Introduction to Data Science

BRIAN D'ALESSANDRO

ADJUNCT PROFESSOR, NYU

FALL 2016

Fine Print: *these slides are, and always will be a work in progress. The material presented herein is original, inspired, or borrowed from others' work. Where possible, attribution and acknowledgement will be made to content's original source. Do not distribute, except for as needed as a pedagogical tool in the subject of Data Science.*

REVIEW - BAYES' RULE

Particularly useful for reasoning about hypotheses (H) given evidence (E)

$$P(H \mid E) = \frac{P(E \mid H) * P(H)}{P(E)}$$

To see this:

$$P(H \mid E) * P(E) = P(EH) = P(E \mid H) * P(H)$$

NYU – Intro to Data Science
Copyright: Brian d'Alessandro, all rights reserved

USEFUL/COMMON APPLICATIONS

- Naïve Bayes
- General probabilistic reasoning
- Regularization
- Dealing with Binomial distributions (i.e., response data)
  - Smoothing probability estimates
  - Multi-Armed Bandits

Copyright: Brian d'Alessandro, all rights reserved

EXAMPLE OF BAYES RULE FOR INFERENCE

- Your friend was tested for a disease D, which has a generally low incidence: P(D) = 0.04%
- A test comes back positive PT, but the test isn't perfect:
  - Sensitivity/TPR (P(PT|D)) = 94%
  - Specificity/TNR (P(!D|!PT)) = 98%
  - FPR (P(D|!PT)) = 2%

How worried should we be?

I.e., what is the likelihood that our friend actually has the disease?

NYU – Intro to Data Science
Copyright: Brian d'Alessandro, all rights reserved

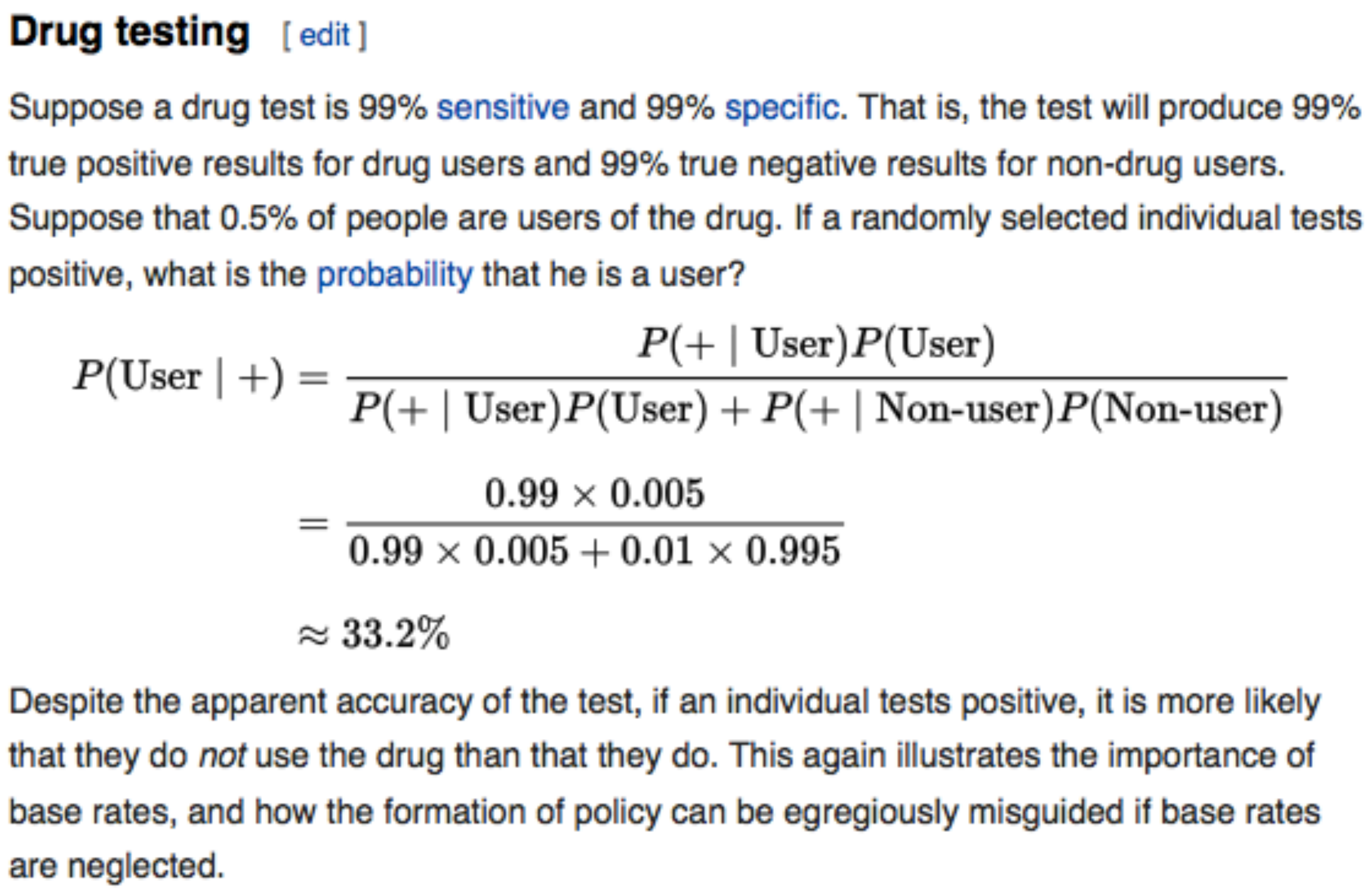
USING BAYES' RULE

$$
\begin{aligned}
P(D \mid PT) &= \frac{P(PT \mid D)P(D)}{P(PT)} \\
&= \frac{P(PT \mid D)P(D)}{P(PT \mid D)P(D) + P(PT \mid ND)P(ND)} \\
&= \frac{(0.94)(0.0004)}{(0.94)(0.0004) + (0.02)(0.9996)} \\
&= 0.018
\end{aligned}
$$

So while P(D|PT) >> P(D), a positive test result still produces a low probability of the disease. How do FPR and TPR affect the final determination of P(D|PT)?

NYU – Intro to Data Science
Copyright: Brian d'Alessandro, all rights reserved

ANOTHER EXAMPLE



Drug testing [edit]

Suppose a drug test is 99% sensitive and 99% specific. That is, the test will produce 99% true positive results for drug users and 99% true negative results for non-drug users. Suppose that 0.5% of people are users of the drug. If a randomly selected individual tests positive, what is the probability that he is a user?

$$
\begin{aligned}
P(\text{User} \mid +) &= \frac{P(+ \mid \text{User})P(\text{User})}{P(+ \mid \text{User})P(\text{User}) + P(+ \mid \text{Non-user})P(\text{Non-user})} \\
&= \frac{0.99 \times 0.005}{0.99 \times 0.005 + 0.01 \times 0.995} \\
&\approx 33.2\%
\end{aligned}
$$

Despite the apparent accuracy of the test, if an individual tests positive, it is more likely that they do *not* use the drug than that they do. This again illustrates the importance of base rates, and how the formation of policy can be egregiously misguided if base rates are neglected.

https://en.wikipedia.org/wiki/Bayes'_theorem

Copyright: Brian d'Alessandro, all rights reserved

COMPUTING PROBABILITIES

Copyright: Brian d'Alessandro, all rights reserved

SIMPLE AND FREQUENTIST

- Collect a sample of *N* events
- Count *S* successes out of the *N* events
- Get:
  - $P = S / N$
- Get a standard error:
  - stderr(P) = sqrt(P*(1-P) / N)

This is simple, scalable and incredibly common – master it!

