Explore – Then – Commit (ETC) Algorithm

Source: (Chapter 6, Bandit Algorithms).

Setting:

→ k arms, i.e., $A = \{1, 2, \ldots, k\}$.

→ $Pa_i \sim \mu_i + 1\text{-subGaussian}$, $i = 1, 2, \ldots, k$

i.e., arm i has mean μ_i, and variance '1', with a subGaussian dist.

→ $\mu_1 > \mu_2 \geq \mu_3 \ldots \geq \mu_k$

↑ strict gap. $(\mu_1 - \mu_2) = \Delta > 0$

→ Unstructured environment E.

(Dependence on variance of arms is hidden, and by normalizing to '1', we are implicitly assuming that arm variance is known)

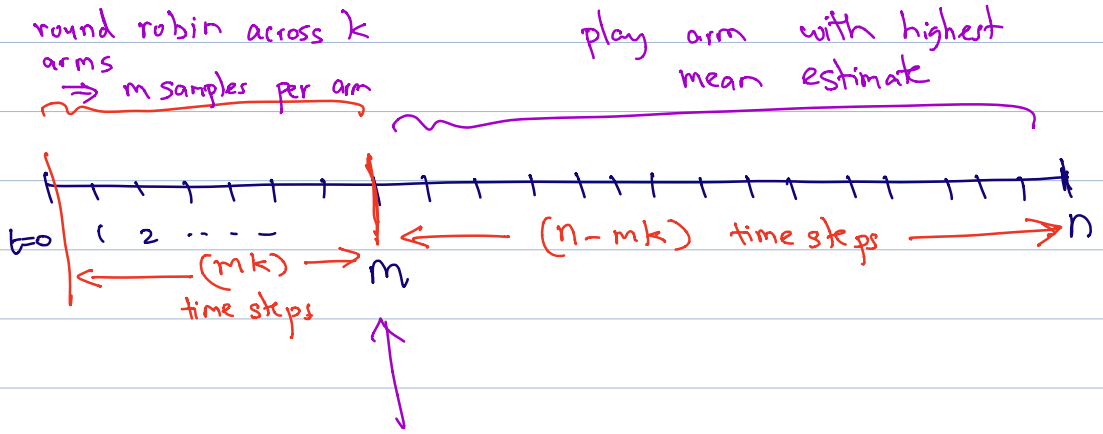
Two parameters that are explicitly considered:

(a) Δ_i = sub-optimality gap between arm i and arm i = $(\mu_1 - \mu_i)$.

(Note: $\Delta_1 = 0$, $\Delta_2 = \Delta$)

(b) n = time horizon (number of time steps).

Explore-Then-Commit (ETC) Algorithm



Algorithm Input: m

(1.) Over first $k \cdot m$ time steps, play the

k arms in round-robin fashion, i.e.,

$$A_t = (t \bmod k) + 1 \quad , \quad t = 1, 2, \ldots, mk$$

(2) At time mk, compute:

$$\hat{\mu}_i(mk) = \frac{1}{m} \sum_{s=1}^{mk} X_s \mathbb{1}_{\{A_s = i\}}$$

$i = 1, 2, \ldots, k$ (compute empirical mean for each arm)

(3) $$A_t = \underset{1 \le i \le k}{\operatorname{argmax}} \hat{\mu}_i(mk) \quad , \quad t \ge mk + 1$$

(tie breaking arbitrary)

Note:

(1) The only input apparently is m. However, for any guarantees on regret

$$m = m(\Delta, n)$$

n → time horizon

Δ → sub optimality gap between arms 1 and 2.

Specifically, for $k=2$

$$m(\Delta, n) = \max\left\{1, \left\lceil \frac{4}{\Delta^2} \ln\left(\frac{n\Delta^2}{4}\right)\right\rceil\right\}$$

Henceforth, $R_n(\pi, \nu) = R_n$ (notation).

