
Introducción al Bootstrap con aplicaciones a modelos de efectos mixtos

El bootstrap es una de las técnicas de remuestreo más conocidas y resulta muy útil para obtener intervalos de confianza en situaciones donde el enfoque clásico (pruebas t o z) fallaría.

¿Qué es el bootstrap?

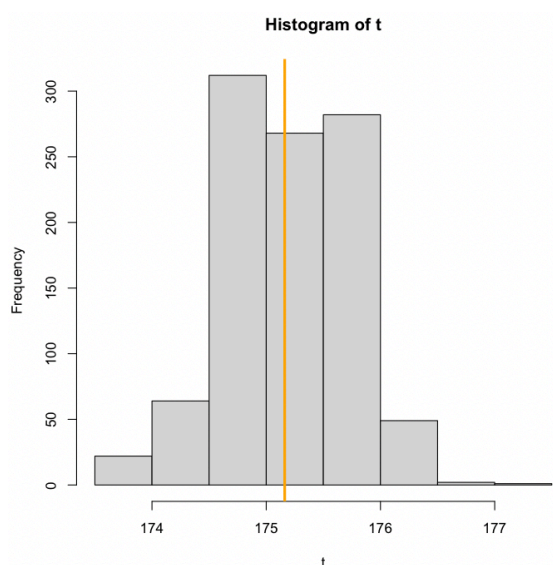
En lugar de escribir ecuaciones, veamos directamente cómo se puede realizar el bootstrap. A continuación, mostraremos un ejemplo sencillo de bootstrap utilizando la altura de 100 personas de la población.

```
> set.seed(20151101)
> height<-rnorm(100,175,6)
```

Ahora, remuestrearemos con reemplazo 1000 veces y calcularemos la mediana:

```
> t0<-median(height)
> t<-sapply(1:1000,function(x) median(sample(x=height,
+   size=100,replace=TRUE))))
> hist(t)
> abline(v=t0,col="orange",lwd=3)
```

Aquí está el histograma que obtenemos:



Y listo, esta es la esencia del bootstrap: remuestrear los datos observados con reemplazo y calcular el estadístico de interés (en este caso, la mediana) varias veces sobre los datos remuestreados para obtener una distribución del estadístico de interés. Esta distribución del estadístico de interés puede utilizarse para calcular, por ejemplo, intervalos de confianza.

Cuándo usar bootstrap

Bootstrap se utiliza para permitir la inferencia sobre el estadístico de interés cuando se desconoce su distribución real. Por ejemplo, en un modelo lineal, el parámetro de interés tiene una distribución conocida a partir de la cual se pueden calcular errores estándar y realizar pruebas formales. Por otro lado, para algunos estadísticos (mediana, diferencias entre dos modelos, etc.), si el analista no desea dedicar tiempo a escribir ecuaciones, el bootstrap puede ser una excelente opción para obtener errores estándar e intervalos de confianza de la distribución bootstrap.

Cuándo falla el bootstrap

Existen algunas situaciones en las que el bootstrap falla:

- (i) El estadístico de interés se encuentra en el límite del espacio de parámetros (como el mínimo o el máximo); la distribución bootstrap no converge (ya que el número de muestras bootstrap aumenta hasta el infinito) a la *distribución real del estadístico*.
- (ii) Si el tamaño de la muestra es pequeño, el bootstrap no aumentará la potencia de las pruebas estadísticas. Si se muestrean muy pocos datos para detectar un efecto de interés, el bootstrap no resolverá el problema por arte de magia; peor aún, el método bootstrap tendrá un rendimiento inferior al de otros.

¿Cuántas muestras de bootstrap?

La respuesta será la mayor cantidad posible. Ten en cuenta que en esta publicación usamos números bajos para acelerar los cálculos en mi computadora “vieja”.

