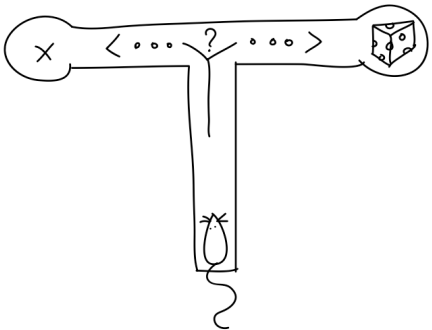
Introduction to Multi-armed Bandit

Runyu Tang

March 8, 2023

Mouse learning a T-maze:
The mice faced the dilemma of choosing to go left or right after starting in the bottom of a T-shaped maze, not knowing each time at which end they would find food.
"Two-armed bandit" machine:
Humans could choose to pull either the left or the right arm of the machine, each giving a random pay-off with the distribution of pay-offs for each arm unknown to the human player.('bandit' because they steal your money)



Example: clinical trials

- You have multiple treatment options for cancer X.
- Cancer X patients arrive sequentially into your clinical trial.
- How do you treat them?
- Want to save as many lives as possible overall.

Example: Feynman's Restaurant Problem

- A restaurant has multiple dishes.
- Every time you come and try a dish you get a noisy observation of its quality.
- You want to maximize the total quality of food you consume.

Whether the 'buy it now' button should be placed at the top of the product page or at the bottom.

- In traditional A/B testing, the objective of the statistician is to decide which website is better.
- When using a bandit algorithm, there is no need to end the trial. The algorithm automatically decides when one version of the site should be shown more often than another.

We need to learn from past to predict the future and optimze.
Core properties:

- Sequentially taking actions of unknown quality
- The feedback provides information about quality of chosen action
- There is no state

It's a special tractable case of reinforcement learning.

Key feature of the solution: balancing *exploitation* and *exploration*.

- Exploitation: use the current knowledge to focus on the action that seems to yield the highest rewards.
- Exploration: explore further the other actions to identify with better precision which action is actually the best.

Optimal policy

$\mathbb{E}[a] = -\$0.5$

$\mathbb{E}[b] = -\$0.2$

$\mathbb{E}[c] = \$0.1$

$\mathbb{E}[d] = \$0.11$

Learn a policy

-1, -1, 5

$\widehat{\mathbb{E}}[a] = 1$

-0.2, -0.2

$\widehat{\mathbb{E}}[b] = -0.2$

-0.5, -0.5, -0.5

$\widehat{\mathbb{E}}[c] = -0.5$

-2, -2

$\widehat{\mathbb{E}}[d] = -2$

- k arms, each with a reward distribution p_i.
- In each period $t = 1, \ldots, T$, we pull an arm $A_t \in \{1, \ldots, k\}$, and observe a reward $x_t \sim p_{A_t}$.
- History: $H_{t-1} = (A_1, x_1, A_2, x_2, \ldots, A_{t-1}, x_{t-1})$.
- Policy: $\pi_t(H_{t-1}) = A_t$

The objective is

$$\max_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} x_t\right]$$

or equvialently, to minimize the regret

$$\min_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} \max_i p_i - p_{A_t}\right]$$

- Bad regret = we keep performing suboptimally and the regret grows linearly T.
  - Namely, $\text{Regret}(T)/T \to c > 0$.
- Good regret = we eventually learn to act optimally and the regret grows sub-linearly in T
  - Namely, $\text{Regret}(T)/T \to 0$.

Try each arm once. Then always pull the arm that appears the best (highest average reward to date).
This can fail spectacularly:

- Consider a two-armed bandit: $x_1 \sim Ber(0.6)$, $x_2 \sim Ber(0.4)$.
- With probability 0.16, we start out by seeing 0 for arm 1 and 1 for arm 2.
- So we always pull arm 2 and accumulate 0.2 regret at each step.
- We have linear regret: $\text{Regret}(T) \geq cT$.

- Exploration phase: try each arm m times
- Exploitation phase: for $t = km + 1, \ldots, n$, pull the arm with the highest average reward.

