

# Module 8

# Multi-armed bandits II

DAV-6300-1: Experimental Optimization

# Review: Randomization

- A/B test: 50/50 between A & B
  - N observations each
- Epsilon-greedy: 90% to best (so far) arm,  $\epsilon=10\%$  to other arms

- Decay  $\epsilon$ :  $\epsilon_n = \frac{kc(BM_0/PS)^2}{n} \iff \epsilon_n \sim 1/n$

- Stop when  $\epsilon$  is small,  $\epsilon < \epsilon_{\text{stop}}$

# Key Terms

- Allocation
- Meta-parameters
- Thompson sampling
- Exploration vs. exploitation

# Meta-parameters

- A/B test: FPR, FNR limits, 5% and 20%
- Epsilon-greedy:  $c, \epsilon < \epsilon_{\text{stop}}$
- *Meta-parameters* determine how the experimental method operates
- Contrast with
  - *Parameters*: Configure the system or model
  - *Hyperparameters*: Regularize a model

Both tuned by  
experimental  
methods

# Meta-parameters

- Meta-parameters too hard to tune
  - Would require rerunning experiments many times
  - If we could afford to do that, we wouldn't need experimental methods.
- Prefer methods with
  - Fewer meta-parameters
  - “Easy” meta-parameters (e.g., understandable, robust)

# Case: Two HFT strategies

- Two strategies, A and B; 100 stocks
- $PS = \$100$
- $\hat{\sigma}_\delta = \$125$
- $N = \left(\frac{2.5\hat{\sigma}_\delta}{PS}\right)^2 = \left(\frac{2.5 \times \$125}{\$100}\right)^2 = 10$  days
- Randomize: Allocate 50 stocks to A, 50 stocks to B
- Run A/B test for two weeks

# Case: Two HFT strategies

- At end, regret:
  - “If we’d run A the whole time we’d have made more money.”, or
  - “If we’d run B the whole time we’d have made more money.”
- \*Really\* want to decide earlier
  - But early stopping increases false positives
- Solution: “Ease into it”

