CS 747, Autumn 2020: Week 2, Lecture 1

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Autumn 2020

Multi-armed Bandits

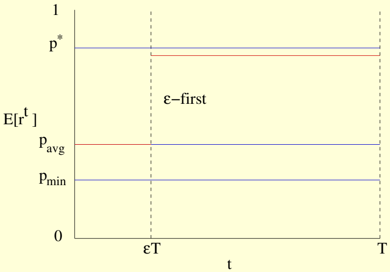
1. Achieving sub-linear regret
2. A lower bound on regret
3. UCB, KL-UCB algorithms
4. Thompson Sampling algorithm
5. Summary and outlook

Multi-armed Bandits

1. Achieving sub-linear regret
2. A lower bound on regret
3. UCB, KL-UCB algorithms
4. Thompson Sampling algorithm
5. Summary and outlook

Review of ϵG1, ϵG2

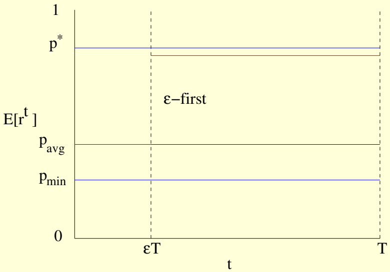
- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.



Review of ϵG1, ϵG2

- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.

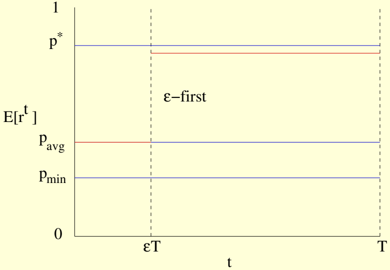
- What would happen if we ran for horizon $2T$ instead of T?



Review of ϵG1, ϵG2

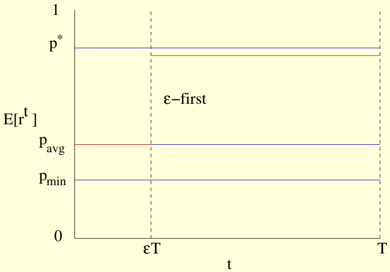
- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.

- What would happen if we ran for horizon $2T$ instead of T? Exploratory phase would last $2\epsilon T$ steps!



Review of ϵG1, ϵG2

- ϵ-first: Explore (uniformly) for ϵT pulls; then exploit.
- What would happen if we ran for horizon $2T$ instead of T? Exploratory phase would last $2\epsilon T$ steps!



$$R_T = Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t] = Tp^\star - \sum_{t=0}^{\epsilon T-1} \mathbb{E}[r^t] - \sum_{t=\epsilon T}^{T-1} \mathbb{E}[r^t]$$

$$= Tp^\star - \epsilon Tp_{\text{avg}} - \sum_{t=\epsilon T}^{T-1} \mathbb{E}[r^t] \geq Tp^\star - \epsilon Tp_{\text{avg}} - (T - \epsilon T)p^\star$$

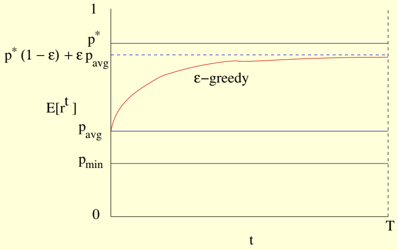
$$= \epsilon(p^\star - p_{\text{avg}})T = \Omega(T).$$

Review of ϵG3

- ϵ-greedy: On each step
  explore (uniformly) w.p. ϵ,
  exploit w.p. $1 - \epsilon$.

Review of ϵG3

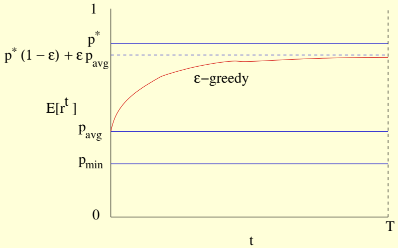
- ϵ-greedy: On each step explore (uniformly) w.p. ϵ, exploit w.p. $1 - \epsilon$.



Review of ϵG3

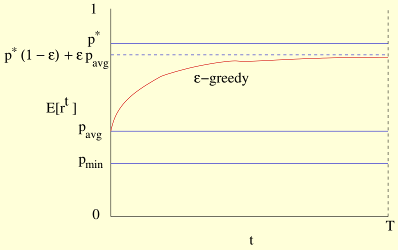
- ϵ-greedy: On each step explore (uniformly) w.p. ϵ, exploit w.p. $1 - \epsilon$.

