

f

Métodos de Remuestreo

Un problema muy común al estimar un parámetro θ usando un estimador $T(X_1, \dots, X_n)$ es determinar la “precisión” de dicho estimador, es decir, cantidades como el sesgo

$$B(T) = E(T) - \theta$$

o la varianza

$$V(T) = E(T - E(T))^2$$

En ocasiones, es muy difícil estimar estas cantidades en forma teórica. Tratando de resolver este problema, en las últimas décadas se han desarrollado métodos de alta demanda computacional, basados en *remuestreo*, es decir, en tomar muestras repetidas de los datos obtenidos y en base a ellas aproximar las medidas de precisión. Estos métodos son posibles gracias al avance de la computación.

Dos de los métodos más importantes basados en remuestreo son el *jackknife* y el *bootstrap*. A continuación haremos una cortísima presentación de ambos.

Jackknife

Este término (que en inglés denomina una navaja de bolsillo con diversas herramientas: destapador, destornillador, tijeras, etc.) fue propuesto por Tukey (1958) para describir un enfoque general para probar hipótesis y calcular intervalos de confianza en situaciones en las cuales no es fácil obtener un método mejor.

Una manera de justificar el jackknife es mirar en forma inusual un procedimiento muy usual: supongamos que tenemos una muestra X_1, \dots, X_n , y sea \bar{X} el promedio de dicha muestra. Consideremos el promedio calculado sin la j -ésima observación,

$$\bar{X}_{-j} = \frac{\sum_{i=1}^n X_i - X_j}{n - 1}$$

Entonces $X_j = n\bar{X} - (n - 1)\bar{X}_{-j}$.

En una situación general, supongamos que deseamos estimar un parámetro θ usando alguna función de los n valores muestrales. Sea $\hat{E}(X_1, \dots, X_n)$ (o \hat{E}) dicho estimador. Si removemos X_j obtenemos un estimador parcial \hat{E}_{-j} . Por analogía con el caso de la media, obtenemos los pseudo-valores:

$$\hat{E}_j^* = n\hat{E} - (n - 1)\hat{E}_{-j}, \quad j = 1, \dots, n$$

El promedio de los pseudo-valores

$$\hat{E}^* = \frac{\sum_{j=1}^n \hat{E}_j^*}{n}$$

es el *estimador jackknife* de θ .

Considerar los pseudo-valores como una muestra aleatoria de estimados independientes sugiere que la varianza de este estimador puede obtenerse como s^2/n , donde s^2 es la varianza muestral de los pseudo-valores; un paso más allá sugiere que un intervalo de confianza aproximado del $100(1 - \alpha)\%$ para θ puede calcularse como $\hat{E}^* \pm t_{\alpha/2, n-1} s / \sqrt{n}$.

Se trata, por tanto, de una manera de transformar muchos problemas de estimación en el problema de estimar la media de una muestra.

Quenouille (1956) había señalado que remplazar un estimador por su versión jackknife remueve sesgo de orden $1/n$. En efecto, supongamos que el valor esperado del estimados \hat{E} de θ es $\theta(1 + A/n)$. Entonces, el valor esperado del estimador parcial de θ , \hat{E}_{-j} será $\theta(1 + A/(n - 1))$, y por tanto el valor esperado del j-ésimo pseudo-valor es $n[\theta(1 + A/n)] - (n - 1)[\theta(1 + A/(n - 1))] = \theta$. Como esto es válido para todos los pseudovalores, el valor esperado del estimador jackknife es θ .

Ejemplo: Estimación de una desviación standard.

Supongamos que una muestra aleatoria de tamaño 20 extraída de cierta población (cuya distribución desconocemos) arroja como resultados 3.56, 0.69, 0.10, 1.84, 3.93, 1.25, 0.18, 1.13, 0.27, 0.50, 0.67, 0.01, 0.61, 0.82, 1.70, 0.39, 0.11, 1.20, 1.21 y 0.72. A partir de esta muestra, necesitamos estimar la desviación standard σ . Supongamos además que se decide usar como estimador $\hat{\sigma} = \sqrt{\sum (x_i - \bar{x})^2 / 20}$.

