Bayesian A/B testing with Python

Bogdan Kulynych

July 14, 2014

A/B testing

Behavioral research study designed to answer specific questions about behavioral interventions. Theoretical developments come from clinical trials.

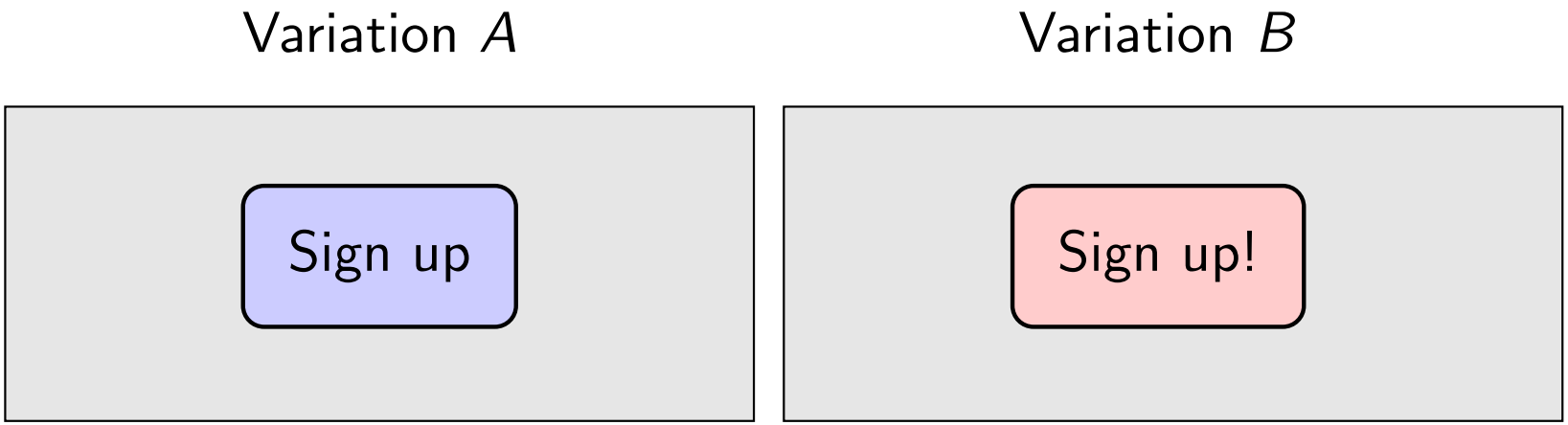
We focus on Web development:

- Small set of (web page) variations: $A, B, C, \ldots$
- Metrics to compare variations: profit gained, time spent on the page, signup rate
- Use of statistical methods to estimate the metrics

To be contrasted to *personalization*.

Classical approach

Suppose there are two variations of Web page design: A and B.



- We want to find out which one will probably produce more signups
- Randomly show visitors one of the variations and log the results

Classical approach

Model

- Let A, B be finite binary populations $A^{(i)}, B^{(i)}$. $a^{(i)}, b^{(i)} \in \{0, 1\}$.
- Fix true signup rates p_A, p_B:

$$A \sim \text{Bernoulli}(p_A), \;\; B \sim \text{Bernoulli}(p_B)$$

- Assume that by logging views and signups we obtain random samples of the populations X_A, X_B:

$$x_A^{(i)} \sim \text{Bernoulli}(p_A), \;\; x_B^{(i)} \sim \text{Bernoulli}(p_B) \text{ are i.i.d RVs}$$

- We want to estimate the difference between true population parameters p_A, p_B. They are *fixed but unknown*.

Classical approach

Hypothesis testing

Two sample t-test is often used:

$$H_0 : p_A = p_B, \quad H_1 : p_A > p_B, \quad H_2 : p_A < p_B$$

Compute T-statistic:

$$T = \frac{\bar{X}_A - \bar{X}_B}{\sigma_{\hat{p}_A - \hat{p}_B}}$$

where $\hat{p}_A = \bar{X}_A$, $\hat{p}_B = \bar{X}_B$ are sample average values, and

