Multi Armed Bandits

Consider a biased coin which gives heads with probability P and tails with probability (1-P) but we dont know P.

There can be many such coins with different probabilities of heads P_1 P_2 P_3 (all unknown)

We want to toss coins N times while maximizing heads. We can choose any coin.

Eventually through experimentation we will find the coin with best P and exploit it
But while experimentation, you also have to maximize the reward. No seperate time for experimentation

Such systems are refered as multi armed bandits

This has applications in →

1. router optimization
2. Clinical trials
3. Game playing
4. Online advertising (A|B Testing)
5. Recommendation system

} part of reinforcement learning

In multi armed bandits, you dont have any training period. You have to learn while in action

Multi armed bandits have binary 0 or 1 reward ε- greedy explore first strategy

for T runs,

Explore few times first

Exploit thereafter

a) If $t < \varepsilon \cdot T$ (Exploration)

- Sample at random

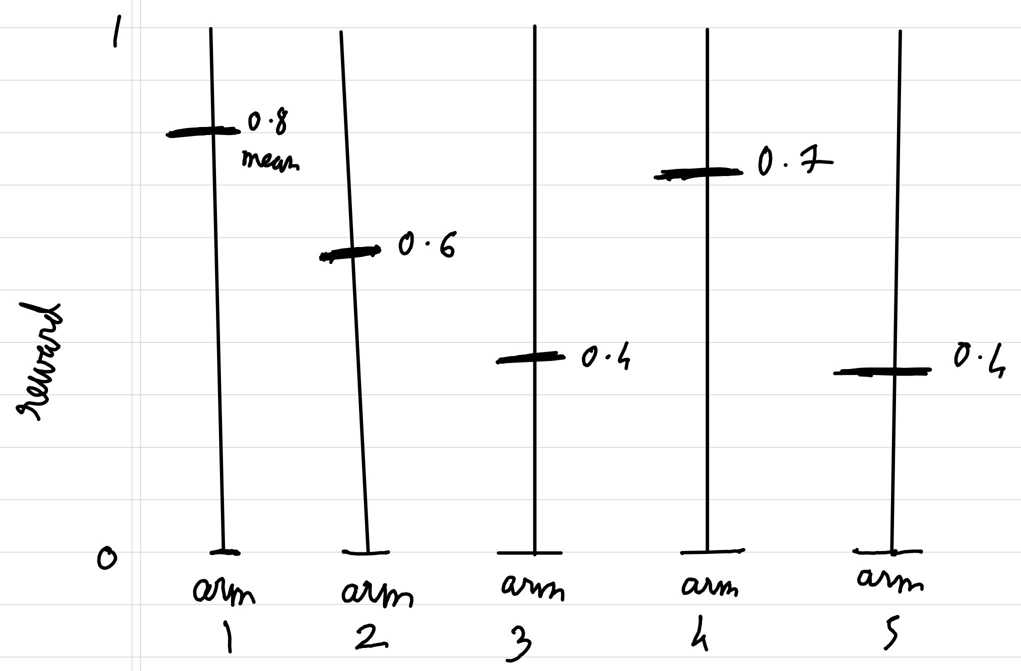
b) at $t = \lfloor \varepsilon \cdot T \rfloor$ - Pick arm with best result

c) If $t > \varepsilon T$ (Exploit)

- pick it for entire run

Once you pick arm, you are stuck with it for life

what after many runs, you realize that the decision was not so good as you had imagined?



Choose arm 1

ε- greedy explore first with updated mean

for T runs,

Explore few times first

Exploit thereafter

a) If $t < \varepsilon \cdot T$ (Exploration)

- Sample at random

b) If $t > \varepsilon T$ (Exploit)

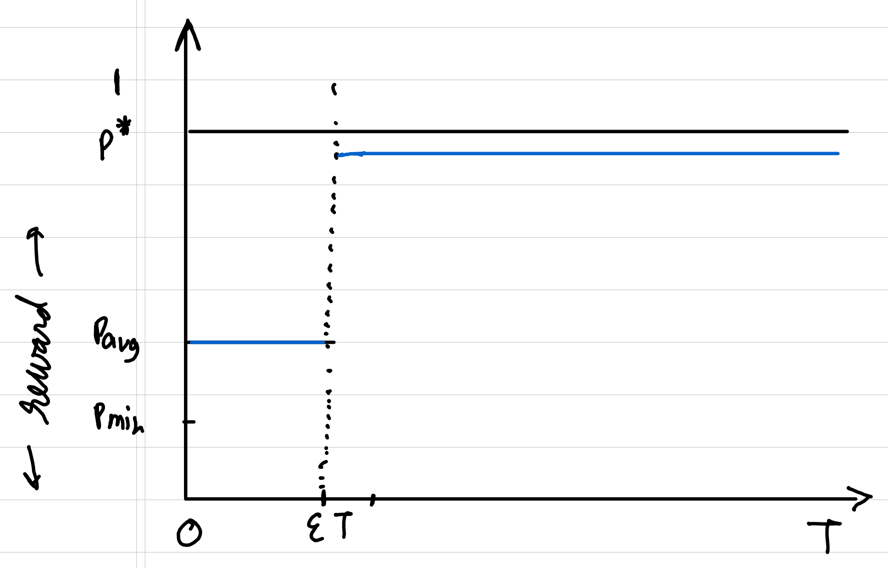
- select current best

- pick it for this run

This strategy allows for change if the mean of selected arm goes down

But this may not ensure that the best bandit has been selected, since we stop exploring after time εT

Regret calculation



p^* → probability of best bandit

P_{avg} → avg of all possibilities

P_{min} → probability of worst bandit

Expected Regret $R_T = T \times p^* - \sum_{t=0}^{T-1} E(r_t)$

Expected reward at time t

for strategies 1 & 2,

$$R_T = \underbrace{T p^*}_{\text{total area}} - \underbrace{\sum_{0}^{\varepsilon T - 1} E(r_t)}_{\text{Exploration area}} - \underbrace{\sum_{\varepsilon T}^{T} E(r_t)}_{\text{exploitation area}}$$

But since in exploitation the p^* may not be fully reached or may reach

ie $\sum_{\varepsilon T}^{T} E(r_t) \leq (T - T\varepsilon) p^*$

Hence

$$R_T = T p^* - \varepsilon T P_{avg} - \sum_{\varepsilon T}^{T} E(r_t) \geq T p^* - \varepsilon T P_{avg} - (T - T\varepsilon) p^*$$

$$\geq \varepsilon (p^* - P_{avg}) T$$

linear in T

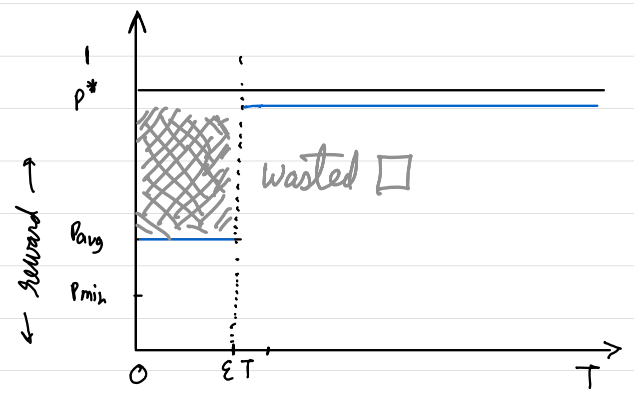
Hence $R_T = \Omega(T)$

What happens if T doubles to 2T?

Exploratory phase would become $2\varepsilon T$

The exploration rectangle will also get stretched by 2

So whatever we are doing, we are wasting the exploration rectangle ∴



Regret is lower bounded by $\Omega(T)$

This means linear regret. As much we increase T a %tage of efforts always get wasted

If $p^* = 0.9$
$P_{avg} = 0.5$
$\varepsilon = 0.1$

Waste $= 0.1 \times (0.9 - 0.5) \cdot T$
$= 0.04T$

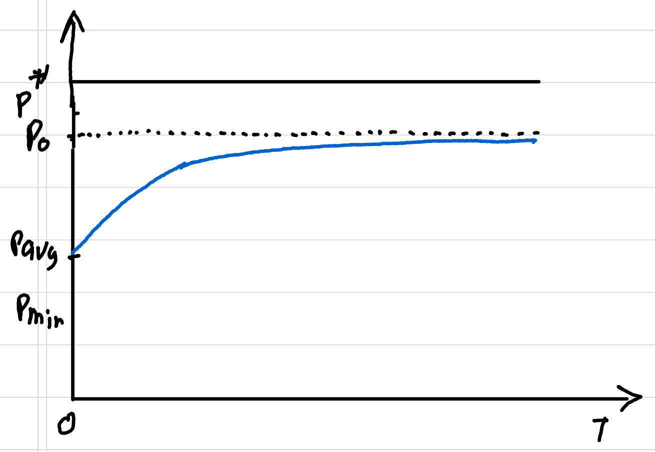
T	Total reward	Waste	%tage
10	9	0.4	3.6 %
100	90	4	3.6 %
1000	900	40	3.6 %
10000	9000	400	3.6 %

Waste grows linearly with T & waste % remains constant

Probabilistic ε- greedy strategy

Explore with probability ε
Exploit with probability $1-\varepsilon$

1) Pick random number p $(0 \le p \le 1)$
2) If $p < \varepsilon \rightarrow$ Explore
3) else exploit



