Bayesian Hypothesis Testing

Moses Marsh

Data Science Immersive
galvanize

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...

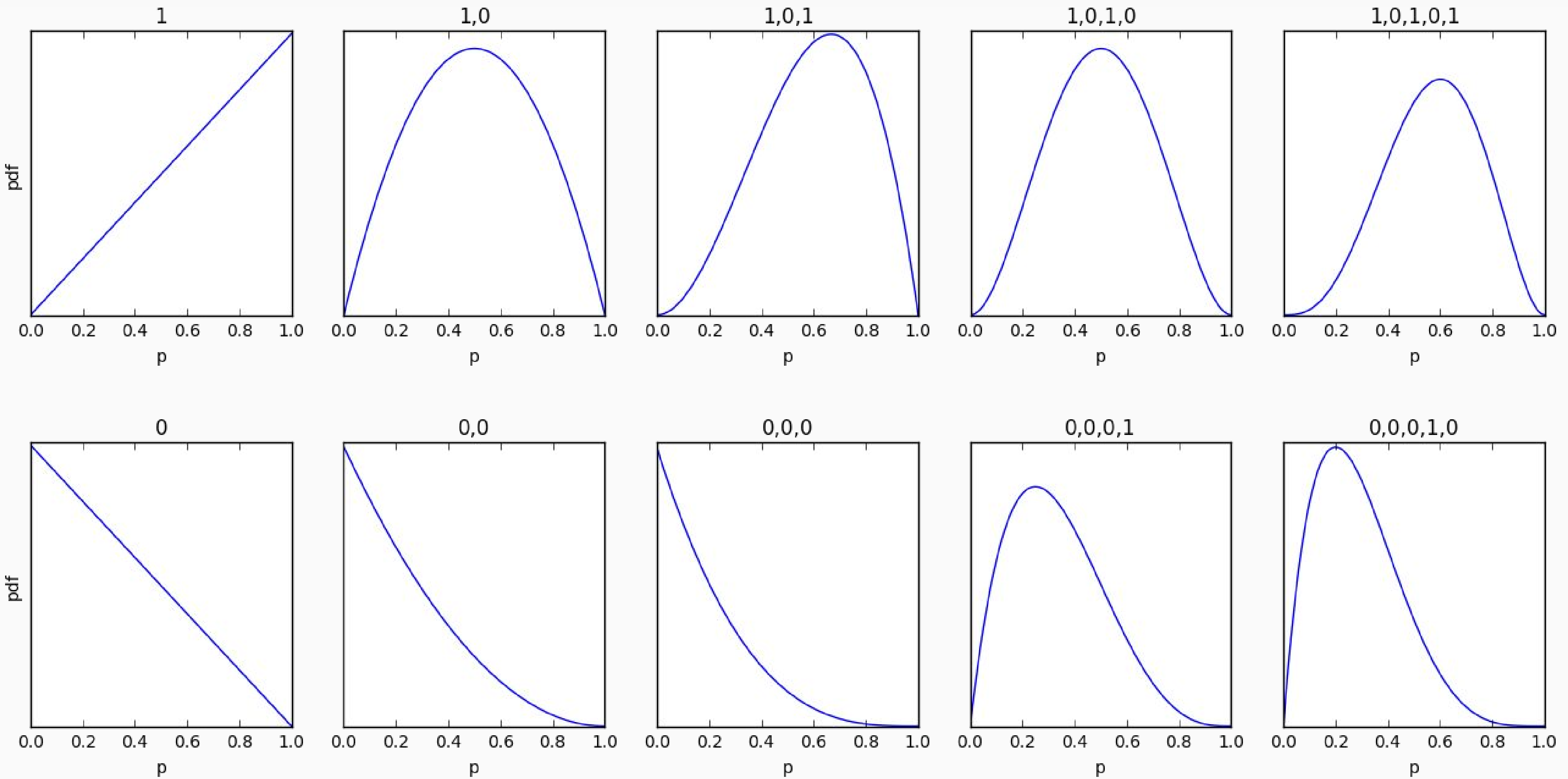
$$\overbrace{\Pr(\theta|y)}^{\text{Posterior Probability}} = \frac{\overbrace{\Pr(y|\theta)}^{\text{Likelihood of Observations}}\overbrace{\Pr(\theta)}^{\text{Prior Probability}}}{\underbrace{\Pr(y)}_{\text{Normalizing Constant}}}$$

Review: Bayesian Inference

$$\underbrace{\Pr(\theta|y)}_{\text{Posterior Probability}} = \frac{\underbrace{\Pr(y|\theta)}_{\text{Likelihood of Observations}}\underbrace{\Pr(\theta)}_{\text{Prior Probability}}}{\underbrace{\Pr(y)}_{\text{Normalizing Constant}}}$$

- 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

Posterior Probability   Likelihood of Observations   Prior Probability

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

Normalizing Constant

- 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**

Bayesian Inference: Distributions

$$Posterior \propto Likelihood \times Prior$$

- 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

Likelihood function

$$likelihood = P(y \mid p)$$

- ***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?

