统计计算 张楠 2019秋: The Bootstrap

(返回 统计计算 张楠 2019秋)

Efron 在1979,1981和1982年的工作中引入和进一步发展了Bootstarp方法, 此后发表了大量的关于 此方法的研究.

Bootstrap方法是一类非参数Monte Carlo方法, 其通过再抽样对总体分布进行估计. 再抽样方法将 观测到的样本视为一个有限总体, 从中进行随机(再)抽样来估计总体的特征以及对抽样总体作出统计推断. 当目标总体分布没有指定时, Bootstrap方法经常被使用, 此时, 样本是唯一已有的信息.

Bootstrap 一词可以指非参数Bootstrap, 也可以指参数Bootstrap(上一讲中). 参数Bootstrap是指总体分布 完全已知, 利用Monte Carlo方法从此总体中抽样进行统计推断; 而非参数Bootstrap是指总体分布完全未知, 利用再抽样 方法从样本中(再)抽样进行统计推断.

可以视样本所表示的有限总体的分布为一个“伪”总体, 其具有和真实总体类似的特征. 通过从此“伪”总体中 重复(再)抽样, 可以据此估计统计量的抽样分布. 统计量的一些性质, 如偏差, 标准差等也可以通过再抽样来估计.

一个抽样分布的Bootstrap估计类似于密度估计的想法. 我们通过一个样本的直方图来估计密度函数的形状. 直方图 不是密度, 但是在非参数问题中, 可以被视为是密度的一个合理估计. 我们有很多方法从已知的密度中产生随机样本, Bootstrap则从经验分布中产生随机样本. 假设$x = (x_1, \cdots, x_n)$为一个从总体分布$F(x)$中观测到得样本, X^*为从x中随机选择的一个样本, 则 $P(X^* = x_i) = \frac{1}{n},\ i = 1, \cdots, n.$ 从x中有放回的再抽样得到随机样本$X_1^*, \cdots, X_n^*$. 显然随机变量$X_1^*, \cdots, X_n^*$ 为$i.i.d$的随机变量, 服从$\{x_1, \cdots, x_n\}$上的均匀分布.

经验分布函数$F_n(x)$是$F(x)$的估计, 可以证明, $F_n(x)$是$F(x)$的充分统计量. 而且另一方面, $F_n(x)$ 本身是$\{x_1, \cdots, x_n\}$上的均匀分布随机变量X^*的分布函数. 因此在Bootstrap中有这个逼近. F_n逼近到F, Bootstrap重复下的经验分布函数F_n^*是F_n的逼近. 从x中再抽样, 等价于从F_n中产生随机样本. 这两种 逼近可以表示为
$$\begin{aligned} F &\to X \to F_n \\ F_n &\to X^* \to F_n^* \end{aligned}$$
从x中产生一个Bootstrap随机样本可以这样实现, 先从$\{1, 2, \cdots, n\}$中有放回的选取n次 得到$\{i_1, \cdots, i_n\}$, 然后得到Bootstrap样本$x^* = (x_{i1}, \cdots, x_{in})$.

假设θ是我们感兴趣的参数(向量), $\hat{\theta}$为θ的估计. 则$\hat{\theta}$的分布的Bootstrap 估计可以通过如下方法得到

1. 对Bootstrap重复的第b次($b = 1, \ldots, B$),
    1. 通过有放回的从$x_1, \cdots, x_n$ 中抽样得到再抽样样本$x^{*(b)} = x_1^*, \ldots, x_n^*$.
    2. 根据$x^{*(b)}$计算$\hat{\theta}^{(b)}$.
2. $F_{\hat{\theta}}(\cdot)$的Bootstrap估计为$\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(B)}$的经验分布函数.

例1 F_n与Bootstrap抽样 假设我们观察到样本 $x = \{2, 2, 1, 1, 5, 4, 4, 3, 1, 2\}$ 从x中再抽样依照选择$1, 2, 3, 4, 5$ 的概率分别为$0.3, 0.3, 0.1, 0.2, 0.1$ 进行. 从而从x中随机选择 的一个样本X^*, 其分布函数就是经验分布函数, 即
$$F_{X^*}(x) = F_n(x) = \begin{cases} 0, & x < 1; \\ 0.3, & 1 \le x < 2; \\ 0.6, & 2 \le x < 3; \\ 0.7, & 3 \le x < 4; \\ 0.9, & 4 \le x < 5; \\ 1, & x \ge 5. \end{cases}$$
注意如果F_n没有靠近F_X, 则重复抽样下的分布也不会靠近F_X. 上例中的样本x实际上是从$Poisson(2)$中随机产生的, 从x中大量重复抽样可以很好的估计F_n, 但是不能很好的估计F_X, 因为无论重复多少次再抽样, 得到的 Bootstrap样本都没有0.