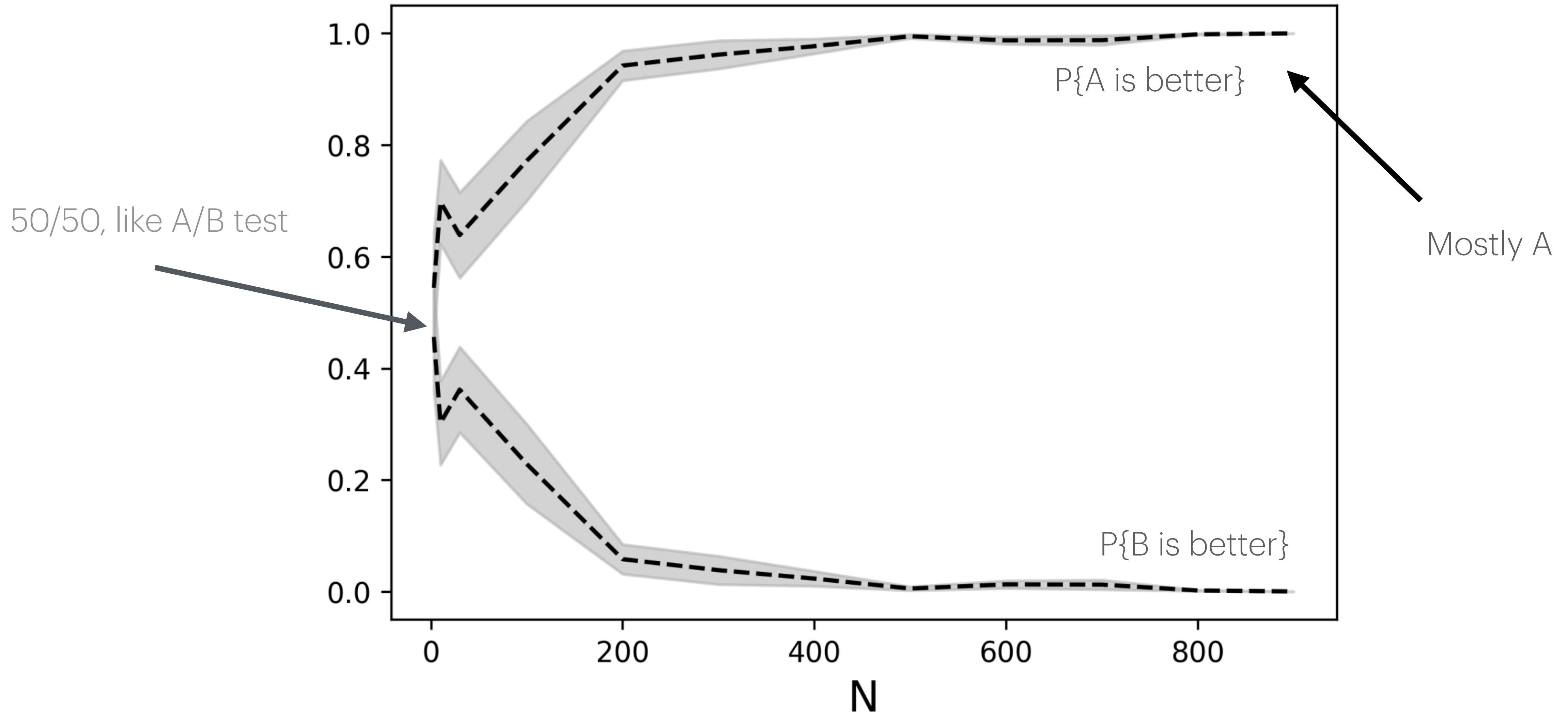
# Dynamic Allocation

- Each day: Calculate  $\mu_a, se_a, \mu_b, se_b$  **so far**
- Estimate probability that A is better than B
  - $p_a = P\{\text{A better}\} = P\{E[y_a] > E[y_b]\}$
  - $p_b = 1 - p_a$
- Allocate  $100 \times p_a$  stocks to A,  $100 \times p_b$  stocks to B
- Start:  $p_a = p_b = 0.5$  ←

50/50, like A/B test



# Dynamic Allocation



# Dynamic Allocation

- Start at 50/50 and gradually transition towards better strategy
  - Spend more time trading better strategy
  - ==> Higher pnl during experiment
  - **Reduces experimentation cost**
- All experimental methods invented to reduce experimentation cost

# Probability A is better

- Recall:  $\mu_a \sim \mathcal{N}(E[y_a], \sigma_a^2/N)$  and  $\mu_b \sim \mathcal{N}(E[y_b], \sigma_b^2/N)$
- Approximate by  $\mathcal{N}(\mu_a, se_a)$  and  $\mathcal{N}(\mu_b, se_b)$
- Draw 10,000 samples from each dist:  $m_{a,b} \sim \mathcal{N}(\mu_{a,b}, se_{a,b})$
- Estimate:  $p_a = \frac{\# \text{ times } m_a > m_b}{10,000}$  and  $p_b = \frac{\# \text{ times } m_b > m_a}{10,000} = 1 - p_a$
- Allocate  $100 \times p_a$  to A,  $100 \times p_b$  to B

# Probability A is better

- Draw 10,000 samples from each from  $m_{a,b} \sim \mathcal{N}(\mu_{a,b}, se_{a,b})$
- Estimate:  $p_a = \frac{\# \text{ times } m_a > m_b}{10,000}$  and  $p_b = \frac{\# \text{ times } m_b > m_a}{10,000} = 1 - p_a$

```
m_a = mu_a + se_a*np.random.normal(size=(10000,))
m_b = mu_b + se_b*np.random.normal(size=(10000,))

p_a = (m_a > m_b).mean()
p_b = 1 - p_a
p_b = (m_b > m_a).mean() # same
```

# Probability an arm is best

- Works for multiple arms, too; A/B/C/...