Likelihood function

$$likelihood = P(y \mid p)$$

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

Binomial distribution

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

Bayesian Inference

$$Posterior \propto Likelihood \times Prior$$

	Binomial	

Prior distribution

$$prior = P(p)$$

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

Prior distribution

$$prior = P(p)$$

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

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

- Can we make a distribution over ***p*** that has this same form?

Prior distribution

$$prior = P(p)$$

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

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

- Can we make a distribution over ***p*** that has this same form?

$$the\_moses\_distribution(p; a, b) \sim p^a (1-p)^b$$

Prior distribution

$$prior = P(p)$$

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

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

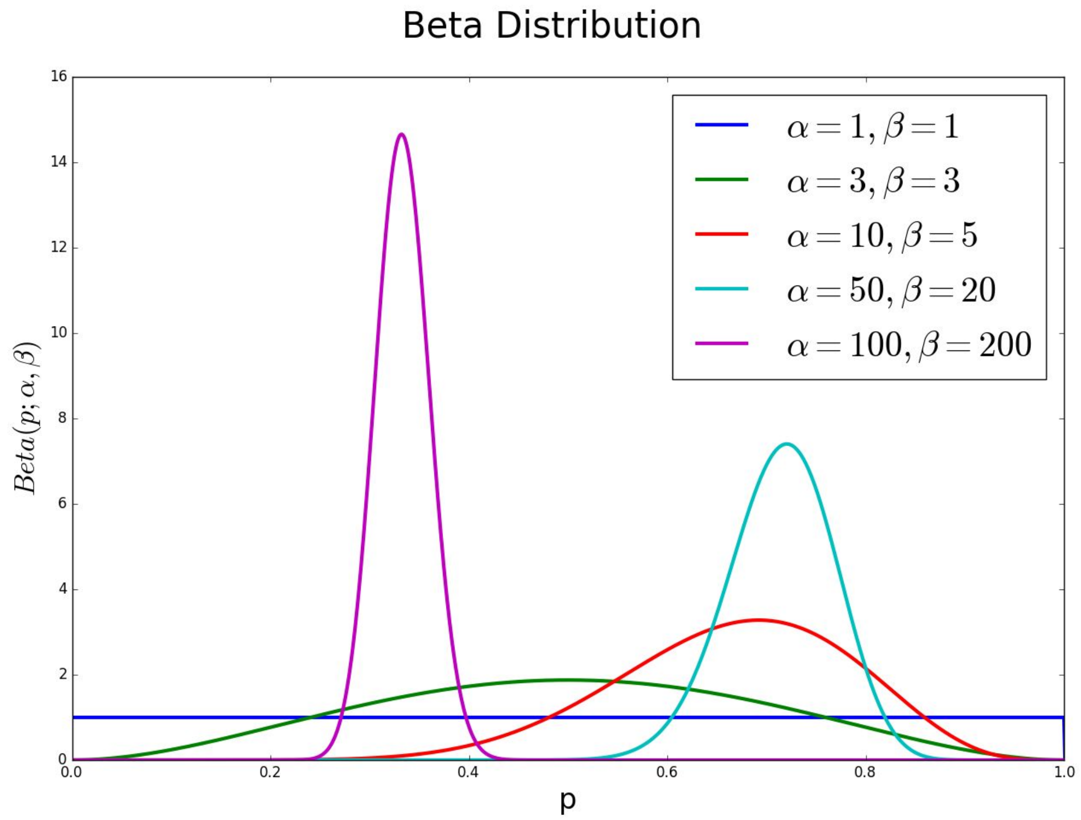
- Can we make a distribution over ***p*** that has this same form?