Theorem: With the setup as above, we have

(a) For any $m \in \{1, 2, \ldots, \frac{n}{k}\}$,

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

(b) For $m = \max\left\{1, \left\lceil \frac{4}{\Delta^2} \ln\left(\frac{n\Delta^2}{4}\right)\right\rceil\right\}$, $k=2$

$$R_n \leq \min\left\{n\Delta, \Delta + \frac{4}{\Delta}\left(1 + \max\left\{0, \ln\left(\frac{n\Delta^2}{4}\right)\right\}\right)\right\}$$

Notes: (a) For large n, $R_n \leq \frac{4}{\Delta}\ln(n) + C$

i.e., the regret scales logarithmly in time (n), and inversely with gap.

Questions: ① $\ln(n)$ scaling $\longrightarrow$ can we do better?

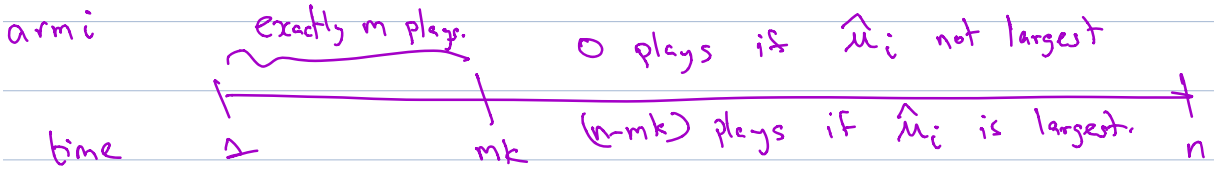
② $\frac{1}{\Delta}$-scaling $\longrightarrow$ can we do better?

ⓑ Algorithm explicitly needs Δ, n for this regret scaling.

Proof: Recall from Regret Decomposition Lemma (note set 3, Chap. 4 in book),

$$R_n = \sum_{i=1}^{k} \Delta_i \, E[T_i(n)],$$

where $$T_i(n) = \sum_{s=1}^{n} \mathbb{1}\{A_s = i\}.$$



$$T_i(n) = m + (n-mk)\,\chi\left\{\hat{\mu}_i \geq \max_{j\neq i} \hat{\mu}_j, \text{ and the breaking chooses } i\right\}$$

$$\leq m + (n-mk)\,\chi\left\{\hat{\mu}_i \geq \max_{j\neq i} \hat{\mu}_j\right\}.$$

where $\hat{\mu}_i \equiv \hat{\mu}_i(mk) \quad \forall\ i = 1, 2, \ldots k$

$$\therefore E[T_i(n)] \leq m + (n-mk)\,P\left(\hat{\mu}_i \geq \max_{j\neq i} \hat{\mu}_j\right)$$

$$\leq m + (n-mk)\,P\left(\hat{\mu}_i \geq \hat{\mu}_1\right)$$

↑ Comparing only to best arm.

$$= m + (n-mk)\,P\left(\hat{\mu}_i - \mu_i + \mu_i \geq \hat{\mu}_1 - \mu_1 + \mu_1\right)$$

$$= m + (n-mk)\,P\left((\hat{\mu}_i - \mu_i) - (\hat{\mu}_1 - \mu_1) \geq \mu_1 - \mu_i\right)$$

$$= m + (n-mk)\, P\left((\hat{\mu}_i - \mu_i) - (\hat{\mu}_1 - \mu_1) \geq \Delta_i \right)$$

$\frac{1}{\sqrt{m}}$ – subGaussian with zero mean

$\hat{\mu}_i$ is the average over exactly m samples of Δ – subGaussian r.v.s.

$= \frac{1}{\sqrt{m}}$ – subGaussian.

further $(\hat{\mu}_i - \mu_i) - (\hat{\mu}_1 - \mu_1)$

indep. $\frac{1}{\sqrt{m}}$ – subGaussian

$\Rightarrow$ difference is $\sqrt{\frac{2}{m}}$ – subGaussian.

$$\therefore P\left((\hat{\mu}_i - \mu_i) - (\hat{\mu}_1 - \mu_1) \geq \Delta_i \right) \leq e^{\frac{-\Delta_i^2}{2(\sqrt{2/m})^2}}$$

$$= e^{-m\Delta_i^2/4}$$

$$\Rightarrow E[T_i(m)] \leq m + (n-mk)\, e^{-m\Delta_i^2/4}$$

$$\Rightarrow R_n = \sum_{i=1}^{k} \Delta_i E[T_i(n)]$$

$$\leq m \sum_{i=1}^{k} \Delta_i + (n-mk) \sum_{i=1}^{k} \Delta_i e^{-m\Delta_i^2/4}$$

