

Bayesian Hypothesis Testing

Moses Marsh

Data Science Immersive



Objectives: answer the following

- What is a prior, posterior, and likelihood?
- How do we apply Bayesian updating to A/B testing?
- What does the Beta distribution represent?
- What are some key differences between frequentist and Bayesian A/B testing?

Review: frequentist p-values

- Remember the one-sentence definition of a p-value?

Review: frequentist p-values

- Remember the one-sentence definition of a p-value?

“The probability of observing data at least as extreme as the observation given the null hypothesis”

$$P(\text{data} \mid \text{null distribution})$$

$$P(y \mid \theta_0)$$

Wouldn't it be nice if, instead, we could give a probability of a **parameter** given the **data**?

Wouldn't it be nice...

$$\text{Pr}(\theta|y) = \frac{\text{Pr}(y|\theta)\text{Pr}(\theta)}{\text{Pr}(y)}$$

Diagram illustrating the components of Bayes' Theorem:

- Posterior Probability: $\text{Pr}(\theta|y)$
- Likelihood of Observations: $\text{Pr}(y|\theta)$
- Prior Probability: $\text{Pr}(\theta)$
- Normalizing Constant: $\text{Pr}(y)$

Review: Bayesian Inference

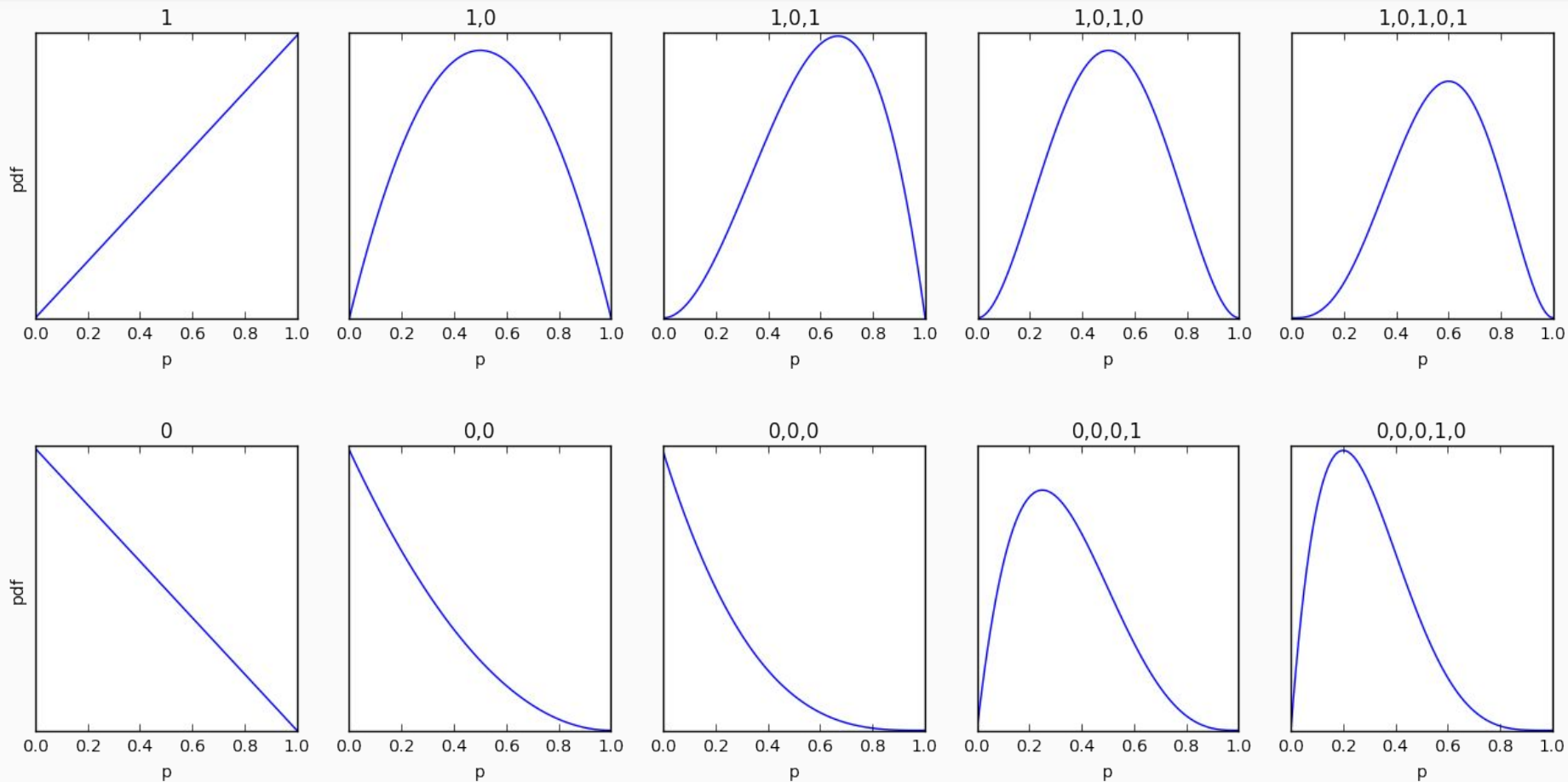
$$\text{Pr}(\theta|y) = \frac{\text{Pr}(y|\theta)\text{Pr}(\theta)}{\text{Pr}(y)}$$

