

Exploration and Exploitation (Bandits)

Last Time

- What is Reinforcement Learning?

Last Time

- What is Reinforcement Learning?
- What are the main challenges in Reinforcement Learning?

Last Time

- What is Reinforcement Learning?
- What are the main challenges in Reinforcement Learning?
- How do we categorize RL approaches?

- 
1. Exploration ^{vs} Exploitation
 2. Credit Assignment
 3. Generalization

Model - Free
Model - Based

Learn π^* , Q
Learn T, R

On Policy
Off Policy
Batch

Tabular
Deep

Exploration policy
Optimized policy

Last Time

Last Time

First RL Algorithm:

Last Time

First RL Algorithm:

Tabular Maximum Likelihood Model-Based Reinforcement Learning

Last Time

First RL Algorithm:

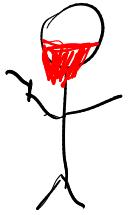
Tabular Maximum Likelihood Model-Based Reinforcement Learning

loop
 choose action a
 gain experience
 estimate T, R
 solve MDP with T, R

Guiding Questions

- What are the best ways to trade off Exploration and Exploitation?

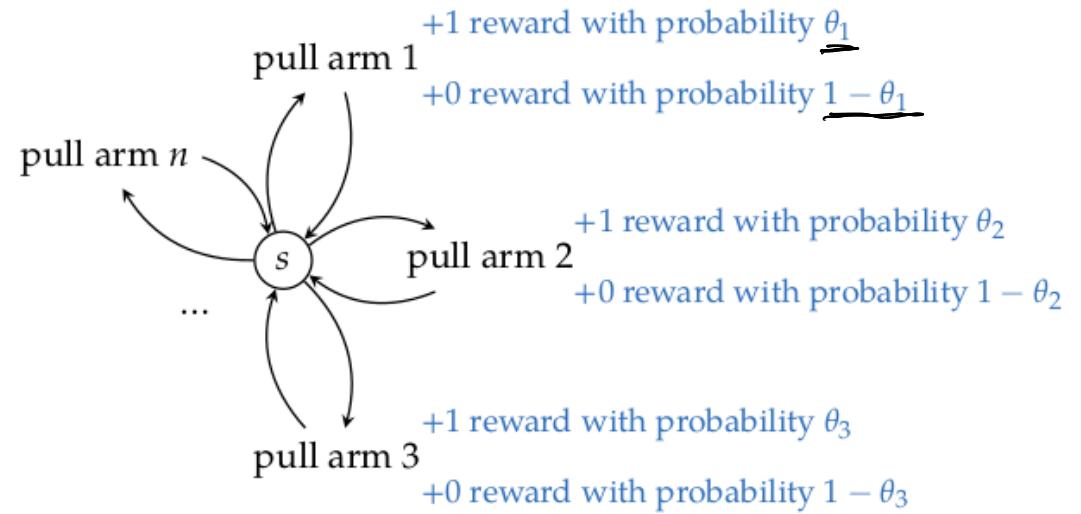
Bandits



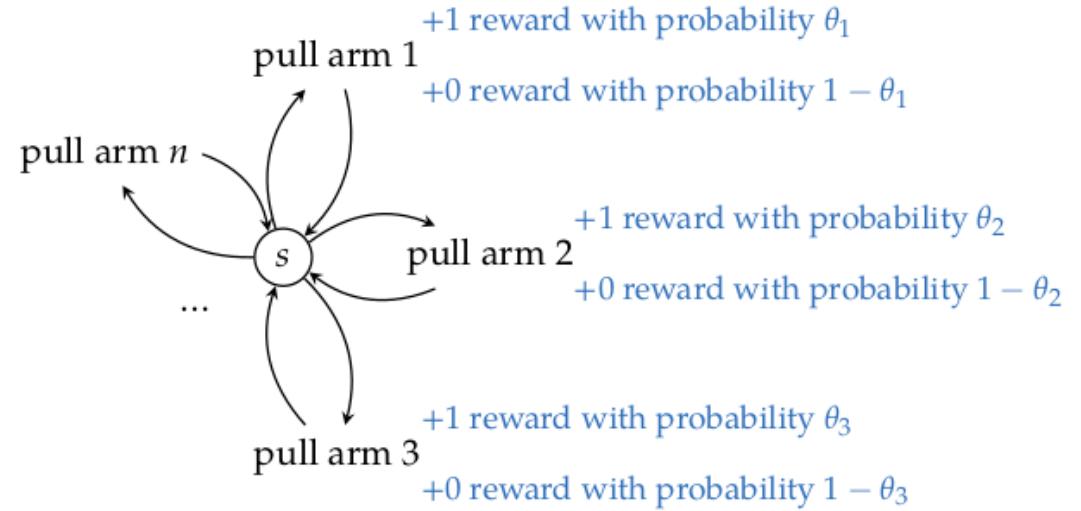
Bandits



Bandits

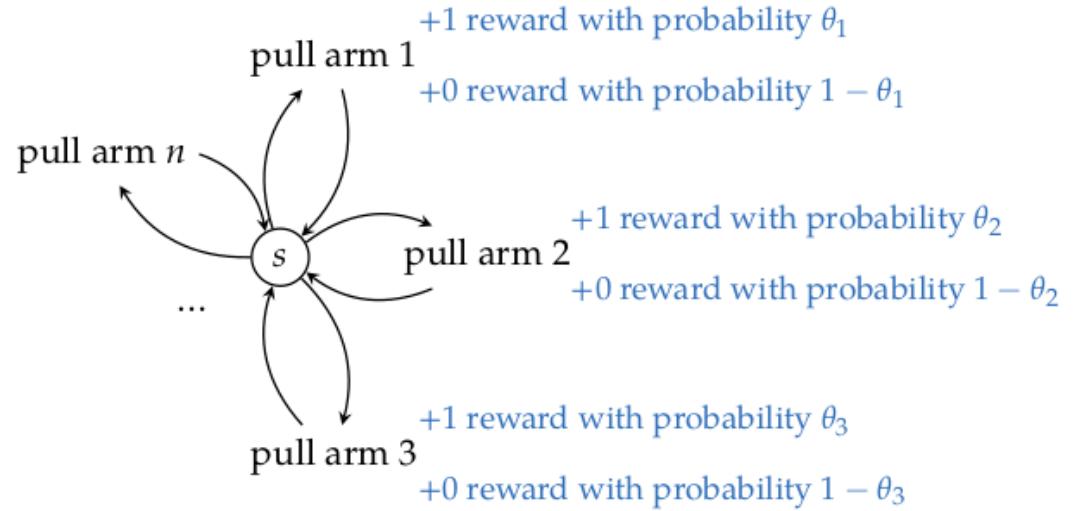


Bandits



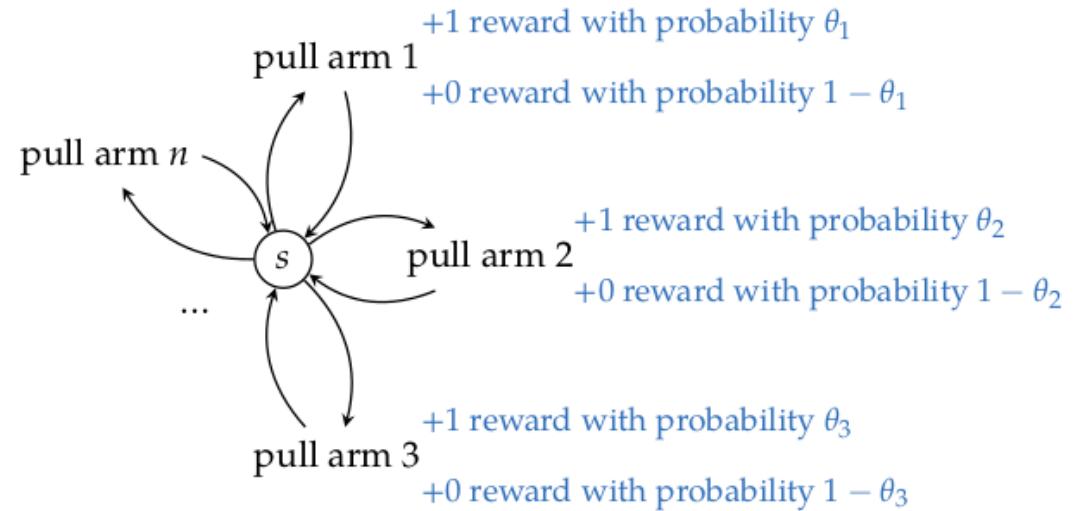
- Bernoulli Bandit with parameters θ

Bandits



- Bernoulli Bandit with parameters θ
- $\theta^* \equiv \max \theta$

Bandits



- Bernoulli Bandit with parameters θ
- $\theta^* \equiv \max \theta$

“According to Peter Whittle, “efforts to solve [bandit problems] so sapped the energies and minds of Allied analysts that the suggestion was made that the problem be dropped over Germany as the ultimate instrument of intellectual sabotage.”