Podemos hacernos tres preguntas.

1. ¿Puede removerse un posible sesgo en el estimador modificándolo?
2. ¿Puede estimarse el error standard del estimador?
3. ¿Puede construirse un intervalo de confianza del 95% para σ ?

Estas tres preguntas pueden responderse usando jackknife. Los cálculos necesarios se muestran a continuación:

```
> obs<-c(3.56,0.69,0.10,1.84,3.93,1.25,0.18,1.13,0.27,0.50,
+ 0.67,0.01,0.61,0.82,1.70,0.39,0.11,1.20,1.21,0.72)
> obs.mat <- matrix(rep(obs,20), ncol= 20,byrow=T)
> diag(obs.mat) <- NA
> sdn <- function(x,...)
+ {return(sqrt(var(x,...) * (length(x)-1)/length(x)))}
> pse.val <- 20*sdn(obs)-19*apply(obs.mat,1,sdn,na.rm=T)
> pse.val
 [1] 3.9358901 0.5577025 0.9437824 0.8123392 5.1806636
 [6] 0.5161162 0.8703026 0.4987377 0.7956859 0.6431032
[11] 0.5649780 1.0345933 0.5892186 0.5201843 0.7099537
[16] 0.7092951 0.9342265 0.5071314 0.5087290 0.5475422
> sd.jac <- mean(pse.val)
 [1] 1.069009
> se.jac <- sqrt(var(pse.val)/20)
> se.jac
 [1] 0.2731823
> ic.jac
 [1] 0.4972316 1.6407859
```

La pregunta de interés es, lógicamente, si el procedimiento produce un buen resultado. En este caso particular, los datos provienen de simular una exponencial con parámetro 1 (y por tanto con desviación standard 1). Observamos que el intervalo de confianza contiene al verdadero valor. Por otro lado, el estimado para la desviación standard está muy cerca del verdadero valor.

Si bien en general no conocemos la distribución de los datos, y mucho menos el valor del parámetro, podemos usar un estudio de simulación para ver cómo se comporta el jackknife para una situación de este tipo. Para ello, simularemos 1000 muestras de 20 observaciones con distribución $\text{exp}(1)$, y repetiremos el procedimiento para cada una de ellas. Para ello usaremos el siguiente programa

```
"expjack" <-
  function(n,k)
  {
    sd.mues <- NULL
    sd.jack <- NULL
    se.jack <- NULL
    for (i in 1:n)
      {
        obs <- rexp(20)
        obs.mat <- matrix(rep(obs,k), ncol=k, byrow=T)
        diag(obs.mat) <- NA
        pse.val <- k*sdn(obs)-(k-1)*apply(obs.mat,1, sdn, na.rm=T)
        sd.mues <- c(sd.mues, sdn(obs))
        sd.jack <- c(sd.jack, mean(pse.val))
        se.jack <- c(se.jack, sqrt(var(pse.val)/k))
      }
    por.incorr <- sum((sd.jack-qt(0.975,k-1)*se.jack)>1 |
                     (sd.jack+qt(0.975,k-1)*se.jack) <1)/n
    return(sd.mues, sd.jack, se.jack, por.incorr)
  }
```

Se obtienen los siguientes resultados:

```
> sal1$por.incorr
[1] 0.2
> mean(sal1$sd.mues)
[1] 0.9444786
> mean(sal1$sd.jack)
[1] 0.9804971
```

Es decir, el verdadero valor del parámetro está en el intervalo de confianza un 80% de las veces, lo cual es insuficiente. Puede verse que esta situación se debe a la falta de normalidad del estimador jackknife, y podría corregirse usando una transformación.

Así mismo, la media del estimador de jackknife es mayor que la de los estimadores sin corregir; esto sugiere que el jackknife corrige el sesgo del estimador.