Diagram illustrating the components of the Bayesian Inference formula:

- Posterior Probability**: $\text{Pr}(\theta|y)$
- Likelihood of Observations**: $\text{Pr}(y|\theta)$
- Prior Probability**: $\text{Pr}(\theta)$
- Normalizing Constant**: $\text{Pr}(y)$

- Coin example
 - y is a set of flips (heads or tails)
 - θ is the coin's probability of coming up heads for a single flip

Posteriors from yesterday's coin



Review: Bayesian Inference

$$\text{Pr}(\theta|y) = \frac{\text{Pr}(y|\theta)\text{Pr}(\theta)}{\text{Pr}(y)}$$

Diagram illustrating the components of the Bayesian Inference formula:

- Posterior Probability**: $\text{Pr}(\theta|y)$
- Likelihood of Observations**: $\text{Pr}(y|\theta)$
- Prior Probability**: $\text{Pr}(\theta)$
- Normalizing Constant**: $\text{Pr}(y)$

- Click-through rate
 - y is a set of visits by unique users to a website, each of which either resulted in a click or not
 - θ is the probability of a click for a single visit
 - **Let's work with this example for the rest of the day**

$$Posterior \propto Likelihood \times Prior$$



- We're going to model each of these terms with an appropriate **distribution**
- We'll see that it makes Bayesian updating easy and fun!
- Our goal is to find an analytical form for the **posterior probability distribution** over all the possible values of the **true click-through rate** p

$$\textit{likelihood} = P(y \mid p)$$

- y here represents a whole data set: “ n visits with k clicks”
- p is the probability of a click for a single visitor

What is the form of the likelihood function?

$$\textit{likelihood} = P(y \mid p)$$

- **y** here represents a whole data set: “**n** visits with **k** clicks”

Binomial distribution

$$P(k \mid p; n) = \binom{n}{k} p^k (1 - p)^{n-k}$$

$$Posterior \propto Likelihood \times Prior$$

Binomial

$$\text{prior} = P(p)$$

- We want to pick a distribution for ***p***, so it must be defined over $[0,1]$
- Hmm...

$$\text{prior} = P(p)$$

- We want to pick a distribution for \mathbf{p} , so it must be defined over $[0,1]$
- Let's look at that binomial distribution again:

$$\text{Binomial}(k; n, p) = \binom{n}{k} p^k (1 - p)^{n-k}$$

- Can we make a distribution over \mathbf{p} that has this same form?