Now, for $k=2$, choose $m = \max\left\{1, \left\lceil \frac{4}{\Delta^2} \ln\left(\frac{n\Delta^2}{4}\right) \right\rceil\right\}$

Substitute in above to get:

$$R_n \leq \min\left\{ n\Delta, \Delta + \frac{4}{\Delta}\left(1 + \max\left\{0, \ln \frac{n\Delta^2}{4}\right\}\right)\right\}$$

Potential Concerns: To get $\ln(n)$ regret scaling, we need knowledge of time horizon n, and gap Δ.

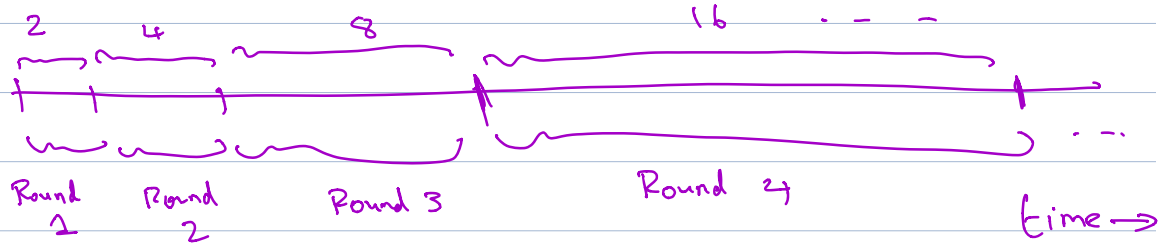
Getting around knowledge of horizon n:

Develop an "outer" algorithm that plays ETC with different parameters over time.

Doubling Trick

Refs: * Problem 6.6 in Bandit Algo. book

* Online learning & Online convex optimization, Shai Shalev-Shwartz, Found. and Trends in ML, Vol 4, No 2, 2011 (see 2.3.1).



→ Round i : 2^i time-steps, $i = 1, 2, \ldots$

(Doubling Trick + ETC) Algorithm :

* Restart ETC algorithm at the begining of each round
* In round i, use $m = m(\Delta, 2^i)$ ("local time horizon")

For notation simplicity, assume $n = 2^r - 2$ for some r, and $k = 2$

↳ # of arms.

Then, we have $(r-1)$ rounds, where round i has 2^i time steps, $i = 1, 2, \dots, (r-1)$

$\therefore$

$R_n =$ (Regret over round 1) + (Regret over round 2) + ...

Importantly, in round i, we are playing a local ETC with parameter $m = m(\Delta, 2^i)$, and whose regret over 2^i time steps is bounded by: $C + \frac{4}{\Delta} \ln(2^i)$

$\therefore$

$$R_n \leq \sum_{i=1}^{r-1} \left(C + \frac{4}{\Delta} \ln(2^i) \right), \text{ where}$$

$$r = \log_2 n$$

$$= \frac{\ln(n)}{\ln(2)}$$

$$\Rightarrow \quad R_n \leq C_1 \ln(n) + \frac{4}{\Delta} \ln(2) \cdot \underbrace{\left(\sum_{i=1}^{r-1} i \right)}_{\leq r^2/2}$$

$$\boxed{R_n \leq C_1 \ln(n) + \frac{4}{\Delta \ln(2)} \cdot (\ln(n))^2}$$

Note: ① This algorithm needs Δ, but does not need n!

② What about other choices of round lengths? How much can we improve; what is the best possible?

Reading HW: What doubling tricks can and can't do for multi-armed bandits, L. Besson and E. Kaufmann, 2018. (on Canvas)

Two other important algorithms: (Proof of regret are HW problems).

ε-Greedy Algorithm

Setting: * k arms, unstructured environment

* arm i: $P_{a_i} \sim \mu_i + \Delta$-subGaussian.

$$\mu_1 > \mu_2 \geq \mu_3 \cdots \geq \mu_k.$$

Algorithm:

Initialize: Play each arm once in round-robin.