Este es un caso especialmente difícil para el jackknife; puede verse que en frecuentemente el jackknife funciona mucho mejor que para este caso (de hecho, el jackknife funciona bien para la desviación de la exponencial si se usa el logaritmo y el tamaño de la muestra es suficientemente grande).

Bootstrap

La idea esencial del bootstrap es que, en ausencia de otro conocimiento sobre una población, la distribución de valores encontrados en una muestra de tamaño n elegida de la población es la mejor guía para la distribución de la población. Por lo tanto, para aproximar lo que sucedería si se obtuviesen nuevas muestras de la población es razonable *remuestrear* la muestra. Este remuestreo se hace con reemplazo (a diferencia del jackknife).

Uno de los principales temas de investigación en el bootstrap es cómo calcular límites de confianza válidos para los parámetros poblacionales. Veremos algunos de los métodos básicos.

El intervalo de confianza standard de bootstrap

σ se estima como la desviación standard de los estimados de un parámetro θ calculados a partir de remuestreo de los valores de la muestra original. El intervalo obtenido es, entonces

$$\hat{\theta} \pm z_{\alpha/2} \sigma$$

Para que este método funcione, deben cumplirse las siguientes condiciones:

1. $\hat{\theta}$ tiene distribución aproximadamente normal.
2. $\hat{\theta}$ es aproximadamente insesgado.
3. El remuestreo da una buena aproximación para σ .

En la práctica es posible evitar el segundo requerimiento si se estima el sesgo como parte del procedimiento de bootstrap.

El número de muestras bootstrap necesarias depende de la situación particular; sin embargo, una muestra de 100 puede ser suficiente para obtener un buen estimado de la desviación standard de un estimador. Otros tipos de intervalo requerirán un número mayor de muestras.

Retomemos el ejemplo anterior. Usaremos 1000 muestras para tratar de responder las tres preguntas anteriores.

```
> boots.matrix_matrix(sample(obs,20000,replace=T),ncol=20)
> boot.vec <- apply(boots.matrix,1,sdn)
> boot.mn <- mean(boot.vec)
[1] 0.974547
> boot.sd <- sqrt(var(boot.vec))
> boot.sd
[1] 0.2365296
> est.bias <- boot.mn - sdn(obs)
> est.bias
[1] -0.0583008
```

Es decir, el sesgo estimado es de aproximadamente -0.06 , de manera que un estimador corregido para la desviación standard es

```
> sdn(obs)-est.bias
[1] 1.091149
```

Pueden obtenerse intervalos de confianza basados en el estimado, o bien en el estimado corregido por sesgo:

```
> boot.ci1 <- c(sdn(obs)-qnorm(0.975)*boot.sd,  
+ sdn(obs)+qnorm(0.975)*boot.sd)  
> boot.ci2 <- c(sdn(obs)-est.bias-qnorm(0.975)*boot.sd,  
+ sdn(obs)-est.bias+qnorm(0.975)*boot.sd)  
> boot.ci1  
[1] 0.5692583 1.4964374  
> boot.ci2  
[1] 0.6275591 1.5547382
```

En este caso, ambos intervalos contienen al verdadero valor del parámetro.

Estudios de simulación basados en 1000 muestras indican que el cubrimiento del intervalo de confianza corregido por sesgo es menor del 75%. Usar el logaritmo mejora este porcentaje a poco más del 80%, pero este cubrimiento es claramente insuficiente.

Otras estrategias para el cálculo de intervalos de confianza se basan en tratar de aproximar los percentiles de la distribución de un estimador usando percentiles generados usando bootstrap.

Supongamos que el deseamos encontrar un intervalo de confianza del $100(1 - \alpha)\%$ para un parámetro θ , para el cual disponemos de un estimado $\hat{\theta}$ basado en una muestra x_1, \dots, x_n . Supongamos además que existe una función monótona creciente f tal que $f(\hat{\theta}) \sim N(f(\theta), 1)$.