$$\text{prior} = P(p)$$

- We want to pick a distribution for \mathbf{p} , so it must be defined over $[0,1]$
- Let's look at that binomial distribution again:

$$\text{Binomial}(k; n, p) = \binom{n}{k} p^k (1 - p)^{n-k}$$

- Can we make a distribution over \mathbf{p} that has this same form?

$$\text{the_moses_distribution}(p; a, b) \sim p^a (1 - p)^b$$

$$\text{prior} = P(p)$$

- We want to pick a distribution for \mathbf{p} , so it must be defined over $[0,1]$
- Let's look at that binomial distribution again:

$$\text{Binomial}(k; n, p) = \binom{n}{k} p^k (1 - p)^{n-k}$$

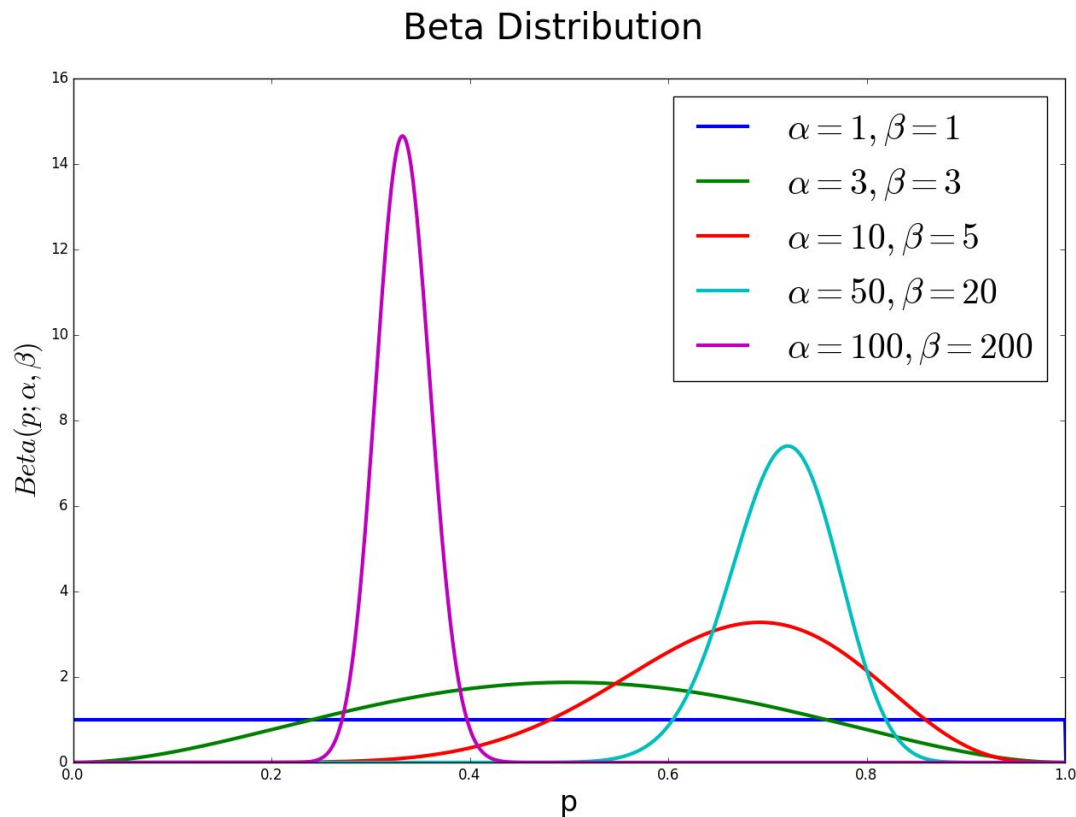
- Can we make a distribution over \mathbf{p} that has this same form?