Greedy Strategy

$$\rho_a = \frac{\text{number of wins}+1}{\text{number of tries}+1}$$

Choose $\operatorname{argmax}_a \rho_a$

Undirected Strategies

Undirected Strategies

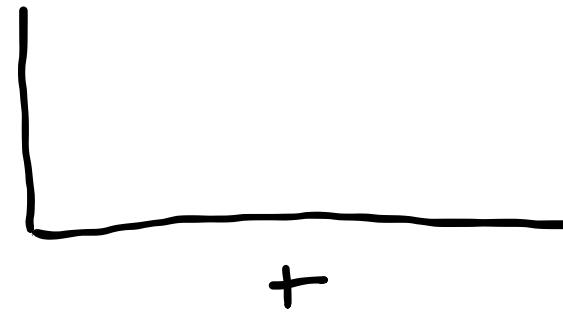
- Explore then Commit

Choose a randomly for k steps

Then choose $\operatorname{argmax}_a \rho_a$

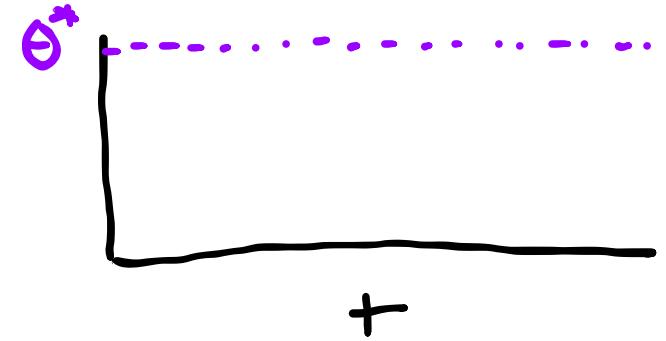
Undirected Strategies

- Explore then Commit
Choose a randomly for k steps
Then choose $\operatorname{argmax}_a \rho_a$



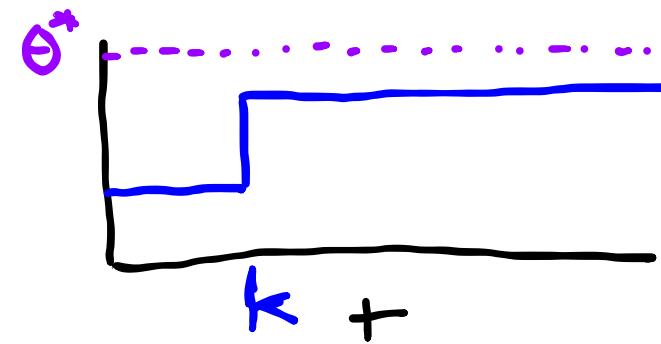
Undirected Strategies

- Explore then Commit
Choose a randomly for k steps
Then choose $\operatorname{argmax}_a \rho_a$



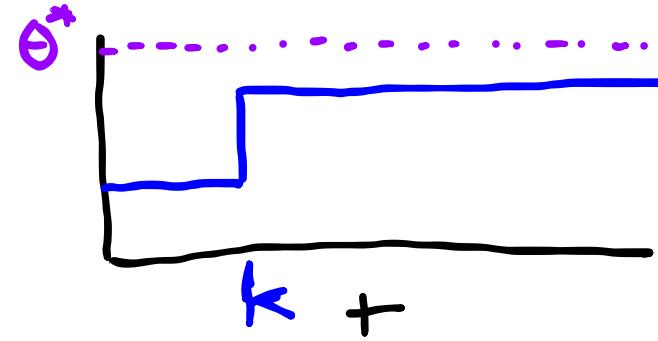
Undirected Strategies

- Explore then Commit
Choose a randomly for k steps
Then choose $\operatorname{argmax}_a \rho_a$



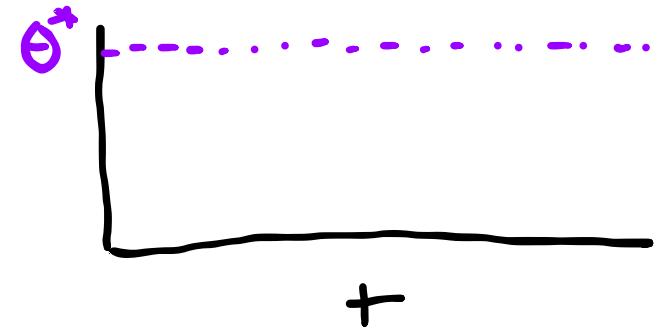
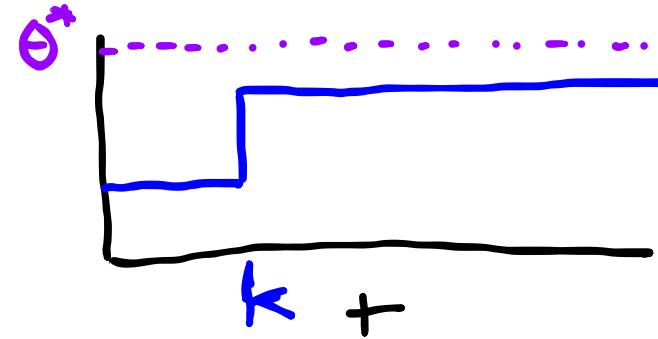
Undirected Strategies

- Explore then Commit
Choose a randomly for k steps
Then choose $\operatorname{argmax}_a \rho_a$
- ϵ - greedy
With probability ϵ , choose randomly
Otherwise choose $\operatorname{argmax}_a \rho_a$



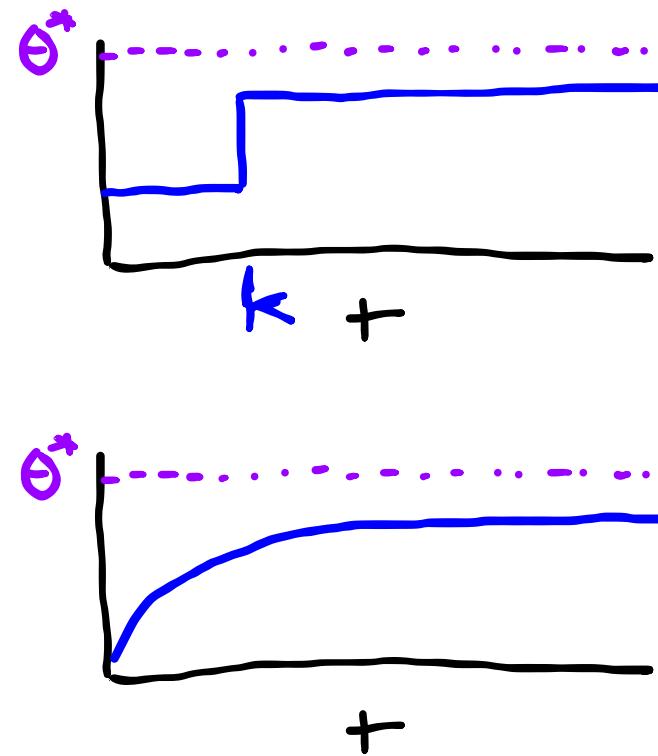
Undirected Strategies

- Explore then Commit
Choose a randomly for k steps
Then choose $\operatorname{argmax}_a \rho_a$
- ϵ - greedy
With probability ϵ , choose randomly
Otherwise choose $\operatorname{argmax}_a \rho_a$



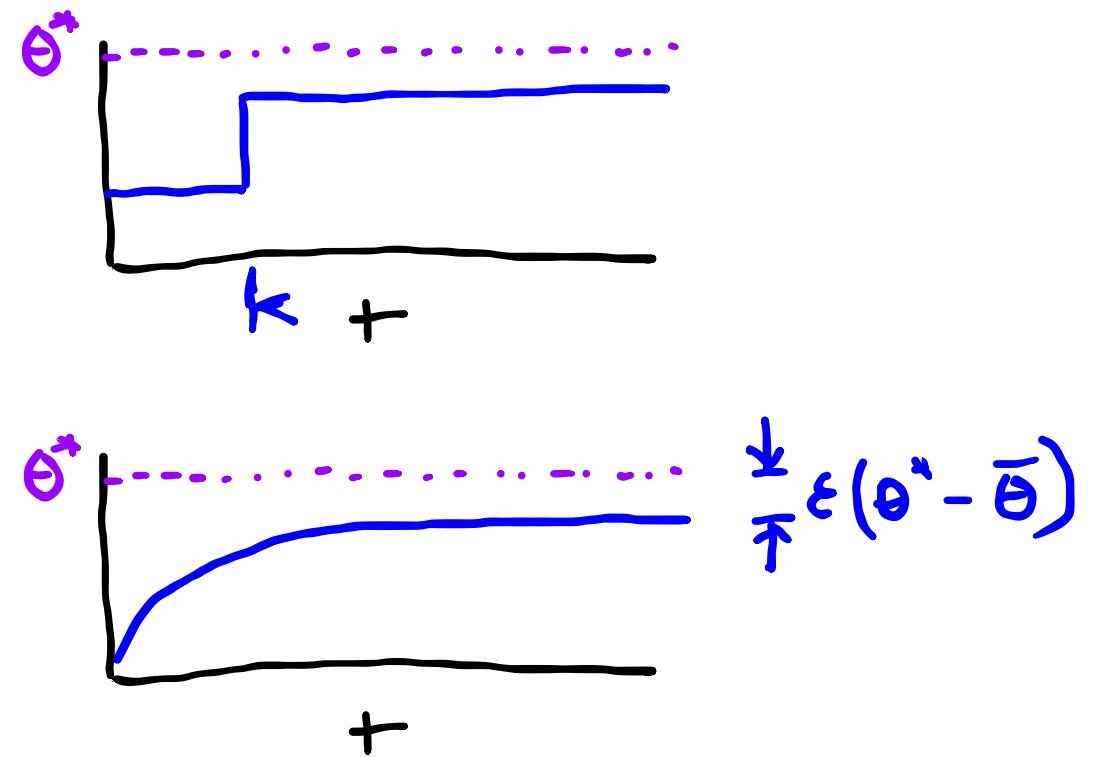
Undirected Strategies

- Explore then Commit
Choose a randomly for k steps
Then choose $\operatorname{argmax}_a \rho_a$
- ϵ - greedy
With probability ϵ , choose randomly
Otherwise choose $\operatorname{argmax}_a \rho_a$



Undirected Strategies

- Explore then Commit
Choose a randomly for k steps
Then choose $\operatorname{argmax}_a \rho_a$
- ϵ - greedy
With probability ϵ , choose randomly
Otherwise choose $\operatorname{argmax}_a \rho_a$



