Lecture 16: Sampling

Lisa Yan
August 1, 2018
Announcements

Problem Set 4 pain poll

Lost?
- Come to office hours!
General problem solving strategies

Step 1
Intuition
events, counting, examples
(if lost: is there another way to think about it?)

Step 2
Math toolbox
(once you have a good handle on what the problem means (step 1), then you can go to step 2)

Step 3
Sanity check!
• Valid probabilities?
• Reasonable expectations?
• Example checks out?
• All independence is justified?
• Assumptions not violated?

Step 4
Reflect
(if problem set: what was this question trying to test? What concepts did you use?)
Summary of last time

Conditional expectation: \[ E_Y[E_X[X|Y]] = E_X[X] \]

- Expectations of complex functions, like \( E[\sum_{i=1}^{X} Y] \)
- Analyze recursive code!

Quicksort:

- While recursive, can be solved as an expectation of a sum of indicator random variables.
- When dealing with a sum of non-trivial indicator probabilities,
  \[ \sum_{x=k}^{n} \frac{1}{ax+b} \approx \int_{x=k}^{n} \frac{1}{ax+b} \, dx \]

(QuickSort is beyond the scope of your HW, but you should understand it)
Where are we going with all of this?

As engineers, we want to:
1. Model things that can be random.
2. Find average values of situations that let us make good decisions.

We perform multiple experiments to help us with engineering.

Counts and averages are both sums of RVs.

Weeks 4 and 5:
Learning how to model multiple RVs

This week (week 6):
• Finding expectation of complex situations
• **Defining sampling and using existing data**
• Mathematical bounds w.r.t modeling the average

Week 7: bringing it all together
• Sampling + average = Central Limit Theorem
• Samples + modeling = finding the best model parameters given data
Goals for today

Sampling

◦ Sample statistics
◦ Bootstrapping: Estimating distribution from samples
◦ Statistical significance and p-values

Probability

Applications
Common types of experiments

**Simulation:**

Create model of system → Many experiment simulations → Report % accuracy

(DNA simulations, network traffic, etc.)

**Machine Learning:**

Learn model of system → Do stuff with model

(image classification, robot operation)

**Social science:**

Pre-test humans (control) → Perform one experiment → Post-test humans → Get statistics

(treatment effect, election polling, etc.)

**Single experiment zero model**
Sampling

Q: Rate Stanford from 1 to 10.

Population:
- Each person’s $X_i$ attribute i.i.d. as $F$, where $E[X_i] = \mu$ and $\text{Var}(X_i) = \sigma^2$

WTF: $\mu$ and $\sigma^2$ of $F$
(Impossible to ask all people)

Sampling: choose $n$ people
- Compute averages to estimate $\mu$ and $\sigma^2$
Averaging example

\[ X_1 = 37 \]
\[ X_2 = 53 \]
\[ X_3 = 34 \]
\[ X_4 = 70 \]
\[ X_5 = 59 \]
\[ X_6 = 29 \]
\[ X_7 = 48 \]
\[ X_8 = 81 \]

(sample with sample size 8)

\[ \bar{X} = \frac{1}{8} \sum_{i=1}^{8} X_i \approx 51.4 \]
Sample mean

Consider $n$ i.i.d random variables $X_1, X_2, \ldots X_n$, where

- $X_i$ have distribution $F$ with $E[X_i] = \mu$ and $\text{Var}(X_i) = \sigma^2$

Define $X_1, X_2, \ldots X_n$ is a sample from distribution $F$

$n$ is our sample size.

Define sample mean:

$$\bar{X} = \frac{1}{n} (X_1 + X_2 + \cdots + X_n) = \frac{1}{n} \sum_{i=1}^{n} X_i$$

- This is a RV!! $\bar{X} = g(X_1, X_2, \ldots, X_n)$
- Depending on our sample, sample mean can differ
- We want sample mean to be unbiased, i.e., $E[\bar{X}] = \mu$

Goal: $\mu$ and $\sigma^2$ of $F$

(if we sampled many times, the average of sample means should be our target $\mu$)
The unbiased sample mean

\[
\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \quad \text{Goal: } \mu \text{ and } \sigma^2 \text{ of } F
\]

Strategy to get \( \mu \):

1. Experiment many times (get \( k \) different samples of size \( n \))
2. Take sample means of all different \( k \) samples
3. Average \( k \) sample means

Prove that this strategy will work (i.e., that \( E[\bar{X}] = \mu \)).