$$the\_moses\_distribution(p; a, b) \sim p^a (1-p)^b$$

- Oh someone already made this one: the Beta distribution

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

Beta distribution



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

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

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

Bayesian Inference

galvanize

$$Posterior \propto Likelihood \times Prior$$

	Binomial	Beta

Posterior

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

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

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

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

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

The posterior is a beta distribution with parameters ***α+k*** and ***β+n-k***

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

Bayesian Inference

$$Posterior \propto Likelihood \times Prior$$

$\boxed{\text{Beta}}$ $\boxed{\text{Binomial}}$ $\boxed{\text{Beta}}$

Conjugate priors

$$Posterior \propto Likelihood \times Prior$$

$\boxed{\text{Beta}}$ $\boxed{\text{Binomial}}$ $\boxed{\text{Beta}}$

- ***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

In summary: ta-da! we have an easy Posterior

galvanize

- If you start with the uniform distribution as a prior (which is the beta distribution with ***α*=*β*=1**) then our posterior is a beta distribution with parameters

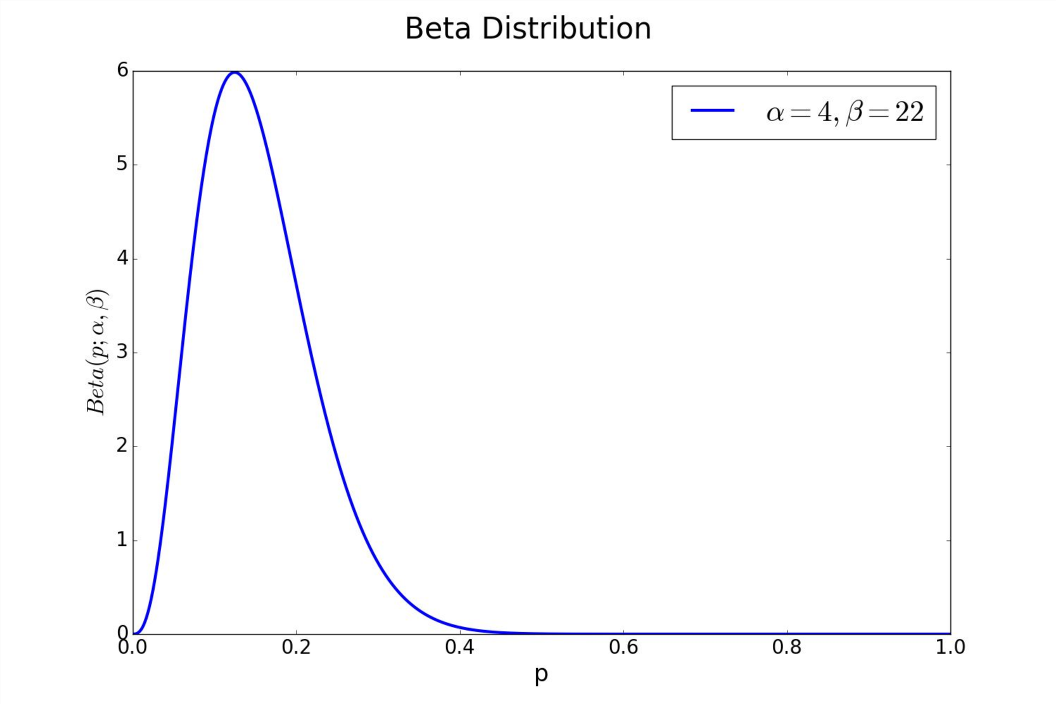
$$\alpha = 1 + k = 1 + (\#\text{ of successes})$$