Directed Strategies



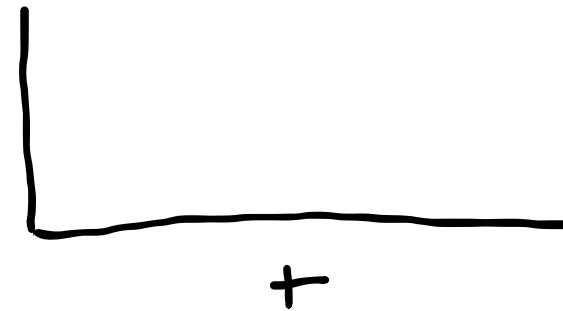
Directed Strategies

- Softmax
Choose a with probability
proportional to $e^{\lambda \rho_a}$



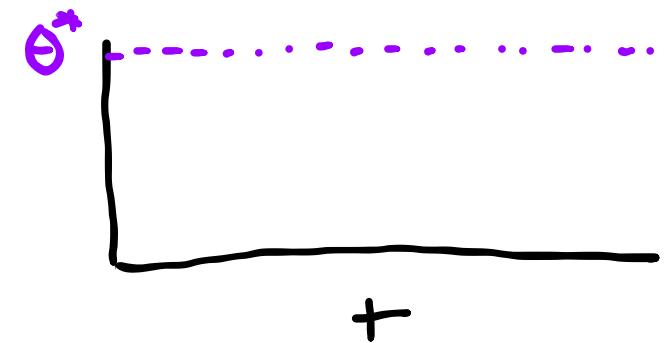
Directed Strategies

- Softmax
Choose a with probability
proportional to $e^{\lambda \rho_a}$



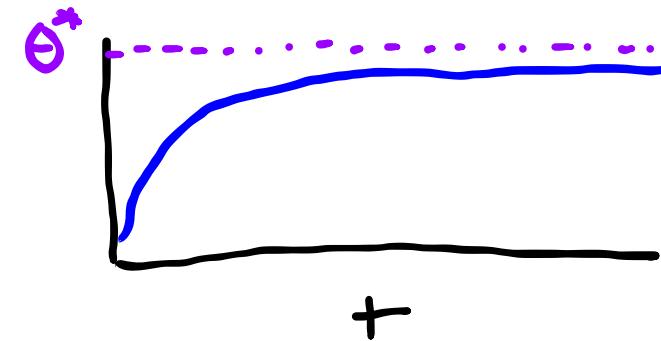
Directed Strategies

- Softmax
Choose a with probability proportional to $e^{\lambda \rho_a}$



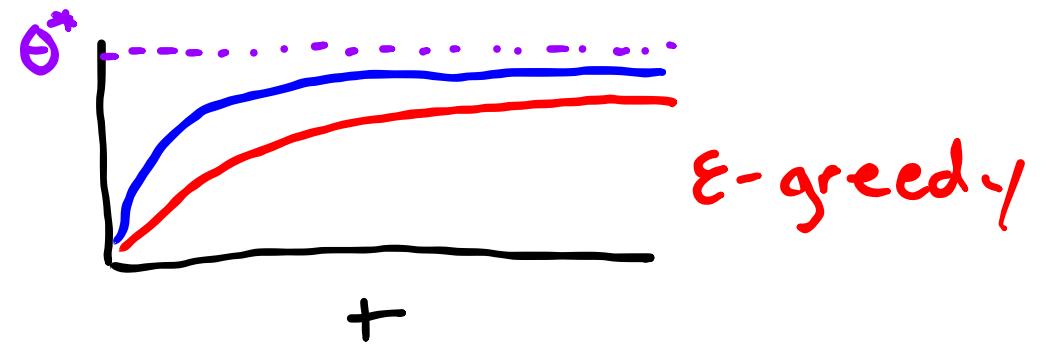
Directed Strategies

- Softmax
Choose a with probability proportional to $e^{\lambda \rho_a}$



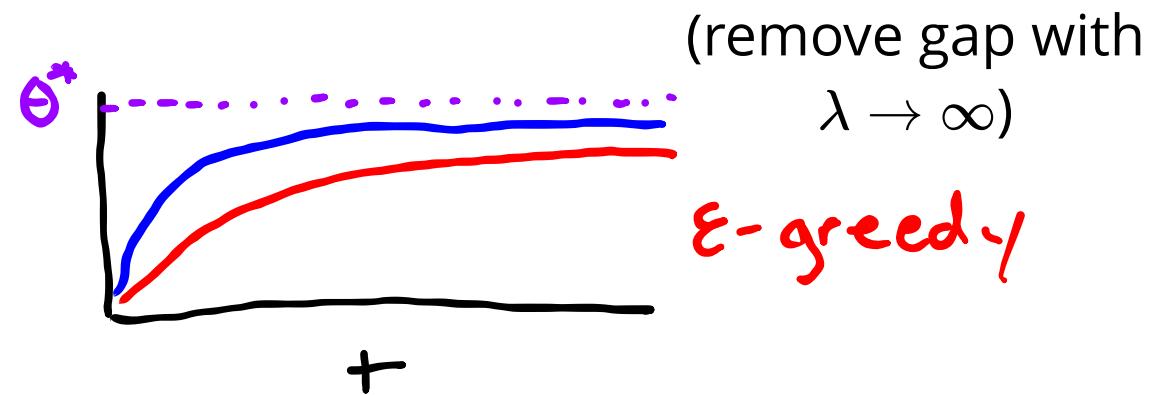
Directed Strategies

- Softmax
Choose a with probability proportional to $e^{\lambda \rho_a}$



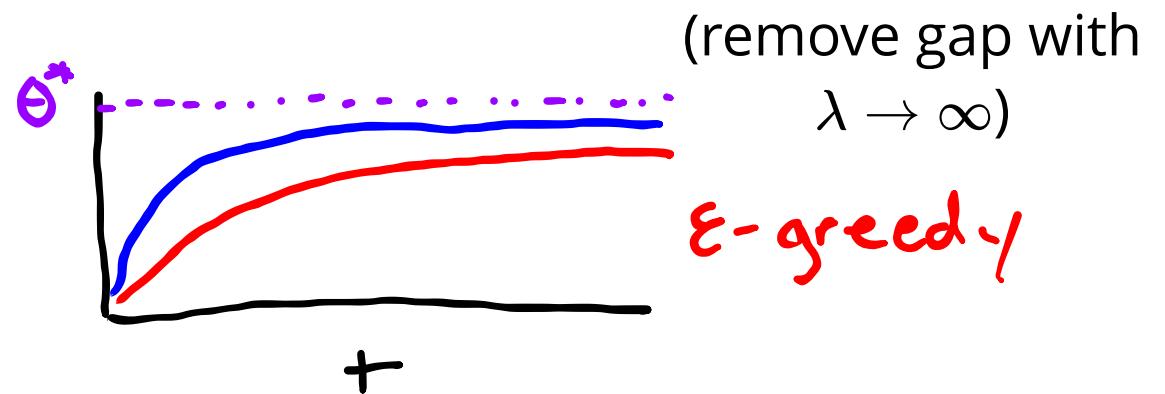
Directed Strategies

- Softmax
Choose a with probability proportional to $e^{\lambda \rho_a}$



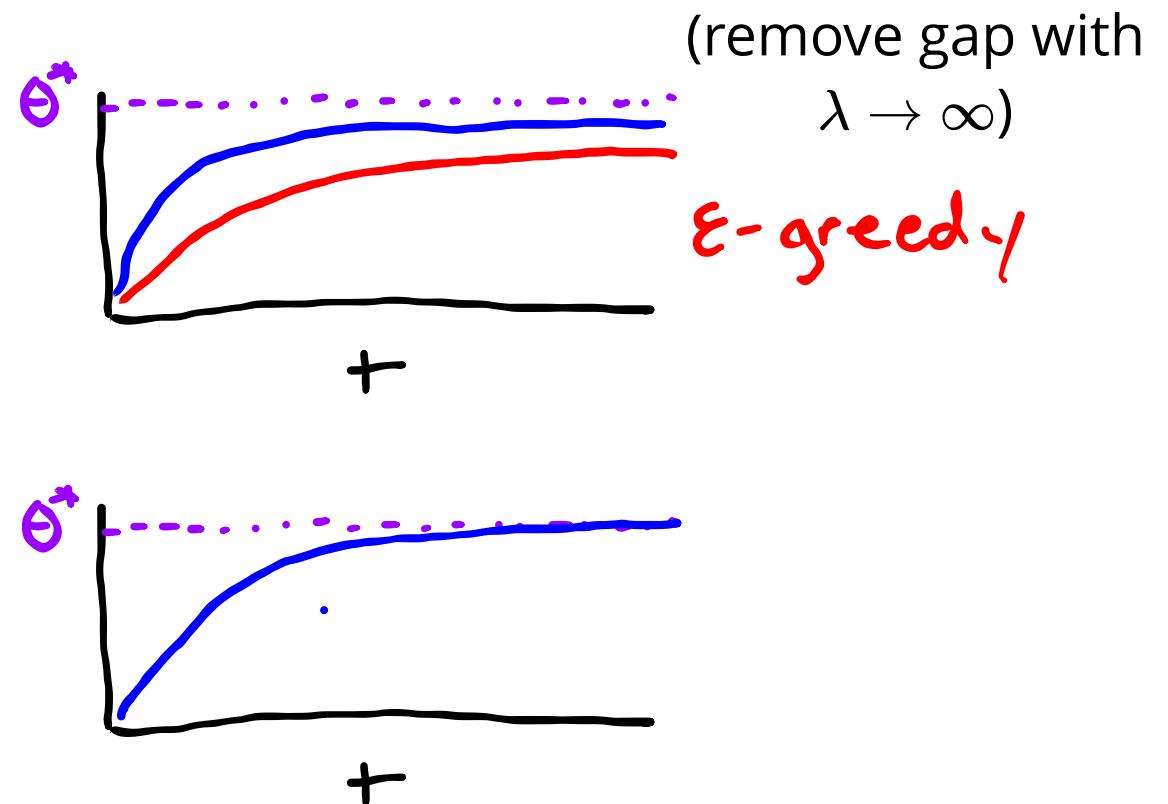
Directed Strategies

- Softmax
Choose a with probability proportional to $e^{\lambda \rho_a}$
- Upper Confidence Bound (UCB)
Choose $\underset{a}{\operatorname{argmax}} \rho_a + c \sqrt{\frac{\log N}{N(a)}}$



