} \right]$$

Theorem

Let $r_{i,t}$ is bounded in $[0,1]$. Let $c = 2$. Then, the regret of UCB1 is

$$R_t = \left[8 \sum_{i:\mu_i<\mu^*} \left(\frac{\log t}{\Delta_i}\right) \right] + \left(1 + \frac{\pi^2}{3}\right) \left(\sum_{i=1}^{K} \Delta_i \right).$$

UCB1 [ACF 02]

Sketch of Proof.

If the agent chooses a suboptimal arm,

$$\hat{\mu}_i + \mathrm{ucb}_i \geq \hat{\mu}^* + \mathrm{ucb}^*.$$

With some modification,

$$\underbrace{\hat{\mu}_i - (\mu_i + \mathrm{ucb}_i)}_{A} + \underbrace{(\mu_i + 2\mathrm{ucb}_i) - \mu^*}_{B} \geq \underbrace{\hat{\mu}^* - (\mu^* - \mathrm{ucb}^*)}_{-C}.$$

Here, at least one of A, B, or C should be nonnegative.

UCB1 [ACF 02]

Sketch of Proof (Cont.)

By Chernoff bound, $Pr(A \geq 0) \simeq 0$ and $Pr(C \geq 0) \simeq 0$.

$$Pr(A) = Pr(\hat{\mu}_i \geq \mu_i + \mathrm{ucb}_i) \leq \exp(-2\frac{2\log t}{n_i} n_i) = t^{-4}$$

Also, $Pr(B \geq 0) = 0$ if $n_i \geq \frac{8 \log t}{\Delta_i^2}$.

$$\mu_i + 2\mathrm{ucb}_i = \mu_i + 2\sqrt{\frac{2\log t}{n_i}} \leq \mu_i + \Delta_i = \mu^*$$

Combining two results,

$$\mathbb{E}[n_i] \leq \frac{8\log t}{\Delta_i^2} + \sum t^{-4} = O(\log t)$$

UCB variants

- UCB1 achived the optimality in order, but not in constant
- Recall: The assymptotic lower bound [LR 85] of the regret is

$$\lim_{T\to\infty} \frac{R_T}{\log T} \geq \sum_{\Delta_i>0} \frac{\Delta_i}{KL(\mathcal{P}_i||\mathcal{P}_{i^*})}$$

- A lot of UCB variants were proposed to achieve the optimality
- Finally, KL-UCB [GC 11] and Bayes-UCB [KCG 12] achieved it
- Proof scheme is similar to UCB1

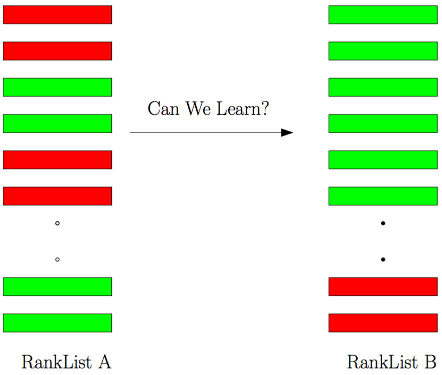
  (1) UCB term goes to zero by AH-like inequality (A/C of UCB1)

  (2) residual term after $O(\log T)$ goes to zero (B of UCB1)

Application 1: Ranking

Learning-to-Rank

- In some applications, we need to find top-K, not the best



source: Gupta's tutorial slide

Relation to Multi-Armed Bandit

- There is an exploration-exploitation dilema

  exploration gather more information about user's preference

  exploitation recommend the top-K list with given information
- It is natural to apply bandit approach

Algorithm [KSWA 2015]

Algorithm 1 UCB-like algorithm for cascading bandits.

// Initialization
Observe $\mathbf{w}_0 \sim P$
$\forall e \in E : \mathbf{T}_0(e) \leftarrow 1$
$\forall e \in E : \hat{\mathbf{w}}_1(e) \leftarrow \mathbf{w}_0(e)$

for all $t = 1, \ldots, n$ **do**
  Compute UCBs $\mathbf{U}_t(e)$ (Section 3.2)

  // Recommend a list of K items and get feedback
  Let $\mathbf{a}_1^t, \ldots, \mathbf{a}_K^t$ be K items with largest UCBs
  $\mathbf{A}_t \leftarrow (\mathbf{a}_1^t, \ldots, \mathbf{a}_K^t)$
  Observe click $\mathbf{C}_t \in \{1, \ldots, K, \infty\}$

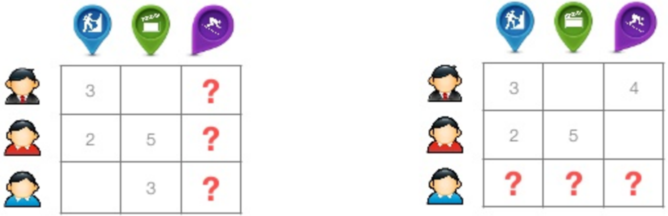
  // Update statistics
  $\forall e \in E : \mathbf{T}_t(e) \leftarrow \mathbf{T}_{t-1}(e)$
  for all $k = 1, \ldots, \min\{\mathbf{C}_t, K\}$ **do**
    $e \leftarrow \mathbf{a}_k^t$
    $\mathbf{T}_t(e) \leftarrow \mathbf{T}_t(e) + 1$
    $\hat{\mathbf{w}}_{\mathbf{T}_t(e)}(e) \leftarrow \dfrac{\mathbf{T}_{t-1}(e)\hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) + \mathbb{1}\{\mathbf{C}_t = k\}}{\mathbf{T}_t(e)}$

source: original paper

Application 2: Recommendation

Cold-Start Problem

- Collaborative filtering is widely used for recommendation
- It is highly effective when there is sufficient data, but suffers when new user enters; which is called cold-start problem



source: Elahi's survey slide

Relation to Multi-Armed Bandit

- There is an exploration-exploitation dilema
  exploration gather more information about user's preference
  exploitation recommend the best item with given information
- It is natural to apply bandit approach

Algorithm [CB 2013]

Algorithm 3 MixNeigh

Require: neighbor estimates $\overline{Y}_a$, c_a for all $a \in S_u$

1: $\overline{\mathbf{X}}, \mathbf{n} \leftarrow \mathbf{0}, \mathbf{0}$
2: **for** $t \geq 1$ **do**
3: **for** $a \in S_u$ **do**
4: $b_a \leftarrow \sqrt{\dfrac{2 \log t}{n_a}}$
5: $\overline{Z}_a \leftarrow \begin{cases} \overline{Y}_a & \text{if } |\overline{X}_a - \overline{Y}_a| < \dfrac{b_a - c_a}{2} \\ \overline{X}_a & \text{otherwise} \end{cases}$
6: $a \leftarrow \arg\max_{a \in S_u} \{\overline{Z}_a\}$
7: **pull** arm a, getting reward $X_{a,t}$
8: $n_a \leftarrow n_a + 1$
9: $\overline{X}_a \leftarrow 1/n_a X_{a,t} + (1 - 1/n_a)\overline{X}_i$

source: original paper

Take Home Message

- Bandit is an insteresting topic for both theorists and practitioners
- The core of bandit is partially observable and exp-exp dilema
- UCB is one great idea to attack the problem

- Single message to keep in mind:

If your research encounters exp-exp dilema,

consider to apply bandit approach!