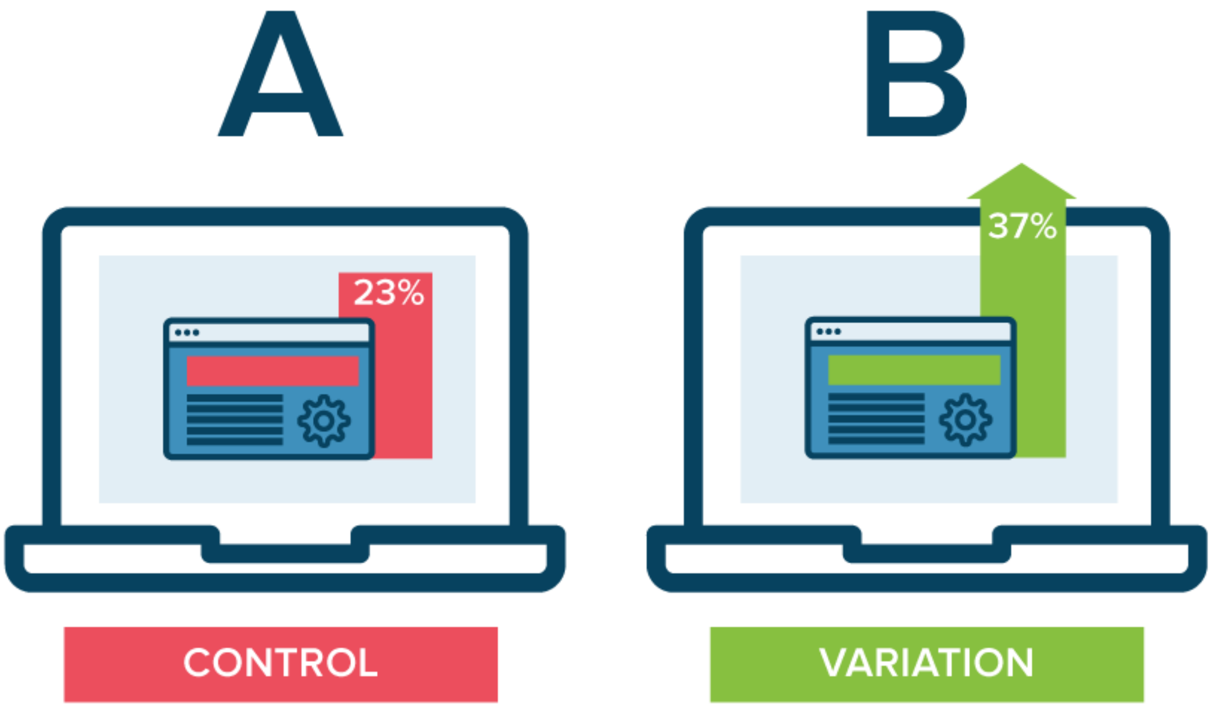
Introduction to Data Science

BAYESIAN EXPERIMENTATION

BRIAN D'ALESSANDRO

AB TESTS

AB testing is a simple and powerful mechanism for optimizing actions*



**Although generally simple to analyze, one should not take for granted implementation costs as well as potential for common pitfalls*

FREQUENTIST APPROACH

- **Compute relevant statistics:**
  - $P_A = S_A / N_A$; $P_B = S_B / N_B$
  - $Z = (P_A - P_B) / \mathrm{sqrt}(P_A*(1-P_A) / N_A+P_B*(1-P_B) / N_B)$
- **Assume Z~N(0,1)**
  - Get P-value = 2*(1-norm.cdf(Z,0,1))
- **If P-value < 0.05:**
    - celebrate
  - Else:
    - Blame implementation team

NOT SO EASY ACTUALLY

- **Most likely you do not have enough data**
  - Sub-populations thus difficult to analyze
  - Getting to significant p-values could take months or years
- **Distributions are never really Normal**
  - Effect sizes are typically small
  - Conversion rates are often low
- **P-values are kind of useless for decision making**
- **There is always pressure for early stopping**