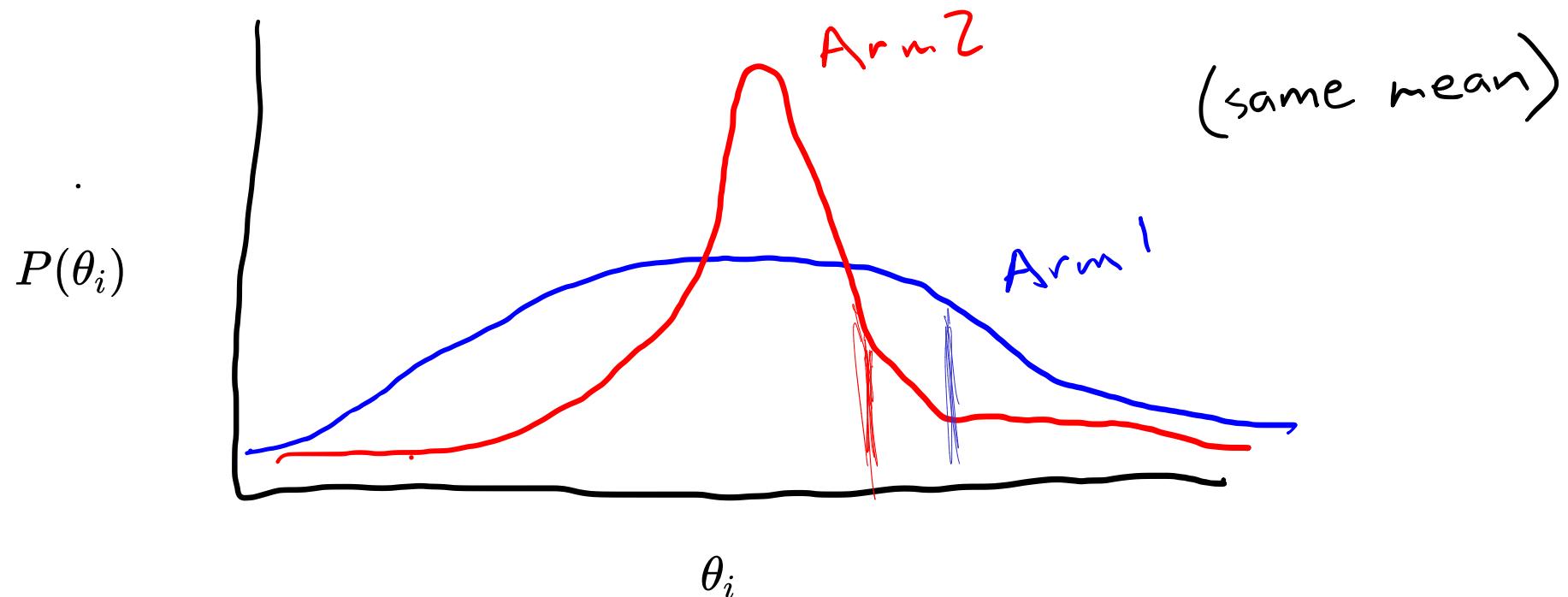
Directed Strategies

- Softmax
Choose a with probability proportional to $e^{\lambda \rho_a}$
- Upper Confidence Bound (UCB)
Choose $\underset{a}{\operatorname{argmax}} \rho_a + c \sqrt{\frac{\log N}{N(a)}}$



Break

Discuss with your neighbor: Suppose you have the following *belief* about the parameters θ . Which arm should you choose to pull next?



Bayesian Estimation

Bayesian Estimation

Bernoulli Distribution

Bayesian Estimation

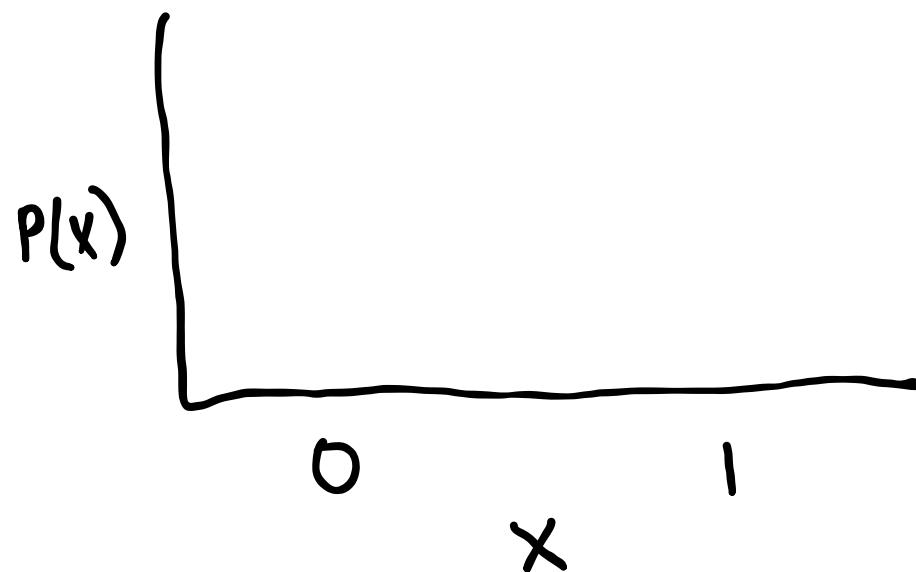
Bernoulli Distribution

$$\text{Bernoulli}(\theta)$$

Bayesian Estimation

Bernoulli Distribution

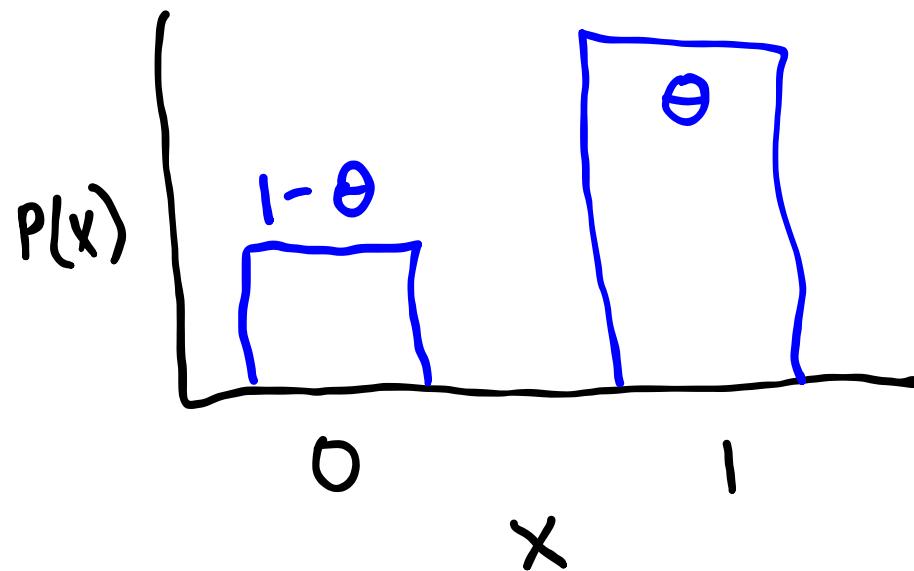
$$\text{Bernoulli}(\theta)$$



Bayesian Estimation

Bernoulli Distribution

$$\text{Bernoulli}(\theta)$$

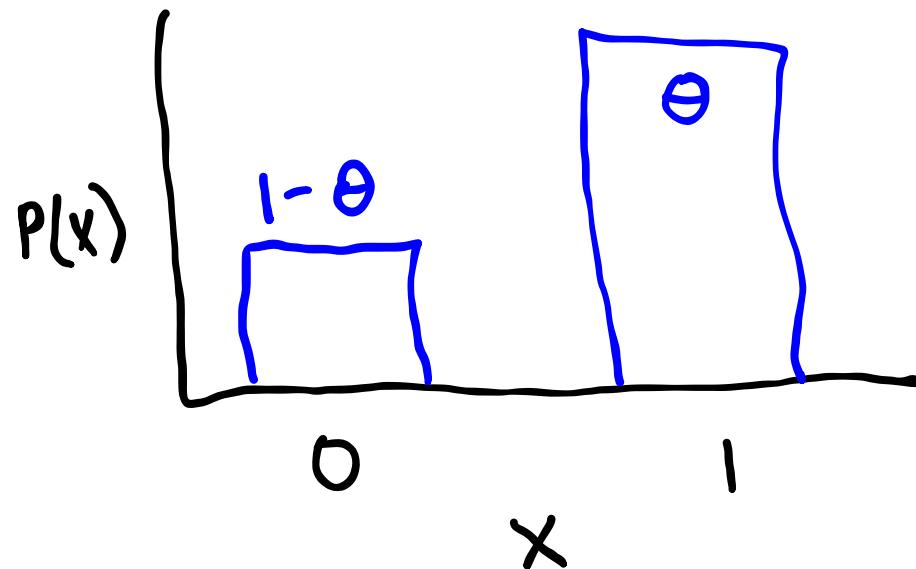


Bayesian Estimation

Bernoulli Distribution

$\text{Bernoulli}(\theta)$

Discussion: Given that I have received w wins and l losses, what should my belief (probability distribution) about θ look like?