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

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

Example

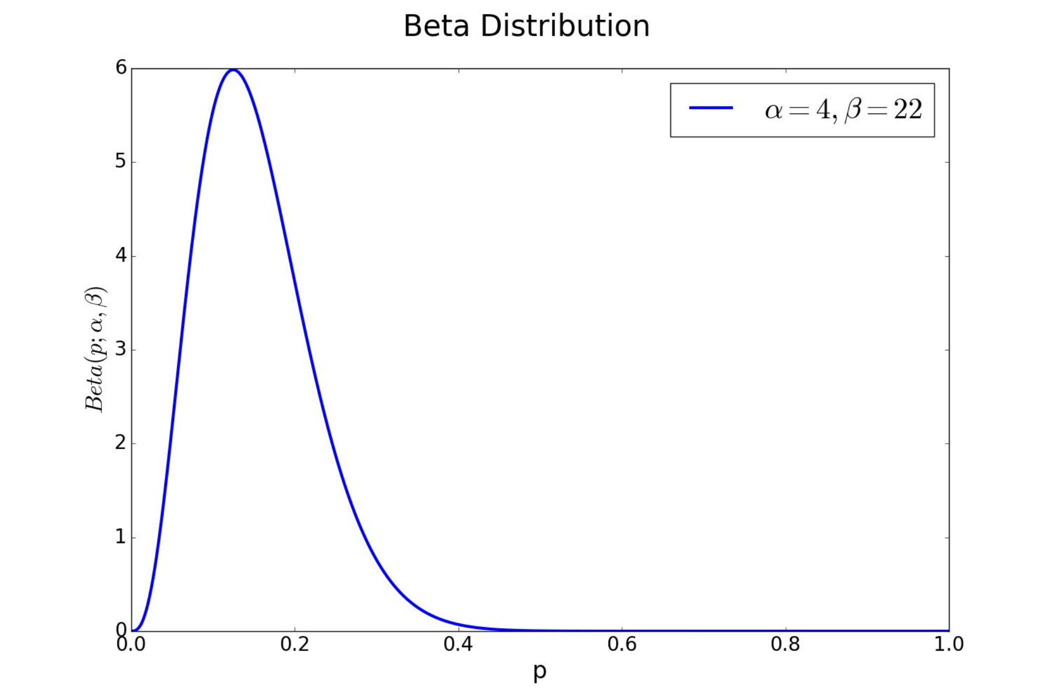
- For example, if you had 24 trials with 3 successes, you'd have this distribution



Statements you can make with this distribution

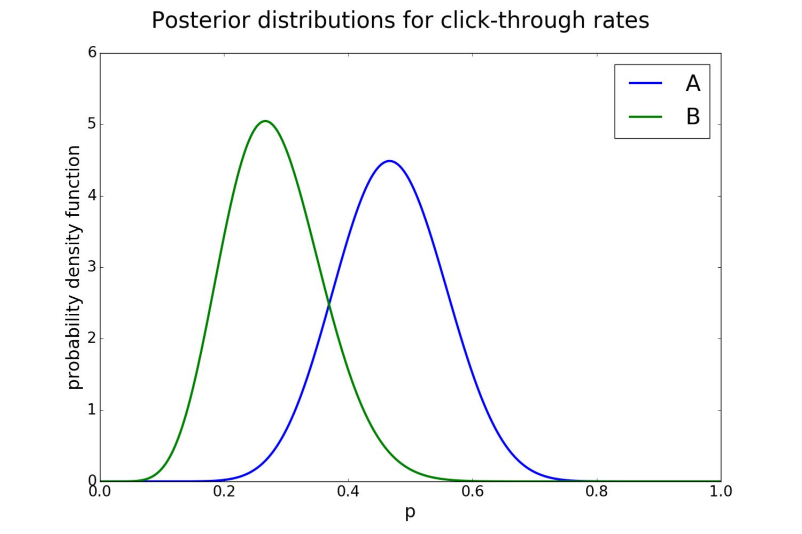
galvanize

- “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**?