- $m_a \sim \mathcal{N}(\mu_a, se_a)$ ,

- $$p_a = \frac{\text{\# times } m_a > m_{a'} \forall a' \neq a}{10,000}$$

```
m_a = mu_a + se_a*np.random.normal(size=(10000,))
m_b = mu_b + se_b*np.random.normal(size=(10000,))
m_c = mu_c + se_c*np.random.normal(size=(10000,))

mx = np.maximum(np.maximum(m_a, m_b), m_c)
p_a = (m_a >= mx).mean()
p_b = (m_b >= mx).mean()
p_c = (m_c >= mx).mean()
```

# One weird trick: Thompson sampling

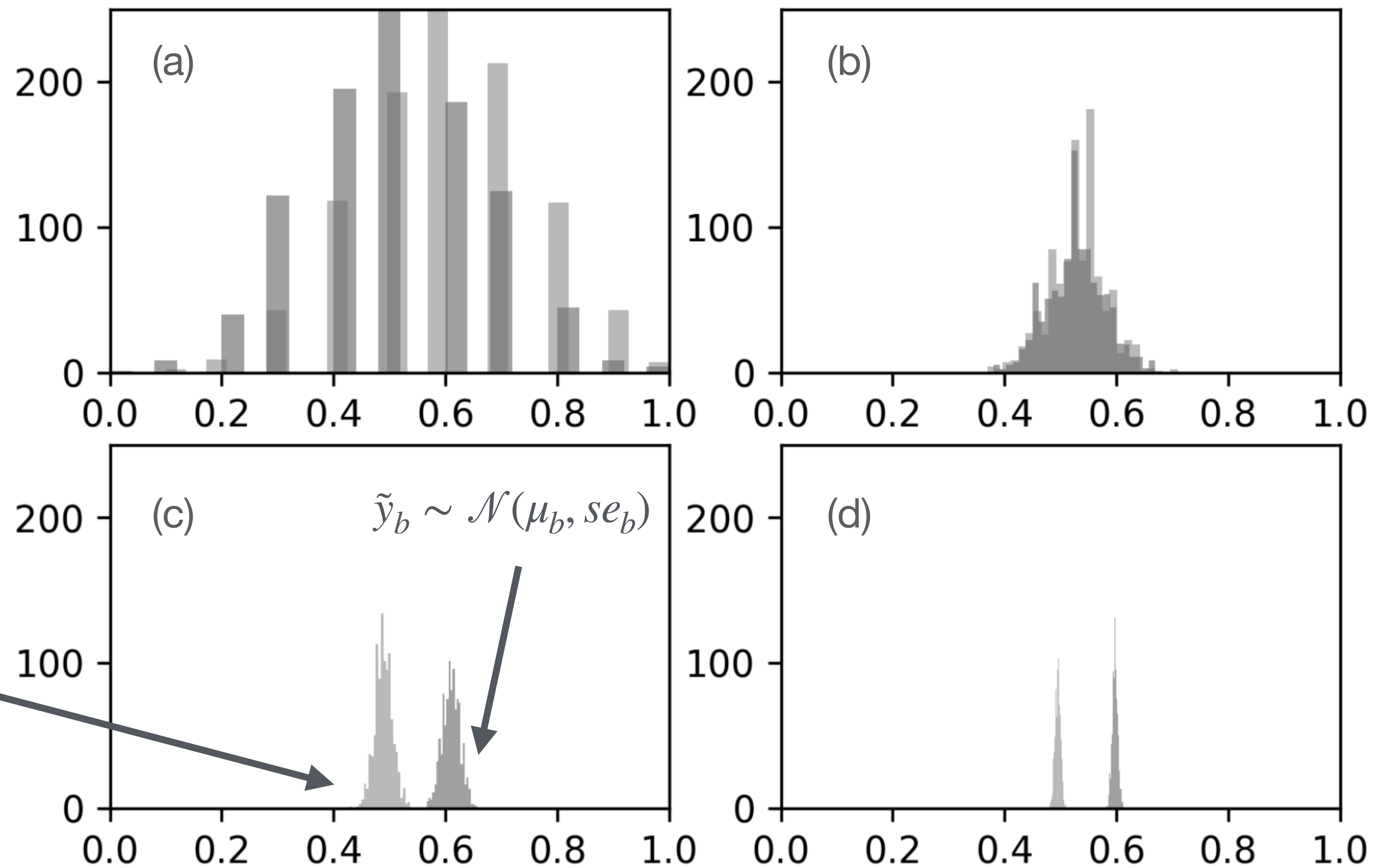
- Draw **one** sample from each of  $m_a \sim \mathcal{N}(\mu_a, se_a)$  and  $m_b \sim \mathcal{N}(\mu_b, se_b)$
- $P\{a>b\} = p_a$ , by definition
- Assign each stock by a single pair of draws
- Called *Thompson Sampling*

```
for i_stock in range(100):  
    m_a = mu_a + se_a*np.random.normal(size=(1,))  
    m_b = mu_b + se_b*np.random.normal(size=(1,))  
    if m_a > m_b:  
        pass # Assign stock to A  
    else:  
        pass # Assign stock to B
```