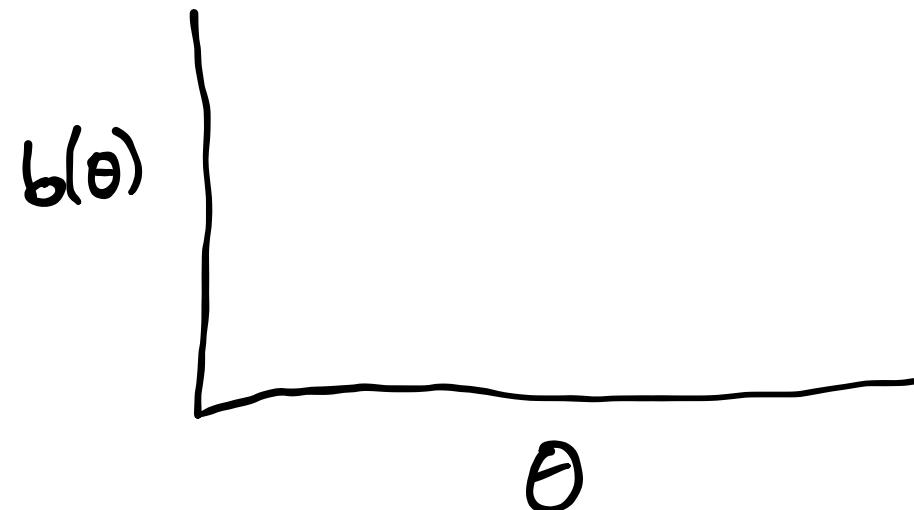
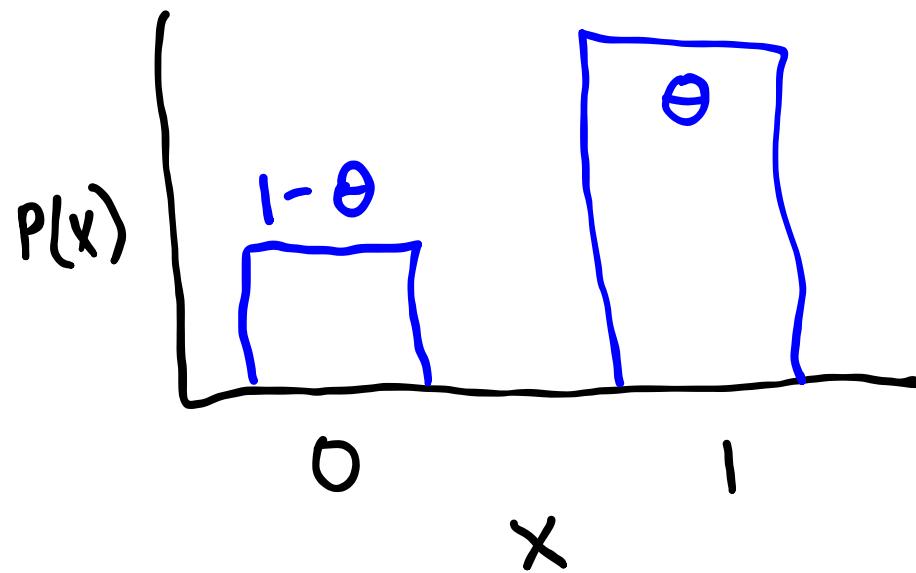


Bayesian Estimation

Bernoulli Distribution

$\text{Bernoulli}(\theta)$

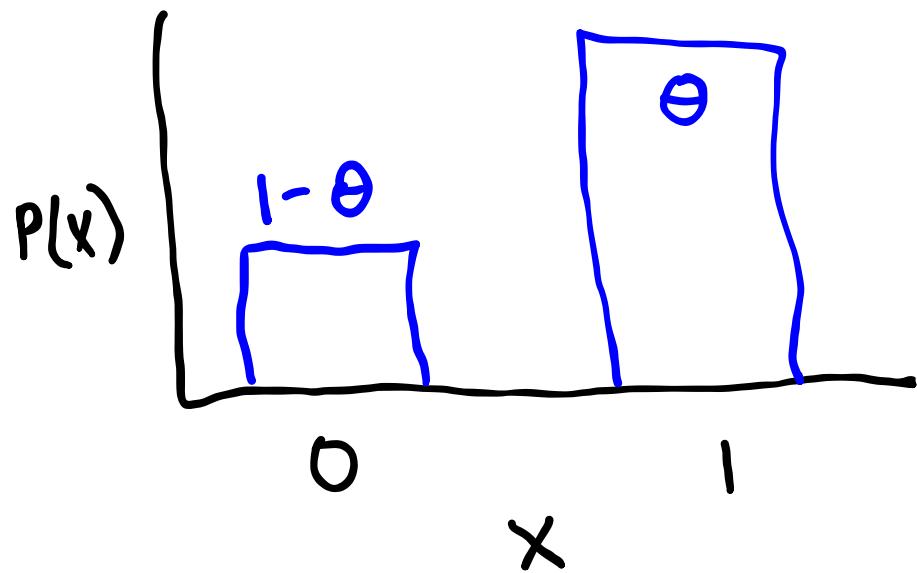
Discussion: Given that I have received w wins and l losses, what should my belief (probability distribution) about θ look like?



Bayesian Estimation

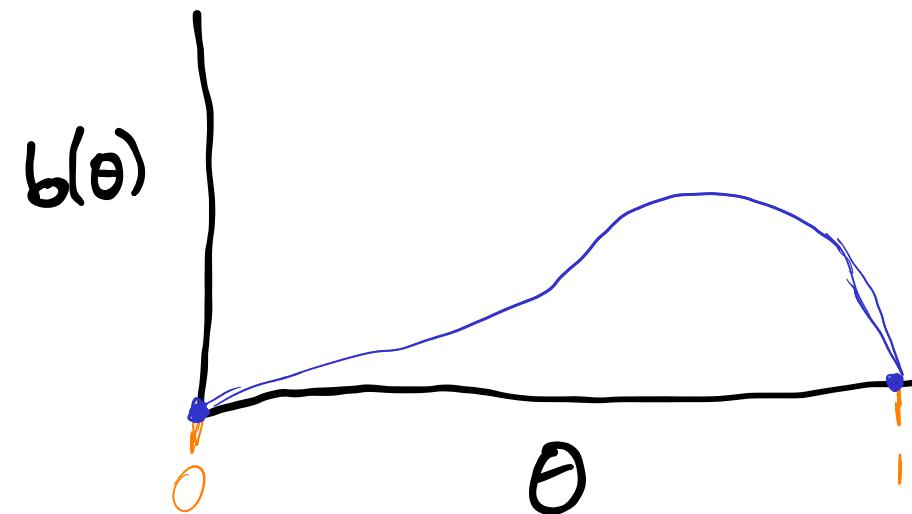
Bernoulli Distribution

$\text{Bernoulli}(\theta)$



Discussion: Given that I have received w wins and l losses, what should my belief (probability distribution) about θ look like?

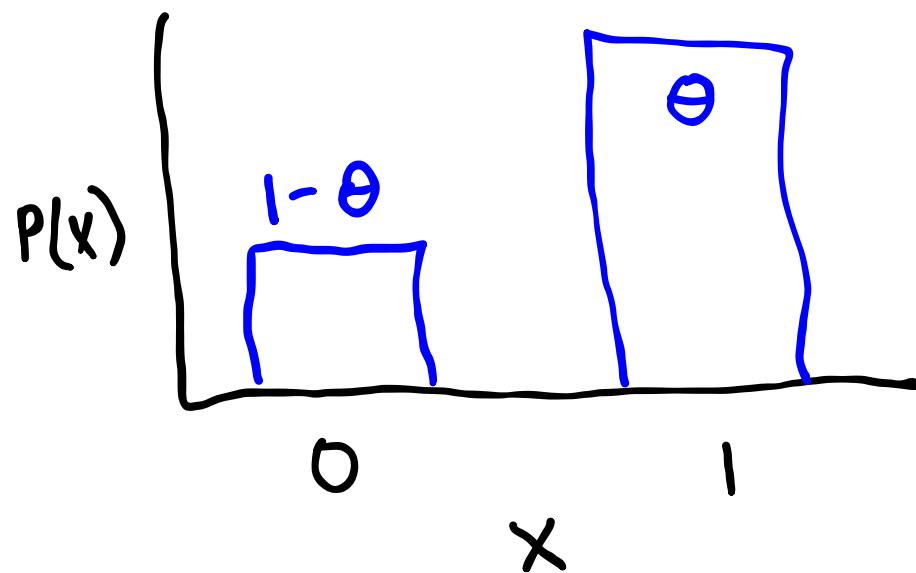
$$w=4, l=1$$



Bayesian Estimation

Bernoulli Distribution

$$\text{Bernoulli}(\theta)$$



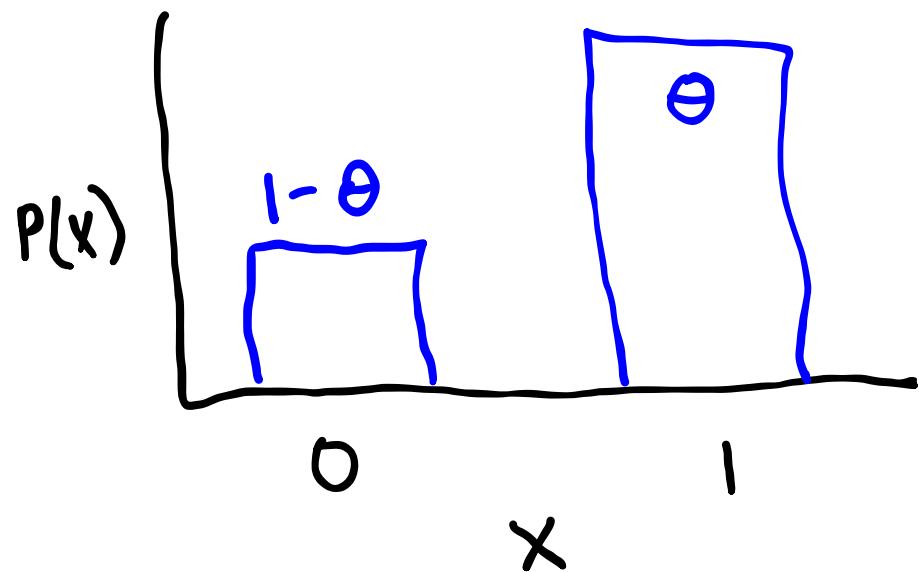
Bayesian Estimation

Bernoulli Distribution

$\text{Bernoulli}(\theta)$

Beta Distribution

(distribution over Bernoulli distributions)