- $\mathbb{E}[r^t]$ can never exceed $p^\star(1 - \epsilon) + \epsilon p_{\text{avg}}$!



Review of ϵG3

- ϵ-greedy: On each step explore (uniformly) w.p. ϵ, exploit w.p. $1-\epsilon$.



- $\mathbb{E}[r^t]$ can never exceed $p^\star(1-\epsilon)+\epsilon p_{\text{avg}}$!

$$
\begin{aligned}
R_T &= Tp^\star - \sum_{t=0}^{T-1} \mathbb{E}[r^t] \\
&\geq Tp^\star - \sum_{t=0}^{T-1} ((\epsilon)p_{\text{avg}} + (1-\epsilon)p^\star) \\
&= \epsilon(p^\star - p_{\text{avg}})T = \Omega(T).
\end{aligned}
$$

How to achieve Sub-linear Regret?

- Two conditions must be met: C1 and C2.

How to achieve Sub-linear Regret?

- Two conditions must be met: C1 and C2.

C1. **Infinite exploration.** In the limit ($T \to \infty$), each arm must be pulled an infinite number of times.

How to achieve Sub-linear Regret?

- Two conditions must be met: C1 and C2.

C1. **Infinite exploration.** In the limit ($T \to \infty$), each arm must be pulled an infinite number of times.

- On the contrary, suppose we start exploiting after pulling each arm a finite U times.
- With probability $(1 - p^{\star})^{U} > 0$, an optimal arm will have empirical mean 0.
- A non-optimal arm may thereafter be "exploited" for ever, giving linear regret.

How to achieve Sub-linear Regret?

C2. **Greed in the Limit**. Let *exploit*(T) denote the number of pulls that are greedy w.r.t. the empirical mean up to horizon T. For sub-linear regret, we need

$$\lim_{T \to \infty} \frac{\mathbb{E}[exploit(T)]}{T} = 1.$$

How to achieve Sub-linear Regret?

C2. **Greed in the Limit**. Let *exploit*(T) denote the number of pulls that are greedy w.r.t. the empirical mean up to horizon T. For sub-linear regret, we need

$$\lim_{T\to\infty} \frac{\mathbb{E}[exploit(T)]}{T} = 1.$$

- Let $\bar{\mathcal{I}}$ be the set of all bandit instances with reward means strictly less than 1.
- **Result.** An algorithm L achieves sub-linear regret on all instances $I \in \bar{\mathcal{I}}$ if and only if it satisfies C1 and C2 on all $I \in \bar{\mathcal{I}}$.

How to achieve Sub-linear Regret?

C2. **Greed in the Limit**. Let $exploit(T)$ denote the number of pulls that are greedy w.r.t. the empirical mean up to horizon T. For sub-linear regret, we need

$$\lim_{T \to \infty} \frac{\mathbb{E}[exploit(T)]}{T} = 1.$$

- Let $\bar{\mathcal{I}}$ be the set of all bandit instances with reward means strictly less than 1.
- **Result.** An algorithm L achieves sub-linear regret on all instances $I \in \bar{\mathcal{I}}$ if and only if it satisfies C1 and C2 on all $I \in \bar{\mathcal{I}}$. In short: "GLIE" $\iff$ sub-linear regret.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.
  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.
  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

  C1 satisfied since each arm pulled at least $\frac{1}{n}\sqrt{T}$ times with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq \frac{T-\sqrt{T}}{T}$.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.
  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

  C1 satisfied since each arm pulled at least $\frac{1}{n}\sqrt{T}$ times with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq \frac{T-\sqrt{T}}{T}$.

- ϵ_t-greedy with $\epsilon_t = \frac{1}{t+1}$.
  On the t-th step, explore w.p. ϵ_t, exploit w.p. $1 - \epsilon_t$.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.
  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

  C1 satisfied since each arm pulled at least $\frac{1}{n}\sqrt{T}$ times with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq \frac{T-\sqrt{T}}{T}$.

- ϵ_t-greedy with $\epsilon_t = \frac{1}{t+1}$.
  On the t-th step, explore w.p. ϵ_t, exploit w.p. $1 - \epsilon_t$.