1: **Input** m.
2: In round t choose action

$$A_t = \begin{cases} (t \bmod k) + 1\,, & \text{if } t \le mk\,; \\ \operatorname{argmax}_i \hat{\mu}_i(mk)\,, & t > mk\,. \end{cases}$$

(ties in the argmax are broken arbitrarily)

where $\hat{\mu}_i(t) = \sum_{s=1}^{t} \mathbb{I}\{A_s = i\} x_s / \sum_{s=1}^{t} \mathbb{I}\{A_s = i\}$ denotes the empirical average reward of arm i up to time t.

Regret analysis

Theorem 1.

When ETC is interacting with any 1-subgaussian bandit and $1 \leq m \leq n/k$,

$$R_n \leq m \sum_{i=1}^{k} \Delta_i + (n - mk) \sum_{i=1}^{k} \Delta_i \exp\left(-\frac{m\Delta_i^2}{4}\right).$$

where $\Delta_i = \mu^* - \mu_i$ *is the suboptimality gap between the mean of action* i *and the optimal action.*

σ-Subgaussian ($\mathbb{E}[X] = 0$): $E(exp(\lambda X)) \leq \exp(-\sigma^2\lambda^2/2), \forall\lambda$.
Property: $\forall\epsilon > 0$, $P(X > \epsilon) \leq \exp(-\epsilon^2/2\sigma^2)$.

In probability theory, concentration inequalities provide bounds on how a random variable deviates from some value (typically, its expected value).

- Markov inequality (X nonnegative, $a > 0$):

$$\Pr(X \geq a) \leq \frac{\mathrm{E}(X)}{a}$$

- Chebyshev's inequality ($E(X)$ and $\mathrm{Var}(X)$ finite):

$$\Pr(|X - \mathrm{E}[X]| \geq a) \leq \frac{\mathrm{Var}[X]}{a^2},$$

- Hoeffding's inequality ($a_i \leq X_i \leq b_i$):

$$\mathrm{P}\left(S_n - \mathrm{E}\left[S_n\right] \geq t\right) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$$

Proof Assume without loss of generality that the first arm is optimal, which means that $\mu_1 = \mu^* = \max_i \mu_i$. By the decomposition given in Lemma 4.5, the regret can be written as

$$R_n = \sum_{i=1}^{k} \Delta_i \mathbb{E}\left[T_i(n)\right] . \tag{6.1}$$

In the first mk rounds, the policy is deterministic, choosing each action exactly m times. Subsequently it chooses a single action maximising the average reward during exploration. Thus,

$$\begin{aligned}\mathbb{E}\left[T_i(n)\right] &= m + (n - mk)\mathbb{P}\left(A_{mk+1} = i\right) \\ &\leq m + (n - mk)\mathbb{P}\left(\hat{\mu}_i(mk) \geq \max_{j \neq i} \hat{\mu}_j(mk)\right) . \end{aligned} \tag{6.2}$$

The probability on the right-hand side is bounded by

$$\begin{aligned}\mathbb{P}\left(\hat{\mu}_i(mk) \geq \max_{j \neq i} \hat{\mu}_j(mk)\right) &\leq \mathbb{P}\left(\hat{\mu}_i(mk) \geq \hat{\mu}_1(mk)\right) \\ &= \mathbb{P}\left(\hat{\mu}_i(mk) - \mu_i - \left(\hat{\mu}_1(mk) - \mu_1\right) \geq \Delta_i\right) .\end{aligned}$$

The next step is to check that $\hat{\mu}_i(mk) - \mu_i - \left(\hat{\mu}_1(mk) - \mu_1\right)$ is $\sqrt{2/m}$-subgaussian, which by the properties of subgaussian random variables follows from the definitions of $(\hat{\mu}_j)_j$ and the algorithm. Hence by Corollary 5.5,

$$\mathbb{P}\left(\hat{\mu}_i(mk) - \mu_i - \hat{\mu}_1(mk) + \mu_1 \geq \Delta_i\right) \leq \exp\left(-\frac{m\Delta_i^2}{4}\right) . \tag{6.3}$$

Substituting Eq. (6.3) into Eq. (6.2) and the regret decomposition (Eq. (6.1)) gives the result. □

Regret analysis

$$R_n \leq m \sum_{i=1}^{k} \Delta_i + (n - mk) \sum_{i=1}^{k} \Delta_i \exp\left(-\frac{m\Delta_i^2}{4}\right).$$

This illustrate exploration-exploitation tradeoff!