Proof:

\[
E[\bar{X}] = E \left[ \frac{1}{n} \sum_{i=1}^{n} X_i \right] \quad (\text{definition of sample mean})
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} E[X_i] \quad (\text{linearity of expectation})
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} \mu = \frac{1}{n} (n\mu) = \mu \quad (E[X_i] = \mu \text{ as defined})
\]
Sample mean

Consider \( n \) i.i.d random variables \( X_1, X_2, \ldots, X_n \), where \( X_i \) have distribution \( F \) with \( \mathbb{E}[X_i] = \mu \) and \( \text{Var}(X_i) = \sigma^2 \)

Sample mean, \( \bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \)

- Mean (unbiased): \( \mathbb{E}[\bar{X}] = \mu \)
- Variance: \( \text{Var}(\bar{X}) = \frac{\sigma^2}{n} \)

Proof of variance:

\[
\text{Var}(\bar{X}) = \text{Var} \left( \frac{1}{n} \sum_{i=1}^{n} X_i \right) = \left( \frac{1}{n} \right)^2 \text{Var} \left( \sum_{i=1}^{n} X_i \right) = \left( \frac{1}{n} \right)^2 \sum_{i=1}^{n} \text{Var}(X_i)
\]

(\( \text{Var}(X) = a^2 \text{Var}(X) \))

\( (X_1, X_2, \ldots, X_n \) independent \)

\[
= \left( \frac{1}{n} \right)^2 \sum_{i=1}^{n} \sigma^2 = \left( \frac{1}{n} \right)^2 n \sigma^2 = \frac{\sigma^2}{n}
\]
Sample variance

Consider $n$ i.i.d random variables $X_1, X_2, \ldots X_n$, where

- $X_i$ have distribution $F$ with $E[X_i] = \mu$ and $\text{Var}(X_i) = \sigma^2$
- We define $X_1, X_2, \ldots X_n$ as a sample from distribution $F$

**def** sample variance:  
$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2$$

Note: $X_i - \bar{X}$ is sample deviation for $i = 1, \ldots, n$

Sample variance as defined is an *unbiased* estimator of the population variance ($\sigma^2$), i.e., $E[S^2] = \sigma^2$  
(proof in lecture notes/next slide)
The unbiased sample variance

Prove that \( E[S^2] = \sigma^2 \) (where \( E[X_i] = \mu, \text{Var}(X_i) = \sigma^2 \)).

\[
E[S^2] = E \left[ \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2 \right] \implies (n-1)E[S^2] = \sum_{i=1}^{n} E[(X_i - \bar{X})^2]
\]

\[
(n-1)E[S^2] = E \left[ \sum_{i=1}^{n} ((X_i-\mu) + (\mu - \bar{X}))^2 \right]
\]

\[
= E \left[ \sum_{i=1}^{n} (X_i - \mu)^2 + \sum_{i=1}^{n} (\mu - \bar{X})^2 + 2 \sum_{i=1}^{n} (X_i - \mu)(\mu - \bar{X}) \right]
\]

\[
= E \left[ \sum_{i=1}^{n} (X_i - \mu)^2 + \sum_{i=1}^{n} (\mu - \bar{X})^2 + 2(\mu - \bar{X}) \sum_{i=1}^{n} (X_i - \mu) \right]
\]

\[
= E \left[ \sum_{i=1}^{n} (X_i - \mu)^2 + n(\mu - \bar{X})^2 + 2(\mu - \bar{X})n(\bar{X} - \mu) \right]
\]

\[
= E \left[ \sum_{i=1}^{n} (X_i - \mu)^2 + n(\mu - \bar{X})^2 + 2(\mu - \bar{X})n(\bar{X} - \mu) \right]
\]

\[
= E \left[ \sum_{i=1}^{n} (X_i - \mu)^2 - n(\mu - \bar{X})^2 \right]
\]

\[
\therefore E[S^2] = \sigma^2
\]
A key difference

Variance of sample mean $\bar{X}$:

$$\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$$

- A single number
- Not computable from sample
- Shrinks with # sample size $n$
- Measures deviation from population mean ($E[\bar{X}] = \mu$)