$$\text{the_moses_distribution}(p; a, b) \sim p^a (1 - p)^b$$

- Oh someone already made this one: the Beta distribution

$$\text{Beta}(p; \alpha, \beta) = \frac{p^{\alpha-1} (1 - p)^{\beta-1}}{B(\alpha, \beta)}$$

Beta distribution



$$E[p] = \frac{\alpha}{\alpha + \beta}$$

$$\text{Mode} = \frac{\alpha - 1}{\alpha + \beta - 1}$$

- Our **prior distribution** is set by our choice of α and β
- $\alpha=\beta=1$ is the uniform distribution

$$Posterior \propto Likelihood \times Prior$$

Binomial

Beta

$$\text{posterior} = P(p \mid y) = P(p \mid n, k)$$

$$\text{posterior} \sim \binom{n}{k} p^k (1-p)^{n-k} \times \frac{p^{\alpha-1} (1-p)^{\beta-1}}{B(\alpha, \beta)}$$

$$\text{posterior} \sim p^k (1-p)^{n-k} \times p^{\alpha-1} (1-p)^{\beta-1}$$

$$\text{posterior} \sim p^{\alpha+k-1} (1-p)^{\beta+n-k-1}$$

$$\text{posterior} = \text{Beta}(p; \alpha + k, \beta + n - k)$$

The posterior is a beta distribution with parameters **$\alpha+k$** and **$\beta+n-k$**

This means we can do all our Bayesian updates at once, instead of updating with one data point at a time!

$$Posterior \propto Likelihood \times Prior$$

Beta

Binomial

Beta

$$Posterior \propto Likelihood \times Prior$$

Beta

Binomial

Beta

- **Conjugate priors** are pairs of distribution families for (likelihood, prior) such that the **posterior** belongs to the same parametric family as the **prior**

Likelihood	Prior	Posterior
Normal	Normal	Normal
Poisson	Gamma	Gamma
Gamma	Gamma	Gamma
Binomial	Beta	Beta
Multinomial	Dirichlet	Dirichlet
Normal	Gamma	Gamma

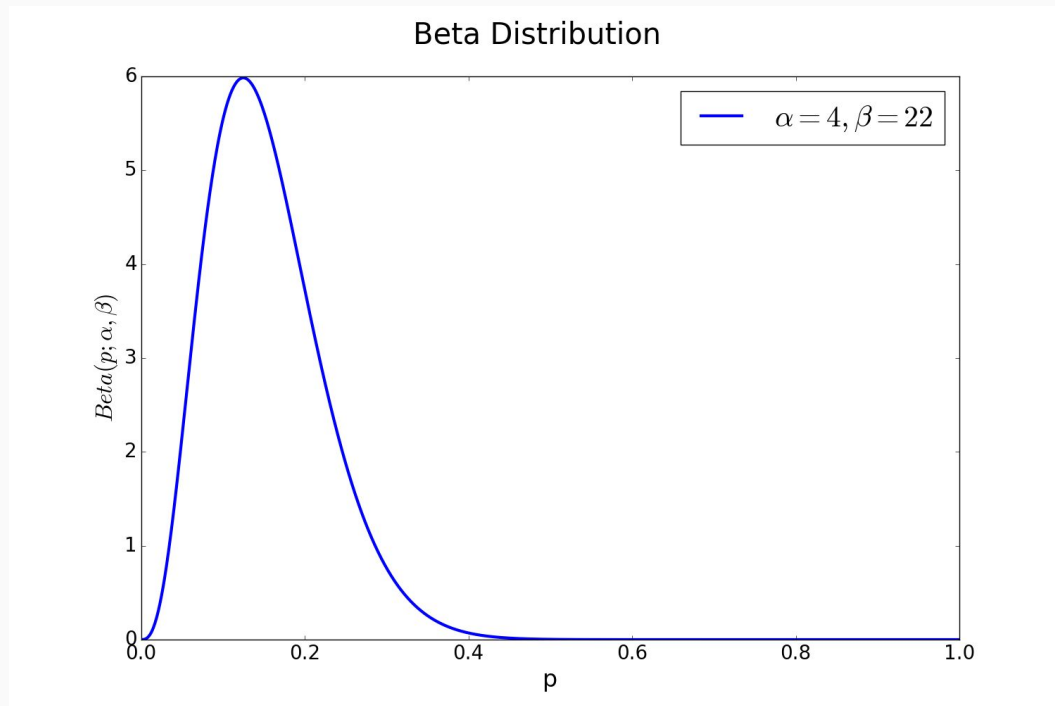
- If you start with the uniform distribution as a prior (which is the beta distribution with $\alpha=\beta=1$) then our posterior is a beta distribution with parameters