Copyright: Brian d’Alessandro, all rights reserved

BUT…

It’s not always so simple.

- Small N, S or P:
  - Normal distribution approximation breaks down
  - High variance in estimates

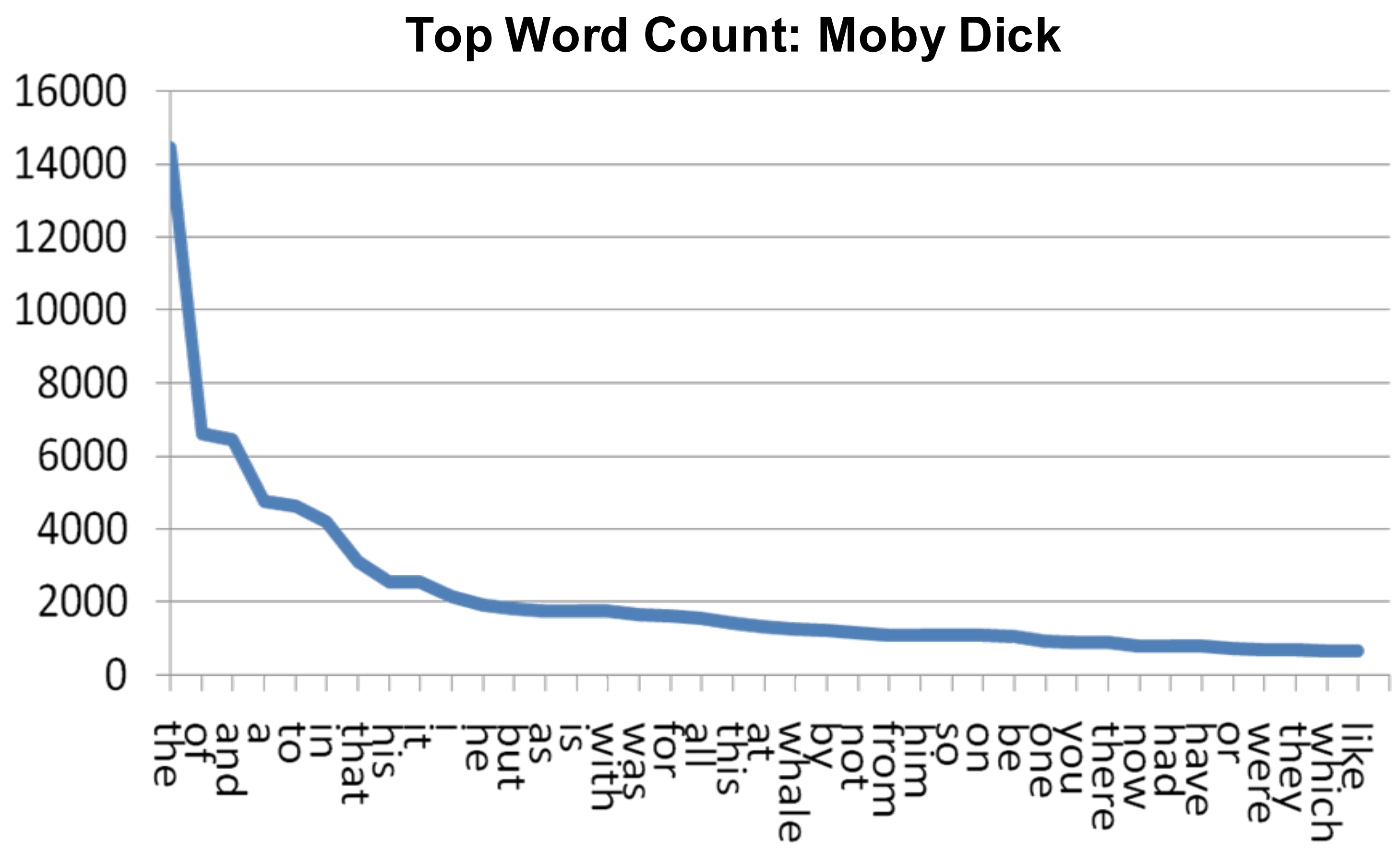
- New data?
  - Nice to have a seamless way to incorporate new evidence.

NYU – Intro to Data Science
Copyright: Brian d’Alessandro, all rights reserved

MOTIVATING EXAMPLE

Let's assume we want P(Word|Y=y). The most popular words in a book/text appear often enough that we can trust our estimate of P(Word|Y=y).

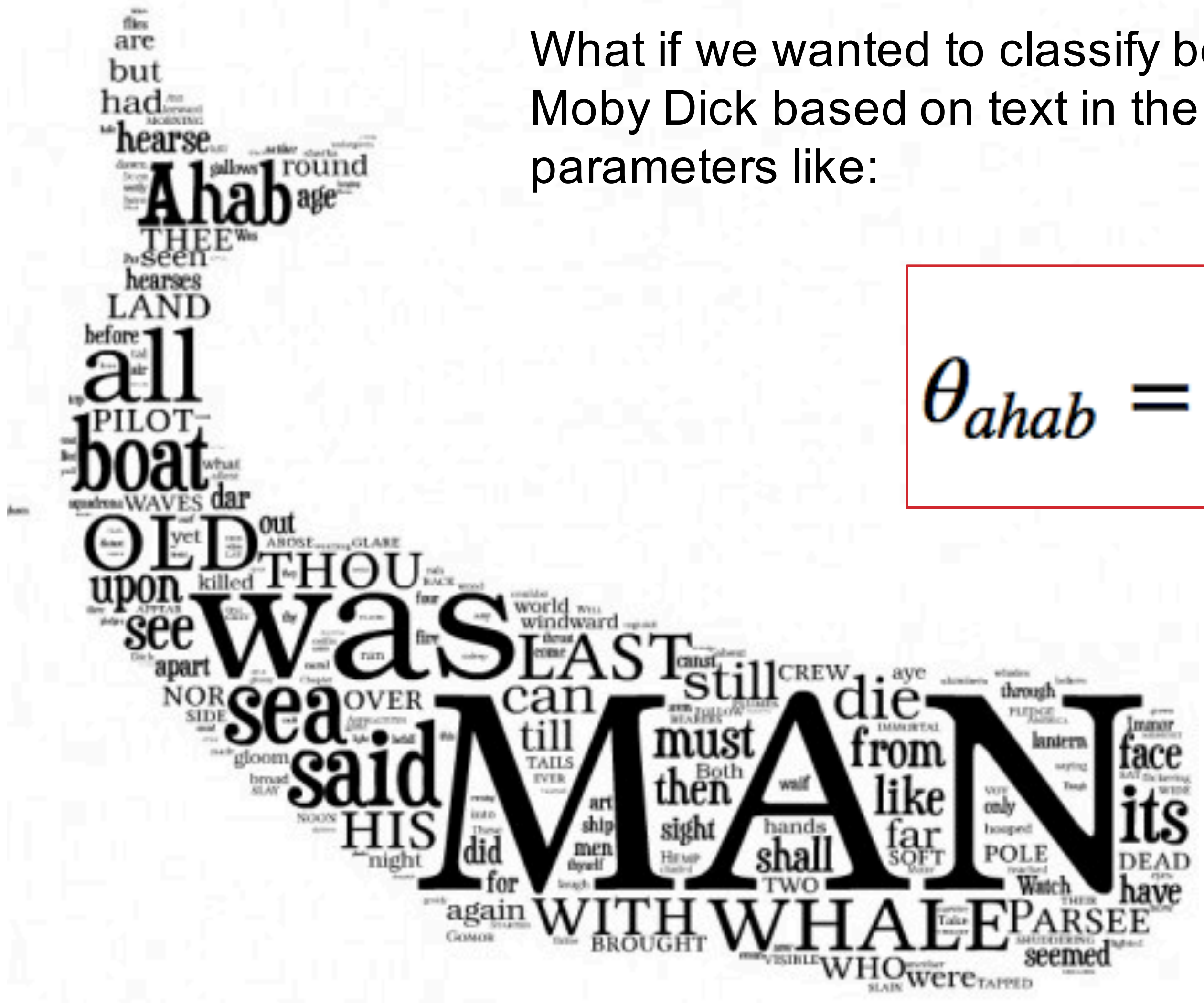
But popular words are often the least discriminative, i.e., P(Word|Y=y)=P(Word)



