The Bootstrap

Generating A Sample Distribution

- We have seen how to simulate in order to conduct tests in R.
- We can do the same thing for other measures of uncertainty.
-
  - Look at the distribution of parameter esti
    simulations
-
- Note: simulation depends on the paramete
- We can also evaluate bias.

Example: Estimating a Variance

Why we divide by $n-1$: consider $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$ for $n = 4$:

```
nsim = 1000
sigest = rep(0,nsim)

for(
  X = rn
  si
}
```

The expected value of the estimate is estimated by

```
mean(sigest)
```

and the bias (since the true variance is 1) is

```
> bias = 1-mean(sigest)
[1] -0.2484821
```

Working With Estimates

Simulation allows us to do a number of things

1. Bias correction: since we know the bias we can correct our estimate

$$\hat{\sigma}^{*2} = \hat{\sigma}^2 - bias$$

2. Standard errors for estimate, from standa simulated samples

```
> sighat.sd = sd(sigest)
[1] 0.6474583
```

3. Confidence intervals based on normal theory:

```
> sighat.nobias + c(-1,1)*qnorm(0.975)*sighat.sd
[1] -0.2116517  2.3263383
```

Alternative Confidence Intervals

Distribution of `sigest` strongly skewed: symmetric confidence intervals not appropriate.

Defining a lower limit $b_{\alpha/2}$:

- $\hat{\sigma}^2$ $\hat{\sigma}^2$
- m
- simulation

```
b.lower = quantile(sigest-sigma
b.upper = quantile(sigest-sigma
```

Analogous for upper limit.

- Confidence interval is then

```
c(sighat - b.lower, sighat - b.upper)
```

Note: no bias correction (why?)

Confidence Intervals Continued

CI's *reverses* and shifts the distribution of $\hat{\sigma}^2$.

- $\hat{\sigma}^2$ has a long right tail (can be much too high)
- So lower side of confidence interval needs to be longer to include true σ^2.

Note: simulation procedure work for any statistic $t(X_1, \ldots, X_n)$ that estimates a parameter θ.

Making Fewer Assumptions

Some important limitations to value of simulation:

- Only valid under the parameters you use to simulate.
  -
  -
  -
- Only valid assuming the distribution you si
  represents the data generating mechanis
  - If our data isn't Gaussian, simulation abo

Maybe we could make more use of the data.

The Bootstrap

Introduced by Efron (1979), $>$ 26,000 citations from all of NSF's funding areas.

Arguably most important statistical development in last 50 years.

Simple idea:

- F

Empirical Estimates of a Distribution

- Cumulative distribution function: $F(x) = P(X \leq x)$
- Estimate from data:

$$\frac{1}{n}$$

- $F_n(x)$ converges to $F(x)$ everywh
- Interpretation of Y drawn from F: *For each i, Y takes value X_i with p*
- Practically: to sample from F_n, choose one X_i at random.
- To sample more "re-sample with replacement": *each time you choose an X_i, keep it in the data set for the next sample*.

Sampling Schemes

Some general terminology (informal)

sample from F generate a random variable X according to distribution F (later in 3520)

r

e

of the data.

Different types of samples

bootstrap resample n observations wi

subsample resample $k < n$ observations without replacement.

Note: bootstrap samples will have repeated values; a subsample of size n is a permuation.

The Bootstrap Recipe

Given $X_1, \ldots, X_n$, and a statistic $t(X_1, \ldots, X_n)$ that estimates a parameter θ:

- For $b = 1, \ldots, B$:
  - Resample $X_1^*, \ldots, X_n^*$ from $X_1, \ldots, X_n$.
  - Record $T_b = t(X_1^*, \ldots, X_n^*)$.
- Use $T_1, \ldots, T_B$ to represent the sampl
- $T_o = t(X_1, \ldots, X_n)$.

sample

Will *resample* objects with or without replacement and will return a vector of required size.


```
# A permutation of the numbers 5:10
sample(5:10)

# A subs
samp

# A bootstrap sample
sample(5:10,replace=TRUE)

# A subsample of size 3 with replacement
sample(5:10,size=3,replace=TRUE)
```


If `N` an integer `sample(N)` is the same as `sample(1:N)`.

Example

Law data: average LSAT and GPA for 15 law schools

Interest is in correlation between LSAT and GPA.


```
cor(law)
obs.cor = cor(law)[1,2]
```

```
nbo
boot.cor = rep(0,nboot)
n = nrow(law)
```

```
for(i in 1:nboot){
    boot.cor[i] = cor(law[sample(n,replace=TRUE),])[1,2]
}
```

Confidence Intervals

Number of possible ways to do confidence intervals.