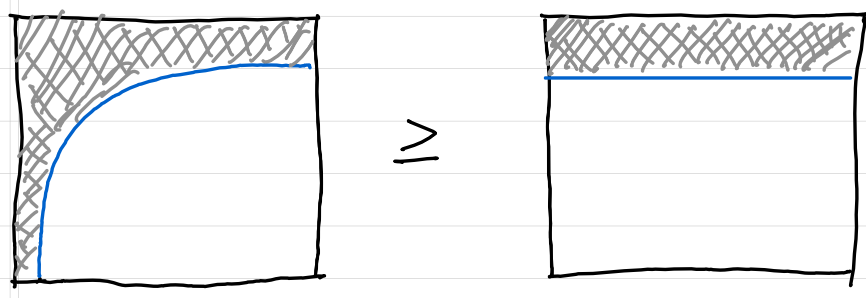
$P_0 = p^*(1-\varepsilon) + \varepsilon P_{avg}$ [Unlike first two, there is a theoritical maximum here]

The maximum probability P_0 it can reach is bounded by its explorations cost

$E(r_t)$ can never exceed P_0

Hence $R_T = Tp^* - \sum_0^{T-1} E(r_t)$

Lets lay an upper bound by considering loss of initial exploration phase

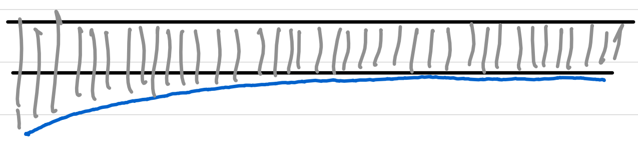


ie the Regret can never go less than the area $T \cdot P_0$

$$
\begin{aligned}
R_T &\le Tp^* - TP_0 \\
&\le Tp^* - Tp^*(1-\varepsilon) + T\varepsilon P_{avg} \\
&\le \varepsilon(p^* - P_{avg})T
\end{aligned}
$$

Hence $R_T = \Omega(T)$ again

Again, the regret is bounded and we are wasting a rectangle



← This □

As T increases, the rectangle will still get wasted.
This is the cost of exploring forever

First two strategies give away a small rectangle in the beginning

This strategy gives it away forever. There is no improvement with T

The area wasted depends upon ε.

If $p^* = 0.9$, $P_{avg} = 0.5$, $\varepsilon = 0.1$ Waste $= 0.1 \times (0.9 - 0.5) \cdot T$
$= 0.04T$

T	Total reward	Waste	% tage
10	9	0.4	3.6 %
100	90	4	3.6 %
1000	900	40	3.6 %
10000	9000	400	3.6 %

Waste grows linearly with T & waste % remains constant

Sublinear Regret

We want that as our T increases, the wasted efforts must get smaller & smaller

Inorder to do that, we need algorithms that fulfill the conditions →

Ⓐ Greed in Limit

As T increases, the ratio of exploitation to total time must become 1

ie the time % wasted in exploration should reduce to 0

$$\lim_{T \to \infty} \frac{E(\text{Exploited } T)}{T} = 1$$

Ⓑ Infinite Exploration

In limit $T \to \infty$, each arm must be pulled an infinite number of times

Because, if we explore an arm only finite fixed U times (fixed regardless of T) then,

there is a small chance that the optimal arm will have reward mean 0 due to bad luck

$$(1-p^*)^U > 0$$

A non optimal arm may thereafter be exploited forever

Hence $$\lim_{T \to \infty} \frac{E(\text{Exploration})}{n} = \infty$$

* An algorithm achieves sub linear regret if and only if it satisfies both above conditions on all bandits

These are called GLIE condition

	GL	IE
ε-greedy 1	×	✓
ε-greedy 2	×	✓
ε-greedy 3	×	✓

GLIE are necessary and sufficient conditions for sublinear regret

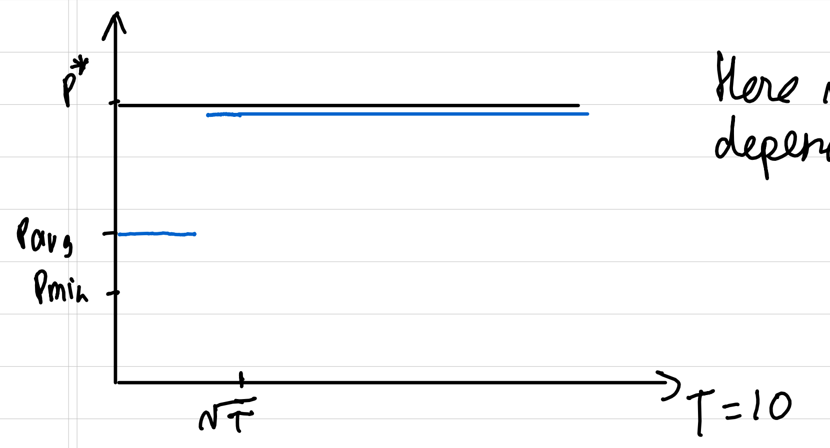
Explore first ε-greedy with GLIE (1^{st} & 2^{nd} strategy)