Entonces

$$\Pr(f(\theta) - z_{\alpha/2} < f(\hat{\theta}) < f(\theta) + z_{\alpha/2}) = 1 - \alpha$$

y por lo tanto, $(f(\hat{\theta}) - z_{\alpha/2}, f(\hat{\theta}) + z_{\alpha/2})$ será un intervalo de confianza del $100(1 - \alpha)\%$ para $f(\theta)$.

Si f es conocida, los límites de un intervalo para θ podrían obtenerse invirtiendo la transformación; lamentablemente, en general f no se conoce.

Sea $f(\hat{\theta}_B)$ un estimador transformado obtenido haciendo bootstrap de los datos originales. Podemos entonces generar tantos valores de $f(\hat{\theta}_B)$ como necesitemos para aproximar la distribución tan bien como queramos, y esta distribución debería ser similar a una $N(f(\hat{\theta}), 1)$, de modo que el intervalo $(f(\hat{\theta}) - z_{\alpha/2}, f(\hat{\theta}) + z_{\alpha/2})$ incluye el $100(1 - \alpha)\%$ central de la distribución. Por tanto, una manera de calcular dicho intervalo es hacer remuestreo de los datos originales y tomar los cuantiles muestrales en lugar de los teóricos.

Más aún, como la transformación es monótona creciente, el ordenamiento de los $f(\hat{\theta}_B)$ es el mismo que el de los $\hat{\theta}_B$, y por lo tanto, los límites de θ correspondientes al intervalo de confianza son simplemente los cuantiles muestrales $\alpha/2$ y $1 - \alpha/2$ de los estimados bootstrap del parámetro.

Dos métodos son basados en estas ideas son:

- **Primer método de percentiles**

Se emplea el remuestreo por bootstrap para generar la distribución bootstrap del parámetro de interés. Entonces, el intervalo de confianza del $100(1 - \alpha)\%$ para el verdadero valor del parámetro viene dado por los cuantiles $\alpha/2$ y $1 - \alpha/2$ de dicha distribución.

En nuestro ejemplo

```
> ci.per.1 <- quantile(boot.vec,c(0.025,0.975))
> ci.per.1
      2.5%      97.5%
0.4664218 1.3994062
```

Estudios de simulación muestran que en este caso el cubrimiento de este intervalo es cercano al 70% para este ejemplo.

- **Segundo método de percentiles**

Se emplea remuestreo por bootstrap para generar una distribución de estimados $\hat{\theta}_B$ para el parámetro θ de interés. Se supone entonces que la distribución bootstrap de la diferencia entre el estimador bootstrap y el estimador de θ en la muestra original, $\varepsilon_B = \hat{\theta}_B - \hat{\theta}$, aproxima a la distribución del error de $\hat{\theta}$. Sobre esta base, la distribución bootstrap de ε_B se usa para encontrar límites ε_L y ε_H , tales que el $100(1 - \alpha)\%$ de los errores está dentro de dichos límites. Finalmente, el intervalo de confianza se calcula como $(\theta - \varepsilon_H, \theta - \varepsilon_L)$.

Esto es equivalente a calcular $(2\theta - \theta_H, 2\theta - \theta_L)$.

En nuestro ejemplo

```
> ci.per.2 <- 2*sdn(obs)-quantile(boot.vec,c(0.975,0.025))
> ci.per.2
      97.5%      2.5%
0.6662896 1.5992740
```

Al hacer un estudio de simulación basado en 1000 muestras con distribución exponencial, se obtuvo un cubrimiento ligeramente mayor al 72% para el intervalo de confianza de 95% nominal.

El cálculo de estos límites de confianza requiere más muestras bootstrap que el cálculo del intervalo de confianza standard.

Efron (1987) sugiere que no tiene sentido tomar más de 100 iteraciones para determinar la media y la desviación standard de la distribución bootstrap; Sin embargo, para los métodos de percentiles son necesarias 1000 muestras o más.

Existen otros métodos de percentiles que buscan corregir los límites de confianza tomando en cuenta el sesgo. Estos métodos no serán discutidos en este curso.