A STILL EASY YET BETTER WAY

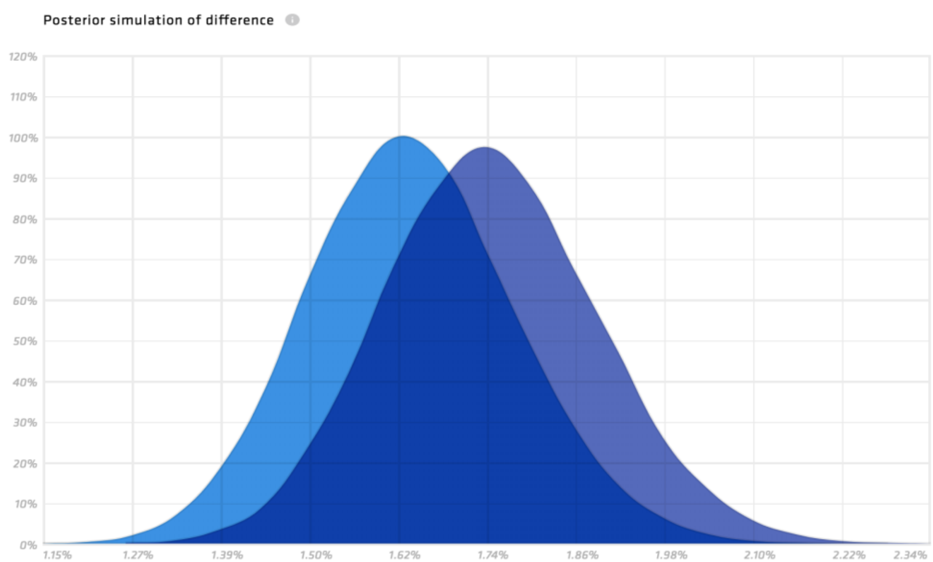
- Establish priors: P_A, P_B ~ Beta(α, β)
- Observe: S_A, N_A, S_A, N_A
- Compute Posteriors:
  - $P_A|Obs \sim Beta(\alpha + S_A, \beta + N_A - S_A)$
  - $P_B|Obs \sim Beta(\alpha + S_B, \beta + N_B - S_B)$

- Run Monte Carlo simulations on Posteriors to get:
  - Mean or distribution of effect sizes
  - P(Test Variant > Control Variant)
  - 95% Credible Intervals
  - Expected value of choosing Test vs Control variant

BETTER DECISION FRAMEWORK

P-values values measure the strength of evidence towards a hypothesis, but do they really help us make optimal decisions? A better path is to choose the option that maximizes your expected value. Beta-binomial posteriors enable such decision making.

$$\text{Expected Benefit(Choose A)} = \int E[Val|Y,A]P(Y|A)dp - \int E[Val|Y,B]P(Y|B)dp$$



This framework allows for much faster decision making and thus shorter testing cycles.

SAFE SEQUENTIAL TESTING VIA BAYESIAN UPDATING

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

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

Need more data?

4. Reset: $P(\Theta) = \text{Beta}(\alpha + S_1, \beta + N_1 - S_1)$
5. Draw N_2 more samples, count S_2 more successes
6. Compute $\Theta = \text{Beta}(\alpha + S_1 + S_2, \beta + N_1 + N_2 - (S_1 + S_2))$.mean()
7. Repeat steps 4 – 7 until satisfied

DYNAMIC A-Z TESTING

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

e.g.:

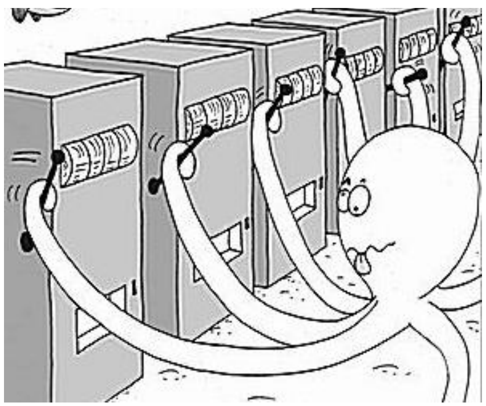
Application	Policy	Reward
Medicine	medical treatment	become healthy
Internet Advertising	a particular ad	conversions
News Recommendation	an article suggestion	clicks on article
A/B Testing	a/b variant	conversions