  C1 satisfied: each arm assured $\sum_{t=0}^{T-1} \frac{1}{n(t+1)} = \theta(\frac{\log T}{n})$ pulls with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq \frac{T-\theta(\log T)}{T}$.

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.
  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

  C1 satisfied since each arm pulled at least $\frac{1}{n}\sqrt{T}$ times with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq \frac{T - \sqrt{T}}{T}$.

- ϵ_t-greedy with $\epsilon_t = \frac{1}{t+1}$.
  On the t-th step, explore w.p. ϵ_t, exploit w.p. $1 - \epsilon_t$.

  C1 satisfied: each arm assured $\sum_{t=0}^{T-1} \frac{1}{n(t+1)} = \theta(\frac{\log T}{n})$ pulls with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq \frac{T - \theta(\log T)}{T}$.
  What happened when we took $\epsilon_t = \epsilon$?

GLIE-ifying ϵ-Greedy Strategies

- ϵ_T-first with $\epsilon_T = \frac{1}{\sqrt{T}}$.
  Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls. Thereafter exploit.

  C1 satisfied since each arm pulled at least $\frac{1}{n}\sqrt{T}$ times with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq \frac{T-\sqrt{T}}{T}$.

- ϵ_t-greedy with $\epsilon_t = \frac{1}{t+1}$.
  On the t-th step, explore w.p. ϵ_t, exploit w.p. $1 - \epsilon_t$.

  C1 satisfied: each arm assured $\sum_{t=0}^{T-1} \frac{1}{n(t+1)} = \theta(\frac{\log T}{n})$ pulls with high probability.
  C2 satisfied since $\mathbb{E}[exploit(T)] \geq \frac{T-\theta(\log T)}{T}$.
  What happened when we took $\epsilon_t = \epsilon$? What will happen by taking $\epsilon_t = \frac{1}{(t+1)^2}$?

Multi-armed Bandits

1. Achieving sub-linear regret
2. A lower bound on regret
3. UCB, KL-UCB algorithms
4. Thompson Sampling algorithm
5. Summary and outlook

A Lower Bound on Regret

- What's the least regret you can get?

A Lower Bound on Regret

- What's the least regret you can get?
  An algorithm that always pulls arm 3 will get zero regret on some instances...

A Lower Bound on Regret

- What's the least regret you can get?
  An algorithm that always pulls arm 3 will get zero regret on some instances. . . but linear regret on other instances!

A Lower Bound on Regret

- What's the least regret you can get?
  An algorithm that always pulls arm 3 will get zero regret on some instances... but linear regret on other instances!
- Paraphrasing Lai and Robbins (1985; see Theorem 2).

> Let L be an algorithm such that for every bandit instance $I \in \bar{\mathcal{I}}$ and for every $\alpha > 0$, as $T \to \infty$:
> $$R_T(L, I) = o(T^\alpha).$$

A Lower Bound on Regret

- What's the least regret you can get?
  An algorithm that always pulls arm 3 will get zero regret on some instances... but linear regret on other instances!
- Paraphrasing Lai and Robbins (1985; see Theorem 2).

Let L be an algorithm such that for every bandit instance $I \in \bar{\mathcal{I}}$ and for every $\alpha > 0$, as $T \to \infty$:

$$R_T(L, I) = o(T^\alpha).$$

Then, for every bandit instance $I \in \bar{\mathcal{I}}$, as $T \to \infty$:

$$\frac{R_T(L, I)}{\ln(T)} \geq \sum_{a: p_a(I) \neq p^\star(I)} \frac{p^*(I) - p_a(I)}{KL(p_a(I), p^\star(I))},$$

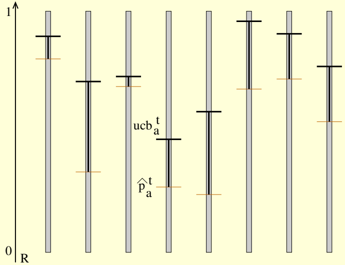
where for $x, y \in [0, 1)$, $KL(x, y) \stackrel{\text{def}}{=} x \ln \frac{x}{y} + (1 - x) \ln \frac{1-x}{1-y}$.