Bayesian A/B Testing

- 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))
```

Frequentist A/B Testing

- 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

Bayesian A/B Testing

- 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
galvanize

Objectives: answer the following

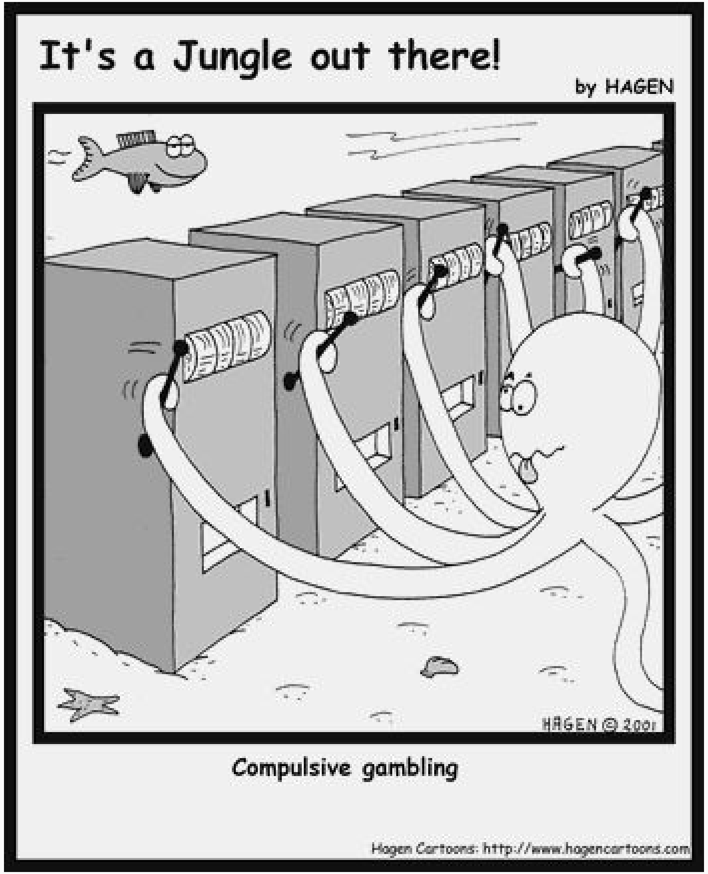
- What are exploitation, exploration, and regret in this context?
- How is this framework related to traditional A/B testing?
- What's your favorite strategy?

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

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

- 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?



Optimizing rewards: CTR again

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

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

- 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

galvanize

- ***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

Terminology: Regret

- 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 $\{p_1, p_2, p_3, \cdots, p_k\}$
- Let $p_{(t)}$ be the true success probability of the bandit chosen at time *t*
- Then our ***regret*** after T rounds is

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

Regret

galvanize

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

- 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

Strategies: epsilon-greedy

- **explore** with some fixed probability ϵ, usually 10% or less
  - generate a random number between 0 and 1. If it is less than ϵ, choose a random bandit
- **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?

Strategies: UCB1

galvanize

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

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

$$n_i = \text{number of rounds played on bandit } i$$

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

$$t = \text{total number of rounds played so far}$$

Strategies: Softmax

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

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

$$\tau = \text{“temperature” or “randomness” parameter, usually around } 0.001$$

- You then choose a bandit by sampling from this probability distribution
    - Coding tip: np.random.choice takes a parameter *p* for specifying probabilities. This is the fastest way to make a discrete random variable & probability mass function

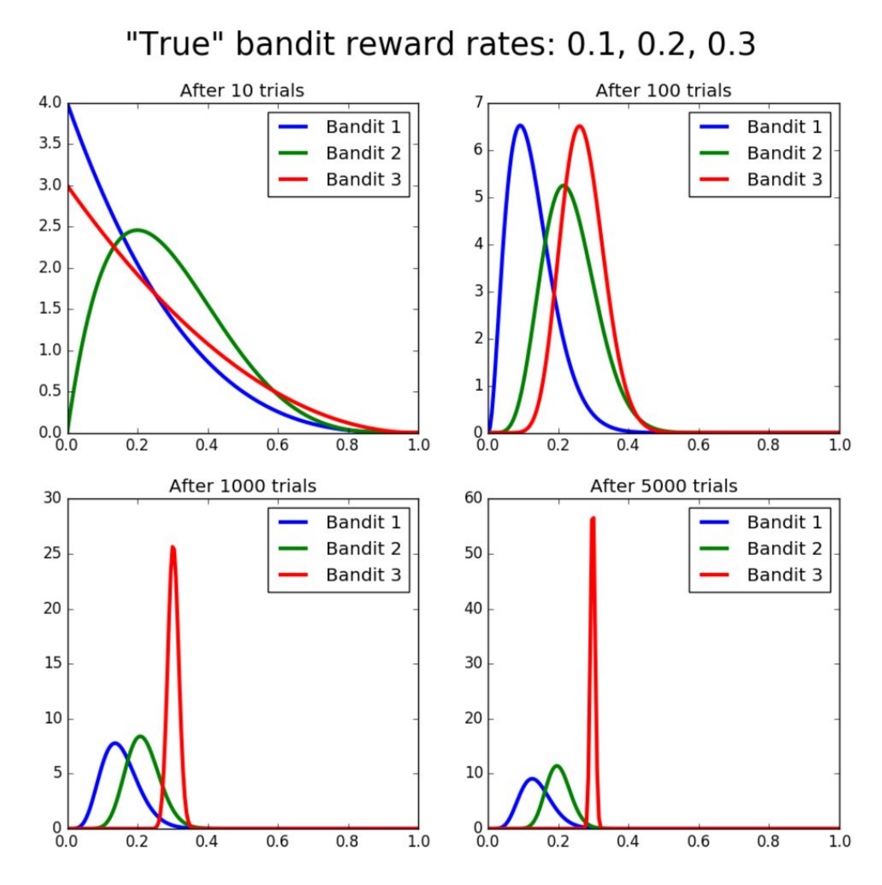
Strategies: Bayesian bandit

- 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



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?