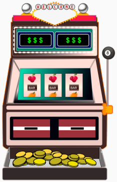
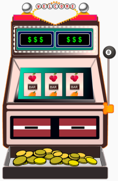
Bandit Algorithms

Flore Sentenac

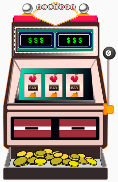
Casino $$$



mean reward 0.5



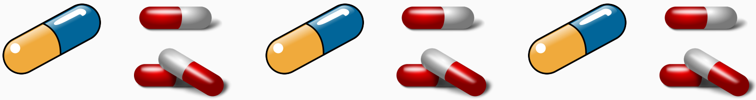
mean reward 0.6



mean reward 0.7

You will be playing a 100 times, what is a good strategy ?

Drug Clinical Trial



Success Success Failure Success Failure Success

Which pill for the next patient ?

Advertisement

Success

Success

Failure

Success

Which picture for the next user ?

Success

Success

Failure

Success

Which song for the next user ?

Problem Formulation

Setting

- K options $= K$ arms
- Each arm $k \in [K]$ has expected reward μ_k
- The values of the means are unknown to the agent
- At iteration t, agent picks arm a_t
- Receives reward $\mu_{a_t} + \epsilon_t$, with ϵ_t sub-gaussian mean 0 noise

GOAL

Maximize cumulative reward over T iterations $\sim$ minimize the regret:

$$R(T) := T \max_{k \in [K]} \mu_k - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{a_t}\right]$$

Exploration/Exploitation trade-off

At iteration t, define the agent's estimate for the average reward of arm k (e.g. empirical mean):

$$\hat{\mu}_k(t).$$

Greedy action: $a^*(t) := \arg\max_{k \in [K]} \hat{\mu}_k(t)$.

Two options:

- Pull $a^*(t) \longrightarrow$ Exploitation,
- Pull another arm $\longrightarrow$ Exploration.

Toy example

Two Bernoulli arms:

Arm A, mean reward 0.6,
Arm B, mean reward 0.5.

- Explore-only strategy: Pull two arms equally until the end,

$$R(T) = 0.1 \times \frac{T}{2}$$

Linear Regret.

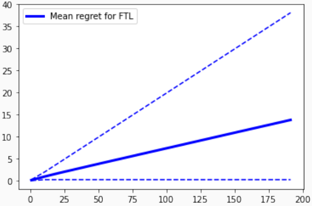
- Exploit-only strategy : Pull the arm with highest empirical mean (ties broken at random).
  With proba. 0.2, at first pull, arm A returns 0 and arm B returns 1,

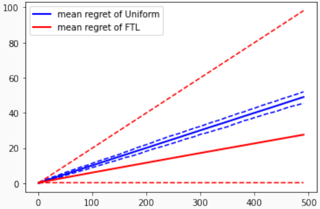
$$R(T) \geq 0.2 \times 0.1 \times T$$

Linear Regret.

Note: this strategy is also called Greedy and Follow-The-Leader (FTL)

Regret of FTL and Uniform





experiments drawn from an unnamed Colab Notebook

Baseline Strategy

Explore uniformly at random for a fixed period (mK iterations).
Then run Greedy.

This is the Explore-Then-Commit (ETC) Algorithm.

Regret Bound

If ETC interacts with a 1-subgaussian bandit and an appropriate m:

$$R(T) \leq 1 + c\sqrt{T},$$

with c a universal constant.

The regret is no longer linear !!

Actually, we cheated...

The value of m giving the previous regret depends on the value of the gap between the optimal arm and the second one.

In practice, m cannot be set to its optimal value

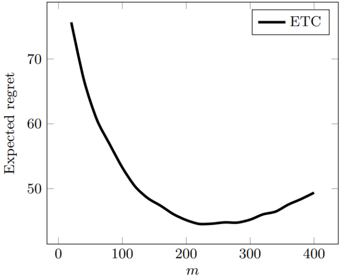


Figure 6.2 Expected regret for ETC over 10^5 trials on a Gaussian bandit with means $\mu_1 = 0, \mu_2 = -1/10$

experiment drawn from Bandit Algorithm book, Lattimore and Szepesvari

The Optimism Principle

UCB Algorithm

At iteration t, define for each arm $k \in [K]$,

- the number of times arm k has been pulled before iteration t,

$$T_k(t-1),$$

- the empirical mean of arm k,

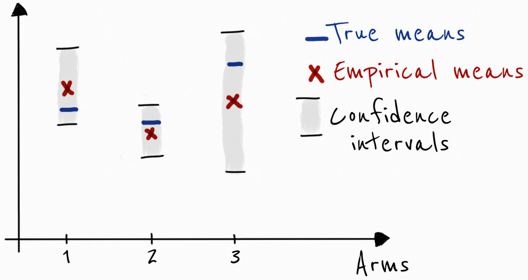
$$\hat{\mu}_k(t),$$

- a "reasonable" upper bound on the true mean of arm k,

$$U_k(t) := \hat{\mu}_k(t) + 2\sqrt{\frac{\log(T)}{T_k(t-1)}}.$$

Pull $\arg\max_{k \in [K]} U_k(t)$

UCB Algorithm



Regret bound

W.l.o.g., assume arm 1 is the optimal arm. Define:

$$\Delta_k := \mu_1 - \mu_k.$$

Theorem

For 1-subgaussian arms, UCB's regret is bounded as

$$R(T) \leq 16 \sum_{\Delta_k > 0}^{K} \frac{\log(T)}{\Delta_k} + 3 \sum_{k=1}^{K} \Delta_k.$$

Proof Sketch

We can decompose the regret:

$$R(T) = \sum_{k=1}^{K} \Delta_k \mathbb{E}\left[T_k(T)\right]$$

A sub-optimal arm k is pulled iff:

1. the upper bound on arm k is larger than the true mean of the optimal arm,
2. the upper bound on the optimal arm is smaller than its true mean.