**def** standard error – approximate variance of sample mean, $SE = \sqrt{\frac{S^2}{n}}$

Sample variance, $S^2$:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2$$

- A random variable
- Computable from sample
- In expectation is the population variance
Summary so far

→ Population mean $\mu$ and population variance $\sigma^2$

→ How do I approximate my population statistics? 
Sample mean $\bar{X}$ and sample variance $S^2$

→ How close is my approximation of the population mean? 
Standard error $SE = \sqrt{\frac{S^2}{n}}$

(approximation of sample mean’s standard deviation $\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$ 
i.e., deviation from population mean)
Python demo

https://github.com/yanlisa/bootstrap_demo
Break

Attendance: tinyurl.com/cs109summer2018
## Sample statistics

<table>
<thead>
<tr>
<th>Population statistic</th>
<th>Sample statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>(each person $X_i \sim F$)</td>
<td>(RVs)</td>
</tr>
<tr>
<td>Population mean $\mu$</td>
<td>Sample mean $\bar{X}$</td>
</tr>
<tr>
<td>Population variance $\sigma^2$</td>
<td>Sample variance $S^2$</td>
</tr>
<tr>
<td>Probabilities</td>
<td>Counts in sample</td>
</tr>
<tr>
<td>...and more!</td>
<td>Standard error</td>
</tr>
<tr>
<td></td>
<td>...and more!</td>
</tr>
</tbody>
</table>
So far...

We want to compute **statistics** about our population...

**Want to find (WTF):**

- P (average distance Lisa can throw candy is b/t 2.1 and 2.5 meters)
- P (this quarter’s midterm average was really different from previous quarter’s)

**Sample:**

- 2.2m, 1m, 2m, 7m, 0.05m, …, (on a particular day)

**General ver.:**

- estimate of the population mean
- probability of this difference in sample means

...but we only have our single sample \((X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)\) to work with.
So far...

If we have many statistics from many samples:

• Average sample statistics from all = our actual population statistic. (if unbiased)

But what if we only have one experiment?

• Find the probability of observing the sample statistic that we got

But we don’t know the underlying distribution $F$!

Solution:

Bootstrapping
Bootstrapping

Legend:
- The only PMF we have (the histogram of our sample)
- Possible PMFs (infinitely many)

Intuitions:
1. We only know what we know (we only have the samples that we have)
2. We can simulate the experiment using technology! (aka computers)
3. Simulation lets us compute a distribution over our statistic over many, many, many experiments

Finding the distribution of our population
Bootstrapping

If we have an unknown distribution $F$:

You can estimate the PMF of the underlying distribution $F$ using your samples.

$$F \approx \hat{F}$$

(the underlying distribution)  (the sample distribution)
Bradley Efron

Bradley Efron is an American statistician (1938–)

Professor at Stanford (since 1964) working on astrophysics and biostatistics

Still teaches classes (STATS 305B)

Efron’s dice: 4 dice A,B,C,D such that

\[
P(A > B) = P(B > C) = P(C > D) = P(D > A) = 2/3
\]
The Bootstrap algorithm

“aka simulate experiment a bunch of times”

```python
def bootstrap(sample):
    pmf = fancy_estimate_distribution(sample)
    results = []
    for i in range(10000):
        resample = pmf.sample(size=len(sample))
        stat = compute_stat(resample)
        results.append(stat)
    return results
```