$\epsilon_T = \frac{1}{\sqrt{T}}$ Explore for $\epsilon_T \cdot T = \sqrt{T}$ pulls
Exploit for $T - \sqrt{T}$ pulls

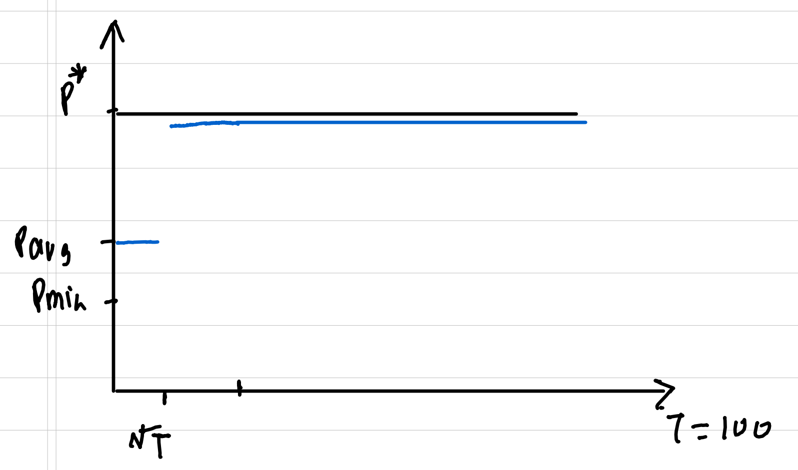
$$E(\text{Exploit}) = \frac{T-\sqrt{T}}{T}$$

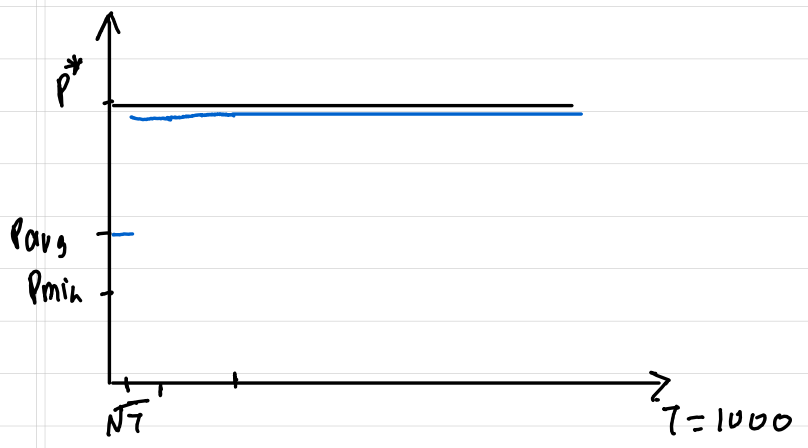
GL $\lim_{T\to\infty} \frac{T-\sqrt{T}}{T} = \lim_{T\to\infty} \frac{1-1/\sqrt{T}}{1} = 1$ ✓

IE $\lim_{T\to\infty} \frac{\sqrt{T}}{n} = \infty$ ✓



Here $\sqrt{T}$ position depends on T





Regret

$$R_T = Tp^* - \sqrt{T}\cdot p_{avg} - \sum_{t=0}^{T-1} E(r^t)$$

$$\geq Tp^* - \sqrt{T}\cdot p_{avg} - (T-\sqrt{T})p^*$$

considering optimal value is found

$$\geq \sqrt{T}(p^* - p_{avg})$$

Hence

$$R_T = \Omega(\sqrt{T})$$

As T grows, the size of the rectangle also grows, but grows lesser at faster rate

Area wasted $\propto \sqrt{T}$

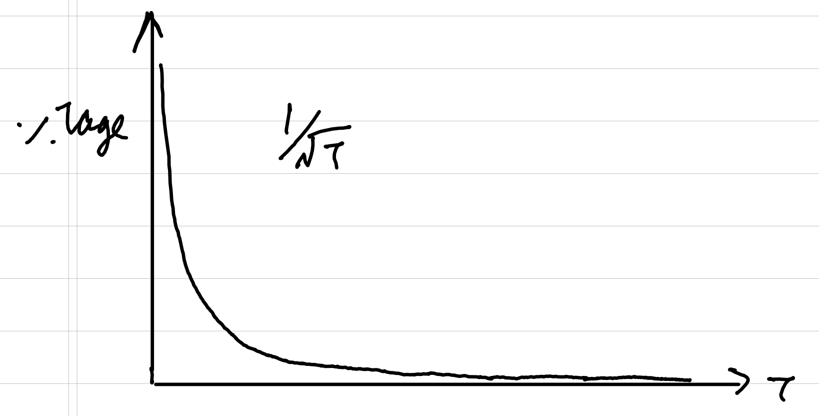
If $p^* = 0.9$, $p_{avg} = 0.5$ waste $= (0.9-0.5)\sqrt{T}$