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

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

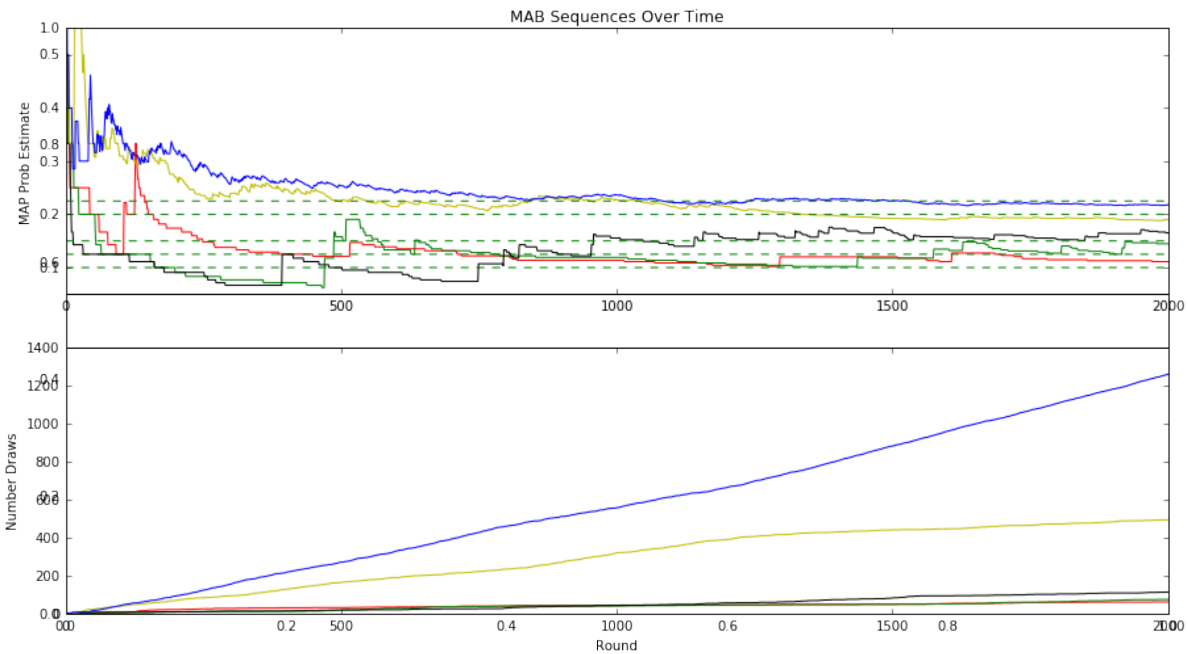
- **ε-Greedy**
  - Choose highest performing policy (1-ε)% of time, and choose
  - Randomly choose other policies ε% of the time

- **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

MAB EVOLUTION

MABs usually take some time before they converge to stable estimates. We can see here that the MAB estimates of individual policy probabilities don't stabilize until many rounds.