Bayesian Estimation

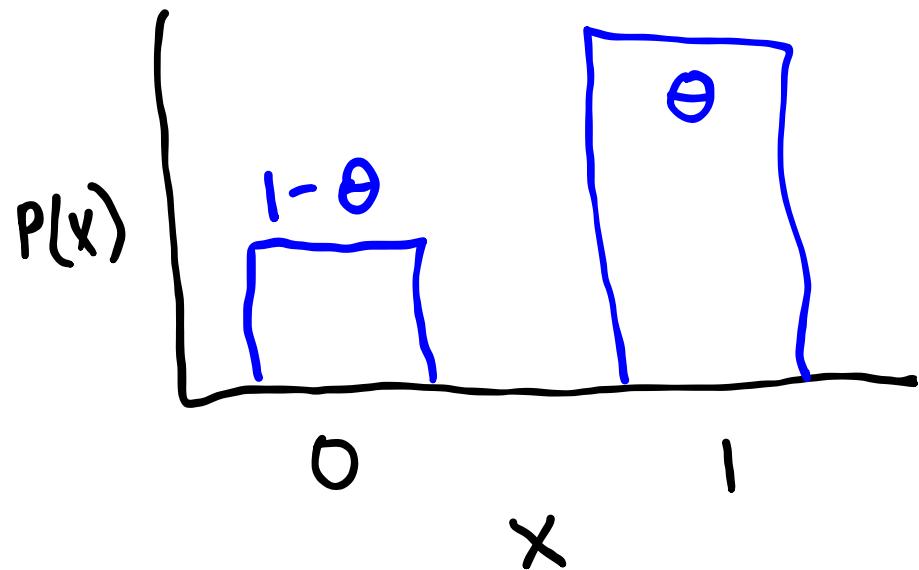
Bernoulli Distribution

$$\text{Bernoulli}(\theta)$$

Beta Distribution

(distribution over Bernoulli distributions)

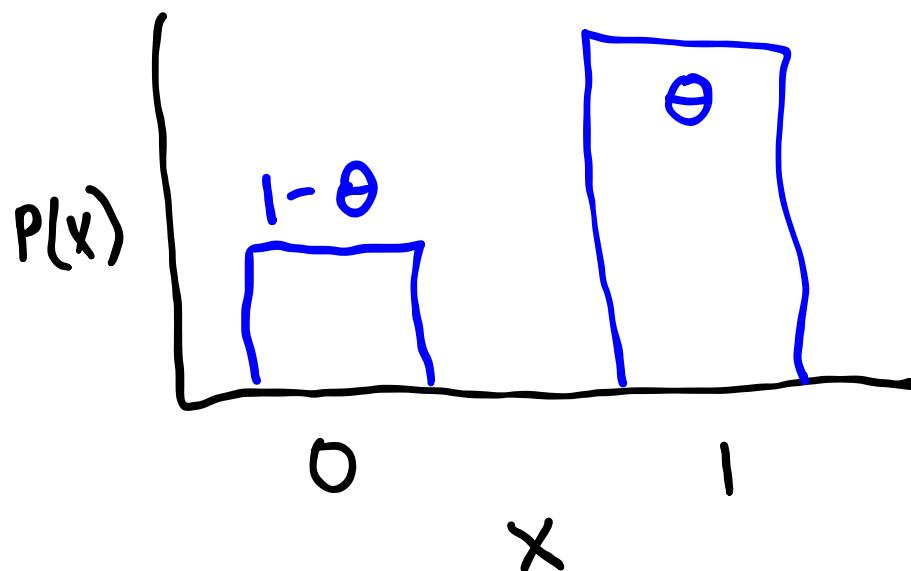
$$\text{Beta}(\alpha, \beta)$$



Bayesian Estimation

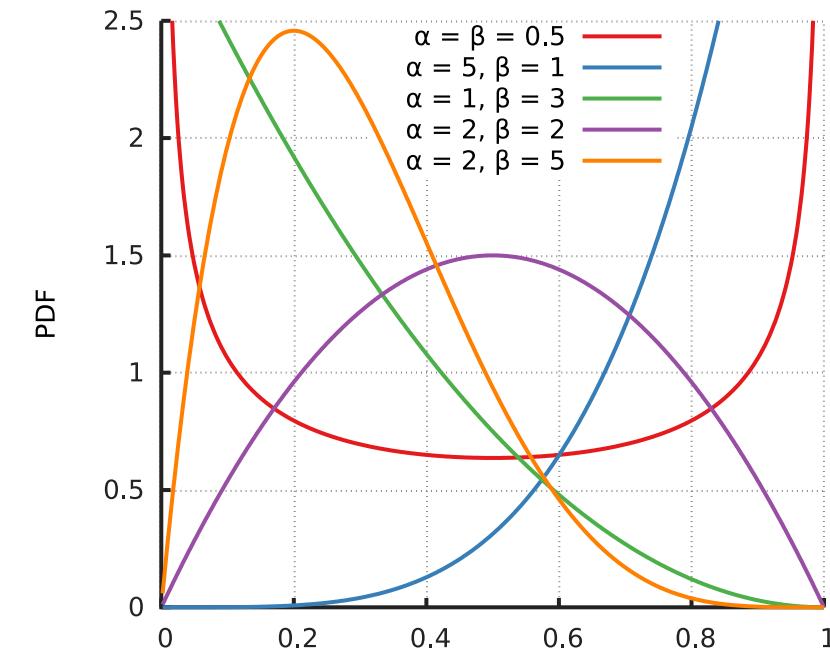
Bernoulli Distribution

$\text{Bernoulli}(\theta)$



Beta Distribution
(distribution over Bernoulli distributions)

$\text{Beta}(\alpha, \beta)$



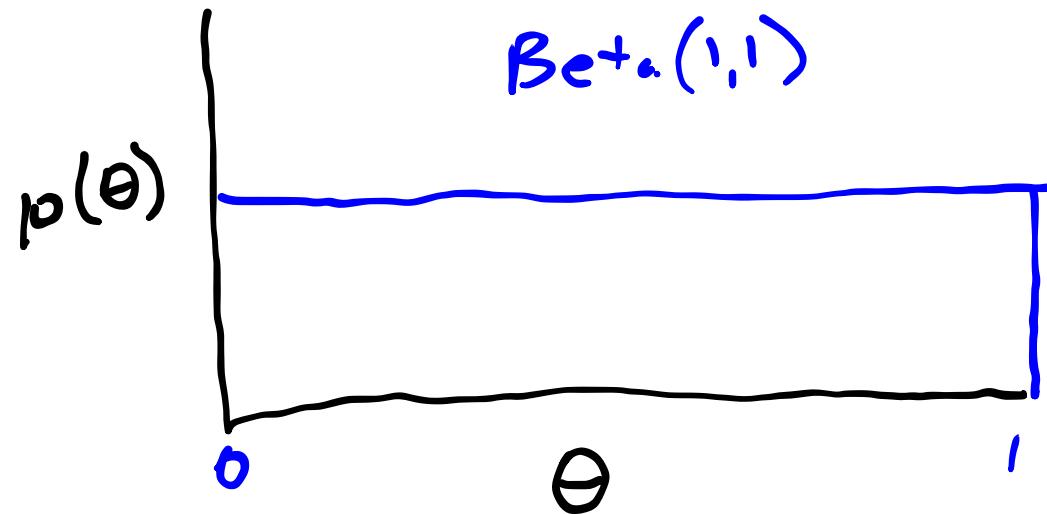
Bayesian Estimation

Bayesian Estimation

Given a $\text{Beta}(1, 1)$ prior distribution

Bayesian Estimation

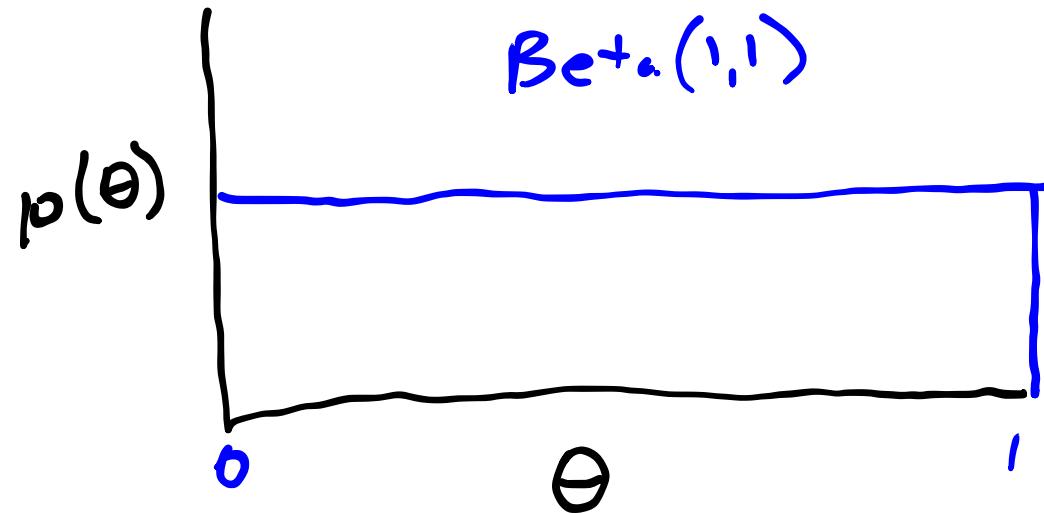
Given a Beta(1, 1) prior distribution



Bayesian Estimation

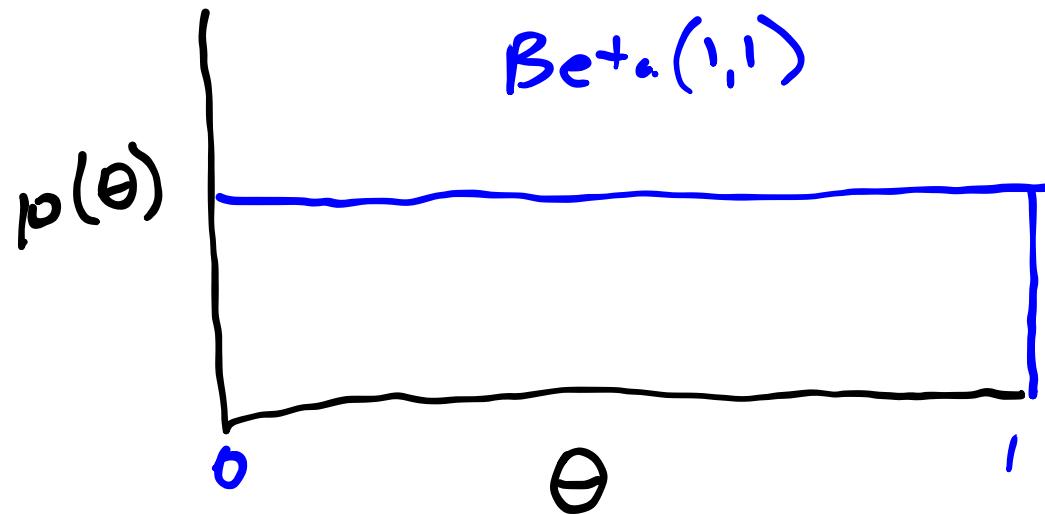
Given a $\text{Beta}(1, 1)$ prior distribution