Source: http://searchengineland.com/the-long-tail-of-search-12198

SPARSITY IS INTERESTING

What we usually want are the rare words, like “whale”, “ahab” and “harpooner”



What if we wanted to classify book reports as relating to Moby Dick based on text in the report? We might estimate parameters like:

$$\boxed{\theta_{ahab} = ln \frac{P(ahab|Y=1)}{P(ahab|Y=0)}}$$

Source: http://ahistoryofnewyork.com/2013/01/moby-dick-big-read-day-117/

yright: Brian d’Alessandro, all rights reserved

PROBABILITY ESTIMATION

Estimating a probability just amounts to counting.

$$p(ahab|Y = y) = \frac{cnt(ahab) \mid y}{cnt(allwords) \mid y}$$

But rare words (events) pose many problems with probability estimation:

- **Zeros**: If the number of samples is low, sometimes the word/event won't ever be observed, so the probability is zero or we end up with a divide-by-zero situation.

- **High-variance:** the variance of a probability estimate is p(1-p)/n. If n or p is low, our estimate won't be very trustworthy. Think about if a word was observed once in a sample. If by chance another example came in with the word, the probability estimate of the word will double!

Copyright: Brian d'Alessandro, all rights reserved

SMOOTHING A PROBABILITY

In the MNB, we added a little extra to the standard maximum likelihood estimate of a Bernoulli probability.

$$P(x_i|y) = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

What is this and where did it come from?
What is the motivation?

NYU – Intro to Data Science
Copyright: Brian d'Alessandro, all rights reserved

SMOOTHING

We fix these problems with a Bayesian method called “smoothing.” Bayesian approaches enable us to blend our prior beliefs (P(*θ*)) with data we observed (through the likelihood *P(X|θ)).*

In many cases we seek to estimate the mode of the posterior distribution, given by:

$$\boxed{\hat{\theta}_{MAP}(X) = \underset{\theta}{\operatorname{argmax}}\, P(X|\theta)P(\theta)}$$

Smoothing is similar, only we take the expected value of the posterior distribution.

Copyright: Brian d’Alessandro, all rights reserved

CHOOSING A PRIOR

1. We can use a prior that assumes X and Y are independent, so P(X|Y=y) = P(X).

2. We can also choose that P(X|Y=y) is equal to some previously estimated quantity

3. We can assume that P(X) = ½, just because.

4. We can also assume that P(X) is uniform for all X (i.e., every word in a document has equal probability).

We will introduce Dirichlet and Beta priors to accommodate all of the above. Ultimately, the optimal prior is an empirical question that can be evaluated using a hold-out set.

Copyright: Brian d'Alessandro, all rights reserved

WORDS AS A MULTINOMIAL

We can think of a word in a corpus as a multinomial discrete random variable X, that can take one of any words in the vocabulary. The likelihood of a particular x_i is simply the number of times x_i appears over the total possible occurrences of any X.

$$\boxed{P(x_i) = \frac{Ni}{N}}$$

This has the constraint that:

$$\boxed{\sum_i P(x_i) = 1}$$

We can see that this is how Multinomial Naïve Bayes is set up. Now, in many cases Ni might be low due to the rarity of the word. We thus want to set up the appropriate prior to smooth our estimate of $P(x_i)$.

Copyright: Brian d'Alessandro, all rights reserved

DIRICHLET PRIOR

The conjugate prior of the Multinomial distribution is the Dirichlet distribution, given by:

$$\boxed{f(x;\alpha) = \frac{\Gamma(\alpha)}{\prod_i \Gamma(\alpha_i)} \prod_i x^{\alpha_i - 1}}$$

Given this prior, the MAP estimate of $P(x_i)$ is then:

$$\boxed{P(x_i) = \frac{N_i + \alpha_i}{N + \alpha}}$$

If we choose α_i=1, then α = |vocabulary|. This assumes a priori that $P(X_i)$ is uniform (i.e., all words have an equal prior probability).

Reference: http://research.microsoft.com/pubs/69588/tr-95-06.pdf

Copyright: Brian d'Alessandro, all rights reserved

BETA DISTRIBUTION

The beta distribution is the conjugate prior of the Binomial distribution. It is a special form of the Dirichlet distribution, where X has only two discrete values. The Dirichlet prior has a specific application to Naïve Bayes because X is often defined as a multinomial. In many cases we only want to deal with a binary random variable, which makes the beta distribution appropriate.

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

Uses the Gamma function to normalize.

Has same algebraic form as the Binomial distribution.

The mean of the beta distribution is given by:

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

Copyright: Brian d'Alessandro, all rights reserved

THE BETA PRIOR

The beta prior gives us greater flexibility to incorporate any prior belief in the probability of binary X occurring. With the beta prior, we assume the prior belief that P(X) is given just by the beta distribution with parameters α and β. The smoothed, MAP estimate of our probability is then:

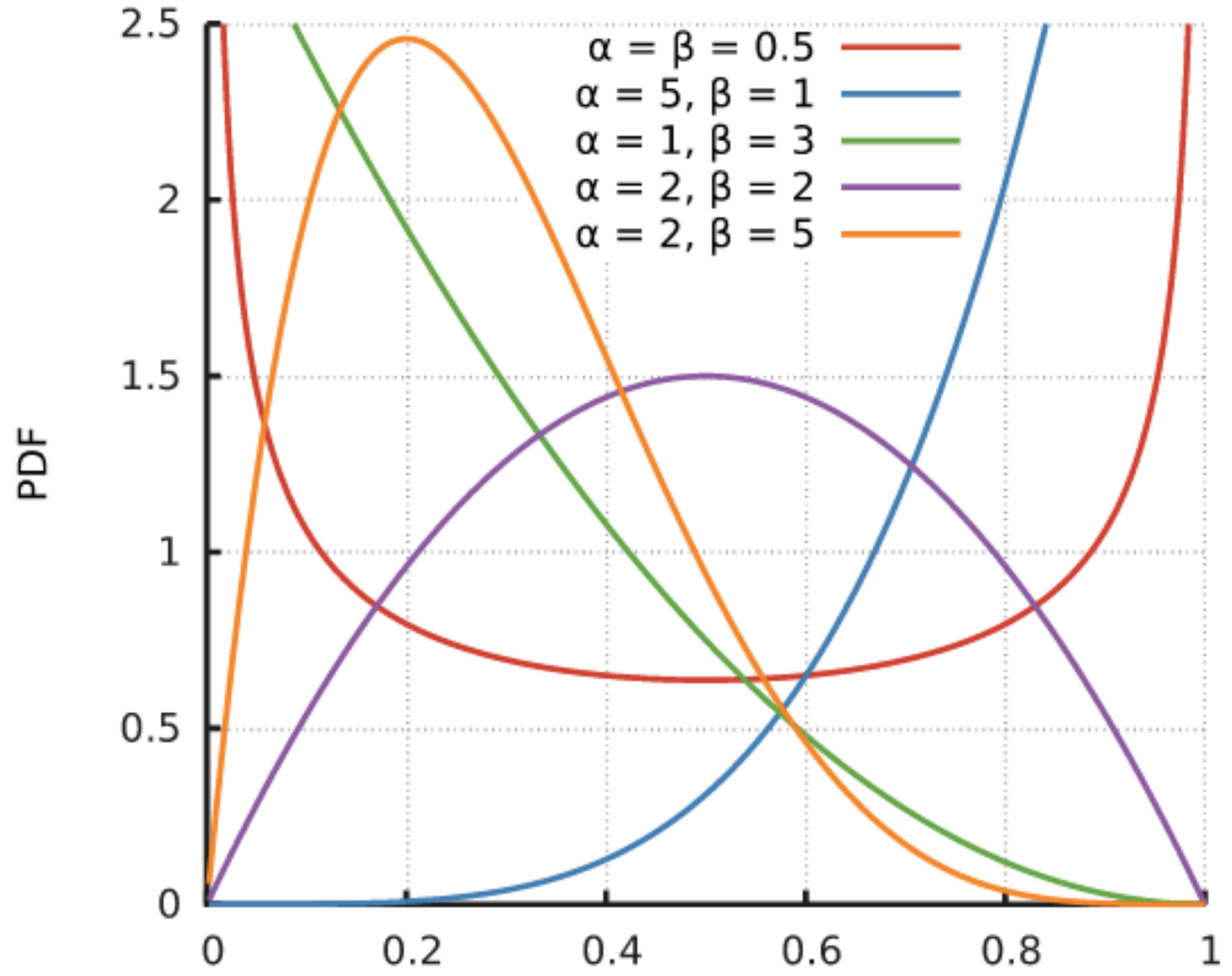
$$P_s(X) = \frac{s + \alpha}{n + \alpha + \beta}$$

The beta prior gives us the ability to estimate our prior directly from the data. I.e., if we want μ = P(X), we can estimate P(X) from the data. We control the degree of smoothing by choosing α and solving for β. The higher α, the more we assert our prior belief into the estimate of θ.

Copyright: Brian d'Alessandro, all rights reserved

THE BETA DISTRIBUTION

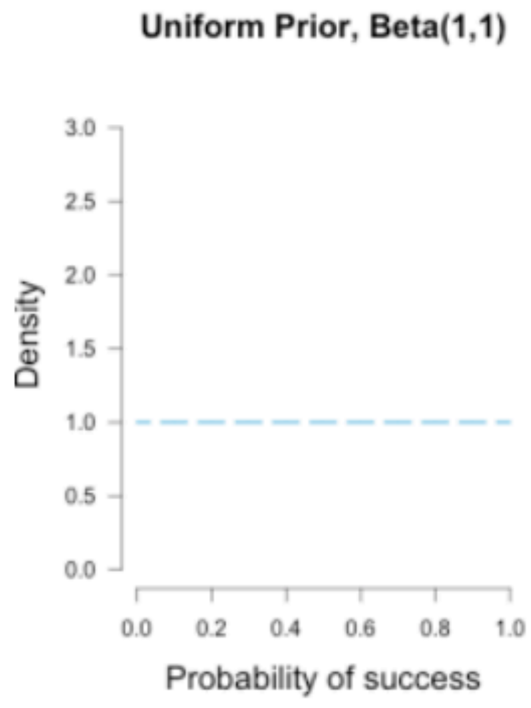
The likelihood of a given P depends on the parameters α and β

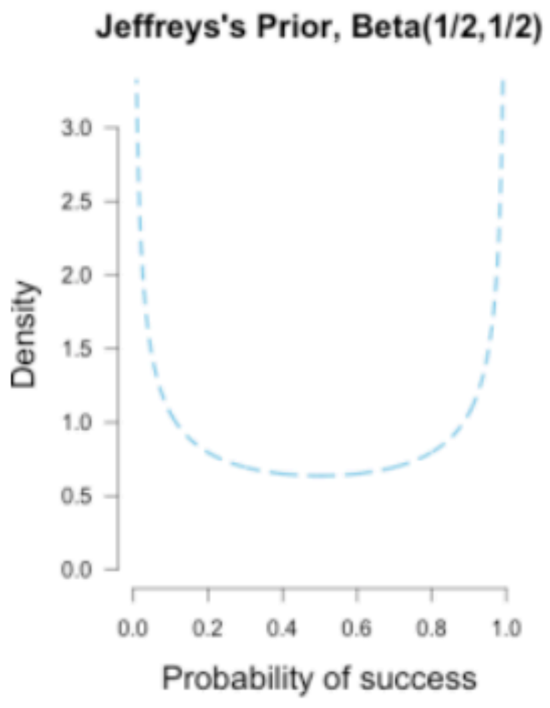


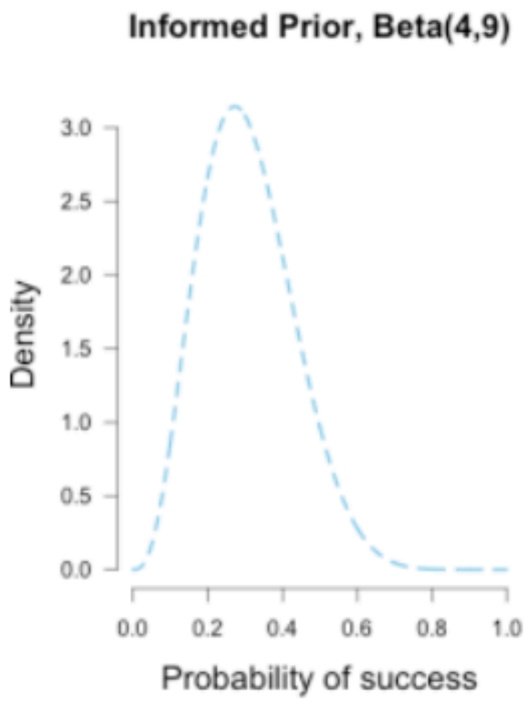
Source: Wikipedia

Copyright: Brian d'Alessandro, all rights reserved

COMMON PRIORS







Best when you have no prior information

Strong belief in either 0/1

Have some data already, use this when updating.

Copyright: Brian d'Alessandro, all rights reserved

BAYESIAN UPDATING

Goal: Compute the mean of a Bernoulli random variable at different intervals in time

Take a Bayesian approach and use the beta-binomial distribution!