(In this class, just use histogram of sample)
The Bootstrap algorithm

“aka simulate experiment a bunch of times”

def bootstrap(sample):
    pmf = fancy_estimate_distribution(sample)
    results = []
    for i in range(10000):
        resample = pmf.sample(size=len(sample))
        stat = compute_stat(resample)
        results.append(stat)
    return results

(In this class, just use histogram of sample)

(# times to perform experiment)
The Bootstrap algorithm

“aka simulate experiment a bunch of times”

def bootstrap(sample):
    pmf = fancy_estimate_distribution(sample)
    results = []
    for i in range(10000):
        resample = pmf.sample(size=len(sample))
        stat = compute_stat(resample)
        results.append(stat)
    return results

(In this class, just use histogram of sample)

(# times to perform experiment) (get a sample)
The Bootstrap algorithm

“aka simulate experiment a bunch of times”

def bootstrap(sample):
    pmf = \textit{fancy\_estimate\_distribution}(sample)
    results = []
    for i in range(10000):
        resample = pmf.sample(size=len(sample))
        stat = \textit{compute\_stat}(resample)
        results.append(stat)
    return results

(In this class, just use histogram of sample)

(# times to perform experiment)
(get a sample)
(sameple mean, etc.)
The Bootstrap algorithm

“aka simulate experiment a bunch of times”

def bootstrap(sample):
    pmf = fancy_estimate_distribution(sample)
    results = []
    for i in range(10000):
        resample = pmf.sample(size=len(sample))
        stat = compute_stat(resample)
        results.append(stat)
    return results

(In this class, just use histogram of sample)

(# times to perform experiment)
    (get a sample)
    (sample mean, etc.)

(returns distribution of your statistic)
We want to compute **statistics** about our population...

Want to find (WTF):

P (average distance Lisa can throw candy is b/t 2.1 and 2.5 meters)

Sample:

2.2m, 1m, 2m, 7m, 0.05m, ..., (on a particular day)

General ver.:

estimate of the **population mean**

(specifically, a **confidence interval** about where we think the population mean may lie)

...but we only have our single sample \((X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)\) to work with.
Understanding the bootstrap

Your real sample (say, candy throws on Monday 7/30)

- N candy throw distances (2.2m, 1m, 2m, 7m, 0.05m, ...)
- Sample mean: 2.3 m

Want to see: \( P(\text{population mean is between } 2.1 \text{ to } 2.5 \text{ m}) \)

1. Make your PMF
   \[ P(X = x) = \frac{\# \text{ points } = x}{N} \]

2. Repeat 10,000 times:
   a. Draw N candy distances with replacement
   b. Compute sample mean and save

3. Now you have a sample mean distribution!

4. Find \# sample means between 2.1 and 2.5 in distribution
def bootstrap(sample):
    pmf = histogram_count(sample)
    results = []
    for i in range(10000):
        resample = pmf.sample(size=len(sample))
        stat = compute_stat(resample)  # sample mean
        results.append(stat)
    return results  # distribution of sample means
def bootstrap(sample):
    pmf = histogram_count(sample)
    results = []
    for i in range(10000):
        resample = pmf.sample(size=len(sample))
        stat = compute_stat(resample)  # sample mean
        results.append(stat)
    return results  # distribution of sample means
Understanding the bootstrap

```python
def bootstrap(sample):
    pmf = histogram_count(sample)
    results = []
    for i in range(10000):
        resample = pmf.sample(size=len(sample))
        stat = compute_stat(resample)  # sample mean
        results.append(stat)
    return results  # distribution of sample means
```

results = [1.9, 2.8]
Understanding the bootstrap

```python
def bootstrap(sample):
    pmf = histogram_count(sample)
    results = []
    for i in range(10000):
        resample = pmf.sample(size=len(sample))
        stat = compute_stat(resample)  # sample mean
        results.append(stat)
    return results  # distribution of sample means
```

results = [1.9, 2.8, 3.7, 0.9, 1.4, 2.3, 2.6, 2.8, ...]
Understanding the bootstrap

means = [1.9, 2.8, 3.7, 0.9, 1.4, 2.3, 2.6, 2.8, ...]