For any $t \geq k$,

$$\underbrace{\hat{\mu}_j(t)}_{\text{empirical mean estimate for arm } j} = \frac{1}{T_j(t)} \sum_{s=1}^{t} X_s \cdot \chi_{\{A_s = j\}}$$

At each time $t \geq (k+1)$:

- With prob. ε_t (indep. of any other event), play arms uniformly at random.

Generate Bernoulli $Z_t \in \{0,1\}$ with $P(Z_t = 1) = \varepsilon_t$

If $Z_t = 1$, Pick arm j w.p. $\frac{1}{k}$,

$j = 1, 2, \cdots, k.$

Play arm k

- With prob. $(1-\varepsilon_t)$, play $A_t = \underset{j}{\operatorname{argmax}}\, \hat{\mu}_j(t-1)$.

arm with best empirical mean.

Choice of ε_t :

$$\varepsilon_t = \min\left\{1, \frac{Ck}{t\,\Delta_{min}^2}\right\}, \quad C > 0 \text{ large enough}$$

Result:

$$R_n \leq C' \sum_{i=1}^{k} \left(\Delta_i + \frac{\Delta_i}{\Delta_{min}^2} \ln \max\left\{ e, \frac{n \Delta_{min}^2}{k} \right\} \right)$$

where $\Delta_{min} = \Delta_2$

Thus, for large n, $k=2$,

$$\boxed{R_n \leq C + \frac{1}{\Delta} \ln(n)}$$

Note: * This algorithm requires knowledge of Δ_{min}, but does not need time horizon n.

* Regret scales as $\left(\frac{1}{\Delta}\right)$, but with knowledge of this parameter.

HW: Analyze the ε-Greedy algo; Problem 6.7

The Elimination Algorithm

(Problem 6.8)

Originally called the improved UCB algorithm—
[1]: UCB Revisted: Improved Regret Bounds for the Stochastic Multi-armed Bandit Problem, P. Auer and R. Ortner, 2010.

Setting: Given knowledge of n, but no knowledge of $\{\Delta_i\}_{i=1}^{k}$.

↓ horizon is known.

The idea:

Initial: Play all arms in round-robin a fixed number of times.

(a) Estimate empirical mean of these arms.

(b) Use a "hypothesis test" to eliminate some of the arms (if their empirical means are "much smaller" than "good" arms).

— With remaining arms, play all arms in round-robin for a fixed number of times

Repeat (a), (b) above.

The algorithm: (Taken from pp. (5) in [1]).

Arms $A = \{1, 2, \cdots, k\}$. , Same setting as before.

Initialize: $\tilde{\Delta}_0 = 1$ $A_0 = \{1, 2, \cdots, k\}$

Repeat following loop as long as $|A_r| > 1$. Once we have $|A_r| = 1$, play remaining arm until time n.

$r = 0, 1, 2, \cdots$

(a) In round r: Play each arm in A_r

$$m_r = \left\lceil \frac{2 \ln(n \tilde{\Delta}_r^2)}{\tilde{\Delta}_r^2} \right\rceil$$

(b) $\hat{\mu}_{ir}$ = estimate of arm i empirical reward using only the samples from round r.

→ estimate is created for each $i \in A_r$ using m_r samples for each arm.

(c) Elimination step:

For each $i \in A_r$, that satisfies:

$$\left(\hat{\mu}_{ir} + \sqrt{\frac{\log(n\tilde{\Delta}_r^2)}{2m_r}}\right)$$

$$< \max_{\substack{j \neq i, \\ j \in A_r}} \underbrace{\left(\hat{\mu}_{jr} - \sqrt{\frac{\log(n\tilde{\Delta}_r^2)}{2m_r}}\right)}_{\text{down rated mean of best arm in this round } r}$$

$$A_{r+1} = A_r \setminus \{i\}.$$

(d) Update $\tilde{\Delta}_{r+1} = \frac{\tilde{\Delta}_r}{2}$.

Note: * This algorithm always uses a deterministic number of samples in any round to form estimates.

* The adaptation here is to dynamically eliminate a set of arms successively.

Result: $R_n \leq C_1 + \frac{C_2}{\Delta} \ln(n)$.

(Read [1] for precise result, and see HW).