Bootstrap no paramétrico y paramétrico usando la biblioteca boot

La biblioteca boot en R es muy práctica para calcular fácilmente intervalos de confianza a partir de muestras de bootstrap. Bootstrap no paramétrico con boot:

```
> library(boot)
> b1<-boot(data=height,statistic=function(x,i) median(x[i]),R=1000)
> boot.ci(b1)
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = b1)

Intervals :

Level	Normal	Basic
95%	(174.0, 176.3)	(174.1, 176.4)

Level	Percentile	BCa
95%	(173.9, 176.2)	(173.9, 176.2)

Calculations and Intervals on Original Scale

Bootstrap paramétrico con boot:

```
> x<-runif(100,-2,2)
> y<-rnorm(100,1+2*x,1)
> dat<-data.frame(x=x,y=y)
```

Ejemplo simple con un modelo lineal

```
> m<-lm(y~x)
```

Nos interesa obtener los intervalos de confianza para el coeficiente del modelo:

```
> foo<-function(out){
+ m<-lm(y~x,out)
+ coef(m)
+ }
```

La función rgen genera un nuevo vector de respuesta a partir del modelo:

```
> rgen<-function(dat,mle){
+ out<-dat
+ out$y<-unlist(simulate(mle))
+ return(out)
+ }
```

Generar 1000 muestras de bootstrap

```
> b2<-boot(dat,foo,R=1000,sim="parametric",ran.gen=rgen,mle=m)
```

Calcular los intervalos de confianza para los dos coeficientes:

```
> boot.ci(b2,type="perc",index=1)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = b2, type = "perc", index = 1)

Intervals :
Level      Percentile
95%      ( 1.039,  1.423 )
Calculations and Intervals on Original Scale
```

Calcular los intervalos de confianza para los dos coeficientes:

```
> boot.ci(b2,type="perc",index=2)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = b2, type = "perc", index = 2)

Intervals :
Level      Percentile
95%      ( 1.832,  2.164 )
Calculations and Intervals on Original Scale
```

En el caso no paramétrico de boot, se esperan dos argumentos en la función que devuelve el estadístico de interés: el primero es el objeto a partir del cual se calculará el estadístico y el segundo es un vector de índice (i), frecuencias (f) o ponderación (pesos) (w) que define la muestra de bootstrap.

En nuestro ejemplo, boot generará una serie de índices (denominados i) con reemplazo, que se utilizarán para crear un subconjunto del vector de altura original. En el caso paramétrico, la función que devuelve el/los estadístico(s) de interés solo necesita un argumento: el conjunto de datos original. A continuación, necesitamos proporcionar otra función (ran.gen) que describa cómo generar los nuevos datos; esta función debe devolver un objeto con el mismo formato que el conjunto de datos original.

Esta función de generación de datos aleatorios necesita dos argumentos: el primero es el conjunto de datos original y el segundo contiene la estimación de máxima verosimilitud del parámetro de interés, que es básicamente un objeto de modelo. El nuevo conjunto de datos generado por la función ran.gen se pasará a la función estadística para calcular el valor bootstrap del estadístico de interés. A continuación, es sencillo obtener los intervalos de confianza del estadístico utilizando boot.ci.

Bootstrap aplicado a modelos de efectos mixtos

Los modelos de efectos mixtos son bastante complejos y las distribuciones o el número de grados de libertad de sus diversas salidas (como los parámetros) no se conocen analíticamente. Por esta razón, el autor del paquete lme4 recomienda el uso de bootstrap para obtener intervalos de confianza en torno a los parámetros del modelo y los valores predichos, así como para obtener valores p a partir de pruebas de razón de verosimilitud.

Un modelo simple de intersección aleatoria:

```
> library(lme4)
> dat<-data.frame(x=runif(100,-2,2),ind=gl(n=10,k=10))
> dat$y<-1+2*dat$x+rnorm(100,0,1.2)[dat$ind]+rnorm(100,0,0.5)
> m<-lmer(y~x+(1|ind),dat)
```

Obtengamos los intervalos de confianza bootstrap para los parámetros del modelo:

```
> b_par<-bootMer(x=m,FUN=fixef,nsim=200)
> boot.ci(b_par,type="perc",index=1)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 200 bootstrap replicates

CALL :
boot.ci(boot.out = b_par, type = "perc", index = 1)

Intervals :
Level      Percentile
95%      (-0.3984,  1.1845 )
Calculations and Intervals on Original Scale
Some percentile intervals may be unstable
```

Obtengamos los intervalos de confianza bootstrap para los parámetros del modelo:

```
> boot.ci(b_par,type="perc",index=2)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 200 bootstrap replicates

CALL :
boot.ci(boot.out = b_par, type = "perc", index = 2)