Simplest: normal approximation

- Compute estimates $T_1, \ldots, T_B$.
- $$\overline{\quad\quad\quad\quad\quad\quad}$$
$$b=1$$
- Compute the confidence interval as

$$(T_o - z_{\alpha/2}\hat{se}(T),\ T_o \qquad \alpha/2$$

for $z_{\alpha/2}$ the normal critical value.

Assumes that T_o really is approximately normally distributed, but it does mean that you don't have to know its variance.

Bias Correction

Some statistics are biassed; to assess this we consider the average bootstrap replicate

$$\bar{T} = \frac{1}{B}\sum_{b=1}^{B} T_b$$

then

and we can even correct our estimate

$$T_o^c = T_o - bias = 2$$

and update confidence intervals

$$(T_o^c - z_{\alpha/2}\hat{se}(T),\ T_o^c + z_{\alpha/2}\hat{se}(T))$$

Example Continued

```
# Estimate the bias
> cor.bias = mean(boot.cor)-obs.cor
[1] -0.004983282

# Bias
> obs
[1] 0.

# Boostrap Standard Error
> cor.se = sd(boot.cor)
[1] 0.1340546

# Bootstrap Corrected Confidence Interval
> obs.cor.c + c(-1,1)*qnorm(0.975)*cor.se
[1] 0.5186155 1.0441000
```

Confidence Intervals II

Can also use the empirical distribution of the bootstrap statistics.

Percentile bootstrap intervals:

$$(T_{(\alpha/2)}, T_{(1-\alpha/2)})$$

wher

Alter

- $\alpha/2$ $\alpha/2$ 2.
- Use bootstrap sample for distribution of θ.
- Yields $b_{\alpha/2} = T_{(1-\alpha/2)} - T_o$ and e

$$(2T_o - T_{(1-\alpha/2)}, 2T_o - T_{(\alpha/2)})$$

Same 'reverse the distribution' effect.

- Unlike simulation-based CIs, bias correction is important here. (Why?)

Continuing Example

```
# Quantiles of Bootstrap Distribution
b0.025 = quantile(boot.cor,0.025)
b0.975 = quantile(boot.cor,0.975)

# Perc
> c(b0
0.46

# Standard Bootstrap Confidence Interva
> 2*obs.cor.c - c(b0.975, b0.025)
0.5978281 1.0968481
```


- Upper limit ≥ 1 can be thresholded (remember, this interval is just meant to capture the "truth", 95% of the time).
- Bootstrap test for $\rho \leq 0.5$ rejects – null hypothesis parameter is not within confidence interval.

Yet Further Intervals

Variants (increasingly elaborate) proposed to improve confidence intervals.

- Bootstrap t–interval

$$(t_0 \quad t^*_{1\ \alpha/2}\hat{se}(t_0), (t_0 \quad t^*_{1\alpha/2}\hat{se}(t_0)$$

  - $\hat{se}(T_b)$: estimate of standard error *for each* bootstrap within a bootstrap.
- Bias-corrected and accelerated bootstr
  both bias and skewness in bootstrap distrib

)/$\hat{se}(T_b)$.

Basic reasoning: using estimates of standard errors requires smaller B, and has better statistical properties than quantiles of bootstrap distribution.

Yet more variants: beyond this course.

When Bootstraps Break

Bootstrap doesn't work for all statistics

- $t(X_1, \ldots, X_n) = \min_j |X_i - X_j|$
  - Minimum distance between points in the data set.
  -
  -

Most cases of failure $t(X_1, \ldots, X_n)$ is not a smooth function of $X_1, \ldots, X_n$ (cannot differentiate with respec

Subsampling often a good alternative (but issue, as it is for this example).

Rare; most cases are pathological (although recent statistical methods are a problem).

Conditionally-Specified Models

Frequently we only describe part of the data generating mechanism.

Eg: regression models

with $_i$

- What about x_i? Treated as fixed (often c experimenter)
- Or, frequently, $x_i \sim h(x)$, but h n
- For large n (and in practice) very little variance in $\hat{\beta}$ due to randomness in x_i.