Bootstrap Estimation of Standard Error

估计量$\hat{\theta}$的标准差的Bootstrap估计，是Bootstrap重复$\hat{\theta}^{(1)},\ldots,\hat{\theta}^{(B)}$的样本标准差：$\hat{se}_B(\hat{\theta}^*)=\sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}^{(b)}-\overline{\hat{\theta}^*})^2}$. 其中$\overline{\hat{\theta}^*}=\frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^{(b)}$. 根据Efron和Tibshirini(1993), 要得到标准差一个好的估计, 重复的次数B并非需要非常大. $B=50$常常已经足够了, $B>200$是很少见的(置信区间除外).

例2 (标准差的Bootstrap估计) bootstrap包里的法律院校数据集law, 记录了15所法律院校入学考试的平均成绩(LSAT)和GPA(乘了100).

```
      1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
LSAT 576 635 558 578 666 580 555 661 651 605 653 575 545 572 594
GPA  339 330 281 303 344 307 300 343 336 313 312 274 276 288 296
```

估计LSAT和GPA之间的相关系数, 并求样本相关系数的标准差的Bootstrap估计.

在本例中
1. 数据是成对的$(x_i,y_i),i=1,\ldots,15$.
2. 可以通过样本相关系数估计相关系数$\hat{\tau}=\frac{n\sum_i xiyi-\sum_i xi\sum_i yi}{\sqrt{n\sum_i x_i^2-(\sum_i x_i)^2}\sqrt{n\sum_i y_i^2-(\sum_i y_i)^2}}$. 3. Bootstrap对这些数据对再抽样.

因此, 算法如下

1. 对Bootstrap重复的第b次($b=1,\ldots,B$),
    1. 通过有放回的从$x_1,\cdots,x_n$中抽样得到再抽样样本$x^{*(b)}=x_1^*,\ldots,x_n^*$. 这里$x_i$或者$x_i^*$为一个向量.
    2. 根据$x^{*(b)}$计算$\hat{\tau}^{(b)}$.
2. $F_{\hat{\tau}}(\cdot)$的Bootstrap估计为$\hat{\tau}^{(1)},\ldots,\hat{\tau}^{(B)}$的经验分布函数.

样本相关系数为$cor(LSAT,GPA)=0.7763745$, 使用Bootstrap估计标准差的程序如下:

```
library(bootstrap)    #for the law data
print(cor(law$LSAT, law$GPA))
#set up the bootstrap
B <- 200              #number of replicates
n <- nrow(law)        #sample size
R <- numeric(B)       #storage for replicates
#bootstrap estimate of standard error of R
for (b in 1:B) {
    #randomly select the indices
    i <- sample(1:n, size = n, replace = TRUE)
    LSAT <- law$LSAT[i]       #i is a vector of indices
    GPA <- law$GPA[i]
    R[b] <- cor(LSAT, GPA)
}
#output
print(se.R <- sd(R))
hist(R, prob = TRUE)
```

$se(\hat{\tau})$的Bootstrap估计为0.1371913, 样本相关系数的标准差的理论值 为0.115.

例3 使用boot函数进行Bootstrap估计标准差 在R中, 包boot里的boot函数可以进行Bootstrap估计. boot 函数中的参数*statistic*是 一个函数, 用来返回感兴趣的统计量值. 这个函数必须至少有两个参数, 其中第一个是数据, 第二个表示Bootstrap抽样中的指标向量, 频率或者权重等. 因此我们首先写一个函数计算$\hat{\theta}^{(b)}$. 用$i=(i_1,\cdots,i_n)$表示指标向量, 则计算相关系数的程序为

```
tau<-function(x,i){
xi<-x[i,]
cor(xi[,1],xi[,2])
}
```

然后我们就可以使用boot函数进行Bootstrap估计:

```
library(boot)       #for boot function
obj <- boot(data = law, statistic = tau, R = 2000)
obj
# alternative method for std.error
y <- obj$t
sd(y)
detach(package:boot)
```

观测到的$\hat{\theta}$值用$t1*$标出. 2000次重复下的Bootstrap标准差估计为 0.1326418.

和boot函数相似功能的函数是bootstrap包里的bootstrap函数. 使用此函数重复上述问题的程序如下

```
library(bootstrap)       #for boot function
n <- 15
theta <- function(i,x){ cor(x[i,1],x[i,2]) }
results <- bootstrap(1:n,2000,theta,law)
sd(results$thetastar) #0.1325971
detach(package:bootstrap)
```

两个函数的用法上有些差异, bootstrap包是收录了Efron & Tibshirani的书里的程序和数据. boot 包是收录了Davson & Hinkley的书里的程序和数据.