1. Chose a suitable prior before taking any samples: $P(\Theta) = \mathrm{Beta}(\alpha, \beta)$
2. Draw N_1 samples and count S_1 successes
3. Compute $\Theta = \mathrm{Beta}(\alpha + S_1, \beta + N_1 - S_1)$.mean()

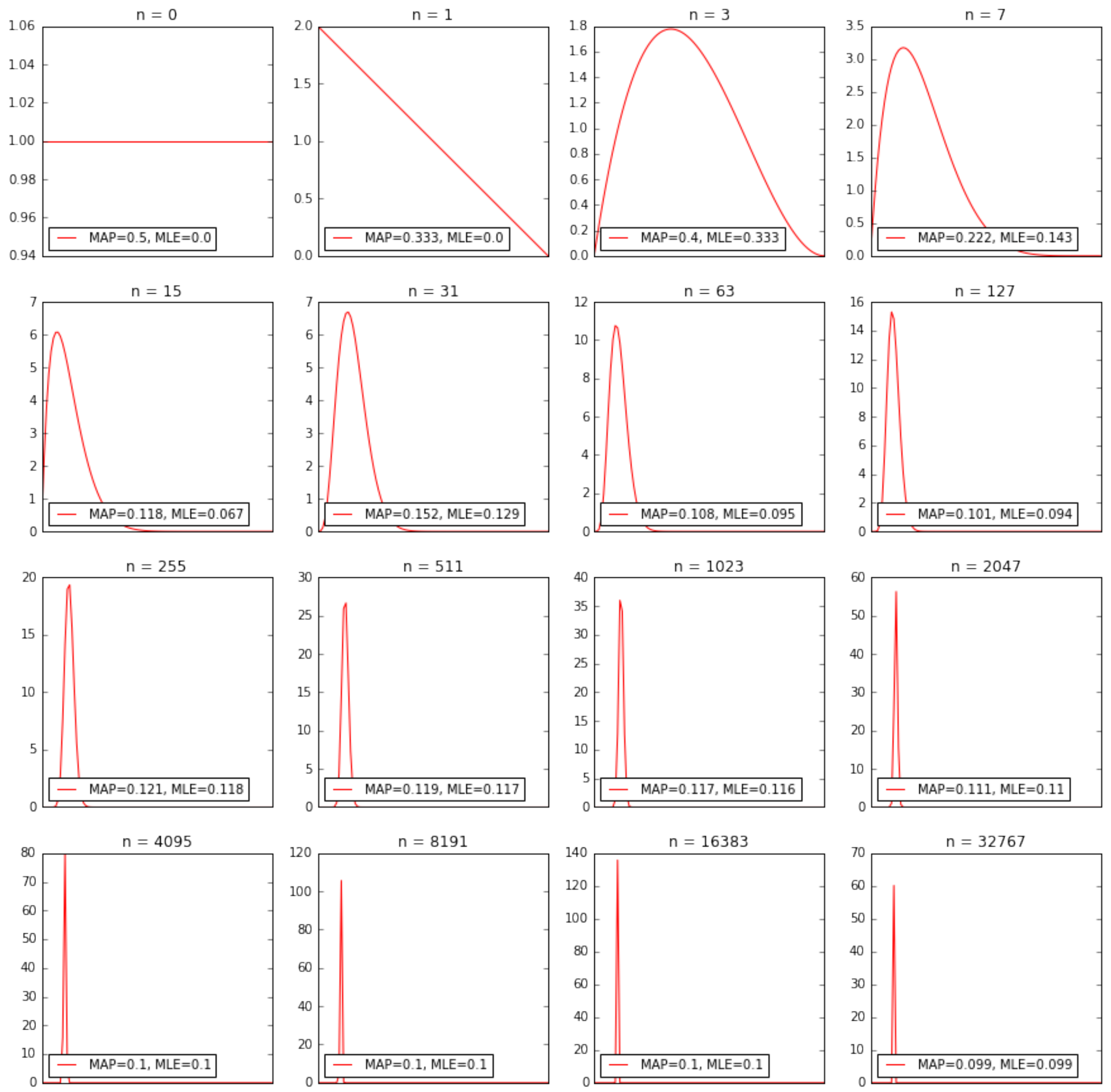
Not satisifed?

4. Reset: $P(\Theta) = \mathrm{Beta}(\alpha + S_1, \beta + N_1 - S_1)$
5. Draw N_2 more samples, count S_2 more successes
6. Compute $\Theta = \mathrm{Beta}(\alpha + S_1 + S_2, \beta + N_1 + N_2 - (S_1 + S_2))$.mean()

7. Repeat steps 4 – 7 until satisfied

Copyright: Brian d'Alessandro, all rights reserved

BAYESIAN UPDATING



NYU – Intro to Data Science
Copyright: Brian d'Alessandro, all rights reserved

MULTI-ARMED BANDIT

A common problem many organizations face is to choose for an entity one of many policies with the goal to maximize some sort of reward.

e.g.:

Application	**Policy**	**Reward**
Medicine	medical treatment	become healthy
Internet Advertising	a particular ad	conversions
News Recommendation	an article suggestion	clicks on article
A/B Testing	a/b variant	conversions

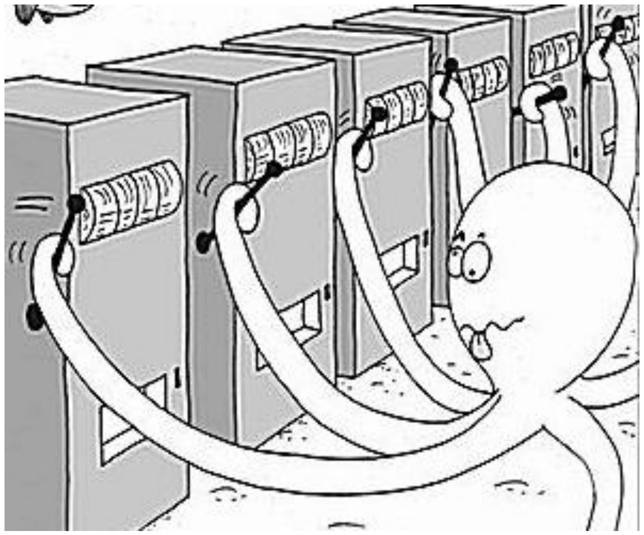
Core problem: at the outset you don't know which policy will yield the maximum reward, so you need a mechanism to both learn the payout of each policy while exploiting policies you think might be better (but you're not 100% certain are).

Copyright: Brian d'Alessandro, all rights reserved

MULTI-ARMED BANDIT

General sketch of a MAB algorithm:

1. Receive an opportunity to enact a policy
2. Choose a policy based on an exploration/exploitation strategy
3. Execute policy and receive feedback
4. Update reward history



Intuitively, a MAB is like a dynamic experiment with 2+ variants. Rather than set a single, fixed split, we incrementally adjust the split based on how confident we are in the rank ordering if the results.

