Bayesian A/B Testing and Multi-Arm Bandit

Brian J. Mann

October 30, 2015

Objectives

- Be able to run a Bayesian A/B Test and know how it's different from frequentist A/B Testing
- Be able to implement a multi-arm bandit version of an A/B test.

Overview

- Morning

1. Review frequentist A/B testing.
2. What is a Bayesian A/B test and how does it work?
3. An example of Bayesian A/B testing.

- Afternoon

1. What is a “multi-arm bandit”??
2. How do we use one to do smarter A/B tests?

Bayesian vs. Frequentist

Frequentist A/B Testing

- Run all experiments and observe the data.
- Significance of the result depends on how many experiments you run (N is a parameter).
- Doesn't tell you how likely it is that A is better than B, just that you are confident A *is* better than B at a certain significance.

Bayes' Theorem

Recall **Bayes' Theorem**

$$P(x|\theta) = \frac{P(\theta|x)P(x)}{P(\theta)}$$

- $P(x|\theta)$: **posterior distribution** of x given observed θ.
- $P(\theta|x)$: **likelihood** of observing θ given x.
- $P(x)$: **prior distribution** of x.

So another way to think of Bayes's Theorem is

$$\textit{posterior} \sim \textit{likelihood} \times \textit{prior}$$

- (likelihood, prior) pairs of distribution families so that the posterior distribution is of the same type as the prior are called **Conjugate Priors**
- One example: prior = Beta, likelihood = Binomial.
- Others are in a table here.

Bayesian A/B Testing

Thomas Bayes

Bayesian A/B Testing

- Update your knowledge of the experiment each time you run it (replace prior with posterior based on observed data).
- Stop the test at any time and have a result, although running for longer will generally give more accurate results.
- Say how likely it is that A is better than B.

Example 1

Suppose you have two versions of a website A and B. Want to examine which is better based on click through rate (CTR).

- Model initial belief (the prior) on the CTR with a distribution. Since the CTR is between 0 and 1, a standard choice is a *Beta Distribution*:

$$\frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$$

- Two parameters α and β.
- α is $1 +$ an initial guess (or result of a previous experiment) for the number of click throughs.
- β is $1 +$ the initial guess (or result of a previous experiment) for the number of non click throughs.
- Can start with $\alpha = \beta = 1$ (uniform distribution).

Example 1 (cont'd)

Run an experiment: show each website to N people. We can model the the likelihood that we observe n clicks out of N trials assuming click through rate x as a *Binomial Distribution*

$$\binom{N}{n} x^n (1-x)^{N-n}$$

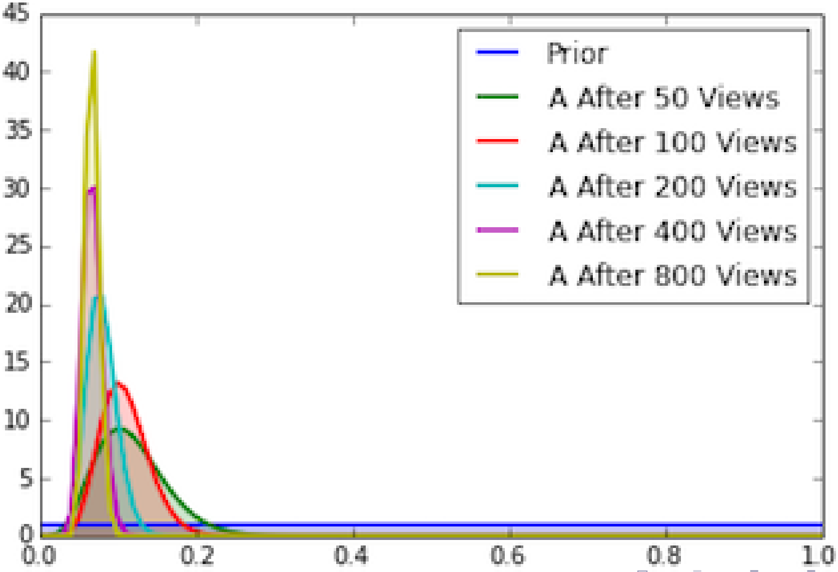
Example 1 (con't)

Bayes' Theorem =>

$$posterior =$$
$$x^n(1-x)^{N-n}x^{\alpha-1}(1-x)^{\beta-1} =$$
$$x^{\alpha+n-1}(1-x)^{\beta+N-n-1}$$

- Beta distribution with parameters $\alpha + n$ and $\beta + N - n$.

Example 1 (cont'd)



Example 1 (cont'd)

Objective: Learn if CTR of A is better than B:

- Show each website to N people to obtain posterior distribution parameters $\alpha = 1 + Clicks$ and $\beta = 1 + N - Clicks$ for each website.
- Simulate: sample from both distributions and find the fraction with $CTR_A > CTR_B =>$ likelihood that A is better than B.
- Can also ask the likelihood that $CTR_A > CTR_B + 0.02$. Can't do this with an Frequentist test!!

Example 1 (cont'd)

```
from scipy import stats

sample_size = 10000
A_sample = stats.beta.rvs(1 + clicks_A,
                          1 + views_A - clicks_A,
                          size=sample_size)
B_sample = stats.beta.rvs(1 + clicks_B,
                          1 + views_B - clicks_B,
                          size=sample_size)
print sum(A_sample > B_sample) / float(sample_size)
```

Multi-Arm Bandit

Traditional A/B Testing

- Equal number of observations for A and B.
- Stop the test and use better site.
- Waste time showing users the site that you'll end up not using.

Multi-Arm Bandit

- Show user the site you think is the best most of the time.
- Update belief about true CTR of that site.
- Repeat the process.

Why is it called that?

Multi-Arm Bandit refers to a mathematical decision problem modeled as n slot machines (bandits) with unknown expected payout.

- Goal: choose order to play the machines => maximize expected payout.
- No *a priori* knowledge about the machines.
- An algorithm to solve the Multi-Arm Bandit minimizes the *regret*

$$regret = \sum_{i=1}^{k}(p_{opt} - p_i) = k * p_{opt} - \sum_{i=1}^{k} p_i$$

- p_{opt} = optimal payout.
- p_i = observed payout.

Epsilon-Greedy algorithm

- With probability ϵ (usually 10%), choose a random bandit.
- With probability $1 - \epsilon$ choose the bandit with the highest expected payout based on past performance.
- Update the performance of the bandit selected.

UCB1 Algorithm

- Choose a bandit where

$$p_A + \sqrt{\frac{2\log N}{n_A}}$$

  is the largest.
- p_A is the expected payout of bandit A
- n_A is the number of times bandit A has been played
- N is the total number of trials so far.

Softmax

- Choose a bandit randomly proportional to their payouts. i.e. if A, B, and C are bandits the probability that you pick A is

$$\frac{e^{CTR_A/\tau}}{e^{CTR_A/\tau} + e^{CTR_B/\tau} + e^{CTR_C/\tau}}$$

- τ is a parameter that controls the 'randomness' of the choice, usually 0.001.

Bayesian Bandit

- Model the win rate of each bandit with a beta distribution.
- $\alpha = (1 +$ number of times the bandit has won)
- $\beta = (1 +$ number of times bandit has lost).
- Take a random sample from each bandit distribution and choose the best one.

Bayesian Bandit

Simulation of 3 bandits with “true” win rates 0.1, 0.3, 0.2: