22S:166
Computing in Statistics

The Jackknife

Lecture 10
October 6, 2017

Kate Cowles
374 SH, 335-0727
kate-cowles@uiowa.edu

The Jackknife

- another resampling-based method of trying to assess bias, standard error, etc. of a sample-based estimate $\hat{\theta}$ of a population characteristic θ
- usually less computationally intensive than the bootstrap
- let n denote sample size of real dataset
- then there are n jackknife samples $\mathbf{x}_{(i)}$, each obtained by leaving out one observation i from the original dataset
  $\mathbf{x}_{(i)} = x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$
- each *jackknife replication* $\hat{\theta}_{(i)}$ is calculated from the jackknife sample $\mathbf{x}_{(i)}$ in the same way that $\hat{\theta}$ was calculated from the whole dataset
- then ith jackknife *pseudovalue* $\tilde{\theta}_i$ is calculated as $n * \hat{\theta} - (n-1) * \hat{\theta}_{(i)}$

Jackknife estimate of standard error of $\hat{\theta}$

- Let $\hat{\theta}_{(.)}$ denote the mean of the jackknife replications $\hat{\theta}_{(i)}$. Then

$$\begin{aligned} \widehat{se}_{jack} &= \sqrt{\frac{n-1}{n} \sum_{i=1}^{n} (\hat{\theta})_{(i)} - \hat{\theta}_{(.)})^2} \\ &= \sqrt{\left(\frac{\Sigma_{i=1}^{n} (\tilde{\theta}_i - \tilde{\theta}_.)^2}{n(n-1)} \right)} \end{aligned}$$

- multiplicative factor $\frac{n-1}{n}$ was chosen to make this work out exactly right for the case of estimating a population mean using $\hat{\theta} = \bar{x}$

Jackknife estimate of bias of $\hat{\theta}$

- $$\widehat{bias}_{jack} = (n-1)(\hat{\theta}_{(.)} - \hat{\theta})$$
- multiplicative factor $(n-1)$ is chosen to make this work out exactly right for the case of estimating a population variance using $\hat{\sigma}^2 = \frac{\Sigma_{i=1}^{n}(x_i - \bar{x})^2}{n}$.
- so (first order) jackknife estimator of θ is

$$\begin{aligned} \hat{\theta}^{jack} &= n\hat{\theta} - (n-1)\ (\hat{\theta}_{(.)}) \\ &= \frac{\Sigma_{i=1}^{n} \tilde{\theta}_i}{n} \end{aligned}$$

- why can we expect $\hat{\theta}^{jack}$ to be less biased than $\hat{\theta}$
  - most estimators are biased
  - bias can be approximated as Taylor series expansion of the estimator

$$E(\hat{\theta} - \theta) \;=\; \frac{a_1}{n} + \frac{a_2}{n^2} + \frac{a_3}{n^3} + \cdots$$

  - then it can be shown that

$$E(\hat{\theta}^{jack} - \theta) \;=\; -\frac{a_2}{n^2} - \frac{a_2 + a_3}{n^3} - \cdots$$

    which eliminates term of order $\frac{1}{n}$ and thus is smaller than the bias of $\hat{\theta}$

Confidence intervals from the jackknife

- confidence intervals are not as commonly computed when using the jackknife as when using the bootstrap
- symmetric level α confidence intervals, with and without bias correction, can be computed as follows
  - (bias-corrected estimate) $\pm\, z_{1-\alpha/2}$ stderr
  - (estimate from orignal sample) $\pm\, z_{1-\alpha/2}$ stderr
- percentile-based confidence intervals don't work for jackknife unless the original sample size is very large

Example with confidence intervals

```
> jackknife.example
function(level)
{
# level is confidence level for intervals

mlevar <- function(v) {
# calculates the mle of variance -- biased estimator!
        n <- length(v)
        ((n-1)/n) * var(v)
}

spatial.dat <- matrix(c(   48, 42, 36, 33, 20, 16, 29, 39,
   42, 38, 42, 36, 20, 15, 42, 33, 22, 20, 41, 43, 45, 34,
   14, 22,  6,  7,  0, 15, 33, 34, 28, 29, 34, 41,  4, 13,
   32, 38, 24, 25, 47, 27, 41, 41, 24, 28, 26, 14, 30, 28,
   41, 40), ncol=2,byrow=T)

n <- nrow(spatial.dat)

orig.stat <- apply(spatial.dat,2, mlevar)
         # apply mlevar func to each column

jack.stats <- matrix( 0, nrow=n, ncol=2 )

for (i in 1:n) {
        jack.stats[i,] <- apply( spatial.dat[-i,],2, mlevar)
}

pseudo <- matrix( 0, nrow=n, ncol=2)

for (i in 1:2) {
  pseudo[,i] <- n * orig.stat[i] - (n-1) * jack.stats[,i]
```



```
}


print(apply( pseudo,2,mean))
jack.mean <- apply( jack.stats,2,mean )
bias <- (n-1) * (jack.mean - orig.stat)

bias.co <- orig.stat - bias

stderr <- apply( pseudo,2, function(v) {sqrt(var(v)/length(v))})

problow <- (1-level)/2
probhigh <- 1-problow

z <- qnorm( c(problow, probhigh ))

unadj.intervals <- rbind( orig.stat + z[1] * stderr, orig.stat,
  orig.stat + z[2] * stderr )

bias.co.intervals <- rbind( bias.co + z[1] * stderr, bias.co,
  bias.co + z[2] * stderr)

ssq <- apply(spatial.dat,2, var)
     #  for comparison, the standard unbiased estimator

list( orig.stat = orig.stat, jack.stats = jack.stats, pseudo = pseudo,
bias = bias, bias.co = bias.co, stderr = stderr,
unadj.intervals = unadj.intervals, bias.co.intervals = bias.co.intervals,
ssq = ssq )
}
```

```
> jackknife.example(0.95)
[1] 178.3954 113.7862
$orig.stat
[1] 171.5340 109.4098

$bias
[1] -6.861361 -4.376391

$bias.co
[1] 178.3954 113.7862

$stderr
[1] 45.02537 22.26572

$unadj.intervals
               [,1]      [,2]
           83.28592  65.76976
orig.stat 171.53402 109.40976
          259.78213 153.04976

$bias.co.intervals
             [,1]      [,2]
         90.14728  70.14615
bias.co 178.39538 113.78615
        266.64349 157.42616

$ssq
[1] 178.3954 113.7862
```