Multi-armed Bandits

1. Achieving sub-linear regret
2. A lower bound on regret
3. UCB, KL-UCB algorithms
4. Thompson Sampling algorithm
5. Summary and outlook

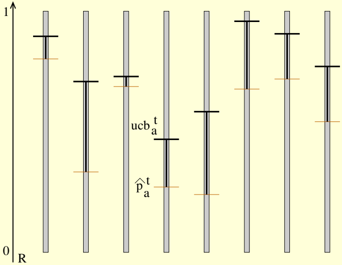
Upper Confidence Bounds

- UCB (Auer et al., 2002)
  - At time t, for every arm a, define $\text{ucb}_a^t = \hat{p}_a^t + \sqrt{\frac{2\ln(t)}{u_a^t}}$.
  - $\hat{p}_a^t$ is the empirical mean of rewards from arm a.
  - u_a^t the number of times a has been sampled at time t.



Upper Confidence Bounds

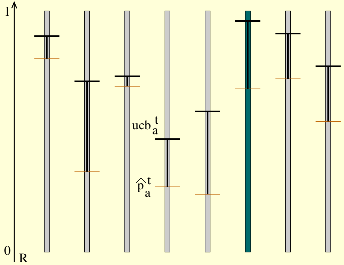
- UCB (Auer et al., 2002)
  - At time t, for every arm a, define $\text{ucb}_a^t = \hat{p}_a^t + \sqrt{\frac{2\ln(t)}{u_a^t}}$.
  - $\hat{p}_a^t$ is the empirical mean of rewards from arm a.
  - u_a^t the number of times a has been sampled at time t.



  - Sample an arm a for which ucb_a^t is maximal.

Upper Confidence Bounds

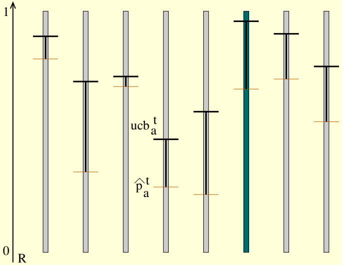
- UCB (Auer et al., 2002)
  - At time t, for every arm a, define $\text{ucb}_a^t = \hat{p}_a^t + \sqrt{\frac{2\ln(t)}{u_a^t}}$.
  - $\hat{p}_a^t$ is the empirical mean of rewards from arm a.
  - u_a^t the number of times a has been sampled at time t.



  - Sample an arm a for which ucb_a^t is maximal.

Upper Confidence Bounds

- UCB (Auer et al., 2002)
  - At time t, for every arm a, define $\text{ucb}_a^t = \hat{p}_a^t + \sqrt{\frac{2\ln(t)}{u_a^t}}$.
  - $\hat{p}_a^t$ is the empirical mean of rewards from arm a.
  - u_a^t the number of times a has been sampled at time t.



  - Sample an arm a for which ucb_a^t is maximal.
- Achieves regret of $O(\log(T))$: optimal dependence on T.

KL-UCB (Garivier and Cappé, 2011)

- Identical to UCB algorithm on previous slide, except for a different definition of the upper confidence bound.

  ucb-kl$_a^t = \max\{q \in [\hat{p}_a^t, 1]$ such that
  $u_a^t KL(\hat{p}_a^t, q) \leq \ln(t) + c \ln(\ln(t))\}$, where $c \geq 3$.

  KL-UCB algorithm: at step t, pull argmax_a ucb-kl$_a^t$.

KL-UCB (Garivier and Cappé, 2011)

- Identical to UCB algorithm on previous slide, except for a different definition of the upper confidence bound.

  $\text{ucb-kl}_a^t = \max\{q \in [\hat{p}_a^t, 1]$ such that
  $u_a^t KL(\hat{p}_a^t, q) \leq \ln(t) + c\ln(\ln(t))\}$, where $c \geq 3$.

  KL-UCB algorithm: at step t, pull $\operatorname{argmax}_a \text{ucb-kl}_a^t$.

- Observe that $KL(\hat{p}_a^t, q)$ monotonically increases with q, and
  - $KL(\hat{p}_a^t, \hat{p}_a^t) = 0$;
  - $KL(\hat{p}_a^t, 1) = \infty$.

  Easy to compute ucb-kl_a^t numerically (for example through binary search).

KL-UCB (Garivier and Cappé, 2011)

- Identical to UCB algorithm on previous slide, except for a different definition of the upper confidence bound.

  $\text{ucb-kl}_a^t = \max\{q \in [\hat{p}_a^t, 1] \text{ such that}$
  $\quad u_a^t KL(\hat{p}_a^t, q) \leq \ln(t) + c \ln(\ln(t))\}$, where $c \geq 3$.