$= 0.4\sqrt{T}$

T	Total reward	waste	% tage
10	9	1.26	14
100	90	4	4.4
1000	900	12.6	1.4
10000	9000	40	0.4

Waste increases with $\sqrt{T}$

% of waste decreases and goes to 0



Hence the ratio $\frac{\text{Exploit}}{\text{Total}}$ goes to 1

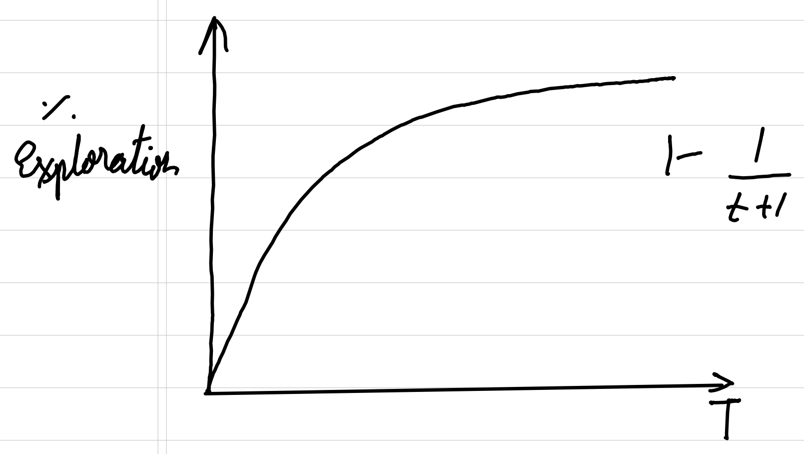
ε-greedy probabilistic with GLIE

Let $\varepsilon = \frac{1}{t+1}$ (small t & not T)

On the t^{th} step, we explore with prob $\frac{1}{t+1}$

Hence the probability of exploration approaches 0

Probability of exploitation approaches 100%



$$\text{Total exploration} = \sum_{t=0}^{T} \frac{1}{t+1} \geq \ln(T+1)$$

$$E(\text{Exploit}) = \frac{T - \sum_{t=0}^{T} \frac{1}{t+1}}{T}$$

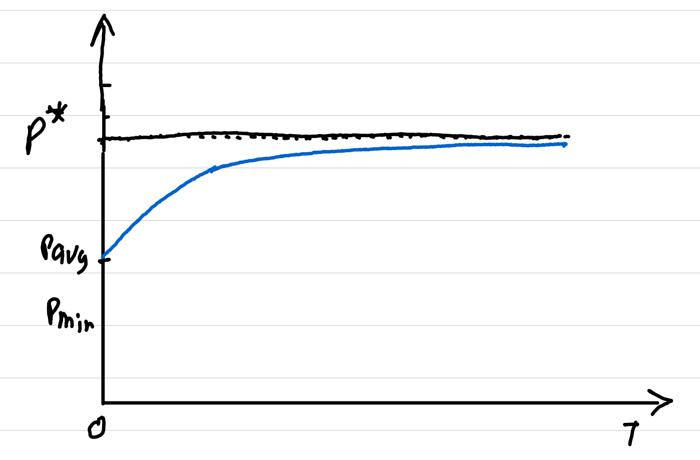
GL: $\lim_{T \to \infty} E(\text{Exploit}) \geq \frac{T - O(\log T)}{T}$

$= 1$ ✓

IE: Each arm is assured $\frac{\varepsilon_t}{n}$ pulls, that is

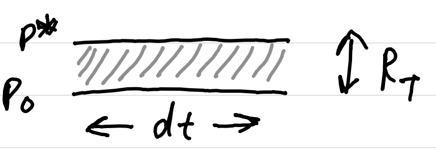
$$\lim_{T \to \infty} \sum_{0}^{T} \frac{1}{(t+1) \cdot n} = \infty \checkmark$$

(Harmonic divergence)