Ejemplo: Se tienen como datos las poblaciones (en miles de habitantes) de $n = 49$ ciudades de los Estados Unidos en 1920 y 1930, los cuales denotaremos u y x .

```
> u
 [1] 138  93  61 179  48  37  29  23  30  2  38  46  71
[14]  25 298  74  50  76 381 387  78  60 507  50  77  64
[27]  40 136 243 256  94  36  45  67 120 172  66  46 121
[40]  44  64  56  40 116  87  43  43 161  36

> x
 [1] 143 104  69 260  75  63  50  48 111  50  52  53  79
[14]  57 317  93  58  80 464 459 106  57 634  64  89  77
[27]  60 139 291 288  85  46  53  67 115 183  86  65 113
[40]  58  63 142  64 130 105  61  50 232  54

> plot(u,x)
```

En este caso, el parámetro de interés es

$$\theta = \frac{E(X)}{E(U)}$$

ya que si conocemos esa fracción, podremos predecir la población de 1930 usando la de 1920.

Parece natural usar como estimador $\hat{\theta} = \bar{x}/\bar{u}$, pero no tenemos manera de determinar su precisión.

```
> r <- mean(x)/mean(u)
> r
 [1] 1.239019
```

Probemos primero el estimador de jackknife

```
> jack1.mat <- matrix(rep(u,49),ncol=49,byrow=T)
> jack2.mat <- matrix(rep(x,49),ncol=49,byrow=T)
> diag(jack1.mat) <- NA
> diag(jack2.mat) <- NA
> psval.r <- 49*r - 48 * apply(jack2.mat,1,mean,na.rm=T)/
+ apply(jack1.mat,1,mean,na.rm=T)
> psval.r
 [1] 0.9657763 1.1303754 1.1757607 1.6152960 1.3879002
 [6] 1.4031611 1.3734039 1.4250897 1.9443954 1.6905337
[11] 1.2860740 1.2007292 1.1526097 1.4874134 0.7119114
[16] 1.2516704 1.2011200 1.1024296 1.1561656 1.0281745
[21] 1.3292747 1.0723439 1.3004313 1.2586739 1.1772519
[26] 1.2169214 1.3389556 0.9510330 1.1384339 0.9470093
[31] 0.9344920 1.2523657 1.2126101 1.0848811 0.9113439
[36] 0.9429642 1.2796740 1.3157452 0.8797605 1.2723904
[41] 1.0822520 1.9364012 1.3772484 1.1055930 1.2120120
[46] 1.3129890 1.2076208 1.5580180 1.3288902
> r.jack <- mean(psval.r)
> r.jack
 [1] 1.237297
> se.r.jack <- sqrt(var(psval.r)/49)
> se.r.jack
 [1] 0.03453431
> r.ci.j <- c(r.jack - qt(0.975,48)*se.r.jack,
+ r.jack + qt(0.975,48)*se.r.jack)
> r.ci.j
 [1] 1.167861 1.306733
```

Calculemos ahora un intervalo de confianza basado en 1000 muestras bootstrap.

```
> u.samp.boot <- sample(u,49000,replace=T)
> x.samp.boot <- x[match(u.samp.boot,u)]
> length(x.samp.boot)
[1] 49000
> u.mat.boot <- matrix(u.samp.boot,ncol=49,byrow=T)
> x.mat.boot <- matrix(x.samp.boot,ncol=49,byrow=T)
> r.boot <- apply(x.mat.boot,1,mean)/apply(u.mat.boot,1,mean)
> r.ci.std <- c(r - qnorm(0.975)*sqrt(var(r.boot)),
+ r + qnorm(0.975)*sqrt(var(r.boot)))
> r.ci.std
[1] 1.169514 1.308523
> r.ci.per1 <- quantile(r.boot,c(.025,.975))
> r.ci.per2 <- 2*r - quantile(r.boot,c(.975,.025))
> r.ci.per1
      2.5%      97.5%
1.175811 1.313200
> r.ci.per2
      97.5%      2.5%
1.164838 1.302226
```