Bootstrap Estimation of Bias

θ的一个估计量$\hat{\theta}$的偏差定义为$bias(\hat{\theta}) = E\hat{\theta} - \theta$. 当$\hat{\theta}$的分布未知或者形式很复杂使得期望的计算不可能(从此分布中抽样变得很困难, Monte Carlo方法不可行), 以及在现实中, 我们也不知道θ的真值时(需要估计), 这种情况下偏差是未知的. 但是我们已经有了样本, $\hat{\theta}$是θ的估计, 而期望$E\hat{\theta}$可以通过Bootstrap方法进行估计. 从而 可以得到偏差的估计: $\widehat{bias}_B(\hat{\theta}) = E^*\hat{\theta}^* - \hat{\theta}$. E^*表示Bootstrap经验分布.

因此一个估计量的偏差的Bootstrap估计, 是通过使用当前样本下的估计量$\hat{\theta}$来估计θ, 而 使用$\hat{\theta}$的Bootstrap重复来估计$E\hat{\theta}$. 对一个有限样本$x = (x_1, \cdots, x_n)$, 有$\hat{\theta}(x)$的B个i.i.d估计量$\hat{\theta}^{(b)}$. 则$\{\hat{\theta}^{(b)}\}$的均值是期望值$E\hat{\theta}^*$的无偏估计, 因此偏差的Bootstrap估计为 $\widehat{bias}_B(\hat{\theta}) = \overline{\hat{\theta}^*} - \hat{\theta}$. 这里$\overline{\hat{\theta}^*} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^{(b)}$. 正的偏差意味着$\hat{\theta}$平均来看过高估计了$\theta$; 而负的偏差意味着$\hat{\theta}$平均来看过低估计了$\theta$. 因此, 一个经过偏差修正(Bias-correction)的估计量为 $\tilde{\theta} = \hat{\theta} - \widehat{bias}_B(\hat{\theta})$. 例**4** Bootstrap偏差估计: 估计上例中样本相关系数的偏差

```
theta.hat <- cor(law$LSAT, law$GPA)
#bootstrap estimate of bias
B <- 2000   #larger for estimating bias
n <- nrow(law)
theta.b <- numeric(B)
for (b in 1:B) {
    i <- sample(1:n, size = n, replace = TRUE)
    LSAT <- law$LSAT[i]
    GPA <- law$GPA[i]
    theta.b[b] <- cor(LSAT, GPA)
}
bias <- mean(theta.b - theta.hat)
bias
```

这个值和例3中的boot函数返回的结果非常相近.

例5 Bootstrap偏差估计: 假设$x = (x_1, \cdots, x_{10}) \sim N(\mu, \sigma^2)$, 求$\sigma^2$的估计量 $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$ 的偏差

```
n<-10
x<-rnorm(n,mean=0,sd=10)

sigma2.hat<-(n-1)*var(x)/n
#bootstrap estimate of bias
B <- 2000   #larger for estimating bias
sigma2.b <- numeric(B)
for (b in 1:B) {
    i <- sample(1:n, size = n, replace = TRUE)
    sigma2.b[b] <-(n-1)*var(x[i])/n
}
bias <- mean(sigma2.b - sigma2.hat)
bias
```

在这种情形下, $\hat{\sigma}^2$过低的估计了参数σ^2.

例6 比值参数估计的偏差的Bootstrap估计. 以包bootstrap里的patch数据为例. 该数据是测量了8个人使用3种不同的药物后血液中某种荷尔蒙的含量. 这三种药物分别是安慰剂, 旧药品(经过FDA审批的), 新药品(某个新工厂相同的工艺下生产的, 按FDA规定, 新工厂生产的药品也要审批). 研究的目的是比较新药和旧药的等价性. 如果可以证明新药和旧药之间的等价性, 则对新药就不需要完全重新向FDA申请审批了. 等价性的标准是对比值参数 $\theta = \frac{E(new)-E(old)}{E(old)-E(placebo)}$. 若$|\theta| \leq 0.20$, 则新药和旧药就等价. 估计$\theta$的统计量为$\bar{Y}/\bar{Z}$. 这两个变量在patch数据中给出. 我们的目标是计算此估计偏差的Bootstrap估计.

```
data(patch, package = "bootstrap")
patch
n <- nrow(patch)  #in bootstrap package
B <- 2000
theta.b <- numeric(B)
theta.hat <- mean(patch$y) / mean(patch$z)
#bootstrap
for (b in 1:B) {
    i <- sample(1:n, size = n, replace = TRUE)
    y <- patch$y[i]
    z <- patch$z[i]
    theta.b[b] <- mean(y) / mean(z)
    }
bias <- mean(theta.b) - theta.hat
se <- sd(theta.b)
print(list(est=theta.hat, bias = bias,
           se = se, cv = bias/se))
```

本页面最后编辑于2019年11月6日 (星期三) 11:47。