Here, P_0 also changes ie theoritical waste

At time t, consider a time slice dt



$$P_{0_t} = (1-\varepsilon_t)P^* + \varepsilon_t P_{avg}$$

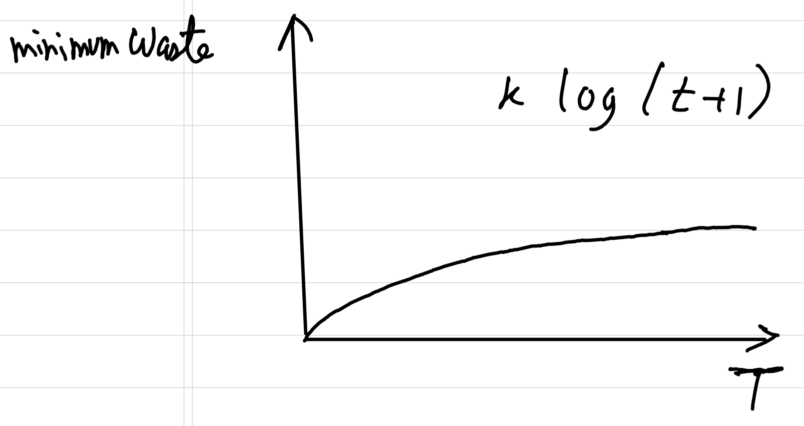
Theoritical minimum $\rightarrow$ $d\gamma = P^* dt - P_0 \, dt$

$$d\gamma = \varepsilon_t (P^* - P_{avg}) \, dt$$

$$\int_0^T d\gamma = \int_0^T \frac{1}{(t+1)} (P^* - P_{avg}) \, dt$$

$$R_T = (P^* - P_{avg}) \log(T+1)$$

Theoritical waste



eg $P^* = 0.9$
$P_{avg} = 0.5$

T	min waste	min waste %
10	0.95	9.5
100	1.84	1.84
1000	2.76	0.276
10000	3.68	0.0368

We are reducing the waste %

Although it is minimum waste the actual wastage also shows similar characteristic because after the model learns the optimal arm, reward will become P_0

Cost of learning the arm is independant of total time T and acts as constant

Hence the regret sublinearly grows with T

Lower Bound on Regret

What is the least complexity of regret?

The least possible bound is given by "the lower bound by Lai & Robbins" (1985)

If $R_T = o(T^{\alpha})$ [simply means sublinear policy] then

$$\frac{R_T}{\ln(T)} \geq \sum_{arm} \frac{p^* - P_{arm}}{P_{arm} \ln \frac{P_{arm}}{p^*} + (1-P_{arm}) \ln\left(\frac{1-P_{arm}}{1-p^*}\right)}$$

Basically, $R_T \geq K \ln(T)$

cannot go beyond $\ln(T)$ for sublinear regret

Lowest is $\Omega(\ln(T))$

That means $\log(\log(T))$ can never happen.

The first condition means that you are not doing something like exploring forever or not learning an arm.

UCB (Upper Confidence Bounds) algorithm

At time t, for every arm a, define UCB as

$$ucb_a^t = \hat{p}_a^t + \sqrt{\frac{2\ln(t)}{u_a^t}}$$

empirical mean of rewards from arm a

number of times arm a has been sampled till time t

Sample an arm a for which ucb_a^t is maximum

As t increases, ucb_a^t increases slightly for every arm due to $\ln(t)$

As mean reward ($\hat{p}_a^t$) obtained increases ucb_a^t increases linearly

As u_a^t increases (whenever the arm is pulled) the ucb_a^t decreases slightly

When arm is pulled small number of times, u_a^t is going to be large hence $\sqrt{\ }$ term is large

So when the arm is not pulled enough number of times there is an incentive to go and explore it because the UCB is higher

An arm that gives higher rewards will also be higher UCB due to higher mean reward.

When an arm has been sampled enough number of times, the $\sqrt{\ }$ factor reduces a lot and UCB becomes equal to empirical mean $\hat{p}_a^t$

"Enough" is defined relatively wrt time by $\ln(t)$ term.

Achieves regret $O(\log(T))$: optimal dependance in T

But does not match the constant of lowest bound

Better than ϵ-greedy

In order to improve upon the constant, the KL UCB algorithm was developed.