Example: Multiple Regression

In the lab, we looked at simple linear regression. For multiple regression

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \epsilon_i$$

where the $\epsilon_i \sim N(0, \sigma^2)$. Also written as

Squa

$$SSE(\beta) = (\mathbf{y} - X\beta)$$

so that the derivative of squared error is

$$\frac{dSSE(\beta)}{d\beta} = 2(X^T X\beta - X^T \mathbf{y})$$

which is zero at

$$\beta = (X^T X)^{-1} X^T \mathbf{y}$$

A Data Set

- 12 children's height, weight and arterial distance from the wrist to the heart.
-
- height and weights.


```
> mod = lm(dist~.,data=heart)
> summary(mod)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 24.50804    5.12461   4.782 0.000998 ***
height       0.05091    0.21060   0.242 0.814396
weight       0.25495    0.10326   2.469 0.035624 *
```

A Simulation

We'll used the estimated coefficients and residual standard error to simulate at the observed covariates.

```
beta = mod$coef    # Start from observed coefficients

# Design matrix
X = as.m

# Predicted values (also from mod$fit)
pred = X %*% beta

# Residual standard error
sigma = summary(mod)$sigma
```

Vectorizing A Simulation

Create a large matrix where response for each data set is in one column.

Recall that $\mathbf{y} = X\beta + \boldsymbol{\epsilon}$, repeat the same $X\beta$ over each column, but cr

```
nsim = 1
Ysim = m
           matrix(rnorm(12*nsim,
```

Now estimate β; the `solve` command in

```
beta.sim = solve( t(X)%*%X, t(X)%*%Ysim
```

Because the estimate is linear in `Ysim` we can obtain them all at once.

Continuing

Simulation results:

Lovely Demonstration of effect of correlated cov

Bootstrap options

- Re-sample (x_i, y_i) pairs and do standard bootstrap.
- Try to re-sample the ϵ_i – corresponds to our model.

Residual Bootstrap

Basically restricted to linear regression models:

$$y = X\beta + \epsilon$$

1.
2.
3. i
4. Add bootstrapped residuals back onto pre
   $y_i^* = \mathbf{x}_i\hat{\beta} + \hat{\epsilon}_i^*, i = 1, \ldots, n$
5. Estimate $\hat{\beta}_b$ for bootstrapped $(x_i,$

Now all the bias, standard error, confidence interval statistics can be calculated with the same recipe.

Why Residual Bootstrap?

More stable, avoids ties in the $\mathbf{x}_i$, doesn't change a fixed design.

Continuing The Example


```
# Estimate errors
eps.hat = heart$dist - pred

# Now bo
eps.                                    UE)]
eps.
```



```
# Create data
Y.boot = matrix(pred,12,nsim,byrow=
```



```
# And re-estimate
beta.boot = solve( t(X)%*%X, t(X)%*%Y.boot)
```

Usual Statistics

Calculate the same statistics as before

```
# Bias
> biases = beta - apply(beta.boot,1,mean)
 (In
 0.2
# Stan
> se.boot = apply(beta.boot,1,sd)
(Intercept)      weight      weight
 4.63035973 0.18945029 0.09076793
```

Biasses are probably not real but

```
> beta.c = beta-biases
24.28460334  0.05857949  0.25263440
```

Confidence Intervals

```
# Lower and Upper Bounds
>lb = apply(beta.boot,1,quantile,0.025)
>ub = apply(beta.boot,1,quantile,0.975)
                      lb          ub
(Int
heig          -0.3
weig           0.069


# Confidence Interval
>cbind(2*beta - ub, 2*beta-ub)
                    [,1]       [,2]
(Intercept) 15.96304893 34.9253459
height      -0.36603873  0.4113947
weight       0.08256726  0.4404885
```

Bootstrap Tests of Significance

We can also test how many times the bootstrap falls below 0

```
> pvals = apply(beta.boot<0,1,mean)
> pvals
(Intercept)      height      weight
```


But r
of the r

More information: correlation of $\hat{\beta}$

```
> cor(t(beta.boot))
             (Intercept)     height     weight
(Intercept)   1.0000000 -0.9074095  0.6355241
height       -0.9074095  1.0000000 -0.8834964
weight        0.6355241 -0.8834964  1.0000000
```

Parametric Bootstrap

Residual bootstrap is not always applicable:

Eg: logistic regression;

$$P(y_i = 1)$$

Does

Instead, estimate $\hat{\beta}$ and create a new data set b

y_i^* according to

$$P(y_i^* = 1) = \frac{}{1 + e^{\mathbf{x}_i \beta}}$$

Isn't this just estimate parameters and then simulate data from the model?

Yes! But naming is part of good salesmanship.

Example

Summary

- Simulation (parametric bootstrap) a tool for evaluating confidence intervals about estimated parameters.
- *Bootstrap*: avoids having to know distribution of data.
-

- Residual bootstrap for linear regression m

But

- Justification is asymptotic: requires enou empirical distribution approximates truth.
- Won't work for every problem or every statistic (but most standard stats are OK).