EXPLORE/EXPLOIT STRATEGIES

- **ε-Greedy**
  - Choose highest performing policy (1-ε)% of time, and choose
  - Randomly choose other policies ε% of the time

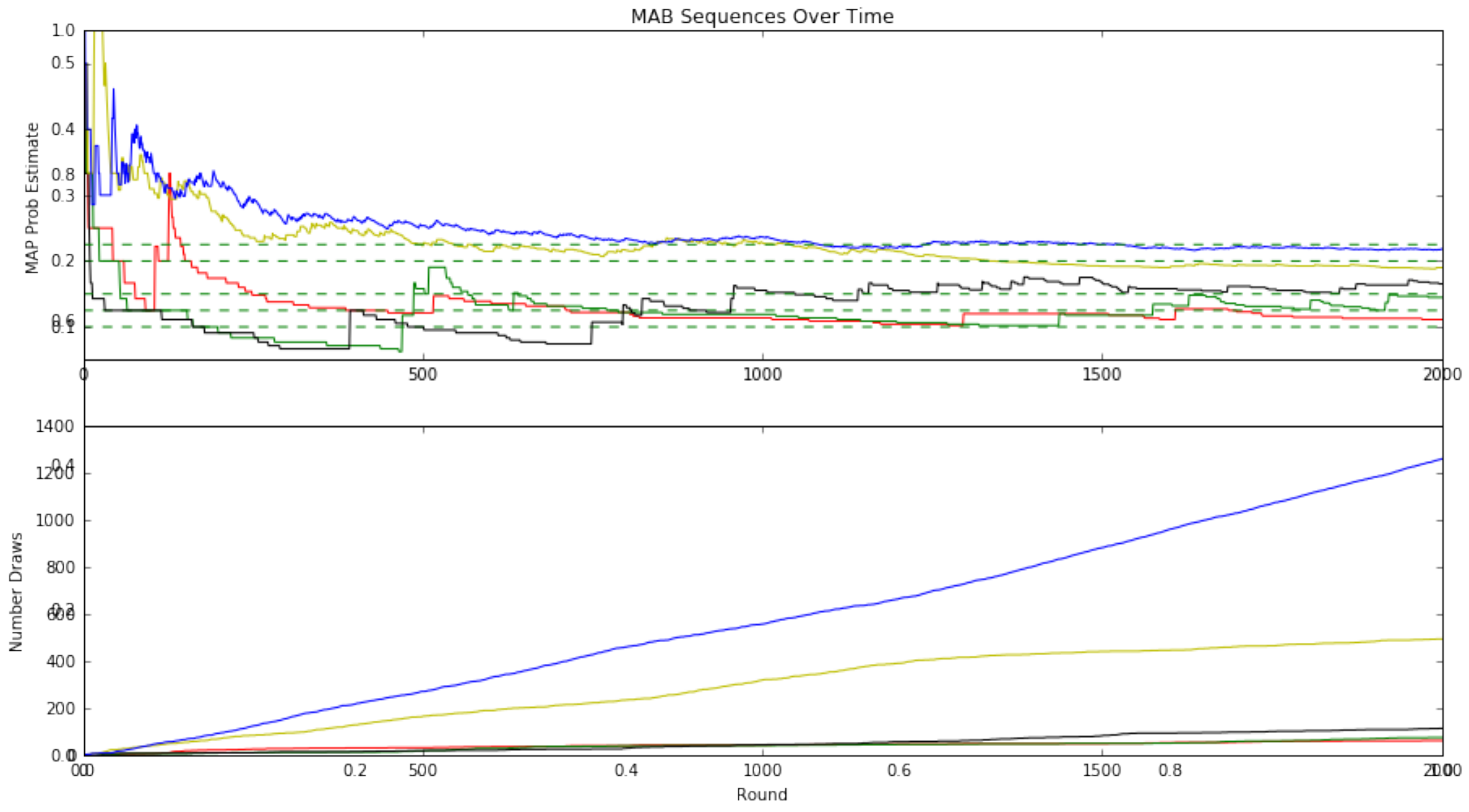
- **Upper Confidence Band (UCB)**
  - For each policy compute upper confidence percentile of the reward statistic (i.e., UCB Score = p + 1.96*sqrt(p*(1-p)/n))
  - Choose policy with highest UCB Score

- **Thompson Sampling**
  - For each policy, generate a random number from the posterior distribution of the reward statistic (i.e. TS Score = Random. Beta($\alpha + S_1$, $\beta + N_1 - S_1$))
  - Choose policy with highest TS Score

NYU – Intro to Data Science
Copyright: Brian d'Alessandro, all rights reserved

MAB EVOLUTION

This is simulated MAB where each policy has a truth in ps = [0.1, 0.125, 0.15, 0.2, 0.225]. We can see that the MAP estimates of individual policy probabilities don't stabilize until many rounds.

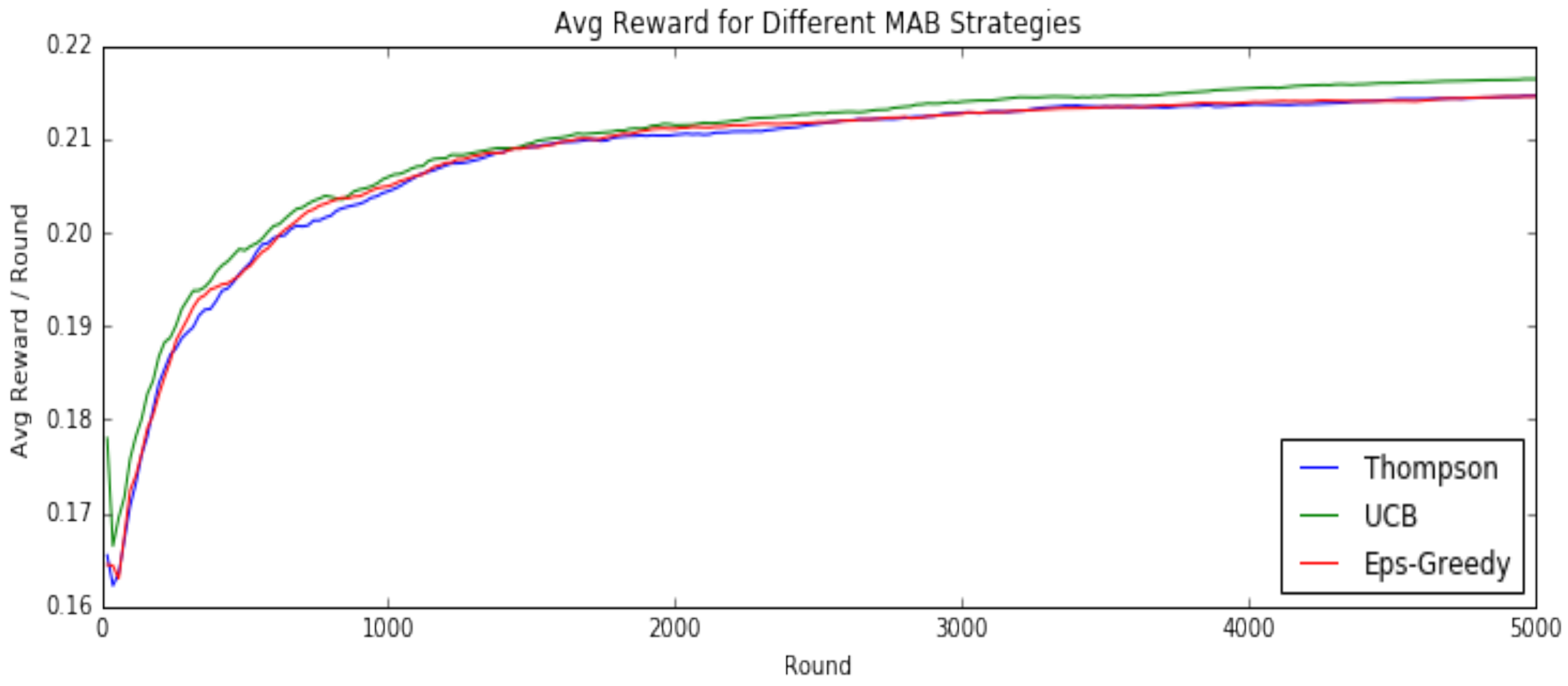


https://github.com/briandalessandro/DataScienceCourse/blob/master/ipython/python35/Lecture_MultiArmedBandit.ipynb

Copyright: Brian d'Alessandro, all rights reserved

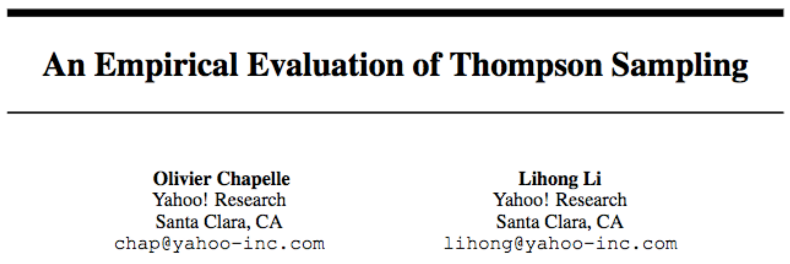
COMPARING STRATEGIES

In this particular simulation and set up, the UCB slightly out-performs the Thompson Sampling and ε-Greedy approaches.