  KL-UCB algorithm: at step t, pull $\operatorname{argmax}_a \text{ucb-kl}_a^t$.

- Observe that $KL(\hat{p}_a^t, q)$ monotonically increases with q, and
  - $KL(\hat{p}_a^t, \hat{p}_a^t) = 0$;
  - $KL(\hat{p}_a^t, 1) = \infty$.

  Easy to compute ucb-kl_a^t numerically (for example through binary search).

- ucb-kl_a^t is a tighter confidence bound than ucb_a^t.

KL-UCB (Garivier and Cappé, 2011)

- Identical to UCB algorithm on previous slide, except for a different definition of the upper confidence bound.

  $$\text{ucb-kl}_a^t = \max\{q \in [\hat{p}_a^t, 1] \text{ such that } u_a^t KL(\hat{p}_a^t, q) \leq \ln(t) + c\ln(\ln(t))\}, \text{ where } c \geq 3.$$

  KL-UCB algorithm: at step t, pull $\text{argmax}_a\, \text{ucb-kl}_a^t$.

- Observe that $KL(\hat{p}_a^t, q)$ monotonically increases with q, and
  - $KL(\hat{p}_a^t, \hat{p}_a^t) = 0$;
  - $KL(\hat{p}_a^t, 1) = \infty$.

  Easy to compute ucb-kl_a^t numerically (for example through binary search).

- ucb-kl_a^t is a tighter confidence bound than ucb_a^t.
  Regret of KL-UCB asymptotically matches Lai and Robbins' lower bound!

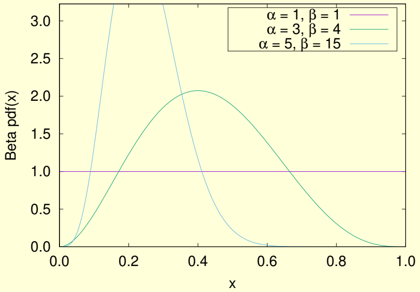
Multi-armed Bandits

1. Achieving sub-linear regret
2. A lower bound on regret
3. UCB, KL-UCB algorithms
4. Thompson Sampling algorithm
5. Summary and outlook

Before Moving on . . . The Beta Distribution

- Beta(α, β) defined on $[0, 1]$.
  Two parameters: α and β.

$$\text{Mean} = \frac{\alpha}{\alpha + \beta}; \quad \text{Variance} = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}.$$

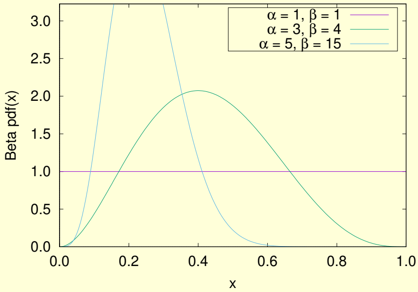


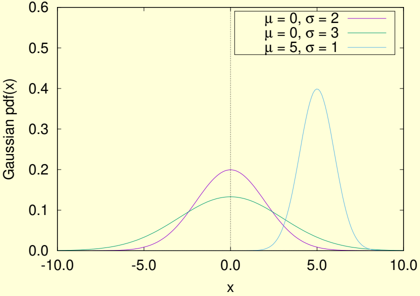
Plots obtained by adapting gnuplot script `http://gnuplot.sourceforge.net/demo/prob.5.gnu`.

Before Moving on . . . The Beta Distribution

- Beta(α, β) defined on $[0, 1]$.
  Two parameters: α and β.

$$\text{Mean} = \frac{\alpha}{\alpha + \beta}; \quad \text{Variance} = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}.$$





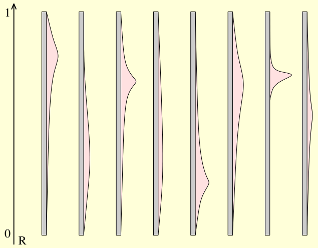
Plots obtained by adapting gnuplot script `http://gnuplot.sourceforge.net/demo/prob.5.gnu`.

Thompson Sampling (Thompson, 1933)

- Algorithm
  - At time t, let arm *a* have s_a^t successes (ones/heads) and f_a^t failures (zeroes/tails).