- Explore too much (m large) then the first term is large.
- Exploit too much (m small) then the second term is large.

If we choose $m \asymp n^{2/3}$, then the regret $R_n = O(Kn^{2/3})$.
Note. the choice of m requires the knowledge of n.

- With probablity ϵ_t, explore an arm uniformly at random.
- With probablity $1 - \epsilon_t$, explore the arm with the highest reward so far.

Again, ϵ_t trade-off between exploration and exploitation.
If $\epsilon_t \asymp t^{-1/3}(K \log t)^{1/3}$, then the regret

$$R_t = O(t^{2/3}(K \log t)^{1/3})$$

.

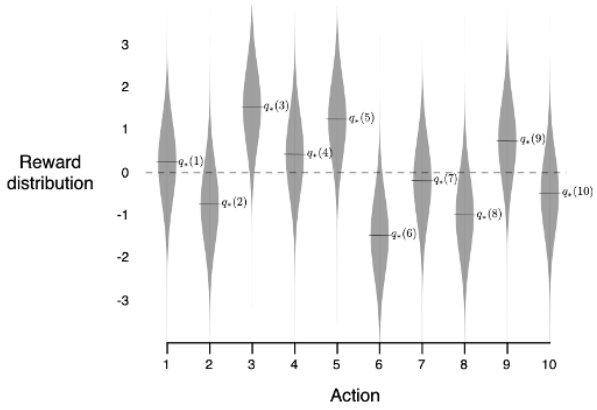
- Similar regret rate to ETC
- But it does not require knowing n, the regret bound holds for any t (*anytime* algorithm)
- the exploration is spread more evenly over time than ETC

The regret of ETC and ϵ-Greedy aren't that great, because they do not adapt exploration to the observed data.

Example

- k=10
- A=1,2,3,4,5,6,7,8,9,10
- $\Pr\{r|a\} \sim N(q(a), 1)$

The optimal policy is to always choose a_3,
But we don't know that initially

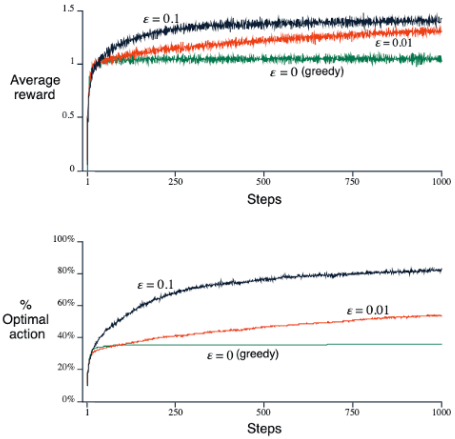


Example

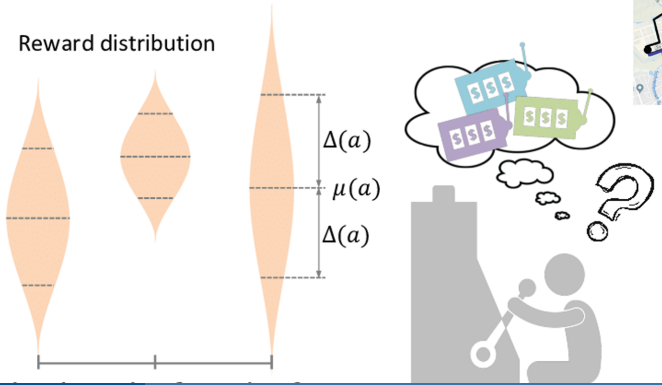
Average performance of ϵ-greedy action-value methods on the 10-armed tested

- $\epsilon = 0$
- $\epsilon = 0.1$
- $\epsilon = 0.01$

Averages over 2000 runs each with 1000 steps All methods used sample averages as their action-value estimates



Based on the idea of "optimism in the face of uncertainty."



Based on the idea of "optimism in the face of uncertainty."

Algorithm:

- compute the empirical mean of each arm and a confidence interval;
- use the upper confidence bound as a proxy for goodness of arm.

Note: confidence interval chosen so that true mean is very unlikely to be outside of confidence interval

1: **Input** k and δ
2: **for** $t \in 1, \ldots, n$ **do**
3: Choose action $A_t = \text{argmax}_i \, \text{UCB}_i(t-1, \delta)$
4: Observe reward X_t and update upper confidence bounds
5: **end for**

We can use

$$\text{UCB}_i(t-1,\delta) = \hat{\mu}_i(t-1) + \sqrt{\frac{2\log(1/\delta)}{N_i(t-1)}}$$

Why?