NYU – Intro to Data Science
Copyright: Brian d'Alessandro, all rights reserved

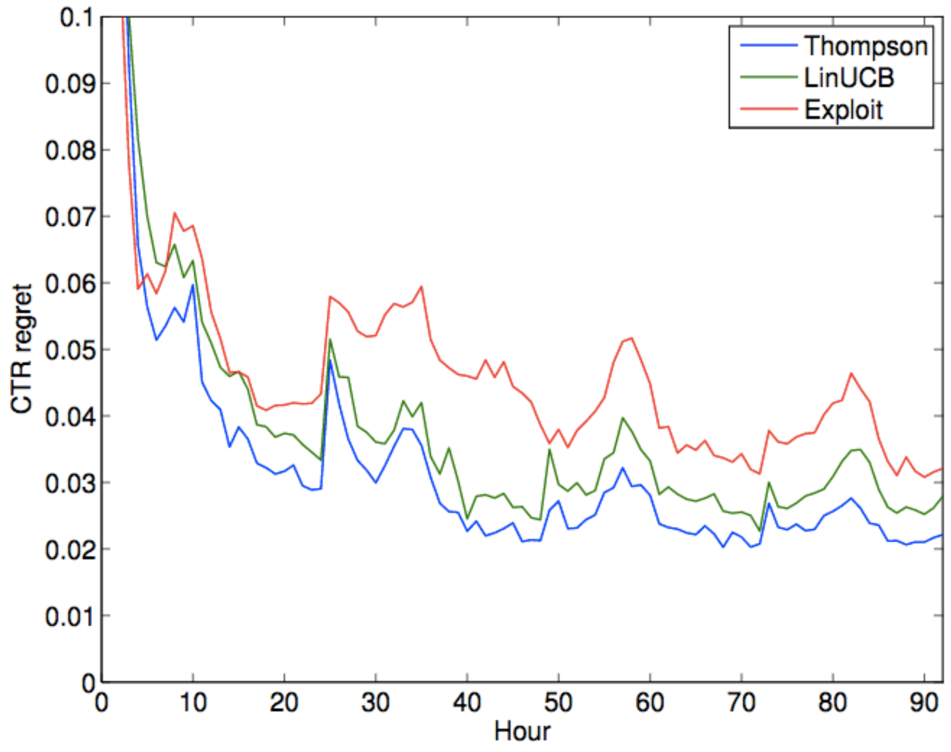
MAB IN PRACTICE



An Empirical Evaluation of Thompson Sampling

Olivier Chapelle
Yahoo! Research
Santa Clara, CA
chap@yahoo-inc.com

Lihong Li
Yahoo! Research
Santa Clara, CA
lihong@yahoo-inc.com



http://www.research.rutgers.edu/~lihong/pub/Chapelle12Empirical.pdf

In this paper the authors show that Thompson Sampling is highly competitive with UCB in both simulations and real-life data sets.

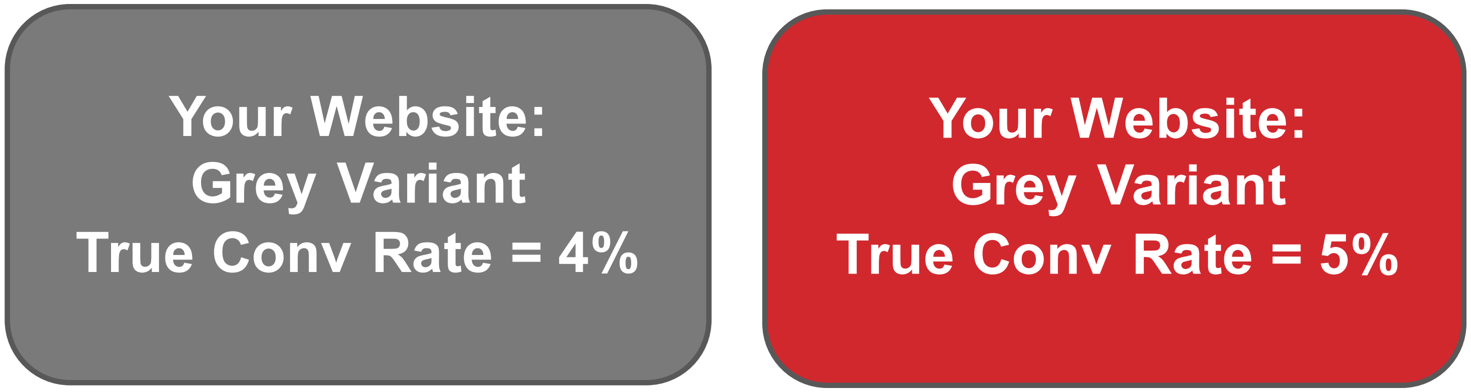
This analysis shows an extension of the MAB, the Contextual Bandit, where we take the posterior of weights in a model P(Y|X) as opposed to just the posterior of a probability.

Copyright: Brian d'Alessandro, all rights reserved

MAB IN PRACTICE (GOOGLE)

We can also use a MAB to speed up experiments and make them less costly: dynamically allocate the split percentage based on the MAB algorithm.

