Advanced A/B Testing

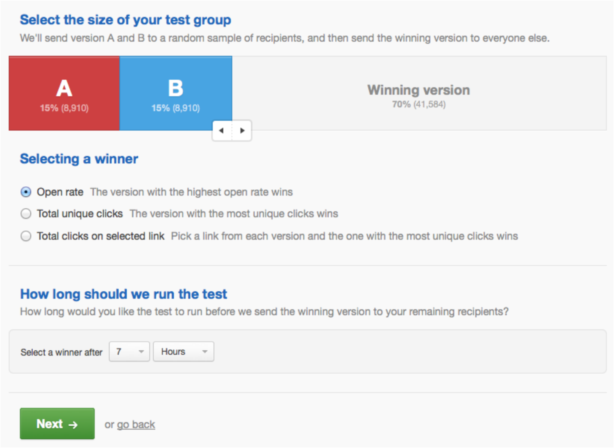
Profit-Maximizing A/B Tests

Elea McDonnell Feit

6/16/2019

Test & Roll

Typical A/B email test setup screen



Hypothesis testing doesn't quite fit this problem

1. Hypothesis tests focus on minimizing Type I error
   - Doesn't matter when we are deciding which of two equal-cost treatments to deploy
2. Populations are limited and hypothesis tests don't recognize this
   - Sample size formulas will suggest sample sizes larger than the population
3. When a hypothesis test is insignificant, it doesn't tell you what to do.
   - Choose randomly? That doesn't make sense!
4. Doesn't allow for unequal group sizes
   - But we see these all the time in media holdout testing

A/B tests as a decision problem

Test

Choose n_1^* and n_2^* customers to send the treatments.
Collect data on response.

Roll

Choose a treatment to deploy to the remaining $N - n_1^* - n_2^*$.

Objective

Maximize combined profit for test stage and the roll stage.

Profit-maximizing sample size

For the case where response is normally distributed with variance s and a symmetric normal prior on the mean response ($m_1, m_2 \sim N(\mu, \sigma)$), the profit maximizing sample size is

$$n_1 = n_2 = \sqrt{\frac{N}{4}\left(\frac{s}{\sigma}\right)^2 + \left(\frac{3}{4}\left(\frac{s}{\sigma}\right)^2\right)^2} - \frac{3}{4}\left(\frac{s}{\sigma}\right)^2$$

If the priors are different for each group (eg a holdout test), the optimal sample sizes can be found numerically. This new sample size formula was recently derived by Feit and Berman (2019) *Marketing Science*.

Test & Roll in math

Response

$$y_1 \sim N(m_1, s) \quad y_2 \sim N(m_2, s)$$

Priors

$$m_1 \sim N(\mu, \sigma) \quad m_2 \sim N(\mu, \sigma)$$

Profit-maximizing sample size

$$n_1 = n_2 = \sqrt{\frac{N}{4}\left(\frac{s}{\sigma}\right)^2 + \left(\frac{3}{4}\left(\frac{s}{\sigma}\right)^2\right)^2} - \frac{3}{4}\left(\frac{s}{\sigma}\right)^2$$

Interpreting the sample size formula

Bigger population (N) $\rightarrow$ bigger test

More noise in the repsonse (s) $\rightarrow$ bigger test

More prior difference between treatments (σ) $\rightarrow$ smaller test

$$n_1 = n_2 = \sqrt{\frac{N}{4}\left(\frac{s}{\sigma}\right)^2 + \left(\frac{3}{4}\left(\frac{s}{\sigma}\right)^2\right)^2} - \frac{3}{4}\left(\frac{s}{\sigma}\right)^2$$

Test & Roll procedure

1. Come up with priors distributions for each treatment
   - Use past data, if you've got it
2. Use the priors to compute the optimal sample size
3. Run the test
4. Deploy the treatment with the higher posterior to the remainder of the population
   - Priors are symmetric $\rightarrow$ pick the treatment with the higher average

Come up with priors

Hierarchical Stan model for past experiments

```
// Stan code for Lewis and Rao 2015 data
// L&R only report the mean and standard deviation for the
data {
  int<lower=1> nexpt; // number of experiments
  real<lower=2> nobs[nexpt]; // sample size for control gro
  real ybar[nexpt]; // observed mean for control group
  real<lower=0> s[nexpt]; // observed standard deviation fo
}
parameters {
  real m[nexpt]; // true mean for control group in experime
  real mu; // mean across experiments
  real<lower=0> sigma; //standard deviation across experime
}
model {
  // priors
  mu ~ normal(0, 10);
  sigma ~ normal(0, 3);
```