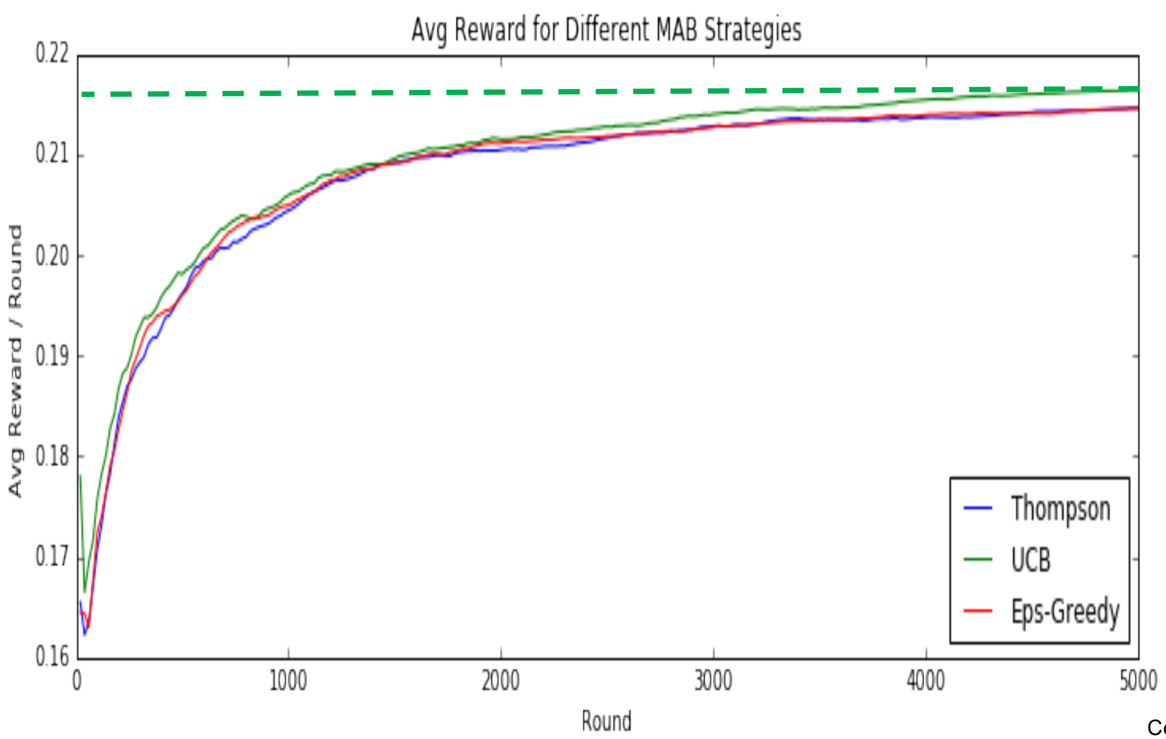


Policies that are chosen less frequently generally take longer to reach convergence, since they effectively get lower sample sizes.

This is simulated MAB where each policy has a truth in ps = [0.1, 0.125, 0.15, 0.2, 0.225].
https://github.com/briandalessandro/DataScienceCourse/blob/master/ipython/python35/Lecture_MultiArmedBandit.ipynb

REGRET ANALYSIS

The objective when choosing a strategy is to minimize the **regret**, where the **regret** is the different between accumulated reward and the reward we would have received had we chosen the optimal strategy from the beginning.



Regret: $$\rho = T\mu^* - \sum_{t=1}^{T} \hat{r}_t$$

Where:

- *T = number of rounds played*
- *μ^* = the best mean reward of all policies*
- *r_t = reward at individual round t*

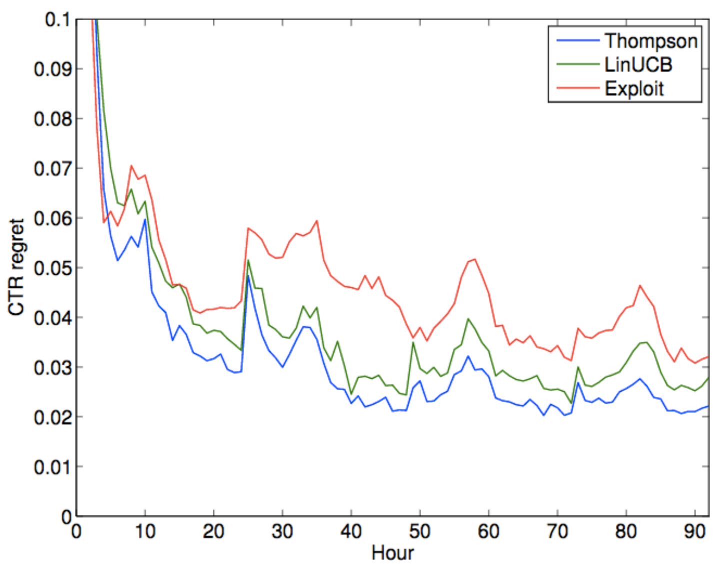
The general goal is to minimize regret, and theoretical analysis of different algorithms focus on time required to reach optimal policy.

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.

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

Your Website:
Grey Variant
True Conv Rate = 4%

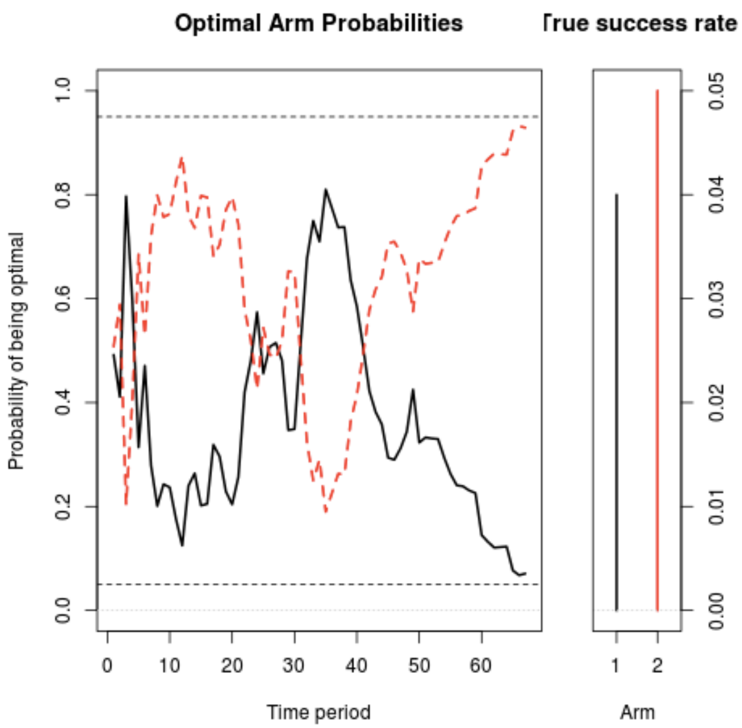
Your Website:
RedVariant
True Conv Rate = 5%

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

MAB IN PRACTICE (GOOGLE)

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

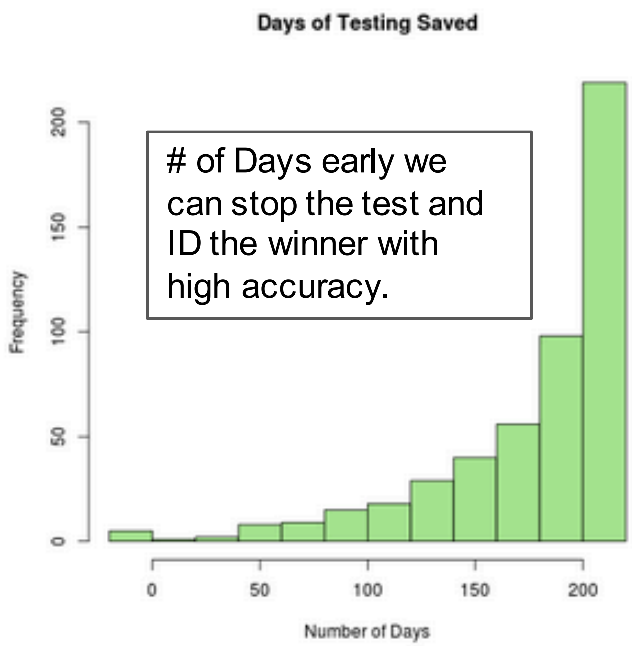


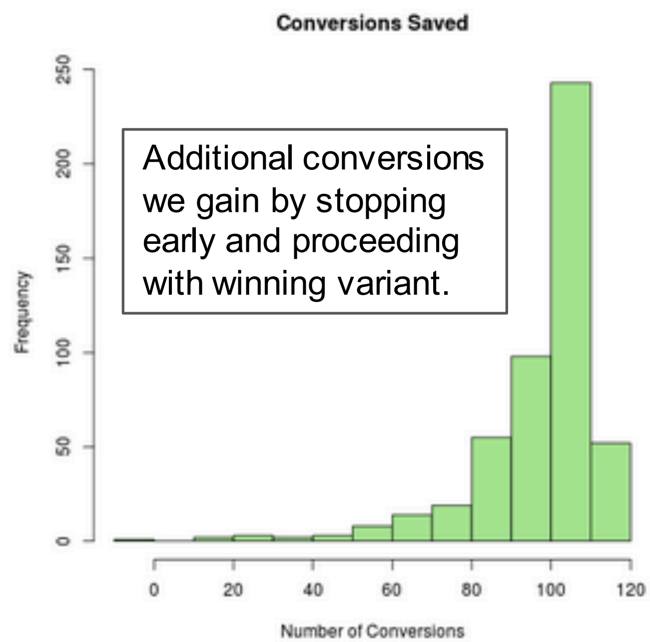
So instead of splitting at 50/50 and waiting for a result, we can dynamically split traffic, where we balance the time to learn a result with exploitation of the optimal variant.

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

MAB IN PRACTICE (GOOGLE)

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





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