A Classical A/B Test



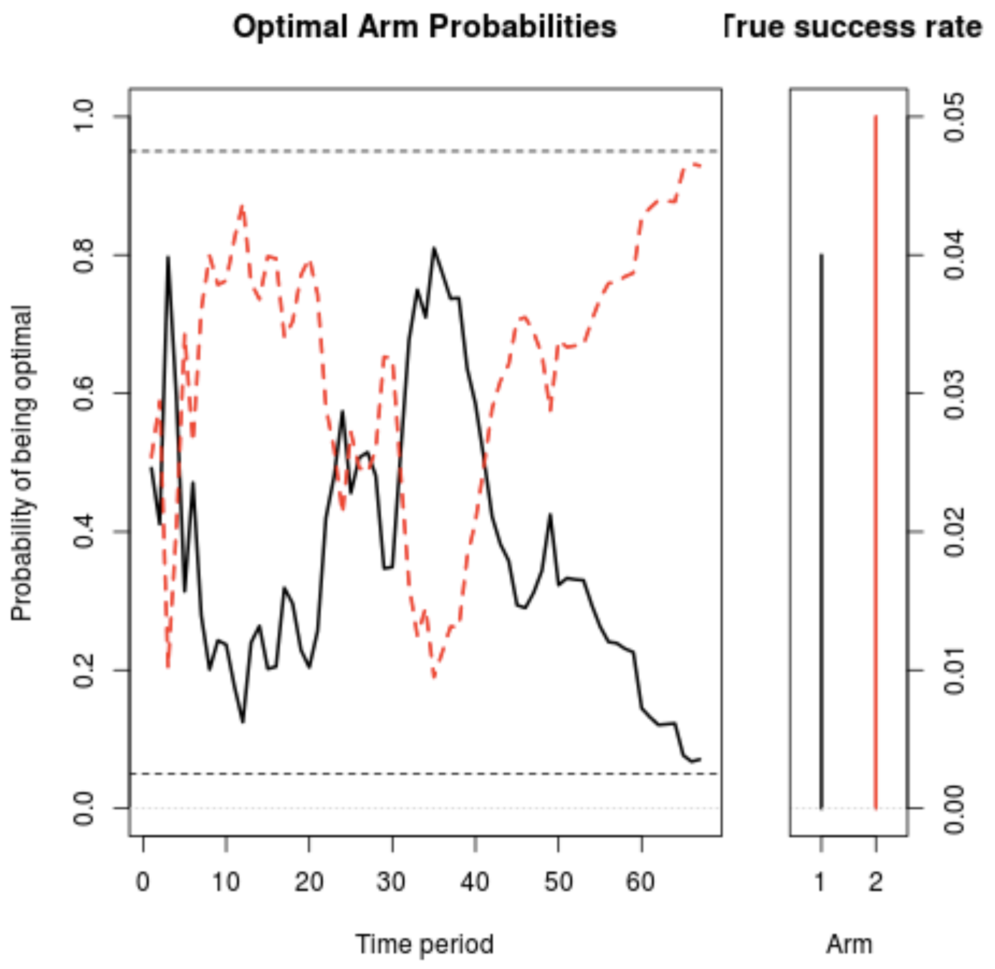
A standard power calculation tells you that you need 22,330 observations (11,165 in each arm) to have a 95% chance of detecting a .04 to .05 shift in conversion rates. Suppose you get 100 sessions per day to the experiment, so the experiment will take 223 days to complete.

https://support.google.com/analytics/answer/2844870?hl=en&ref_topic=1745207

Copyright: Brian d'Alessandro, all rights reserved

MAB IN PRACTICE (GOOGLE)

Instead of the usual 50/50 split, the MAB reallocates traffic based on cumulative observed performance.

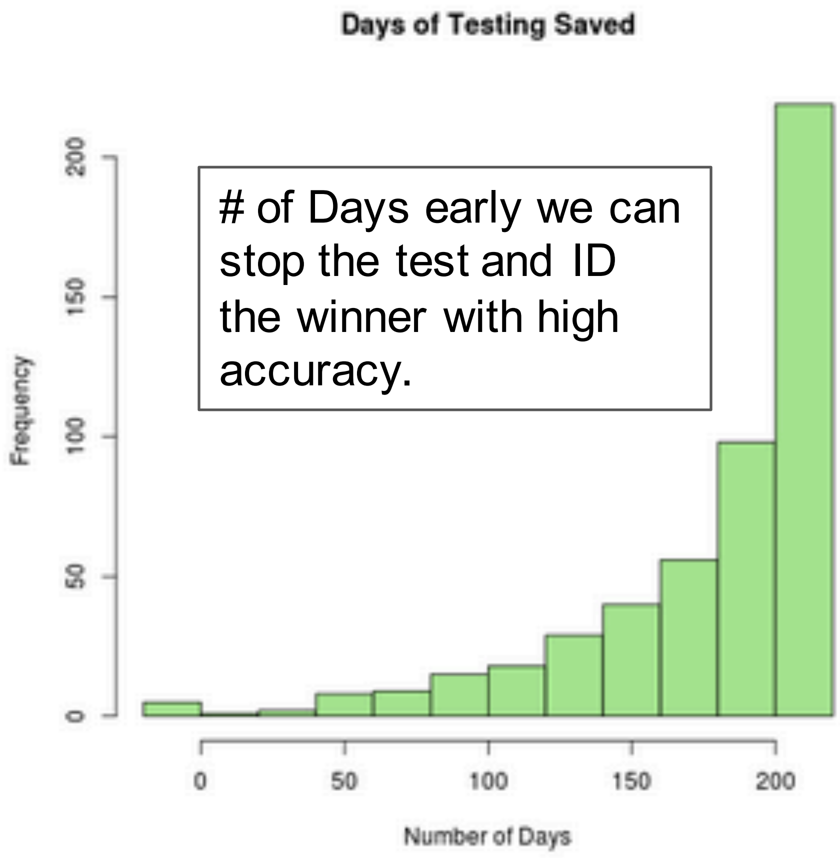


https://support.google.com/analytics/answer/2844870?hl=en&ref_topic=1745207

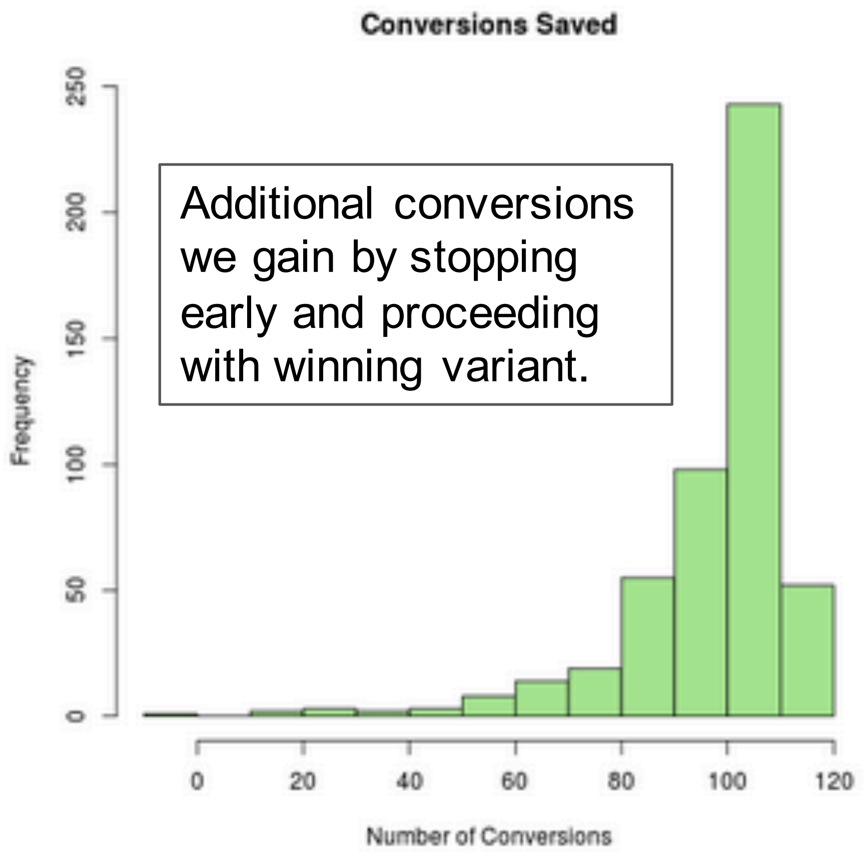
Copyright: Brian d'Alessandro, all rights reserved

MAB IN PRACTICE (GOOGLE)

Simulations can help show the benefit both in time and value.



# of Days early we can stop the test and ID the winner with high accuracy.



Additional conversions we gain by stopping early and proceeding with winning variant.

https://support.google.com/analytics/answer/2844870?hl=en&ref_topic=1745207

Copyright: Brian d'Alessandro, all rights reserved