Proof Sketch

Bad event "Some upper bound does not hold" happens with small probability.
By Hoeffding's inequality, for any t, k:

$$\mathbb{P}\left(\mu_k > U_k(t)\right) \leq \frac{1}{T^2}.$$

If for any t, k

$$\mu_k \leq U_k(t),$$

then for any sub-optimal arm $k > 1$:

$$T_k(T) \leq \frac{16 \log(T)}{\Delta_k^2}.$$

Experiments

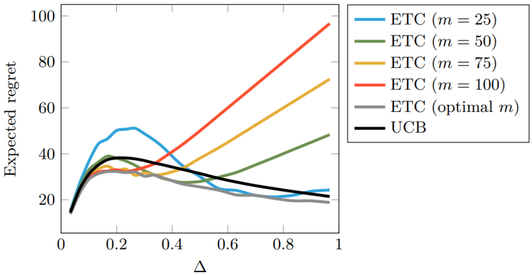


Figure 7.1 Experiment showing universality of UCB relative to fixed instances of ETC

experiment drawn from Bandit Algorithm book, Lattimore and Szepesvari

Bayesian Approach: Thompson Sampling

Algorithm

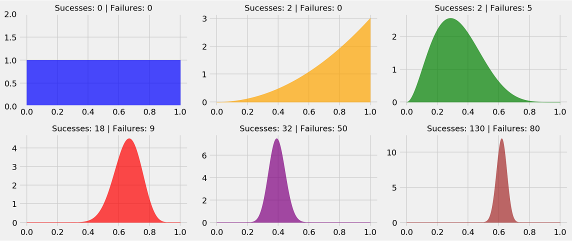
- Input a prior distribution for the arms' parameters
- At every iteration, compute posterior,
- Sample from the posterior,
- Play optimal arm according to sampled parameters.

Example with Bernoulli arms and Beta priors

- Prior Beta(α_k, β_k) for every arm.
  If $\alpha = \beta = 1$, the prior is $U([0, 1])$.
- Success of arm k: $\beta \leftarrow \beta + 1$,
- Failure of arm k: $\alpha \leftarrow \alpha + 1$,

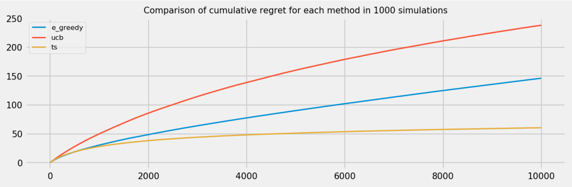
Note 1 : here, Bernoulli and Beta are conjugate of each other, hence the easy update. It is not always the case.

Note 2 : as α_k and β_k increase, the variance of Beta(α_k, β_k) decreases.



experiment drawn from https:gdmarmerola.github.iots-for-bernoulli-bandit

Comparison of the 3 strategies on the Bernoulli bandit with 4 arms



experiment drawn from https:gdmarmerola.github.iots-for-bernoulli-bandit

Beyond Vanilla Multi-Armed Bandits

Linear Contextual Bandits

Iteration
$t = 1$

context $x_{1,1}$

context $x_{2,1}$

context $x_{3,1}$

Iteration
$t = 2$

context $x_{1,2}$

context $x_{2,2}$

context $x_{3,2}$

Linear Contextual Bandits

- K arms (may be very large),
- For every arm k, at every iteration t, a d-dimensional feature vector (context) $x_{k,t}$,
- Hidden parameter θ,
- Agent picks $x_t \in \{x_{1,t}, \dots, x_{N,t}\}$, observe $r_t = x_t \cdot \theta + \epsilon_t$,
- Optimal arm depends on context: $x_t^* = \arg\max_k x_{k,t} \cdot \theta$,
- Goal : minimize regret $\sum_t (x_t^* - x_t) \cdot \theta$

Combinatorial Bandits

- K arms,
- At iteration t, agent draws an ensemble of arm, $\mathcal{A}_t \subset [K]$ subject to a combinatorial constraint

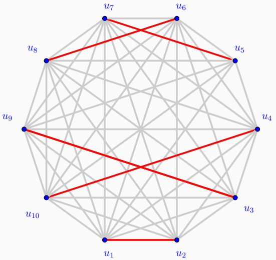
$$\mathcal{A}_t \in \mathcal{C},$$

- Receives reward:

$$\sum_{a \in \mathcal{A}_t} x_a(t),$$

- Observes $x_a(t)$ for each $a \in \mathcal{A}_t$.

Matching Bandit



- **Set of arms** : edges of the graph $(\mathcal{U}, \mathcal{E})$
- **Combinatorial constraint** : the selected arms form a matching (no vertex is selected twice)

$+$ Rank one structure:
Expected reward for arm (u_i, u_j),
$\mathbb{E}[x_{ij}(t)] = u_i u_j$

Sentenac, F., Yi, J., Calauzenes, C., Perchet, V., and Vojnovic, M. Pure Exploration and Regret Minimization in Matching Bandits. In International Conference on Machine Learning, 2021.

Multi-player Bandits

Second-by-second packet routing
Dropped packets have to be resent in next rounds

→ Learning in repeated games **with carryover**?

Flore Sentenac, Etienne Boursier, and Vianney Perchet. Decentralized learning in online queuing systems. arXiv preprint arXiv:2106.04228, 2021.

Thank you!