KL-UCB is one of the best possible complexity for Multi armed bandits

Beta distribution

Beta distribution is continouse univariate distribution that can take values between 0-1

Total area under curve is 1

It is probability of probabilities. Useful when the probability of an event is not known

Beta distribution models the belief of probability

It tells you "what is the probability that an unknown probability p takes on a specific value?"

$$f(x, \alpha, \beta) = \boxed{\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}} x^{\alpha-1}(1-x)^{\beta-1}$$

Success Parameter — α

failure Parameter — β

$\frac{1}{B(\alpha,\beta)}$ Beta function

Normalization to ensure sum 1

$\Gamma(z)$ is the gamma function

$\Gamma(n+1) = n\Gamma(n) = n!$ for integers

$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ for continouse cases

Hence for discrete values of α & β

$$\frac{1}{B(\alpha,\beta)} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} = \frac{(\alpha+\beta-1)!}{(\alpha-1)!\,(\beta-1)!}$$

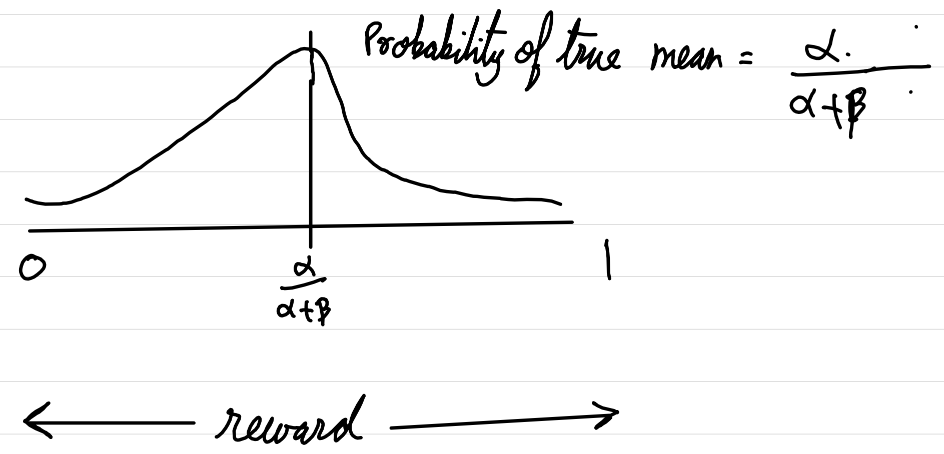
Note $\rightarrow$ $\int_0^1 x^{\alpha-1}(1-x)^{\beta-1} = B(\alpha,\beta)$

Usage $\rightarrow$ 1) Assign a prior "belief" probability
2) Experiment. Note success & failure
3) Update α & β with results

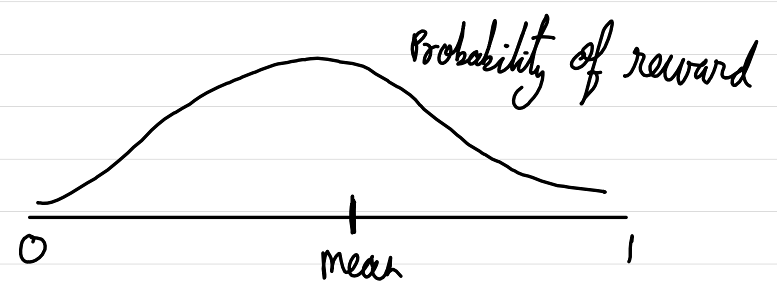
$$\text{Mean} = \frac{\alpha}{\alpha+\beta}$$

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

Beta models probability of mean $= \frac{\alpha}{\alpha+\beta}$



Note that this is different from gaussian



Gaussian bell models probability of reward.
It assumes a known mean.
Beta on the other hand models probability of mean being a value.
Useful when mean is unknown.

Thompson Sampling (1933)

Works on Beta distribution

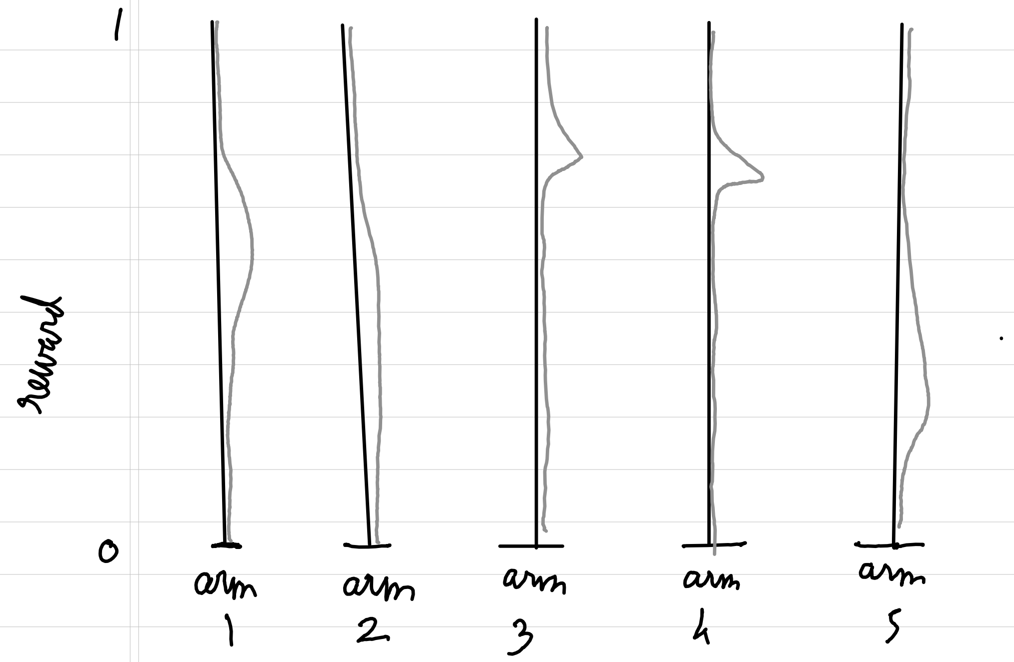
At time t, let arm a have s_a^t successes and f_a^t failures