Fit hierarchical model to past experiments

```
lr <- read.csv("display_LewisRao2015Retail.csv")
# data taken from tables 1 and 2 of Lewis and Rao (2015)
c <- c(1:3,5:6) # include only advertiser 1 and eliminate 
d1 <- list(nexpt=length(c), nobs=lr$n1[c], ybar=lr$m[c], s=
m1 <- stan(file="test_roll_model.stan", data=d1, seed=2003
           iter=10000)
```

```
##
## SAMPLING FOR MODEL 'test_roll_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 1.7e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per tr
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 1: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 1: Iteration: 3000 / 10000 [ 30%]  (Warmup)
```

Fitted model

```
summary(m1)$summary[,c(1,3,5,8)]
```

```
##              mean         sd        25%      97.5%
## m[1]     9.490377 0.08467993   9.433258   9.655464
## m[2]    10.500796 0.10008523  10.433861  10.698501
## m[3]     4.860131 0.06181156   4.818066   4.981305
## m[4]    11.470157 0.07017636  11.422815  11.609507
## m[5]    17.615434 0.09047702  17.553716  17.792088
## mu      10.352358 2.00230021   9.138484  14.169450
## sigma    4.398852 1.17817514   3.540116   7.193112
## lp__   -13.182626 1.90558956 -14.218789 -10.511587
```

Compute optimal sample size

```r
source("nn_functions.R")
```

```
## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel
```

```r
(n <- test_size_nn(N=1000000, s=mean(d1$s), mu=10.36044, s
```

```
## [1] 11390.89 11390.89
```

Evaluate the test

```
(eval <- test_eval_nn(n=n, N=1000000, s=mean(d1$s), mu=10.3
```

```
##          n1       n2 profit_per_cust   profit profit_test
## 1 11390.89 11390.89        12.72715 12727151    236029.3
##   profit_rand profit_perfect profit_gain      regret err
## 1    10360440       12840843   0.9541639 0.008853939 0.0
##   tie_rate
## 1        0
```

Compare to sample size for hypothesis test

Null hypothesis test size to detect difference between:
- display ads that have no effect - display ads that are exactly worth the costs (ROI = 0 versus ROI = -100).

```
margin <- 0.5
d <- mean(lr$cost[c])/margin
(n_nht <- test_size_nht(s=mean(d1$s), d=d))
```

```
## [1] 4782433
```

Sample size for hypothesis test with finite population correction

```
(n_fpc <- test_size_nht(s=mean(d1$s), d=d, N=1000000))
```

```
## [1] 452673.4
```

```
(eval_fpc <- test_eval_nn(c(n_fpc, n_fpc), N=1000000,
                          s=mean(d1$s), mu=10.36044, sigma=
```

```
##          n1       n2 profit_per_cust   profit profit_test
## 1 452673.4 452673.4        10.59508 10595077      9379790
##   profit_rand profit_perfect profit_gain    regret error
## 1    10360440       12840877  0.09459509 0.1748946 0.011
##   tie_rate
## 1        0
```

Comparison of display ad tests

	n_1	n_2	Expected Sales ($000) Test	Roll	Overall	Regret	Roll Error
No Test (Random)	-	-	-	-	10,360	19.32%	50.0%
Standard Hyp. Test*	4,782,433*	4,782,433*	n/a	n/a	n/a	n/a	n/a
Hyp. Test FPC**	452,673	452,673	9,380	1,125	10,595	17.5%	1.1%
Test & Roll	11,391	11,391	236	12,491	12,727	0.89%	6.9%
Thompson Sampling	-	-	-	-	12,803	0.29%	-
Perfect Information	-	-	-	-	12,840	0%	-

Multi-armed bandits

Multi-armed bandits

Multi-armed bandits are a dynamic profit-maximizing approach that is more flexible than a test & roll experiment. They are often referred to as the “machine learning for the A/B testing world.”