$$\alpha = 1 + k = 1 + (\# \text{ of successes})$$

$$\beta = 1 + n - k = 1 + (\# \text{ of failures})$$

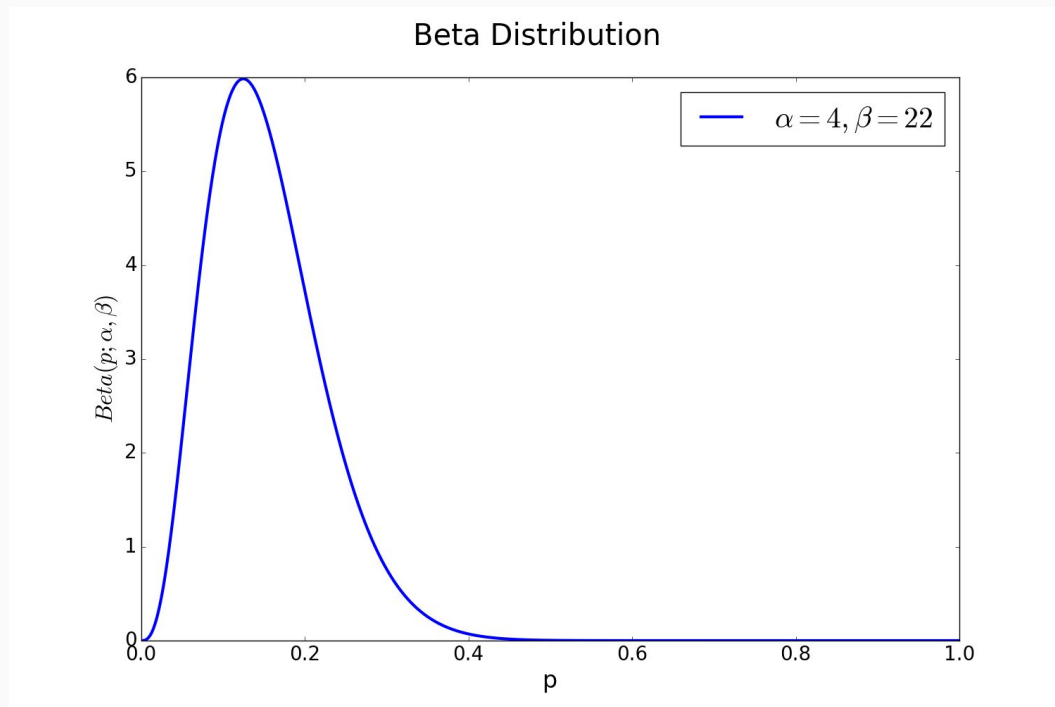
$$\textit{Posterior} = P(p \mid n, k) = \textit{Beta}(p; \alpha, \beta) = \frac{p^{\alpha-1} (1-p)^{\beta-1}}{B(\alpha, \beta)}$$

- For example, if you had 24 trials with 3 successes, you'd have this distribution



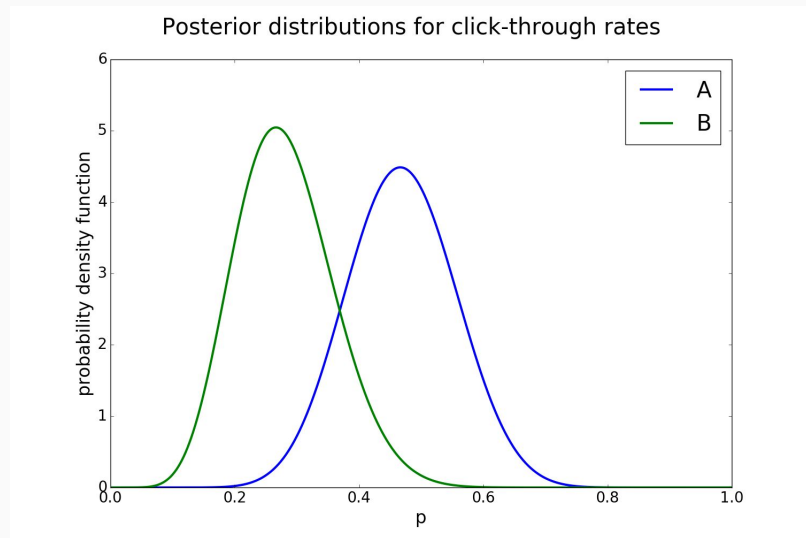
Statements you can make with this distribution

- “The probability that the true CTR is less than 0.15 is 53%”
- “There is a 95% probability that the true CTR lies between 0.045 and 0.312”
 - that’s a **credible interval**



Bayesian A/B Testing

- Randomly send users to two versions of our site (A and B)
- Calculate/update the posterior distributions for each click through rate, p_A and p_B
- Say we end up with the two beta distributions on the right. How would you get the probability that p_A is greater than p_B ?



- We **sample** from each distribution and see how often p_A is greater than p_B

```
# let's draw values from those distribution models
```

```
sample_size = 10000
```

```
# model for A, fed with the right values
```

```
A_sample = stats.beta.rvs(1 + clicks_A,  
                           1 + views_A - clicks_A,  
                           size=sample_size)
```

```
# model for B, fed with the right values
```

```
B_sample = stats.beta.rvs(1 + clicks_B,  
                           1 + views_B - clicks_B,  
                           size=sample_size)
```

```
# let's find out the probability that A is better than B
```

```
print np.mean(A_sample > B_sample)
```

```
# we can also find the probability that  $p_A$  is larger than  $p_B$  by 0.05
```

```
print np.mean(A_sample > (B_sample + 0.05))
```

- Define a metric (e.g., click through rate), null & alternative hypotheses
- Set the study parameters (significance level, power, number of observations)
- Run the test, **wait until it is done**, then analyze results
- Report p-value, confidence interval
- Reject or fail to reject the null hypothesis

- Define a metric (e.g., click through rate)
- Define a prior distribution of the metric
- Run the test, **continually monitoring results**
- **At any time** calculate the probability that $CTR_A > CTR_B$