Fix an arm i, with probability at $1-2\delta$, (by the subgaussian assumption), we have

$$|\mu_i - \hat{\mu}_i(t)| \leq \sqrt{2\log(\frac{1}{\delta})/N_i(t)}$$

Regret analysis

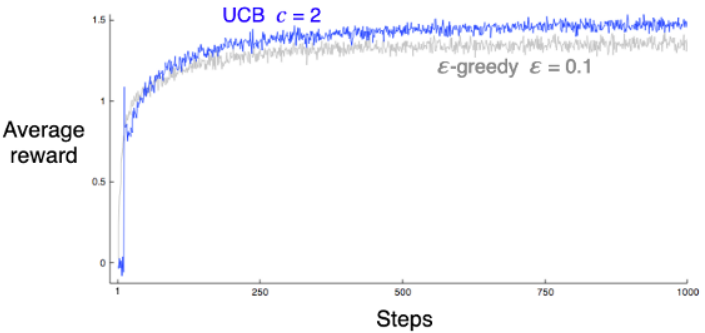
Theorem 2.

Consider a UCB Algorithm on a stochastic k-armed 1-subgaussian bandit problem. For any horizon n, if $\delta = 1/n^2$, then

$$R_n \leq 3\sum_{i=1}^{k} \Delta_i + \sum_{i:\Delta_i>0} \frac{16 \log n}{\Delta_i}.$$

Choosing $\delta \asymp 1/n^2$, we have $R_n = O(\sqrt{Kn \log n})$. (Recall that ETC has $O(n^{2/3})$ regret.)

We still need the knowledge of n. In practice, we may use $\mathrm{UCB}_i(t-1, \delta) = \hat{\mu}_i(t-1) + \alpha\sqrt{\frac{2\log t}{N_i(t-1)}}$ with a tuning parameter $\alpha > 0$.


UCB $c = 2$
ε-greedy $\varepsilon = 0.1$
Average reward
1.5
1
0.5
0
1
250
500
750
1000
Steps

- So far we have made no assumptions about the reward distribution R
- Bayesian bandits exploit prior knowledge of rewards $p(R)$
- They compute posterior distribution of rewards $p(R|H_t)$
- Use posterior to guide exploration

$$\mathrm{BR}_n = \mathbb{E}\left[\sum_{t=1}^{n} (\mu_{A^*} - \mu_{A_t})\right]$$

- Thompson Sampling is an alternative Bayesian method with similar regret guarantees.
- Before the game starts, the learner chooses a prior over a set of possible bandit environments. In each round, the learner samples an environment from the posterior and acts according to the optimal action in that environment.

Input Bayesian bandit environment $(\mathcal{E}, \mathfrak{B}(\mathcal{E}), Q, P)$
for $t = 1, 2, \ldots, n$ **do**
    Sample $\nu_t \sim Q(\cdot \mid A_1, X_1, \ldots, A_{t-1}, X_{t-1})$
    Choose $A_t = \operatorname{argmax}_{i \in [k]} \mu_i(\nu_t)$
end for

Note. the goal is to optimize *Bayesian regret* now.

Let's consider a Beta distribution prior over the mean rewards of the Bernoulli bandits:

$$p(\theta_k) = \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \theta_k^{\alpha_k - 1}(1 - \theta_k)^{\beta_k - 1}$$

$$\Gamma(n) = (n-1)!$$

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{p(D)}$$

The posterior is also a Beta! Because beta is conjugate distribution for the Bernoulli distribution.
A closed form solution for the bayesian update, possible only for conjugate distributions!

$$(\alpha_k, \beta_k) \leftarrow \begin{cases} (\alpha_k, \beta_k) & \text{if } x_t \neq k \\ (\alpha_k, \beta_k) + (r_t, 1 - r_t) & \text{if } x_t = k. \end{cases}$$

Regret analysis

Theorem 3.