Beta distribution $f(s_a^t + 1, f_a^t + 1)$ represents a belief about the true distribution of a

$$\text{Mean} = \frac{s_a^t + 1}{s_a^t + f_a^t + 2}$$

$$\text{Variance} = \frac{(s_a^t + 1)(f_a^t + 1)}{(s_a^t + f_a^t + 2)^2 (s_a^t + f_a^t + 3)}$$

Arm a true mean is unknown, but we have belief that it is $s_a^t + 1$ & $f_a^t + 1$



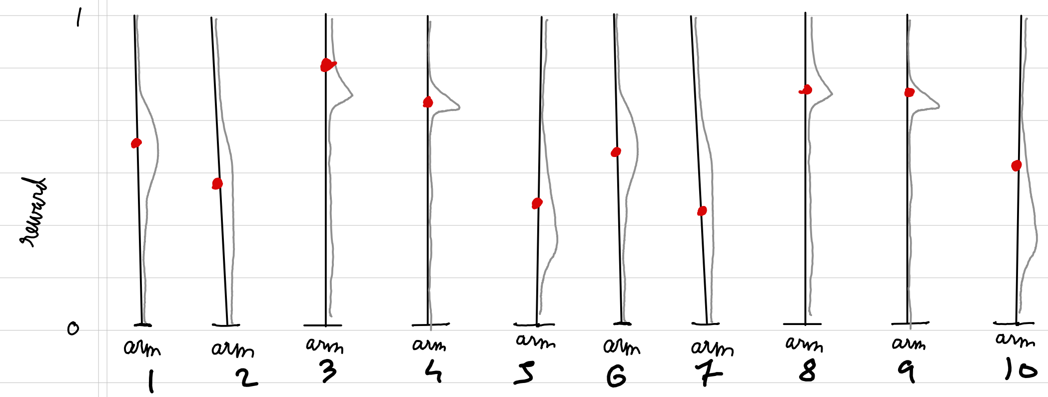
Now we dont have fixed means. Note that these are the probabilities of means and not rewards.

Expected reward over time now has mean as well as variance

A single throw is still bernoli $1 \cdot P + 0 \cdot (1-P)$

Thompson Sampling algorithm

step 1 → from every arm pick x_a^t, a random number sampled from beta distribution



step 2 → Pull arm a for which x_a^t is maximum.

step 3 → Update α & β of beta. It will change only for the arm we are pulling

Very effective in practise

for unexplored arms, variance is lower, hence there is probability that the x_a^t will be high

Thompson sampling in practise is slightly more effective than KL-UCB

It is a randomized algorithm with complexity par of Lai lower bound

Hoeffdings inequality (1963)

Let X be a random variable bounded in $[0,1]$ with $E(x) = \mu$

Let $x_1, x_2 \dots x_n$ be iid samples of X

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_n \rightarrow \text{sample mean}$$

Then,
$$P(\bar{x} \geq \mu + \varepsilon) \leq e^{-2n\varepsilon^2}$$
&
$$P(\bar{x} \leq \mu + \varepsilon) \geq e^{-2n\varepsilon^2}$$

1. For given mistake probability δ and tolerance ε how many samples n of X do we need to gurantee that with probability $1-\delta$, the empirical mean $\bar{x}$ will not exceed the mean μ by ε or more?

$$P(\bar{x} \geq \mu + \varepsilon) \leq e^{-2n\varepsilon^2} \leq \delta$$

$\therefore$ at limiting condition

$$e^{-2n\varepsilon^2} = \delta$$

$$\therefore n = \frac{-1}{2\varepsilon^2} \ln(\delta)$$

$$n = \frac{\ln(1/\delta)}{2\varepsilon^2}$$ pulls are sufficient (ie. samples)

2. We have n samples of X then with probability at least $1-\delta$, the empirical mean $\bar{x}$ exceeds the true mean μ by at most ε

Then $\varepsilon = \sqrt{\frac{1}{2n} \ln(1/\delta)}$