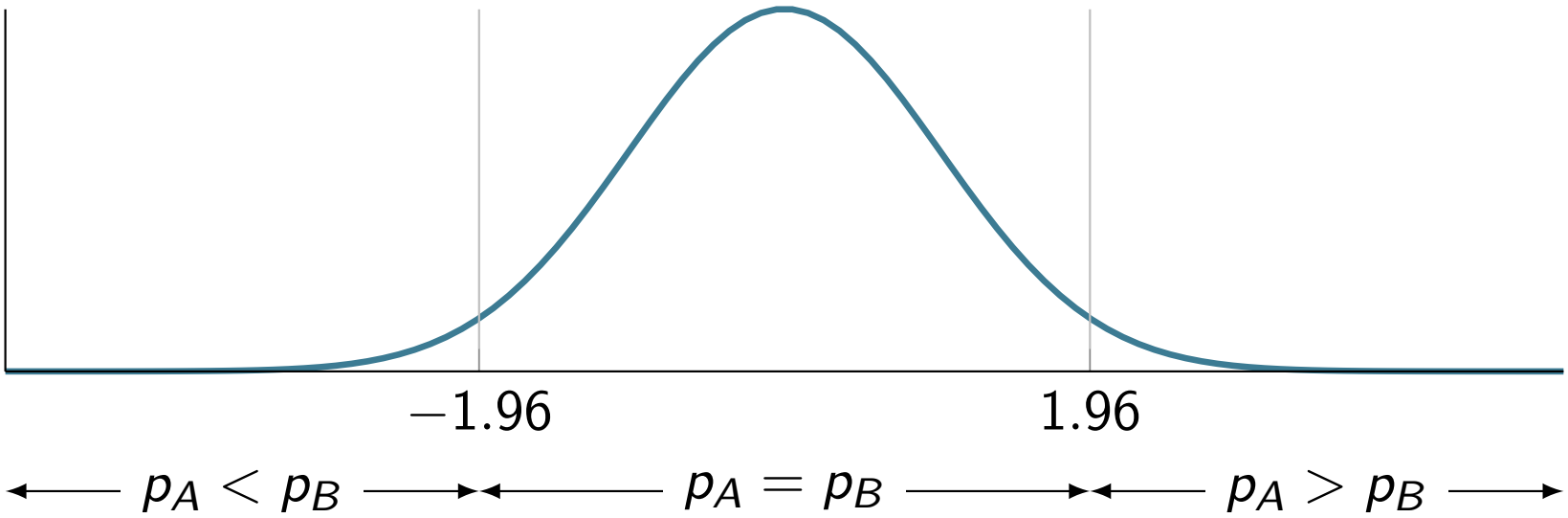
$$\sigma^2_{\hat{p}_A - \hat{p}_B} = \frac{\sigma^2_{X_A}}{n_{X_A}} + \frac{\sigma^2_{X_B}}{n_{X_B}}$$

Classical approach

Hypothesis testing

Under $H_0 : p_A = p_B$ T-statistic is t-distributed for small sample sizes, and standard normal distributed if sample sizes are big enough.

$P(X \mid H_0)$ can be found then. Example for $\alpha = 0.95$ and standard normal distribution:



Classical approach

Problems

- Easy to misuse: α controls only type I errors, type II are often forgotten. For every parameter value, there's a certain sample size that needs to be obtained before drawing any conclusions.
- Reliance on large numbers: LLN and CLT.
- Reasoning based on P(data | hypothesis) and finite amount of hypotheses about the fixed parameters. P(parameters | data) is arguably more appropriate and more general, but doesn't make sense within the given model (*parameters are fixed*).
- Confidence intervals like $I : P(\text{parameter} \in I \mid \text{data}) = \gamma$ often misinterpreted: it generally does not mean that *probability of parameter being in the interval* I *is* γ, since parameters are fixed.

Bayesian approach

Model

- *Let true signup rates p_A, p_B be independent random variables.* Let *prior* distributions of p_A and p_B be Beta-distributed:

$$p_A \sim \text{Beta}(\alpha_A, \beta_A), \;\; p_B \sim \text{Beta}(\alpha_B, \alpha_B)$$

- Assume that the likelihood of data obtained by logging views and signups is binomial. Let number of signups $k_A = |\{x_A^{(i)} = 1\}|$, number of views $n_B = |X_A|$:

$$\text{P}(X_A \mid p_A) = \text{Binomial}(k_A; n_A, p_A).$$

  Analogically, for p_B.