Let $(\mathcal{E}, \mathcal{B}(\mathcal{E}), Q, P)$ be a k-armed Bayesian bandit environment such that for all $\nu \in \mathcal{E}$ and $i \in [k]$, the distribution $P_{\nu i}$ is 1-subgaussian (after centering) with mean in $[0, 1]$. Then the policy π of Thompson sampling satisfies

$$BR_n(\pi, Q) \leq C\sqrt{kn \log n}$$

where $C > 0$ is a universal constant.

- ETC: $O(n^{2/3})$ regret
- $\epsilon-$greedy: $O(t^{2/3})$ regret
- UCB: $O(\sqrt{n \log n})$ regret
- Thompson sampling: $O(\sqrt{n \log n})$ regret

Interactive bandits:
`https://pavlov.tech/2019/03/02/animated-multi-armed-bandit-policies/`

- **Adversarial bandits:** the learner must choose an action a_t and observe a reward r_t. The adversary chooses the reward distribution.
- **Contextual bandits**: the learner observes a context x_t at each time step. The learner must choose an action a_t and observe a reward r_t.
- **Non-stationary bandits**: the reward distribution changes over time.

- Lattimore, T., & Szepesvári, C. (2020). Bandit Algorithms. *Cambridge University Press*. https://tor-lattimore.com/downloads/book/book.pdf
- Slivkins, A. (2019). Introduction to Multi-Armed Bandits. ArXiv. https://arxiv.org/pdf/1904.07272.pdf

- Selling horizon: $[0, T]$
- Demand: Poisson process $\lambda_t = \lambda(p(t))$.
- Feasible prices: $[\underline{p}, \bar{p}]$
- Inventory level: x

Algorithm:

- A "learning" phase (exploration) of length τ is used first, in which κ prices are tested.
- Then, a "pricing" phase (exploitation) fixes a "good" price based on demand observations in the first phase.

Omar Besbes, Assaf Zeevi, (2009) Dynamic Pricing Without Knowing the Demand Function: Risk Bounds and Near-Optimal Algorithms. *Operations Research* 57(6):1407-1420.

ALGORITHM 1. $\pi(\tau, \kappa)$

Step 1. Initialization:

(a) Set the learning interval to be $[0, \tau]$, and the number of prices to experiment with to be κ. Put $\Delta = \tau/\kappa$.

(b) Divide $[\underline{p}, \overline{p}]$ into κ equally spaced intervals and let $\{p_i, i = 1, \ldots, \kappa\}$ be the left endpoints of these intervals.

Step 2. Learning/experimentation:

(a) On the interval $[0, \tau]$ apply p_i from $t_{i-1} = (i-1)\Delta$ to $t_i = i\Delta$, $i = 1, 2, \ldots, \kappa$, as long as inventory is positive. If no more units are in stock, apply p_∞ up until time T and STOP.

(b) Compute

$$\hat{d}(p_i) = \frac{\text{total demand over } [t_{i-1}, t_i)}{\Delta}, \quad i = 1, \ldots, \kappa.$$

Step 3. Optimization:

Compute
$$\hat{p}^u = \arg\max_{1 \leqslant i \leqslant \kappa} \{p_i \hat{d}(p_i)\},$$
$$\hat{p}^c = \arg\min_{1 \leqslant i \leqslant \kappa} |\hat{d}(p_i) - x/T|, \tag{9}$$

and set $\hat{p} = \max\{\hat{p}^c, \hat{p}^u\}$. (10)

Step 4. Pricing:

- Denote n as the market size: $x_n = nx$ and $\lambda_n(\cdot) = n\lambda(\cdot)$. Then, as $n \to \infty$, the regret is asymptotically optimal.
- Set $\tau \asymp n^{-1/4}$ and $\kappa \asymp n^{1/4}$The regret bound locates in $[\frac{C'}{n^{1/2}}, \frac{C(\log n)^{1/2}}{n^{1/4}}]$

Topics

- Dynamic pricing
- Product assortment
- Advertising placement

Operational concerns:

- finite switches (limited number of prices to test))
- non-stationary reward (demand changes over time)
- switching cost (cost of switching prices)

People to follow

- Simchi Levi: http://slevi1.mit.edu/
- Assaf Zeevi: https://www0.gsb.columbia.edu/faculty/azeevi/
- Xi Chen: https://pages.stern.nyu.edu/~xchen3/
- Ningyuan Chen: http://individual.utoronto.ca/ningyuanchen/
- Yunzong Xu: https://xyz.mit.edu/research