What is probability that the population mean is in the range 2.1 to 2.5?

confidence interval of X%: \( P(2.1 \leq \text{pop. mean} \leq 2.5) = X\% \)
What is the confidence interval associated with the belief that the average Water Pokemon speed stat is within ±1 bootstrapped standard error of the sample mean?

https://github.com/yanlisa/bootstrap_demo

Interpret: What is $P(| \text{pop. mean} - \text{sample mean} | < 1 \text{ boot SE})$?

Interpret: How off is our estimate of the population mean?
Bootstrap

1. Assume your sample distribution is population distribution.
2. Get many samples and calculate some statistic
3. Get a statistic distribution
4. Report probabilities on your statistic distribution

Use cases:
1. **Confidence interval** of population mean
2. Whether two samples are significantly different
3. Many more...

(we just did this)
(next)
(in your future)
So far...

We want to compute statistics about our population...

Want to find (WTF):

\[ P \left( \text{this quarter’s midterm average was really different from previous quarter’s} \right) \]

Sample:

- This quarter: 82.5%, 46.3%, ...
- Last quarter: 73.2%, 90.1%, ...

General ver.:

probability of this difference in sample means

(specifically, a p-value about how unlikely we think this difference arises due to chance)

...but we only have our single sample \( (X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) \) to work with.
Statistical significance

Sampling by nature will have different fluctuations.

def statistical significance of a result – whether this result would be very unlikely to happen due to chance.

Want to see: Is this statistic that we obtained significant?

Interpretation: What is the probability of seeing this statistic if the only differences between different observations are due to chance alone?
**p-values**

def **Alternate hypothesis**: There is a significant difference in sample means of groups A and B, tosses of a coin, etc.  
(our goal)

def **Null hypothesis**, \(H_0\): coin is fair, groups are from same populations, etc.

def **p-value**: What is \(P(\text{getting a result } | \ H_0)\)?  
(what we assume)

- Small \(p\) (< .01): likelihood that observation was due to random sample error is low \(\Rightarrow\) statistically significant  
(reject null)
- Any other \(p\): Cannot say anything about significance of observation  
(do not reject null)

Note that \(p\)-value is not an error rate! \(P(\text{observation } | \ H^0) \neq P(H^0)\)
Bootstrap for p-values

How often will we get a difference greater than the observed difference?

```python
def bootstrap_p(sample1, sample2):
    observed_diff = np.abs(np.mean(sample1) - np.mean(sample2))
    null_sample = sample1 + sample2
    null_pmf = histogram_count(null_sample)
    count = 0
    for i in range(10000):
        resample1 = null_pmf.sample(size=len(sample1))
        resample2 = null_pmf.sample(size=len(sample2))
        resample_diff = np.abs(np.mean(resample1) - np.mean(resample2))
        if resample_diff >= observed_diff:
            count += 1
    return count/10000  # p value
```

- **original statistic**
- **combine samples**
- **compute statistic**
- **update result**
- **return p-value**
Another bootstrap example

Two samples:

• N exam scores from this quarter, sample mean $\bar{X} = x_1$
• M exam scores from last quarter, sample mean $\bar{X} = x_2$

WTF: whether these sample means were significant different

Interpretation for bootstrap: find distribution of sample mean differences, then find probability we observe $>\text{the actual } |x_2 - x_1|$

1. Make your PMF
   
   $P(X = x) = \frac{(\# \text{ last quarter} == x) + (\# \text{ this quarter} == x)}{N + M}$

2. Repeat 10,000 times:
   a. Draw two groups of size N and M
   b. Compute difference in sample means and check if difference $> |x_2 - x_1|$

3. Compute p value
   
   $P(\text{difference} > |x_2 - x_1| | \text{same distribution})$

4. If p < 0.01, reject null hypothesis $\Rightarrow$ quarters significantly different
Python demo

1. Do Water Pokemon and Normal Pokemon have significantly different speed stats?

2. Are Beta(3,6) and Beta(6,3) significantly different?

(demo not posted; complete bootstrap function for HW)