Intervals :
Level      Percentile
95%      ( 1.834,  2.014 )
Calculations and Intervals on Original Scale
Some percentile intervals may be unstable
```

O alternativamente:

```
> confint(m,parm=c(3,4),method="boot",nsim=200,boot.type="perc")
```

```
Computing bootstrap confidence intervals ...
      2.5 %   97.5 %
(Intercept) -0.2112445  1.359807
x            1.8411784  2.041298
```

Obtengamos los intervalos de confianza bootstrap alrededor de las curvas de regresión:

```
> new_dat<-data.frame(x=seq(-2,2,length=20))
> mm<-model.matrix(~x,data=new_dat)
> predFun<-function(.) mm%*%fixef(.)
> bb<-bootMer(m,FUN=predFun,nsim=200) #do this 200 times
```

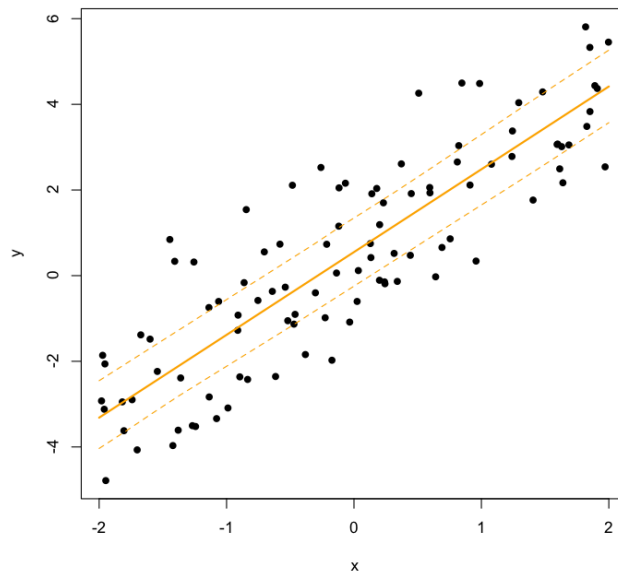
Como hicimos esto 200 veces, el IC del 95 % estará delimitado por los valores 5 y 195:

```
> bb_se<-apply(bb$t,2,function(x) x[order(x)][c(5,195)])
> new_dat$LC<-bb_se[1,]
> new_dat$UC<-bb_se[2,]
> new_dat$pred<-predict(m,newdata=new_dat,re.form=~0)
```

Gráfica de los resultados

```
> plot(y~x,dat,pch=16)
> lines(pred~x,new_dat,lwd=2,col="orange")
> lines(LC~x,new_dat,lty=2,col="orange")
> lines(UC~x,new_dat,lty=2,col="orange")
```

Aquí está el gráfico:



Finalmente, obtengamos los p-valores bootstrap de la prueba de razón de verosimilitud entre dos modelos.

```
> library(pbkrtest)
> m_0<-update(m,.-x)
> PBmodcomp(m,m_0,nsim=200)

Bootstrap test; time: 7.78 sec; samples: 200; extremes: 0;
large : y ~ x + (1 | ind)
y ~ (1 | ind)
      stat df    p.value
LRT    269.3  1 < 2.2e-16 ***
PBtest 269.3    0.004975 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dibujar intervalos de confianza alrededor de las curvas de regresión es complicado debido a los valores estimados de efectos aleatorios, que conllevan su propia incertidumbre (consulta `dotplot(ranef(m,condVar=TRUE))` para verlo).

El bootstrap es una forma eficiente de tener en cuenta estas incertidumbres, ya que las desviaciones aleatorias se vuelven a calcular en cada extracción. Finalmente, se pueden obtener p-valores para el efecto de un término de efecto fijo mediante un método bootstrap paramétrico, como se describe aquí e implementado en la función `PBmodcomp` del paquete `pbkrtest`. En la salida de `PBmodcomp`, los p-valores bootstrap se encuentran en la línea `PBtest`; la línea `LRT` indica el p-valor estándar, asumiendo una distribución de chi-cuadrada para el valor `LRT`.

Conclusión

Dado que las computadoras son cada vez más rápidas, el bootstrap permite obtener una estimación fiable del intervalo de confianza (siempre que el tamaño de la muestra original sea lo suficientemente grande) sin depender de distribuciones hipotéticas. El bootstrap también se puede extender a un marco bayesiano.