Relationship between jackknife and bootstrap

- unless n is large, jackknife is less computationally intensive
- but if jackknife uses fewer samples than bootstrap, then jackknife is using less information
- jackknife may be considered an approximation to the bootstrap
- jackknife does not work well if the quantity being estimated is not "smooth"
  - e.g. median can change sharply with deletion of a single observation

Jackknife for cities population example

```
> jackex
function()
{
library(boot)  # load library and dataset city
n <- nrow(city)
attach(city)
thetahat <- rep(0,n)
thetawig <- rep(0,n)
thetahatn <- mean(x) / mean(u)
for(i in 1:n) {
   thetahat[i] <- mean(x[-i])/mean(u[-i])
   thetawig[i] <- n * thetahatn - (n-1) * thetahat[i]
}
thetahat.dot <- mean(thetahat)
thetawig.dot <- mean(thetawig)
se.thetan <- sqrt(sum((thetawig - thetawig.dot)^2) / (n * (n-1)))
se.thetan2 <- sqrt((n-1) *sum((thetahat - thetahat.dot)^2) / n)
unbiased2 <- n * thetahatn - (n-1) * thetahat.dot

list(thetahatn = thetahatn, thetahat = thetahat, thetawig = thetawig,
se.thetan = se.thetan, se.thetan2 = se.thetan2, thetawig.dot = thetawig.dot,
unbiased2 = unbiased2)
}
>  jackex()
$thetahatn
[1] 1.520312

$thetahat
 [1] 1.653386 1.588665 1.561313 1.546638 1.516892 1.509121
     1.510638 1.499190
 [9] 1.413115 1.446708
```



```
$thetawig
 [1] 0.3226469 0.9051360 1.1513115 1.2833853 1.5510980
     1.6210354 1.6073803
 [8] 1.7104184 2.4850922 2.1827488

$se.thetan
[1] 0.194791

$se.thetan2
[1] 0.194791

$thetawig.dot
[1] 1.482025

$unbiased2
[1] 1.482025
```

Jackknife for sample mean as estimator of population mean

```
function()
{
blood.flow <- read.table("blood.flow.dat")
print(blood.flow)
n <- nrow(blood.flow)
thetahat <- rep(0, n)
thetawig <- rep(0, n)
muhatn <- mean(blood.flow[, 1]) #
for(i in 1:n) {
thetahat[i] <- mean(blood.flow[ - i, 1])
thetawig[i] <- n * muhatn - (n - 1) * thetahat[i]
}
thetawig.dot <- mean(thetawig)
se.muhat <- sqrt(sum((thetawig - thetawig.dot)^2)/(n * (n - 1)))
list(muhatn = muhatn, thetahat = thetahat, thetawig = thetawig,
thetawig.dot = thetawig.dot, var.muhat = var.muhat)
}
   V1  V2
1 115 138
2 170 172
3 142 159
4 138 147
5 280 166
6 470 345
7 480 387
8 141 131
9 390 375
$muhatn:
[1] 258.4444
```



```
$thetahat:
[1] 276.375 269.500 273.000 273.500 255.750 232.000 230.750 273.125 242.000

$thetawig:
[1] 115 170 142 138 280 470 480 141 390

$thetawig.dot:
[1] 258.4444

$se.muhat:
[1] 50.25936
```


Jackknife for sample correlation coefficient as estimator of population correlation coefficient

```
function()
{
blood.flow <- read.table("blood.flow.dat")
print(blood.flow)
n <- nrow(blood.flow)
thetahat <- rep(0, n)
thetawig <- rep(0, n)
thetahatn <- cor(blood.flow)[1, 2] #
for(i in 1:n) {
thetahat[i] <- cor(blood.flow[ - i,  ])[1, 2]
thetawig[i] <- n * thetahatn - (n - 1) * thetahat[i]
}
thetawig.dot <- mean(thetawig)
stderr.thetahat <- sqrt(sum((thetawig - thetawig.dot)^2)/(n * (n - 1)))
list(thetahatn = thetahatn, thetahat = thetahat, thetawig = thetawig,
thetawig.dot = thetawig.dot, stderr.thetahat = stderr.thetahat)
}
   V1  V2
1 115 138
2 170 172
3 142 159
4 138 147
5 280 166
6 470 345
7 480 387
8 141 131
9 390 375

$thetahatn:
[1] 0.9448479
```



```
$thetahat:
[1] 0.9405566 0.9434308 0.9433717 0.9407706 0.9766233 0.9395961 0.9205287
[8] 0.9395995 0.9583461

$thetawig:
[1] 0.9791780 0.9561844 0.9566569 0.9774661 0.6906443 0.9868617 1.1394014
[8] 0.9868350 0.8368616

$thetawig.dot:
[1] 0.9455655

$stderr.thetahat:
[1] 0.04085297
```