Source: personal photo from Ceasar’s Palace, Las Vegas

Multi-armed bandit process/problem

1. Define treatment probabilities p_k
2. Asssign one or a few units to treatments with probability for each treatment k
3. Collect data
4. Adjust p_k's based on the data
5. Repeat

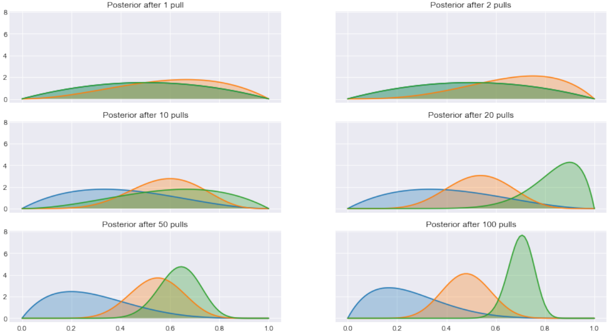
Thompson sampling

A popular approach multi-armed bandit problems was proposed by Thompson in 1933.

1. Start with prior distributions on the performance of each treatment
2. Assign units to treatments based on the probability that the treatment is best
3. Collect data
4. Update priors
5. Repeat

There are other methods that work better in specific contexts, but Thompson sampling is very robust.

Thompson sampling for 3 treatments



Source: eigenfoo.xyz

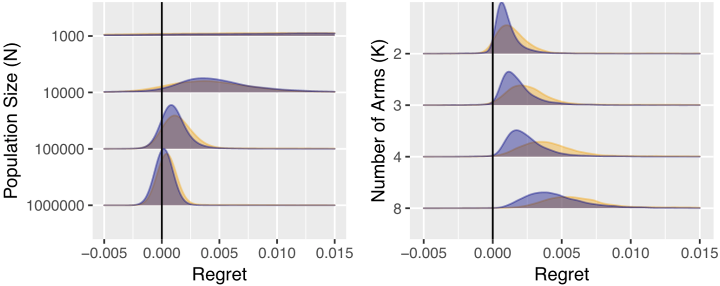
How do Thompson sampling and Test & Roll compare?

Both methods are profit-maximizing. We can compare them based on how much profit they generate.

Thompson sampling is less constrained, so will always produce more profit on average.

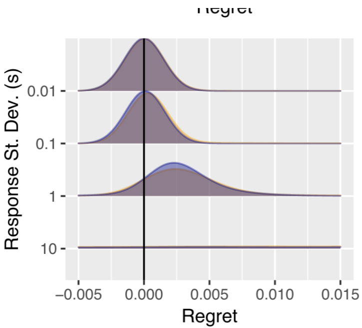
Statisticans are a pessimistic lot, so we prefer to compute **regret** for an algorithm, which is the difference between profit with perfect information and profit with the algorithm.

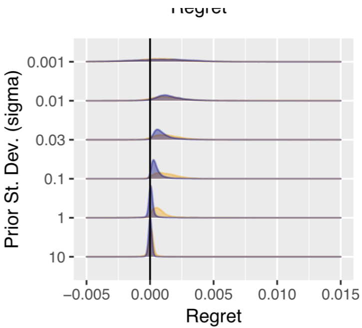
Comparison of regret for Thompson sampling and Test & Roll



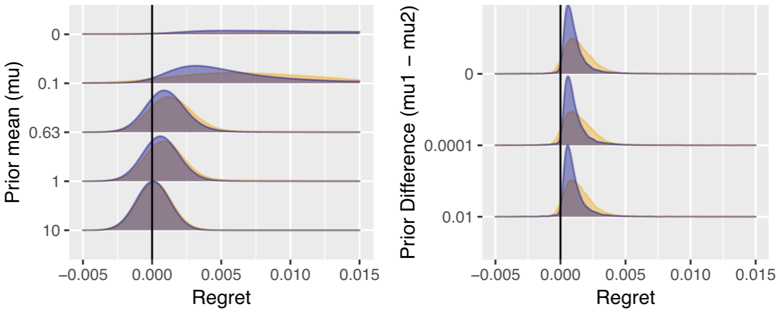
Source: Feit and Berman 2019

Comparison of Thompson sampling and Test & Roll





Comparison of Thompson sampling and Test & Roll



Why do Test & Roll?

- Works when response takes a long time to measure
  - Long purchase cycles
- Works when iterative allocation would be time-consuming
  - Email, catalog and direct mail
- Reduces complexity for website tests
  - Don't need bandit interacting with site

Test & Roll profit-maximizing sample size can be used as a conservative estimate of how long to run a bandit algorithm.

Things you just learned

- Test & Roll experiments
  - Profit-maximizing sample size
- Multi-armed bandits
  - Thompson sampling