# Thompson sampling

## Exploration vs. exploitation

- Gets easier to tell the distributions apart as # observations grows
- (a)  $\rightarrow$  (d) increasing N



# Thompson sampling with three arms

[http://andamooka.org/~dsweet/EOCourse/Thompson\\_Sampling.mov](http://andamooka.org/~dsweet/EOCourse/Thompson_Sampling.mov)



# Thompson Sampling: Stopping

- Stop when  $\max\{p_a\} > 0.95$ 
  - When the largest  $p_a$  is very large,  $p_a^* > p_{\text{stop}}$
- Stopping criterion easy to determine
  - $p_a$  more intuitive than  $t$

“Users” less inclined to misunderstand  
and stop early

# Recap: Thompson Sampling

- Method:
  - **Draw once** from each arm's model of BM:  $\{m_a \sim \mathcal{N}(\mu_a, se_a)\}$
  - Run arm  $a^* = \arg \max_a \{m_a\}$
  - Allocate  $\propto p_a$
  - Stop when  $p_{a^*} > p_{stop}$

# Recap: Thompson Sampling

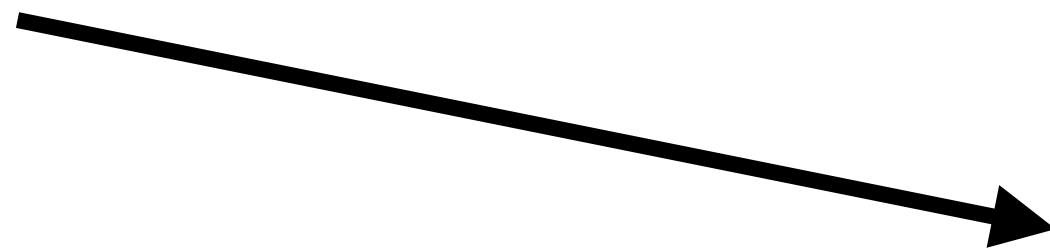
- Advantages:
  - Only one meta-parameter,  $p_{stop}$  — and it's "easy"
  - Increases BM **while experimenting**, reducing experimentation cost
  - Fast: One sample / arm / allocation
    - (Well, faster than estimating  $p_a$  for all arms)

No slower than A/B testing  
or  $\epsilon$ -greedy

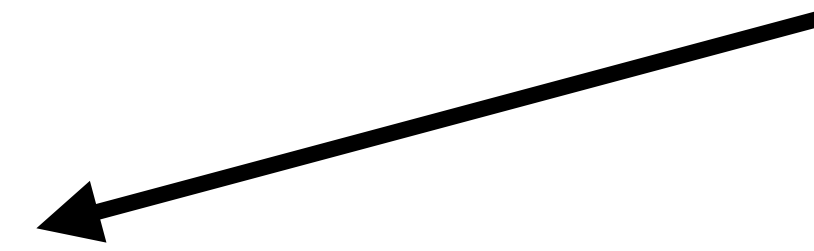
# One-Armed Bandit



No arm here



Only one arm



“Bandit” == Steals  
your money

This is a  
slot machine

# Multi-Armed Bandit Problem

- “Multi-Armed” Bandit ==> A slot machine with multiple arms?  $\_ (ツ) \_ /$
- Instead, imagine multiple slot machines
  - Each with one arm
  - Payout/*reward* distribution differs from machine to machine
  - May pull one arm at a time
- Why we call A/B/C... “arms”

# Multi-Armed Bandit (MAB) Problem

- Goal
  1. Identify the machine with the highest reward
  2. Maximize the cumulative reward
- In MAB terms
  1. Minimize *instantaneous regret*
  2. Minimize *cumulative regret*

# Multi-Armed Bandit (MAB) Problem

- Solution methods
  - Epsilon-greedy
  - Thompson sampling
- Also, UCB:
  - Run arm with largest  $\mu_a + se_a$
  - “Optimism under uncertainty”

# Multi-Armed Bandit Problem

“Originally considered by Allied scientists in World War II, it proved so intractable that, according to Peter Whittle, the problem was proposed to be dropped over Germany so that German scientists could also waste their time on it.”

[https://en.wikipedia.org/wiki/Multi-armed\\_bandit](https://en.wikipedia.org/wiki/Multi-armed_bandit)

On the Likelihood that One Unknown Probability Exceeds Another  
in View of the Evidence of Two Samples

William Thompson, 1933

<https://www.jstor.org/stable/2332286>

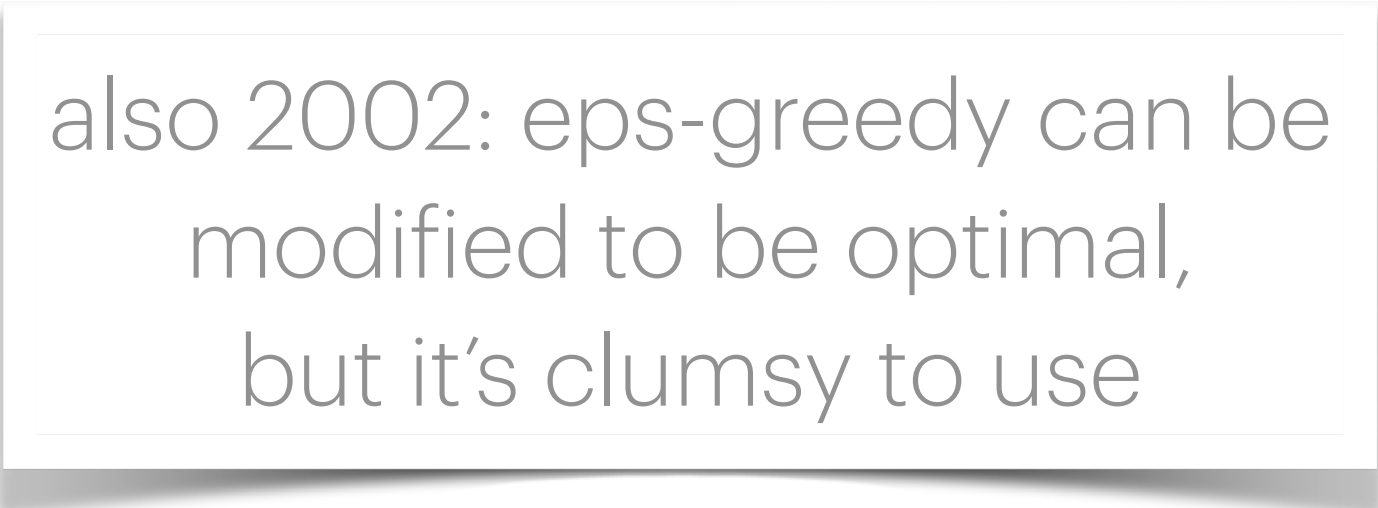


# History of Thompson Sampling

- 1933: Thompson sampling introduced  
<https://www.jstor.org/stable/2332286>
- 1940's: "MAB is impossibly hard"  
[https://en.wikipedia.org/wiki/Multi-armed\\_bandit](https://en.wikipedia.org/wiki/Multi-armed_bandit)
- 2002: UCB is optimal  
<https://link.springer.com/article/10.1023/a:1013689704352>
- 2012-2017: Thompson sampling is optimal, and better than UCB in practice  
<http://proceedings.mlr.press/v23/agrawal12/agrawal12.pdf>  
<https://arxiv.org/abs/1209.3353>  
<http://www.columbia.edu/~sa3305/papers/j3-corrected.pdf>



Took a while



also 2002: eps-greedy can be modified to be optimal, but it's clumsy to use

# Summary

- MAB methods maximize BM during experiment
  - By adapting to observations
- Thompson sampling is simple, optimal, and empirically performant