The posterior distribution of θ is
 $\text{Beta}(w + 1, l + 1)$

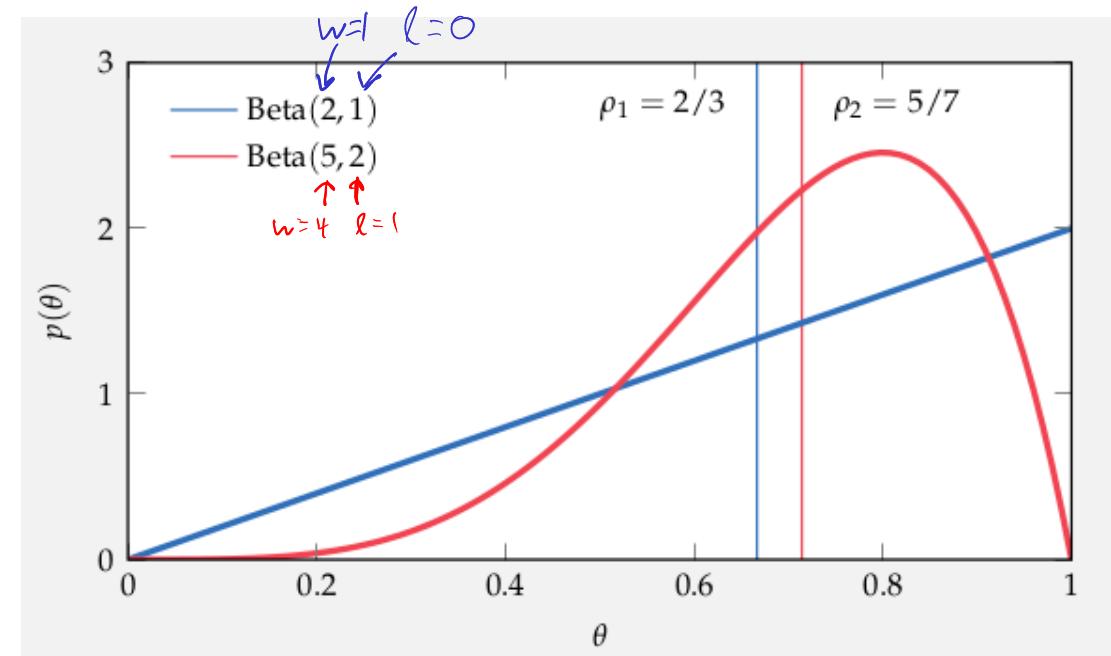


Bayesian Estimation

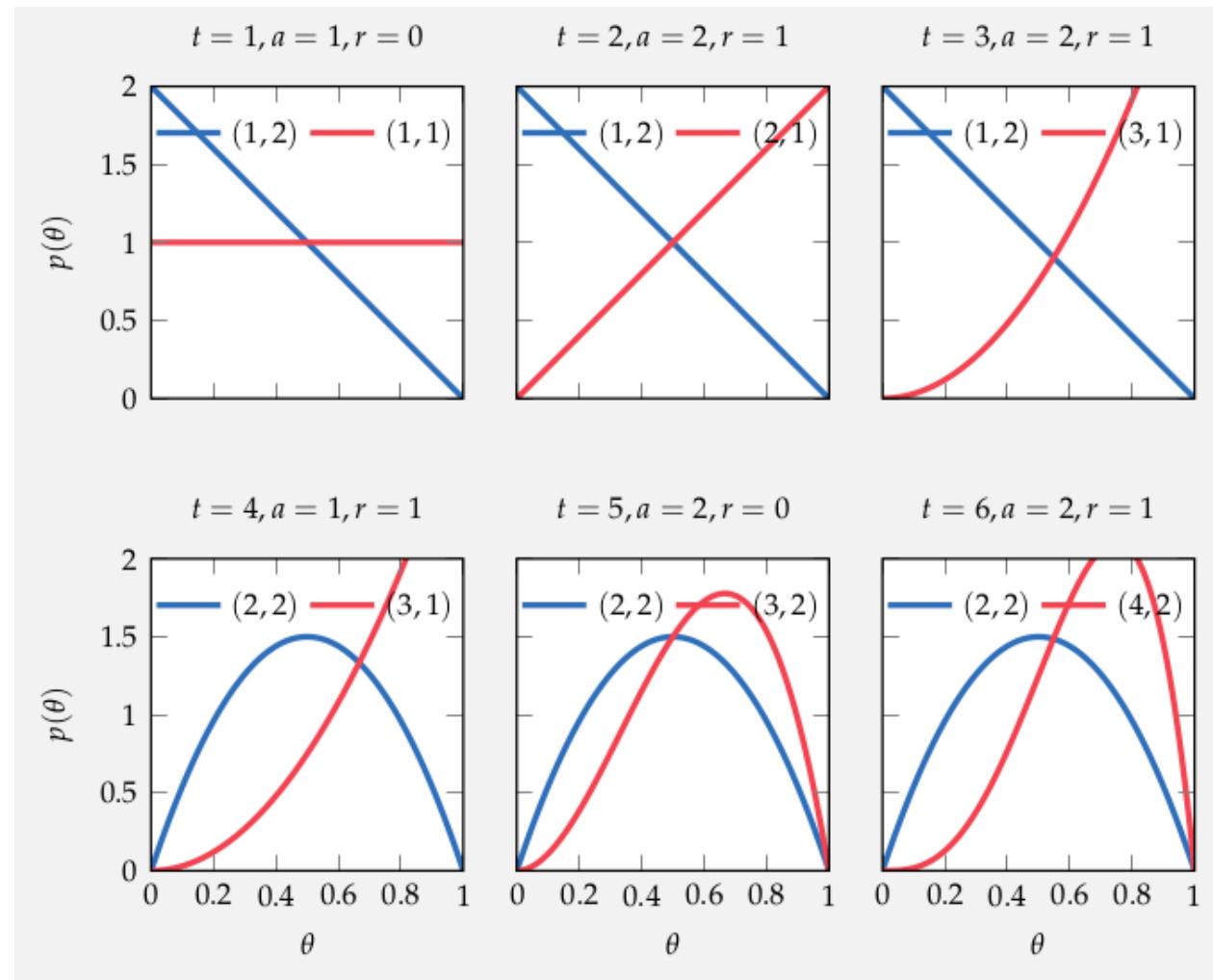
Given a $\text{Beta}(1, 1)$ prior distribution



The posterior distribution of θ is
 $\text{Beta}(w + 1, l + 1)$



Bayesian Estimation



t = time

a = arm pulled

r = reward

Bayesian Bandit Algorithms

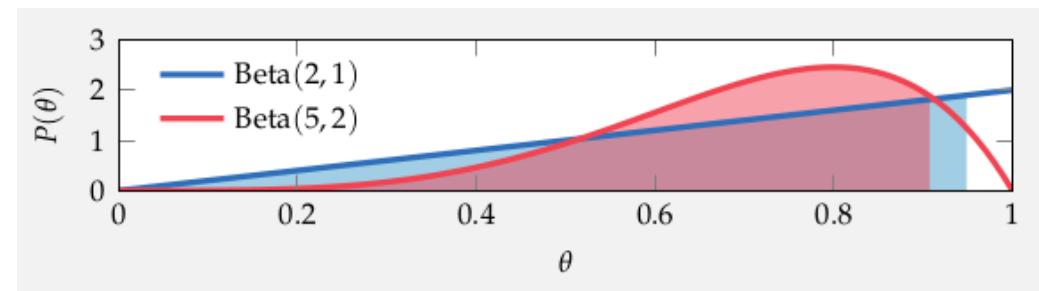
Bayesian Bandit Algorithms

- Quantile Selection
Choose a for which the α quantile of
 $b(\theta)$ is highest