Multi-Armed Bandit Strategies

Moses Marsh

Data Science Immersive



Objectives: answer the following

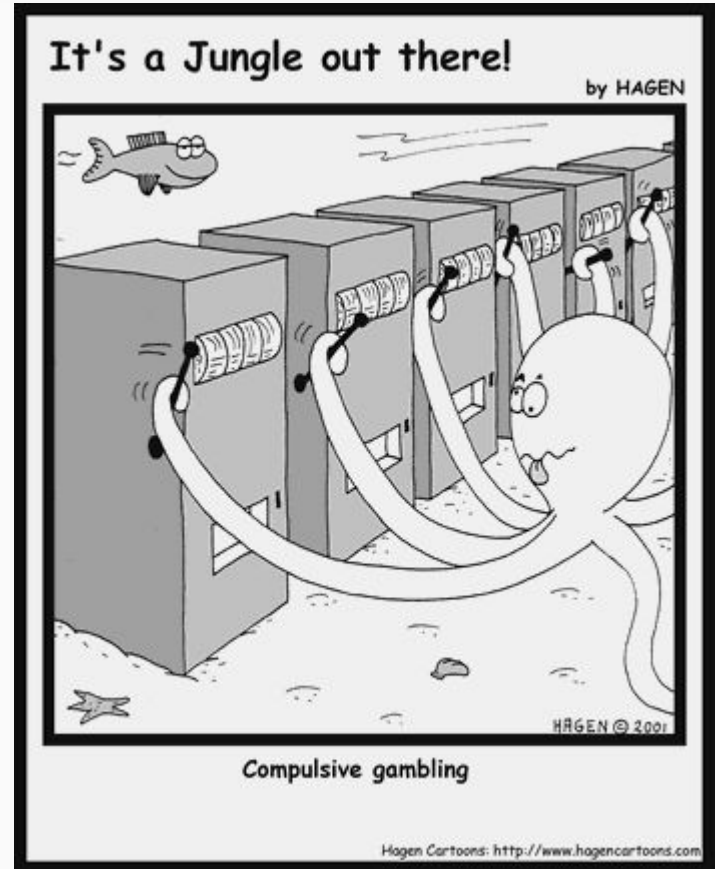
- What are exploitation, exploration, and regret in this context?
- How is this framework related to traditional A/B testing?
- What's your favorite strategy?

Optimizing rewards

- Terminology: a slot machine is called a “one-armed bandit”
- Imagine there are k slot machines (bandits), each with a **probability of payout** for a single pull

$$\{p_1, p_2, p_3, \dots, p_k\}$$

- How do I find out which bandit has the highest probability? What's my best course of action? How do I make the most money from this situation?



- Say you've got k versions of your landing page, each with a click-through probability for a single visit

$$\{p_1, p_2, p_3, \dots, p_k\}$$

- You send users at random to every page (*bandit*), but soon find that some pages are outperforming others. You don't like losing business by sending users to ugly pages (*bandits*), but you may be stuck waiting for statistical significance from your complicated multiple-hypothesis tests.
- Now your job depends on solving this problem.

Exploration vs Exploitation

- **exploration**: collecting more data for each bandit to get a better sense of all the success probabilities
- **exploitation**: using whichever bandit has performed the best so far
- Every strategy for optimization will have to balance exploration and exploitation.

Traditional A/B testing

- Starts with ***pure exploration***: both bandits get the same number of users. This is the testing phase.
- Shifts to ***pure exploitation***: whichever bandit performed better is then shown to all users forever.

The Multi-Armed approach

- Show the best-performing site (*bandit*) **most** of the time (several strategies will be discussed for defining exactly how much time)
- As the experiment runs and users see more sites (*bandits*), update your beliefs about which site is best
- each site (***bandit_i***) will have:
 - a number of visits (*rounds* or *pulls*) n_i
 - a number of successes (*wins*) w_i
 - an observed success rate

$$\hat{p}_i = \frac{w_i}{n_i}$$

- Run until a clear victor emerges

- Dynamic A/B Testing
- Budget allocation among competing projects
- Clinical trials
- Adaptive routing to minimize network delays
- Reinforcement learning (ask Cary)

- We quantify our failure to pick the best bandit with **regret**: the difference between the maximum expected reward (if we had picked the best bandit every time) and the expected reward of all the bandits we actually picked
- For each round (*user*), we pick a bandit and observe whether or not it resulted in a success
- Let p^* be the max of $\{p_1, p_2, p_3, \dots, p_k\}$
- Let $p_{(t)}$ be the true success probability of the bandit chosen at time t
- Then our **regret** after T rounds is

$$r = Tp^* - \sum_{t=1}^T p_{(t)}$$

$$r = Tp^* - \sum_{t=1}^T p_{(t)}$$

- We want a strategy that minimizes regret
- A **zero-regret strategy** is defined as one who's average regret per round, r/T , goes to **zero** in the limit where the number of rounds **T** goes to **infinity**.
- The interesting thing is that a zero-regret strategy does **not** guarantee that you will never choose a suboptimal outcome.
- Instead it guarantees that as you continue to play you will tend to choose the optimal outcome.
- Note that actually calculating regret requires knowing the true bandit probabilities

- **explore** with some fixed probability ϵ , usually 10% or less
- **exploit** at all other times: choose the bandit that has the highest observed success rate so far
- for each bandit, update \hat{p}_i after each round

Is this a zero-regret strategy?