Thompson Sampling (Thompson, 1933)

- Algorithm
  - At time t, let arm a have s_a^t successes (ones/heads) and f_a^t failures (zeroes/tails).
  - $Beta(s_a^t + 1, f_a^t + 1)$ represents a "belief" about the true mean of arm a.
  - Mean = $\frac{s_a^t+1}{s_a^t+f_a^t+2}$; variance = $\frac{(s_a^t+1)(f_a^t+1)}{(s_a^t+f_a^t+2)^2(s_a^t+f_a^t+3)}$.

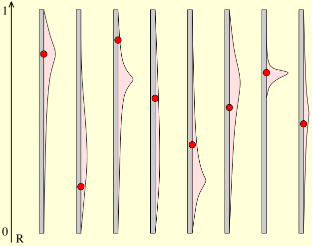


Thompson Sampling (Thompson, 1933)

- Algorithm
  - At time t, let arm a have s_a^t successes (ones/heads) and f_a^t failures (zeroes/tails).
  - $Beta(s_a^t + 1, f_a^t + 1)$ represents a "belief" about the true mean of arm a.
  - Mean = $\frac{s_a^t+1}{s_a^t+f_a^t+2}$; variance = $\frac{(s_a^t+1)(f_a^t+1)}{(s_a^t+f_a^t+2)^2(s_a^t+f_a^t+3)}$.

- Computational step: For every arm a, draw a sample $x_a^t \sim Beta(s_a^t + 1, f_a^t + 1)$.

- Sampling step: Pull arm a for which x_a^t is maximal.

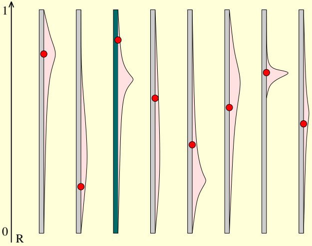


Thompson Sampling (Thompson, 1933)

- Algorithm
  - At time t, let arm a have s_a^t successes (ones/heads) and f_a^t failures (zeroes/tails).
  - $Beta(s_a^t + 1, f_a^t + 1)$ represents a “belief” about the true mean of arm a.
  - Mean = $\frac{s_a^t+1}{s_a^t+f_a^t+2}$; variance = $\frac{(s_a^t+1)(f_a^t+1)}{(s_a^t+f_a^t+2)^2(s_a^t+f_a^t+3)}$.

- Computational step: For every arm a, draw a sample $x_a^t \sim Beta(s_a^t + 1, f_a^t + 1)$.

- Sampling step: Pull arm a for which x_a^t is maximal.

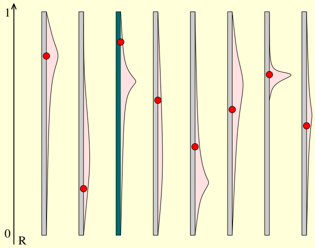


Thompson Sampling (Thompson, 1933)

- Algorithm
  - At time t, let arm a have s_a^t successes (ones/heads) and f_a^t failures (zeroes/tails).
  - $Beta(s_a^t + 1, f_a^t + 1)$ represents a “belief” about the true mean of arm a.
  - Mean = $\frac{s_a^t+1}{s_a^t+f_a^t+2}$; variance = $\frac{(s_a^t+1)(f_a^t+1)}{(s_a^t+f_a^t+2)^2(s_a^t+f_a^t+3)}$.

- Computational step: For every arm a, draw a sample $x_a^t \sim Beta(s_a^t + 1, f_a^t + 1)$.

- Sampling step: Pull arm a for which x_a^t is maximal.



- Achieves optimal regret (Kaufmann et al., 2012); is excellent in practice (Chapelle and Li, 2011).

Multi-armed Bandits

1. Achieving sub-linear regret
2. A lower bound on regret
3. UCB, KL-UCB algorithms
4. Thompson Sampling algorithm
5. Summary and outlook

Summary

- We desire low, sub-linear regret on all bandit instances.
- Possible if and only if algorithm satisfies GLIE conditions.
- If an algorithm gives sub-polynomial regret on all instances, it must give super-logarithmic regret on all instances (Lai and Robbins, 1985).

- UCB algorithm achieves logarithmic dependence on T.
- KL-UCB also improves the accompanying constant, thereby matching the lower bound (asymptotically).
- Thompson Sampling, a qualitatively different randomised algorithm, also matches regret lower bound.
- UCB, KL-UCB, Thompson Sampling all examples of optimism in the face of uncertainty principle.

- Next week: concentration inequalities, analysis of UCB, KL-UCB, Thompson Sampling, other bandit formulations.