Bayesian Bandit Algorithms

- Quantile Selection
Choose a for which the α quantile of $b(\theta)$ is highest

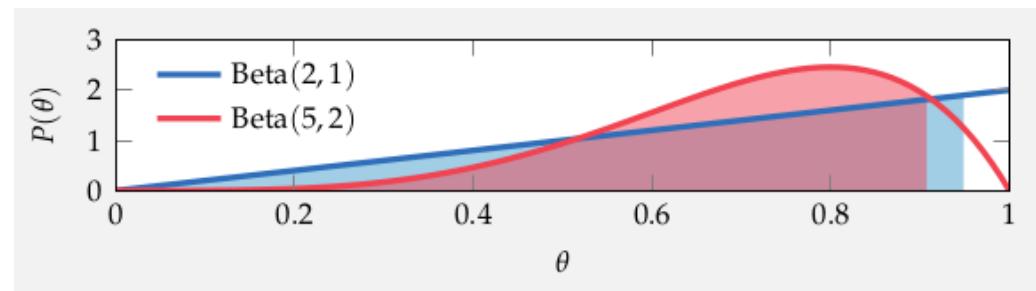
$$\alpha = 0.9$$



Bayesian Bandit Algorithms

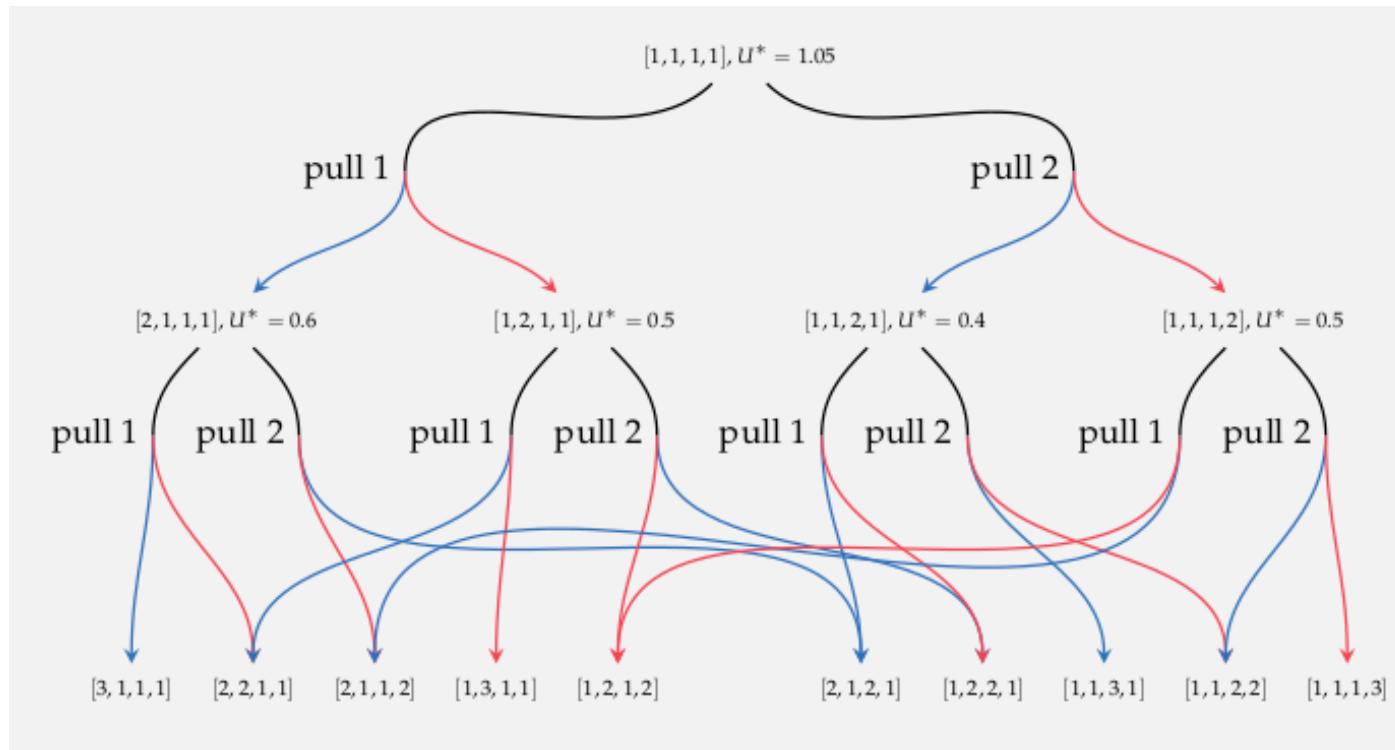
- Quantile Selection
Choose a for which the α quantile of $b(\theta)$ is highest

$$\alpha = 0.9$$



- Thompson Sampling
Sample $\hat{\theta}$
Choose $\operatorname{argmax}_a \hat{\theta}_a$

Optimal Algorithm - Dynamic Programming



Regret Analysis

Roughly:

Regret Analysis

$$\text{Regret}(n) \equiv \theta^* n - \sum_{t=1}^n r_t$$

Roughly:

Regret Analysis

$$\text{Regret}(n) \equiv \theta^* n - \sum_{t=1}^n r_t$$



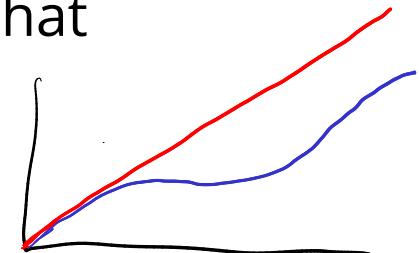
Roughly:

Regret Analysis

$$\text{Regret}(n) \equiv \theta^* n - \sum_{t=1}^n r_t$$



Recall: $f(n) = O(g(n))$ means that there exists a $C > 0$ and $N > 0$ such that $f(n) < C g(n)$ for all $n > N$.



Roughly:

Regret Analysis

$$\text{Regret}(n) \equiv \theta^* n - \sum_{t=1}^n r_t$$



Recall: $f(n) = O(g(n))$ means that there exists a $C > 0$ and $N > 0$ such that $f(n) < C g(n)$ for all $n > N$.

Roughly:

- $O(n)$ regret means you might keep picking the wrong arm forever

Regret Analysis

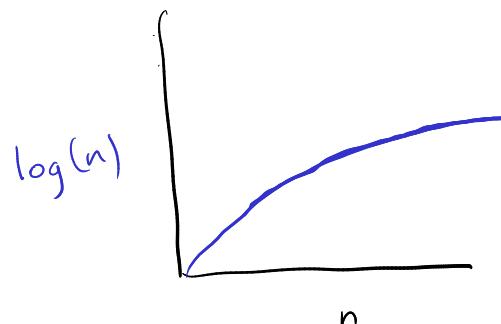
$$\text{Regret}(n) \equiv \theta^* n - \sum_{t=1}^n r_t$$



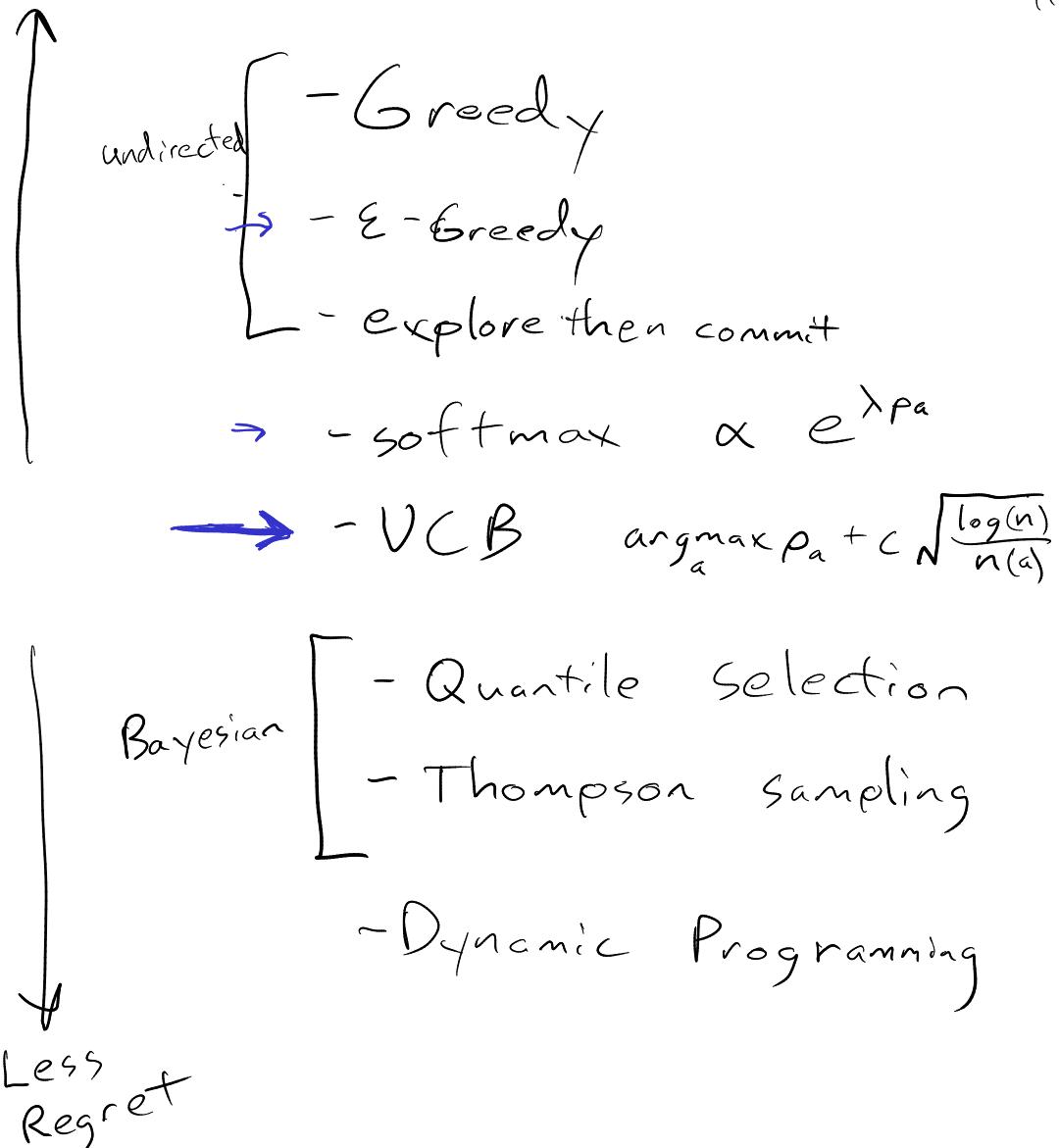
Recall: $f(n) = O(g(n))$ means that there exists a $C > 0$ and $N > 0$ such that $f(n) < C g(n)$ for all $n > N$.

Roughly:

- $O(n)$ regret means you might keep picking the wrong arm forever
- $O(\log(n))$ regret means that you keep learning



Easier to implement



Review

"Optimal in limit"
(parameter)

No

$\epsilon \rightarrow 0$

$K \rightarrow \infty$

$\lambda \rightarrow \infty$



Regret

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(\log(n))$



$O(\log(ns))$

$O(\log(n))$

(only for finite horizon)

Guiding Questions

- What are the best ways to trade off Exploration and Exploitation