- At round t , choose the bandit that maximizes the following expression:

$$\hat{p}_i + \sqrt{\frac{2 \ln(t)}{n_i}}$$

n_i = number of rounds played on bandit i

$\hat{p}_i = \frac{w_i}{n_i}$, w_i = number of successes for bandit i

t = total number of rounds played so far

- Here we create a **probability** of choosing a bandit according to the following formula

$$P(\text{choosing bandit } i) = \frac{e^{\hat{p}_i / \tau}}{\sum_{j=1}^k e^{\hat{p}_j / \tau}}$$

τ = “temperature” or “randomness” parameter, usually around 0.001

- You then choose a bandit by sampling from this probability distribution

- Use Bayesian updating to make a beta distribution for each bandit, where

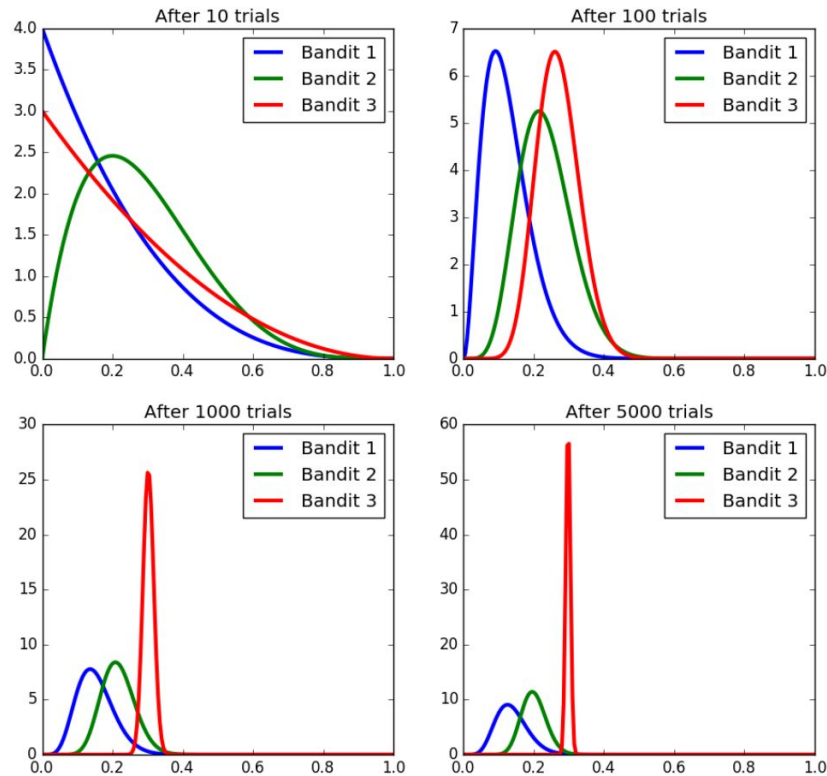
$$\alpha = 1 + (\# \text{ of times that bandit has won})$$

$$\beta = 1 + (\# \text{ of times that bandit has lost})$$

- Then sample from each bandit's posterior distribution and play the one that gave you the highest probability

Bayesian bandits: simulation

"True" bandit reward rates: 0.1, 0.2, 0.3



Why multi-armed-bandit?

In pairs discuss the following:

- What advantage does multi armed bandit give us over standard a/b testing?
- Why doesn't everyone use multi armed bandits?