- We want to find $P(p_A > p_B \mid X)$, $P(p_A < p_B \mid X)$ and *lifts* $\frac{p_B - p_A}{p_A}$ and $\frac{p_A - p_B}{p_B}$.

Bayesian approach

Bayes theorem

$$
\begin{aligned}
\mathrm{P}(p_A \mid X_A) &= \frac{\mathrm{P}(X_A \mid p_A) \cdot \mathrm{P}(p_A)}{\mathrm{P}(X_A)} \propto \mathrm{P}(X_A \mid p_A) \cdot \mathrm{P}(p_A) \\
&= \binom{n_A}{k_A} p_A^{k_A} (1-p)^{n_A - k_A} \; \frac{1}{B(\alpha_A, \beta_A) p_A^{1-\alpha_A} (1-p_A)^{1-\beta_A}} \\
&= \mathrm{Beta}(\alpha_A + k_A, \beta_A + n_A - k_A)
\end{aligned}
$$

$\mathrm{P}(p_A \mid X_A)$ is called *posterior* distribution. We can trivially compute point estimates $\mathrm{E}[p_A \mid X_A]$, *credible intervals* $I : \mathrm{P}(p_A \in I \mid X_A) = \gamma$.

Using Monte Carlo methods, we can compute $P(p_A < p_B \mid X)$ and lifts.

Bayesian approach

Summary

- Similar assumptions (independent Bernoulli trials, implied by Binomial likelihood)
- No reliance on big numbers (in theory)
- Instead of finding P(data | hypothesis) for a set of predefined hypotheses about the parameters, we integrate over all possible values of parameters and get P(parameter | data). This is more general approach than hypothesis testing.
- We can find *credible intervals* which are easy to interpret right.
- Using Monte Carlo techniques, we can calculate any function of the parameters we want (like lift), and assume any likelihoods we want (at the cost of computation)

Bayesian approach

Code

We can implement Bayesian A/B testing using *numpy* and *scipy*, and optionally *PyMC* for Monte Carlo methods.

```
import numpy as np
from scipy import stats

data = {
'A': { 'views': 42, 'signups': 2 },
'B': { 'views': 85, 'signups': 11 }
}

posteriors = { variation: stats.beta(logs['signups'],
    logs['views'] - logs['signups'])
      for variation, logs in data.items() }
```

Bayesian approach

Code (cont.)

- Point estimates:

```
print(posteriors['A'].mean())
```

$$\mathrm{E}[p_A \mid X] = 5.81\%, \ \mathrm{E}[p_B \mid X] = 13.37\%.$$

- 95%-Credible intervals

```
print(posteriors['A'].ppf(0.025), posteriors['A'].
    ppf(0.975))
```

$$\mathrm{P}(1.00\% < p_A < 14.41\%) = 0.95,$$
$$\mathrm{P}(7.07\% < p_B < 21.28\%) = 0.95$$

Bayesian approach

Code (cont.)

- Monte Carlo approach to compute $P(p_B > p_A \mid X) \approx \frac{1}{n} \sum_i \mathbb{I}[y_A^i > y_B^i]$ and expected lift:

```
size = 10000
samples = { variation: posterior.sample(size) for
    variation, posterior in posteriors.items() }

dominance = np.mean(samples['B'] > samples['A'])
lift = np.mean((samples['B'] - samples['A']) /
    samples['A'])
```

Variation B performs better, so $P(p_B > p_A) = 92.90\%$.
Expected lift of signup rate under variation B is 271.68%

Bayesian approach

trials library

I wrote a small library for running Bayesian A/B testing called *trials* that can do all of the above.

```
from trials import Trials

test = Trials(['A', 'B'], vtype='bernoulli')

test.update({
    'A': (2, 40),
    'B': (11, 79),
})
```

Bayesian approach

trials library — statistics

- $\mathrm{P}(p_A > p_B \mid X)$:

```
dominances = test.evaluate('dominance')
```

- Expected lift $\mathrm{E}(\frac{p_B - p_A}{p_A} \mid X)$:

```
lifts = test.evaluate('expected lift')
```

- Lift 95%-credible interval:

```
intervals = test.evaluate('lift CI', level=95)
```

Available statistics for Bernoulli experiments: expected posterior, posterior CI, expected lift, lift CI, empirical lift, dominance

Bayesian approach

trials library

Get it on `github.com/bogdan-kulynych/trials`.

Suggestions and corrections welcome.