

UNIVERSIDAD DE SEVILLA

FACULTAD DE MATEMÁTICAS

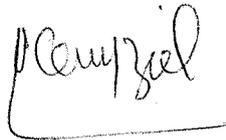
Departamento de Estadística e Investigación Operativa

UNIVERSIDAD DE SEVILLA
FACULTAD DE MATEMÁTICAS
E-41013

PROCEDIMIENTOS ORTOGONALES

APLICADOS AL BOOTSTRAP

Memoria presentada por
Mohamed Gheziel
Para optar al grado de Doctor.



Mohamed Gheziel



Fdo. Fernando López Blázquez



Fdo. Antonia Castaño Martínez

VºBº de los Directores

Sevilla, Febrero 2004

043

412

066 250
15-12-03

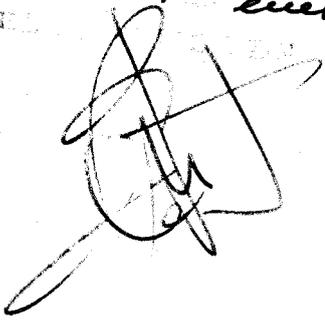
Plena Ruffin

PROCEDIMIENTOS ORTOGONALES APLICADOS AL BOOTSTRAP

MOHAMED GHEZIEL

Febrero 2004

*Estadística e I.O.
Fac. de Matemáticas
19-12-03
29-enero-2004
29 enero de 2004*



Índice General

1	INTRODUCCIÓN	1
1.1	Notaciones	3
1.2	Estimación de una distribución	5
1.2.1	Distribución empírica	5
1.2.2	Estimador paramétrico	5
1.3	Introducción al bootstrap	6
2	MÉTODOS TRADICIONALES	15
2.1	Sesgo y Varianza	15
2.2	Aproximación normal	17
2.3	Estadísticos definidos mediante un funcional	19
2.4	Método delta	20
2.4.1	Caso paramétrico	20
2.4.2	Caso no paramétrico	21
2.4.3	Desventajas de los métodos tradicionales	25
3	DESARROLLOS ORTOGONALES Y TRANSFORMADAS DE FOURIER	27
3.1	Desarrollos ortogonales	27
3.1.1	Existencia	27
3.1.2	Expresión de un estimador	30
3.1.3	Sistema ortogonal en familias NEF	31
3.1.4	Polinomios ortogonales en otras familias	40
3.2	Transformadas de Fourier	41
3.2.1	Operador discreto de Fourier	42
3.2.2	Transformadas Rápidas de Fourier, FFT	43
3.2.3	Coefficientes de Fourier	45

4	ESTIMACIÓN BOOTSTRAP	47
4.1	Estimación Bootstrap del Sesgo y de la Varianza	49
4.2	Estimación Bootstrap de la Distribución y de los Cuantiles	53
4.3	Error en la estimación Bootstrap	55
4.3.1	Introducción	55
4.3.2	Error de Muestreo	56
4.3.3	Error de Remuestreo	57
4.4	Número de simulaciones B	61
4.4.1	Aproximación del error estandar	62
4.4.2	Aproximación de los cuantiles	64
4.5	Método Jakknife-after-Bootstrap	65
5	°MÉTODOS COMPUTACIONALS	67
5.1	Bootstrap Balanceado	68
5.1.1	Introducción	68
5.1.2	Estimación bootstrap usual y balanceada	69
5.2	El Método centrado	73
5.2.1	Estimación del sesgo	74
5.2.2	Estimación de la varianza	76
5.3	Bootstrap Lineal	78
5.4	Método Bootstrap Antitético	81
6	MÉTODOS ORTOGONALES: BOOTSTRAP PARAMÉTRICO	83
6.1	Caso unidimensional	84
6.1.1	Estimación bootstrap	86
6.1.2	Resultados numéricos	93
6.2	Caso multiparamétrico	99
6.2.1	Resultados numericos	101
7	MÉTODOS ORTOGONALES: BOOTSTRAP NO PARAMÉTRICO	107
7.1	Desarrollos basados en la fdD empírica	108
7.1.1	Polinomios ortogonales asociados a F_n	108
7.1.2	Estimación bootstrap	110
7.1.3	Resultados numéricos	111
7.2	Desarrollos basados en la distribución Multinomial	116
7.2.1	Sistema ortogonal en la familia Multinomial	119
7.2.2	Resultados numéricos	122

8	ESTIMACIÓN BOOTSTRAP DE LA DENSIDAD Y DE LA DISTRIBUCIÓN VÍA FÓRMULAS DE INVERSIÓN	131
8.1	Introducción	131
8.2	Fórmulas de inversión	132
8.3	Aproximación de la densidad	133
8.3.1	Resultados numéricos	136
8.4	Aproximación bootstrap de la función de distribución	144
8.4.1	Cálculo de la distribución a partir de la aproximación de la densidad	144
8.4.2	Cálculo directo de la distribución	146
8.4.3	Resultados numéricos	147
9	IMPLEMENTACIÓN EN EL ORDINADOR	153
9.1	El paquete Estima	155
9.2	El Paquete NPestima	165
9.3	El Paquete Repita	173
9.4	Otros Programas	176
9.4.1	Familia NEF	176
9.4.2	Estimación de la Densidad y la Distribución	177
9.5	Códigos fuente de los programas	179
9.5.1	Estima.	179
9.5.2	NPestima.	195
9.5.3	Repita.	213
9.5.4	NEF	216
9.5.5	FFTestima.	223

Capítulo 1

INTRODUCCIÓN

Basándonos en una muestra (que también llamaremos datos) de tamaño n , X_1, \dots, X_n , procedente de una cierta población, el objetivo del análisis estadístico es extraer toda la información posible que conlleva la muestra. Esta información nos permite deducir las propiedades requeridas de la población que ha dado lugar a los datos, y que suponemos que es generada por una distribución F desconocida, llamada *distribución de interés*. La propiedad o la característica de la población requerida por el analista, generalmente, puede ser representada como una función de la población, digamos $\theta = t(F)$, y se llama *parámetro de interés*. Por ejemplo si se desea estimar la media de la población θ , entonces $t(F)$ se refiere a la media de la población generada por F . El análisis, generalmente, está basado en unas funciones de la muestra $T_n = T(X_1, \dots, X_n)$, llamados *Estadísticos*, o a veces, *Estimadores* si el propósito de los mismos es aproximar de forma razonable algún parámetro desconocido. Estos Estadísticos se seleccionan de acuerdo con unos criterios, como es el de máxima verosimilitud, de suficiencia o de mínima varianza. Por ejemplo, en muchos casos, la media muestral \bar{X} es preferida, entre cualquier estimador, para estimar la media de la población. El Estadístico seleccionado para el análisis será llamado *Estadístico de Interés*.

Por ser una función de la muestra, un Estadístico es una variable aleatoria que tiene una cierta distribución de probabilidad denominada *Distribución muestral* del Estadístico. En la mayoría de los procedimientos estadísticos, se necesita cierto conocimiento de la distribución muestral del estadístico seleccionado para llevar a cabo los análisis. El grado de conocimiento requerido depende del tipo de análisis. Por ejemplo, la construcción de un intervalo de confianza, o un contraste de hipótesis, necesitan el conocimiento de la distribución muestral misma o por lo menos de sus cuantiles estratégicos.

Por otro lado, en problemas de estimación, es imprescindible saber algo sobre la precisión del estimador, puesto que cada estimación puede tener un error de estimación. En este caso, el conocimiento de ciertas medidas de precisión, como la varianza, el sesgo o el error cuadrático medio, es requerido. Evidentemente, estas medidas de precisión del estimador son características de su distribución muestral. Dichas medidas de precisión, también, nos ayudan a elegir entre varios estimadores candidatos. En general, la distribución muestral de un estadístico tanto como sus características dependen de la distribución de la población F , por lo que son desconocidas y se tienen que ser aproximadas a partir de los datos disponibles.

Lamentablemente, en la mayoría de las situaciones es muy difícil, o es imposible, tener expresiones explícitas en función de los datos para las medidas de precisión, como es el caso de la varianza de un estimador dado. Tradicionalmente, cuando es posible obtener una expresión explícita teórica de cierta medida de precisión, o una expresión aproximada, entonces dicha medida se aproxima sustituyendo las cantidades desconocidas, en su expresión, por sus análogas empíricas. Los procedimientos tradicionales también utilizaban aproximaciones asintóticas para aproximar medidas de precisión. Eso sí, este último método es válido cuando el número de los datos es bastante elevado. En la práctica, no es siempre posible obtener muestras suficientemente grandes que para usar métodos asintóticos. Por este motivo los métodos tradicionales se encontraban muy limitados. Así, los investigadores siempre se preocuparon por ofrecer soluciones más flexibles y aplicables en situaciones más difíciles.

En 1949, QUENOUILLE introdujo un método, que posteriormente fue llamado *Jackknife*, para estimar el sesgo de un estimador. Dicho método se basa en el remuestreo, eliminando en cada paso un dato de la muestra original. Una década después, concretamente en 1958, TUKEY adaptó el método *Jackknife* para estimar también la varianza de un estimador. Desde entonces, el *Jackknife* ha tenido éxito y se estableció como herramienta principal, usada por los estadísticos para estimar el sesgo y la varianza en situaciones complicadas cuando los métodos tradicionales fallaban. En 1979, BRADLEY EFRON generalizó el concepto del *Jackknife* ofreciendo así un nuevo método de remuestreo, nos referimos al *Método Bootstrap*. La idea del método *Bootstrap* es esencialmente la misma idea del método *Jackknife*, la cuál es bastante evidente de modo que podría haber aparecido muchos años antes. Sin embargo, si hubiera sido así, el nuevo método no habría tenido el éxito que tuvo en los años ochenta y que sigue teniendo hasta ahora. La razón es que el método *bootstrap* se basa en el uso intensivo del ordenador. Y en los años sesenta y setenta el mundo de informática no estaba suficientemente desarrollado como para llevar a cabo la idea del método *bootstrap*.

Como hemos dicho, el bootstrap es un método, o más bien, una herramienta muy útil de la inferencia estadística que se basa en la computación intensiva. La revolución informática que empezó a finales de los setenta y a principios de los ochenta ayudó mucho a que el método saliera adelante con todo el éxito que merece. Así, el bootstrap despertó un gran interés entre los investigadores durante toda la década de los ochenta, por lo que, el método fue adaptado a muchas situaciones de interés estadístico, como es la estimación, la construcción de intervalos de confianza, el contraste de hipótesis, la regresión etcétera. El método bootstrap es capaz de responder a muchas cuestiones en situaciones complicadas sin necesidad de cálculos teóricos ni de fórmulas previas. Por ejemplo, este método nos permite obtener gran información sobre un estadístico dado, como su sesgo, su varianza o su error cuadrático medio, usando sólo los datos disponibles, la muestra. También nos permite construir intervalos de confianza de un parámetro o decidir entre dos hipótesis, sin necesidad de aproximaciones asintóticas.

1.1 Notaciones

En esta sección presentamos las notaciones más comunes usadas en esta memoria. Los parámetros desconocidos serán notados generalmente por las letras griegas θ y β . Las letras F y f serán reservadas respectivamente para la función de distribución de la población de interés y su función de densidad que suponemos desconocidas parcialmente (como en el caso paramétrico) o totalmente. Las letras X e Y serán usadas para variables aleatorias generalmente desconocidas. Así mismo, X_1, X_2, \dots, X_n denotará una muestra de variables aleatorias independientes idénticamente distribuidas (iid) de tamaño n , escribiremos $X_1, X_2, \dots, X_n \rightsquigarrow F$ iid para declarar que la muestra proviene de la distribución F . A veces usaremos \mathcal{X}_n para notar una muestra de tamaño n . Asimismo, con datos nos referiremos a la muestra o a una realización de la misma.

Con $\hat{\theta}_n = \hat{\theta}(X_1, \dots, X_n)$ o simplemente $\hat{\theta}$ denotaremos un estimador del parámetro θ . Por \hat{F} denotaremos la distribución estimada de F , que puede ser el estimador no paramétrico que será la distribución empírica, F_n , asociada a X_1, X_2, \dots, X_n , o el estimador paramétrico, $F_{\hat{\theta}}$, de F en situaciones paramétricas cuando $F = F_{\theta}$. Para el estimador de θ usaremos también la notación $T_n = T(X_1, \dots, X_n) = \hat{\theta}$, para reflejar la dependencia de $\hat{\theta} = T_n$ de los datos X_1, X_2, \dots, X_n , o bien la notación $T_n = T(\hat{F})$ para reflejar la dependencia de $\hat{\theta}$ de una distribución estimada \hat{F} , concretamente cuando θ tiene la representación $\theta = T(F)$, para un cierto funcional T . Con G_n y g_n denotaremos la distribución y la densidad de T_n , el estadístico de interés o de una

variable aleatoria $R_n(T_n, F)$ (o $R_n(\mathcal{X}_n, F)$), variable de interés, que es función de T_n como por ejemplo la variable estudentizada basada en T_n o simplemente la variable $T_n - \theta$.

La esperanza, varianza, probabilidad etc serán notadas como es usual por $E(\cdot)$, $\text{Var}(\cdot)$, $P(\cdot)$ etc. Cuando necesitemos apuntar a que los cálculos de dichas cantidades se hacen respecto a una distribución H , escribiremos $E_H(\cdot)$, $\text{Var}_H(\cdot)$, $P_H(\cdot)$ etc, y si $H = H_\xi$ es parametrizada por el parámetro ξ , escribiremos simplemente $E_\xi(\cdot)$, $\text{Var}_\xi(\cdot)$, $P_\xi(\cdot)$ etc. en lugar de $E_{H_\xi}(\cdot)$, $\text{Var}_{H_\xi}(\cdot)$, $P_{H_\xi}(\cdot)$ etc.

Las variables con una estrella como X^* y Y^* denotaran variables aleatorias con distribución $F^* = \hat{F}$, la estimada de F . Entonces, X_1^*, \dots, X_n^* sera una muestra extraída de \hat{F} y se llama *muestra bootstrap*. Las notaciones como $E^*(\cdot)$, $\text{Var}^*(\cdot)$ y $P_*(\cdot)$ serán la esperanza, varianza y probabilidad bajo la distribución estimada $F^* = \hat{F}$, ie, $E^*(\cdot) = E_{\hat{F}}(\cdot)$, $\text{Var}^*(\cdot) = \text{Var}_{\hat{F}}(\cdot)$ y $P_*(\cdot) = P_{\hat{F}}(\cdot)$ etc. Estas últimas notaciones también corresponden a la esperanza, varianza y probabilidad condicional siendo conocida la muestra original X_1, \dots, X_n . Asimismo, notaciones como $\hat{\theta}^* = T_n^*$ corresponden a $T(X_1^*, \dots, X_n^*)$ la réplica del estadístico $\hat{\theta} = T_n = T(X_1, \dots, X_n)$ calculada usando la muestra X_1^*, \dots, X_n^* . La letra B será usada para el número de simulaciones usadas, por lo que es el número de replicas de T_n , notadas como $T_n^{*1}, \dots, T_n^{*B}$. Cuando hablamos de los estadísticos ordenados ponemos los índices entre paréntesis como $X_{(1)} < \dots < X_{(n)}$ o $T_n^{*(1)} < \dots < T_n^{*(B)}$. Reservamos la letra Z para denotar la variable normal estándar, $N(0, 1)$, cuya distribución sera por Φ y su densidad ϕ . Las letras Q ó Q_n generalmente denotarán variables pivotes como variables estudentizadas cuya distribución es conocida. Una distribución degenerada en el punto x sera notada por δ_x como es usual. A menudo usamos $\mu = \mu_H = \mu_X$, y $v = v_H = v_X$ para notar la media y la varianza de una población representada por X cuya distribución es H . Y como es usual $\bar{X} = \bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ y $S^2 = S_n^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$ son la media y la varianza muestrales. El signo \sim en expresiones como $X \sim F$ donde X es una variable aleatoria y F es una función de distribución, significa que X esta distribuida "aproximadamente" como F . El signo \simeq en ecuaciones como $A \simeq B$ significa que A es "aproximadamente" igual a B , y el signo \equiv , como es habitual, significa "equivalente". El signo $\#$ como en $\#A$, donde A es un conjunto significan "Cardinal" de A , es decir, el número de los elementos de A , y el signo \mathbb{I} en \mathbb{I}_A es el "Indicador" del conjunto A , es decir la función definida por $\mathbb{I}_A(y)=1$ si y es un elemento de A y $\mathbb{I}_A(y)=0$ en caso contrario.

1.2 Estimación de una distribución

Sea X_1, \dots, X_n una muestra iid con distribución F . Por el papel fundamental que juega la "distribución estimada" de la distribución de interés en la teoría del Bootstrap, definiremos aquí los principales estimadores utilizados usualmente en dicha teoría. Las principales estimaciones de F se restringen al estimador no paramétrico F_n que es la distribución empírica asociada a los datos X_1, \dots, X_n , y el estimador paramétrico $F_{\hat{\beta}}$ cuando $F = F_{\beta}$ pertenece a una clase paramétrica, parametrizada con el parámetro β , donde $\hat{\beta}$ es un estimador consistente de β y suele ser el estimador de máxima verosimilitud

1.2.1 Distribución empírica

La distribución empírica asociada a la muestra X_1, \dots, X_n se define como la función de distribución de una variable aleatoria X^* que toma los n valores $\{X_1, \dots, X_n\}$ con igual probabilidad n^{-1} . Es decir

$$X^* \rightsquigarrow F_n \implies P\{X^* = X_j\} = \frac{1}{n}, \quad j = 1, \dots, n.$$

Entonces, para cualquier punto x ,

$$F_n(x) = \frac{\#\{j : X_j \leq x\}}{n}. \quad (1.1)$$

Podemos expresar a $F_n(x)$ en términos de distribuciones degeneradas en x_j o en términos de los indicadores de $\{X_j \leq x\}$ como sigue

$$F_n(x) = \frac{1}{n} \sum_{j=1}^n \delta_{X_j}(x) = \frac{1}{n} \sum_{j=1}^n \mathbb{I}_{\{X_j \leq x\}}.$$

Esta distribución juega un papel muy importante en la teoría bootstrap, ya que cuando no hay suficiente información sobre la distribución de interés F , siempre se recurre a F_n .

1.2.2 Estimador paramétrico

Supongamos que sabemos que la muestra X_1, \dots, X_n proviene de una cierta distribución F_{β} con forma funcional conocida salvo el parámetro β que es desconocido. β puede ser un escalar o un vector de parámetros. Primero, estimamos β mediante

un estimador consistente $\hat{\beta} = \hat{\beta}(X_1, \dots, X_n)$ por ejemplo el estimador de máxima verosimilitud o el umvue de β . Después, sustituimos el parametro β en la expresión de F_β por su estimación $\hat{\beta}$, obtendremos el estimador de $F = F_\beta$,

$$\hat{F} = F_{\hat{\beta}}. \quad (1.2)$$

Este estimador es un estimador paramétrico de F . Cuando $\hat{\beta}$ es el estimador de máxima verosimilitud de β , entonces $F_{\hat{\beta}}$ es también el estimador de máxima verosimilitud de F_β . Este estimador paramétrico juega un papel muy importante en la teoría bootstrap siempre cuando un modelo paramétrico para la distribución de interés esté disponible.

1.3 Introducción al bootstrap

En esta sección introduciremos el método bootstrap mediante dos ejemplos muy sencillos en los que los cálculos teóricos son fáciles de llevar a cabo. Esto nos permitirá comparar los resultados del método bootstrap con los teóricos. Nuestro objetivo de esta sección es la introducción al método bootstrap mediante una ilustración numérica sencilla, aunque la explicación del método se hará en capítulos posteriores. El primer ejemplo consiste en datos para los cuales un modelo paramétrico parece ser adecuado. Al contrario, en el segundo ejemplo ningún modelo para los datos esta disponible.

Ejemplo A. En una fábrica de componentes eléctricos se ha medido el tiempo de vida de 12 bombillas seleccionadas de forma aleatoria. Los resultados obtenidos son:

$$135, 20, 7, 86, 420, 5, 45, 103, 230, 10, 94, 100. \quad (1.3)$$

Entonces la muestra X_1, \dots, X_{12} consiste en estos $n = 12$ tiempos de vida de las 12 bombillas. La fabrica esta interesada a estimar el tiempo medio, μ , de vida de sus bombillas. A este tipo de datos se le puede asociar un modelo exponencial. Es decir que podemos suponer que la muestra en (1.3) proviene de una cierta distribución exponencial con media, μ ,

$$F_\mu(x) = \begin{cases} 1 - \exp(-x/\mu), & \text{si } x > 0 \\ 0, & \text{si } x \leq 0. \end{cases} \quad (1.4)$$

Aquí la expresión explícita de la distribución de interés F_μ es conocida excepto el parámetro μ . La *Figura 1-1* muestra la representación de los valores de la muestra

ordenada en el eje vertical en frente de los cuantiles de la distribución exponencial estándar en el eje horizontal

$$F_{\mu}^{-1}\left(\frac{j}{n+1}\right)\Bigg|_{\mu=1} = -\log\left(1 - \frac{j}{n+1}\right).$$

En la figura, la recta corresponde a los cuantiles del modelo $F_{\bar{X}}$ frente a los de la distribución exponencial estándar.

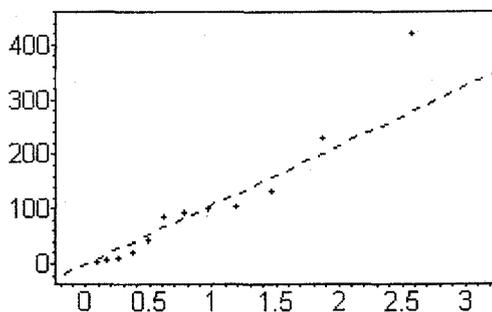


Figura 1.1

Podemos ver que los datos se ajustan razonablemente al modelo elegido, $F_{\bar{X}}$, aunque en nuestra situación el tamaño de la muestra es bastante pequeño como para confiar en el modelo. Este tipo de gráficos en la *Figura 1-1* nos permiten verificar el ajuste de los datos al modelo elegido. Otra alternativa más evidente es dibujar la función de distribución empírica junto con el modelo paramétrico elegido para los datos.

Evidentemente, el parámetro de interés es $\theta = \mu$. Sea $\hat{\theta} = T(X_1, \dots, X_{12}) = \bar{X}$ el estadístico de interés. Su valor observado es $\bar{X} = 104.58$. Ahora bien nos queda la incertidumbre sobre la elección de dicho estadístico como estimador de μ .

Por lo general en esta memoria, siempre suponemos que la selección del estimador de interés ya está hecha. Nuestro objetivo es estudiar sus medidas de precisión como su varianza $v = v_n$ y su sesgo $s = s_n$, o construir un intervalo de confianza para θ por lo que necesitamos calcular la distribución, G_n , de $\hat{\theta}$. Todo esto, el método bootstrap es capaz de ofrecerlo sin necesidad de fórmulas explícitas para v_n , s_n y G_n como veremos más adelante.

En esta sección nos limitamos a ofrecer resultados numéricos obtenidos por los métodos bootstrap y mediante ciertos métodos teóricos. Los detalles de la estimación bootstrap serán presentadas en el Capítulo 3 con más cuidado. En nuestro ejemplo, las fórmulas teóricas para la varianza y el sesgo de $\hat{\theta}$ vienen dadas por

$$v = \text{var}_{\mu} \bar{X} = \frac{\text{var}_{\mu} X_1}{12} = \frac{\mu^2}{12} \text{ y } s = \text{sesg}_{\mu}(\bar{X}) = E_{\mu}(\bar{X}) - \mu = 0 \quad (1.5)$$

ya que bajo F_μ la varianza de X_1 es μ^2 . A partir de estas fórmulas podemos concluir por lo menos que el estimador elegido es insesgado. La fórmula de la varianza en (1.5) es exacta pero desconocida porque μ lo es. Es natural pensar en estimarla, a primera vista, por el estimador natural

$$\hat{v} = \frac{\bar{X}^2}{12} = 911.4728 \quad (1.6)$$

Ahora veremos que este estimador no es nada mas que el estimador bootstrap teórico bajo el modelo paramétrico. En la teoría bootstrap, primero se estima la distribución F_μ por $F_{\bar{X}}$ y luego para hacer cálculos consideremos que la muestra proviene de $F_{\bar{X}}$ en lugar de F_μ . Es decir, para estimar por ejemplo la varianza v y el sesgo s de $\hat{\theta}$, consideremos una muestra X_1^*, \dots, X_{12}^* iid con distribución $F_{\bar{X}}$ y estimamos a v y s respectivamente por la varianza, v^* , y el sesgo, s^* , de $\hat{\theta}^* = T(X_1^*, \dots, X_{12}^*) = \bar{X}^*$. Obtendremos

$$\begin{aligned} v^* &= \text{var}_{\bar{X}}(\bar{X}^*) = \frac{\text{var}_{\bar{X}}(X_1^*)}{12} = \frac{\bar{X}^2}{12} = 911.4728 \text{ y} \\ s^* &= \text{sesg}_{\bar{X}}(\bar{X}^*) = E_{\bar{X}}(\bar{X}^*) - \bar{X} = 0. \end{aligned} \quad (1.7)$$

Antes de todo, notemos que el método bootstrap nos ofrece la misma conclusión que, teóricamente, tenemos sobre el sesgo. Es decir, la fórmula del sesgo en (1.7), al igual que en (1.5), nos dice que $\hat{\theta}^*$ es insesgado, lo que quiere decir que nuestro estimador original es insesgado. Por otra parte, el estimador bootstrap de la varianza v^* en (1.7) es exactamente el mismo estimador que el estimador paramétrico natural \hat{v} de v en (1.6).

Como hemos dicho antes, en este sencillo ejemplo, la obtención de fórmulas explícitas para las medidas de precisión del estimador de interés es inmediata. Sin embargo, en la mayoría de las situaciones ésto no es posible. Supongamos, por ejemplo, que nuestro interés en este ejemplo fuera la estimación del parámetro $\theta = \log(\mu)$ usando el estadístico $\hat{\theta} = \log(\bar{X})$. Entonces, la obtención de fórmulas explícitas para la varianza y el sesgo de $\hat{\theta}$ como en (1.5) no es fácil. Entonces, encontrar estimadores intuitivos como hemos hecho para obtener a \hat{v} en (1.6) es imposible. Aunque pasemos a trabajar bajo el modelo $F_{\bar{X}}$ con el fin de obtener estimadores para la varianza y el sesgo de $\hat{\theta}$, como hemos hecho para obtener v^* y s^* en (1.7), no lograremos fórmulas explícitas. En esta situación, los métodos tradicionales usan el método delta para aproximar v y s . Sin embargo, el uso del modelo $F_{\bar{X}}$ por el método bootstrap, aunque no nos ha llevado a fórmulas exactas para v^* y s^* como en (1.7), siempre nos permite aproximarlas mediante muestreo múltiple a partir de $F_{\bar{X}}$, como veremos en

el *Capítulo 4*. En el *Capítulo 6*, presentaremos un método basado sobre los desarrollos ortogonales para aproximar v^* y s^* , este método mejora considerablemente el método bootstrap estándar.

Volviendo a nuestro ejemplo, donde $\theta = \mu$ y $\hat{\theta} = \bar{X}$. Con el fin de comparar el método de muestreo múltiple, para aproximar v^* y s^* , con los valores exactos en (1.7), hemos simulado $B = 20$ muestras de tamaño 12 a partir de $F_{\bar{X}}$, y hemos calculado las $B = 20$ replicas de $\hat{\theta} = \bar{X}$ correspondientes a cada muestra, digamos

$$\hat{\theta}^{*1} = \bar{X}^{*2}, \dots, \hat{\theta}^{*20} = \bar{X}^{*20} \quad (1.8)$$

Tomamos a la varianza muestral, $\hat{v}_{MC,20}^*$, y al sesgo muestral, $\hat{s}_{MC,20}^*$, correspondientes a la muestra en (1.8) como aproximaciones a v^* y s^* respectivamente. Los resultados numéricos nos han dado

$$\hat{v}_{MC,20}^* = 720.262 \text{ y } \hat{s}_{MC,20}^* = -2.438$$

Usando la misma muestra de replicaciones que en (1.8) los métodos ortogonales del *Capítulo 6* nos han dado las aproximaciones

$$\tilde{v}_{Ort,20}^* = 911.47279 \text{ y } \tilde{s}_{Ort,20}^* = 10^{-7}.$$

Comparando estos últimos resultados con los valores teóricos en (1.7) se ve que son prácticamente iguales. Los resultados para diferentes números de simulaciones están resumidos en la *Tabla 1-1*, donde $B = \infty$ corresponde a los estimadores bootstrap exactos v^* y s^* en (1.7) que llamaremos estimadores bootstrap ideales.

B	\hat{v}_{MC}^*	\tilde{v}_{Ort}^*	\hat{s}_{MC}^*	\tilde{s}_{Ort}^*
10	466.808	911.47276	5.334	10^{-7}
20	720.262	911.47279	-2.438	0
50	727.32	911.47279	4.759	10^{-7}
100	900.666	911.47279	0.031	0
200	877.649	911.47279	-1.345	0
∞	911.4728		0	

Tabla 1-1.

Comparando numéricamente \hat{v}_{MC}^* y \tilde{v}_{Ort}^* al estimador bootstrap teórico $v^* = 911.4728$, y \hat{s}_{MC}^* y \tilde{s}_{Ort}^* al estimador bootstrap teórico $s^* = 0$ se ve claramente que los estimadores obtenidos mediante métodos ortogonales son muy exactos respecto a los estimadores bootstrap estándares, aun cuando el número de simulaciones es muy bajo (por ejemplo $B = 10$). Generalmente los estimadores estándares como

\hat{v}_{MC}^* y \hat{s}_{MC}^* se aproximan a sus valores esperados, que son los estimadores ideales v^* y s^* , a medida que el número de simulaciones crece. Por lo tanto, para números de simulaciones no suficientemente grandes podemos obtener aproximaciones no fiables. Sin embargo, hemos notado, que el uso de métodos ortogonales de los capítulos 6 y 7 nos dan aproximaciones muy exactas aunque con número de simulaciones muy bajo.

En la *Figura 1-2* hemos calculado cuatro replicas de $\hat{v}_{MC,B}^*$ y $\hat{s}_{MC,B}^*$ repitiendo cuatro veces las simulaciones a partir $F_{\bar{X}}$, para cada uno de los ocho números $B = 10, 20, 50, 100, 200, 300, 400$ y 500 . Notemos como varían las estimaciones para cada una de las cuatro repeticiones. A medida que el número B crece las estimaciones se acercan a las líneas horizontales que representan la estimación bootstrap ideal.

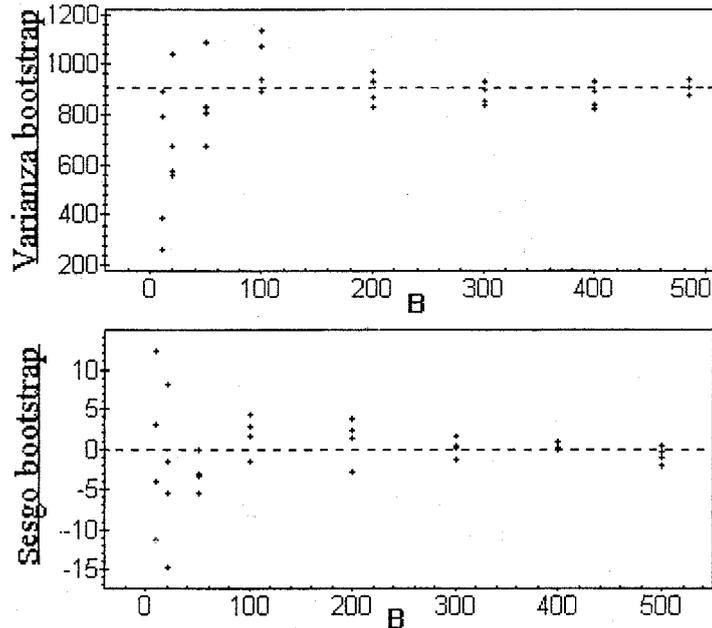


Figura 1-2.

Nota. En el ejemplo A hemos usado el estimador paramétrico, $F_{\bar{X}}$, para la distribución de interés porque este estimador nos permite aprovechar la información que tenemos sobre el aspecto de F_{μ} . Sin embargo, nada nos impide usar el estimador no paramétrico F_n de F_{μ} . Bajo el modelo no paramétrico F_n obtendremos las

siguientes estimaciones bootstrap ideales para v y s :

$$v^* = \frac{\text{var}_{F_n}(X_1^*)}{12} = \frac{1}{12^2} \sum_{j=1}^{12} (X_j - \bar{X})^2 = 1081.34 \text{ y}$$

$$s^* = E_{F_n}(\bar{X}^*) - \bar{X} = 0.$$

Lo mismo, para aproximar v^* y s^* cuando faltan fórmulas explícitas podemos usar F_n para hacer simulación.

Ejemplo B. En este ejemplo los datos consisten en $n = 49$ pares. Cada par $Y_i = (U_i, X_i)$ consiste en los números de habitantes de una ciudad i elegida al azar en USA en 1920, U_i , y 1930, X_i . Los datos están registrados en la *Tabla 1-2*

U	138	93	61	179	48	37	29	23	30	2
X	143	104	69	260	75	63	50	48	111	50
U	67	71	25	286	74	50	161	256	94	36
X	67	79	57	317	93	58	232	288	85	46
U	76	381	387	78	60	507	50	77	64	40
X	80	464	459	106	57	634	64	89	77	60
U	120	172	66	46	121	44	64	56	40	116
X	115	183	86	65	113	58	63	142	64	130
U	36	46	243	43	43	38	45	136	87	
X	54	53	291	61	50	52	53	139	105	

Tabla 1-2.

El número total de habitantes en 1930 será estimado por el número total de habitantes en 1920 multiplicado por la razón de las medias $\theta = E(X)/E(U)$. En esta situación no parece claro la disponibilidad de un modelo paramétrico para la distribución conjunta de $Y = (U, X)$. Es natural estimar entonces θ con $\hat{\theta} = \bar{X}/\bar{U}$, el cociente de las medias muestrales. Pero ¿Cómo de preciso es nuestro estimador $\hat{\theta}$ de θ ? Es decir que una vez seleccionado el estimador queda incertidumbre sobre él. Para responder a la cuestión anterior tenemos que estudiar ciertas medidas de precisión de T_n como es su varianza, v , y su sesgo, s .

Si tuviésemos un modelo paramétrico para la distribución de X , entonces los cálculos teóricos, como en el ejemplo A nos podrían llevar a estimar v y s . Sin

embargo, el modelo no paramétrico, F_n , está siempre disponible para llevar a cabo los cálculos.

Por conveniencia, consideremos las primeras diez observaciones de pares (u, x) en la *Tabla 1.2*. Trataremos de estimar ciertas características de $\hat{\theta}$, como su varianza, v , su sesgo, s , y su distribución, G , simulando a partir del modelo no paramétrico F_{10} que es la distribución empírica asociada a dichas diez observaciones. La *Tabla 1.3* muestra los resultados correspondientes a la simulación de 9 muestras a partir F_{10} , cada muestra es de tamaño $n = 10$.

i	1	2	3	4	5	6	7	8	9	10	
u	138	93	61	179	48	37	29	23	30	2	
x	143	104	69	260	75	63	50	48	111	50	
Número de apariciones de cada observación											
Datos	1	1	1	1	1	1	1	1	1	1	Estadístico
Replicas											$\hat{\theta} = 1.520$
1	1	2	0	1	0	1	1	1	2	1	$\hat{\theta}_1^* = 1.596$
2	1	1	0	1	1	2	1	2	0	1	$\hat{\theta}_2^* = 1.484$
3	0	1	1	2	1	0	3	0	1	1	$\hat{\theta}_3^* = 1.589$
4	2	3	0	2	1	1	0	0	1	0	$\hat{\theta}_4^* = 1.329$
5	1	0	2	1	1	2	0	1	0	1	$\hat{\theta}_5^* = 1.513$
6	0	1	0	3	0	1	0	2	1	2	$\hat{\theta}_6^* = 1.678$
7	0	1	2	0	1	2	0	2	2	0	$\hat{\theta}_7^* = 1.717$
8	0	0	2	0	1	1	0	4	1	1	$\hat{\theta}_8^* = 1.900$
9	1	0	0	1	2	3	1	0	2	1	$\hat{\theta}_9^* = 1.654$

Tabla 1-3.

La estimación de la varianza y del sesgo de $\hat{\theta}$ basándonos en las 9 réplicas registradas en la última columna de la *Tabla 1-3* ha dado

$$v_9^* = 0.026 \text{ y } s_9^* = 0.0868.$$

Podemos, por ejemplo, también aproximar la distribución de $\hat{\theta}$ por la distribución empírica, G_9^* , asociada a las 9 réplicas $\hat{\theta}_b^*$, $b = 1, \dots, 9$ en la *Tabla 1-3*. Otra aproximación alternativa para G_9^* , puede ser la normal $N(s_9^*, v_9^*)$, aunque esta última no la podemos considerar precisa debido al pequeño número, 9, de réplicas de $\hat{\theta}$ disponibles en este ejemplo.

Notemos también que podemos usar las mismas simulaciones, obtenidas anteriormente, para aproximar la distribución de $\hat{\theta} - \theta$ o de $Q = (\hat{\theta} - \theta)/\hat{v}$. En la *Figura 1-3*, podemos ver los histogramas bootstrap correspondientes a $\hat{\theta}^*$ y Q^* . Podemos notar, por ejemplo, que el histograma de $\hat{\theta}^*$ no es simétrico y que tiene la cola larga a la derecha, por lo que podemos juzgar a que la aproximación normal $N(s_{\theta}^*, v_{\theta}^*)$, dicha antes, es imprecisa.

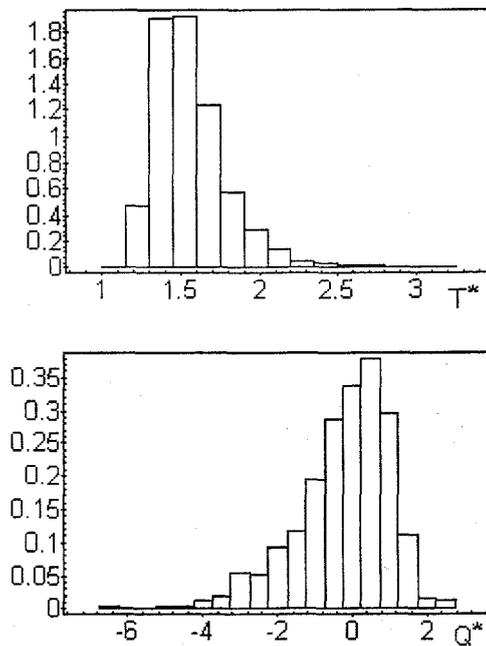


Figura 1-3.

Para el histograma de $Q^* = (\hat{\theta}^* - \hat{\theta})/\hat{v}^*$ hemos usado la estimación delta de la varianza (Ver *Capítulo 2*)

$$\hat{v}^* = v_L^* = \frac{1}{n^2} \sum_{j=1}^n \frac{(X_j^* - \hat{\theta}^* U_j^*)^2}{\bar{U}^{*2}}$$

Las estimaciones de la varianza y el del sesgo bootstrap de $\hat{\theta}$ basadas sobre diversos

números de simulaciones B estan registrados en la *Tabla 1.3*.

B	\hat{v}_{MC}^*	\hat{v}_{Ort}^*	\hat{s}_{MC}^*	\hat{s}_{Ort}^*
70	0.050	0.046	-0.020	0.043
200	0.055	0.048	0.045	0.039
500	0.049	0.048	0.042	0.038
Ideal ($B=10000$)	0.0479		0.0363	

Tabla 1.3.

En el *Capítulo 7* veremos que el error cuadrático medio cometido usando los estimadores \hat{v}_{Ort}^* y \hat{s}_{Ort}^* , basados sobre métodos ortogonales, es menor que el cometido usando los estimadores \hat{v}_{MC}^* y \hat{s}_{MC}^* , basados sobre el método bootstrap estándar.

Capítulo 2

MÉTODOS TRADICIONALES

Probablemente, el paso crítico en el análisis estadístico es el uso de los datos para estimar o aproximar ciertas medidas de precisión, como la varianza, el sesgo o el error cuadrático medio, de un estimador dado. En este capítulo, veremos los métodos más comunes usados en los procedimientos tradicionales para estimar o aproximar medidas de precisión. En el procedimiento tradicional, generalmente, una medida de precisión se estima por la análoga empírica de una fórmula explícita teórica de la medida en cuestión, calculada bajo el modelo de interés (desconocido). Cuando la obtención de una fórmula exacta es difícil, se recurre a la búsqueda de una fórmula aproximada para la medida. A menudo, se consideran expresiones asintóticas para las medidas de precisión. Como ilustración, consideremos la estimación del sesgo y la varianza.

2.1 Sesgo y Varianza

Sea X_1, \dots, X_n una muestra aleatoria (m.a.) de variables aleatorias (v.a.) independientes e idénticamente distribuidas (i.i.d.) con función de distribución F y función de densidad f desconocidas. Supongamos que queremos estimar una característica θ de la población generada por F . Sea $T_n = T(X_1, \dots, X_n)$ el estadístico seleccionado para estimar θ . Teóricamente, el sesgo y la varianza de T_n como estimador de θ vienen definidos por:

$$\text{sesg}(T_n) = \int T(x) \prod_{i=1}^n dF(x_i) - \theta \quad (2.1)$$

y

$$\text{var}(T_n) = \int \left[T(x) - \int T(y) \prod_{j=1}^n dF(y_j) \right]^2 \prod_{i=1}^n dF(x_i) \quad (2.2)$$

respectivamente, donde $x = (x_1, \dots, x_n)$ y $y = (y_1, \dots, y_n)$.

Cuando T_n tiene una expresión sencilla es posible obtener una expresión explícita para su sesgo y su varianza aunque quizás dependiendo de ciertos valores desconocidos relacionados con F , como se vió en el ejemplo A. La estimación del sesgo o de la varianza de T_n se hace entonces sustituyendo dichas cantidades desconocidas por sus estimaciones. Por ejemplo, si $T_n = \bar{X}_n^2$ entonces su varianza viene dada por

$$\text{var}(\bar{X}_n^2) = \frac{4\mu^2 m_2}{n} + \frac{4\mu m_3}{n^2} + \frac{m_4}{n^3}, \quad (2.3)$$

con $\mu = EX_1$ y $m_k = E(X_1 - \mu)^k$, $k = 2, 3, 4$. En este caso, los valores desconocidos son, evidentemente, μ y m_k $k = 2, 3, 4$. Estos valores pueden ser estimados por

$$\hat{\mu} = \bar{X}_n \text{ y } \hat{m}_k = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X}_n)^k, \quad k = 2, 3, 4.$$

Desafortunadamente, las fórmulas como en (2.1) o (2.2) no siempre son útiles que para obtener fórmulas como en (2.3). Un ejemplo sencillo para ello lo tenemos en el ejemplo B. Para la mayoría de los estadísticos que aparecen en la práctica, la fórmula de la varianza (2.2) es demasiado complicada como para llevar a cabo la estimación de $\text{var}(T_n)$, buscando fórmulas explícitas. Consideremos por ejemplo, la estimación de la varianza del estadístico *Media Trimmed* definido como:

$$\bar{X}_n^\alpha = \frac{1}{n - 2[n\alpha]} \sum_{i=[n\alpha]+1}^{n-[n\alpha]} X_{(i)}$$

donde $[x]$ representa la parte entera de x . Una expresión explícita para la varianza de este estimador no existe. Sin embargo, para esta situación los métodos tradicionales ofrecen otras alternativas para calcular la varianza de tal estimador. Dichas alternativas consisten en intentar buscar expresiones aproximadas para $\text{var}(T_n)$, o más bien, considerar expresiones asintóticas para $\text{var}(T_n)$. Bajo ciertas condiciones de regularidad, a menudo, se puede establecer que

$$\lim_{n \rightarrow \infty} [n \cdot \text{var}_F(T_n)] = v(F) = v(\theta), \quad (2.4)$$

donde $v(\cdot)$ es una función simple de F o de un parámetro θ desconocido, como ocurre en situaciones paramétricas. Por ejemplo si $T_n = \bar{X}_n^2$, se puede ver, además de (2.3), que $\lim_{n \rightarrow \infty} n \text{ var}(T_n) = 4m_2\mu^2$, por lo que se puede aproximar $\text{var}(T_n)$ por $4m_2\mu^2/n$ que es simplemente el término líder en la parte derecha de (2.3). En el caso de la *media trimmed*, una expresión explícita de la varianza no existe, pero se puede probar que (2.4) se verifica con

$$v(F) = \frac{2}{(1-2\alpha)^2} \left[\int_0^{F^{-1}(1-\alpha)} x^2 dF(x) + \alpha F^{-1}(1-\alpha) \right],$$

donde

$$F^{-1}(t) = \inf\{x : F(x) \geq t\}.$$

Entonces podemos estimar $\text{var}(\bar{X}_n^\alpha)$ por $v(\hat{F})/n$ con \hat{F} una estimación de F , por ejemplo la empírica F_n .

2.2 Aproximación normal

Muchas aproximaciones, en los procedimientos clásicos, se basan en la teoría normal. Consideremos la estimación de la distribución $G_n(x)$ de una variable aleatoria $R_n = R_n(\mathcal{X}, F)$ relacionada con el estadístico de interés T_n . Por ejemplo, R_n puede ser la diferencia $T_n - \theta$ o la variable estandarizada $(T_n - \theta)/S_n$ donde S_n es una estimación de la desviación estandar de T_n . En muchas situaciones, concretamente para los parámetros de localización, la estimación de G_n se basa en la teoría normal. Así, $G_n(x)$ se aproxima por una cierta distribución $\Psi_\sigma(x)$ que posiblemente depende de algún parámetro desconocido σ relacionado con la distribución de interés F . Por ejemplo, cuando $R_n = T_n - \theta$, $\Psi_\sigma(x)$ suele ser $\Phi(x\sqrt{n}/\sigma)$, con Φ la distribución normal estandar, y cuando $R_n = (T_n - \theta)/S_n$, $\Psi_\sigma(x)$ suele ser la distribución $\Phi(x)$. La estimación de $G_n(x)$ se reduce, entonces, a la estimación del parámetro σ por un cierto estimador consistente $\hat{\sigma}$ dando lugar a $\hat{G}_n(x) = \Psi_{\hat{\sigma}}(x)$ como una estimación de $G_n(x)$. En particular, la construcción de un intervalo de confianza para θ suele estar basado en la aproximación $\Phi(x)$ a la distribución de $R_n = (T_n - \theta)/S_n$. Esto da lugar al intervalo estandar de confianza $1 - 2\alpha$,

$$T_n - S_n z_\alpha \leq \theta \leq T_n + S_n z_\alpha \quad (2.5)$$

donde z_α es el cuantil α de la distribución normal Φ . Sin embargo, este tipo de aproximación es bastante débil en muchas situaciones sobre todo cuando el tamaño muestral n no es suficientemente grande, como suele ocurrir en práctica.

Sea X_1, \dots, X_n una muestra aleatoria simple procedente de la familia $\text{exp}(2)$, y $T_n = \bar{X}_n$. Los Gráficos en la *Figura 2-1* muestran como la densidad de \bar{X}_n se aproxima a la densidad normal a medida que n crece.

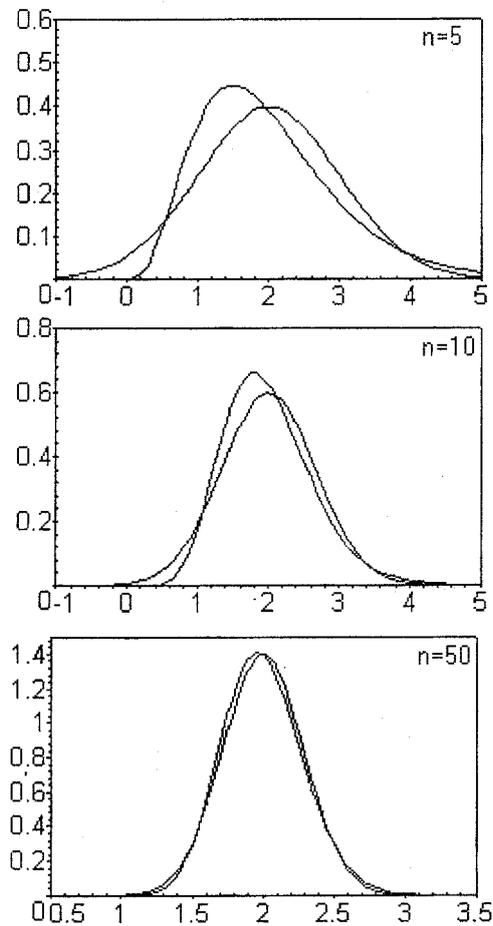


Figura 2.1

En la *Figura 2-2* se compara la verdadera densidad de \bar{X} con $n = 10$ frente a la aproximación normal representada por los puntos, y la aproximación Bootstrap basada en 500 replicas representada por el histograma. Observamos que la aproxi-

mación Bootstrap es más precisa que la aproximación normal.

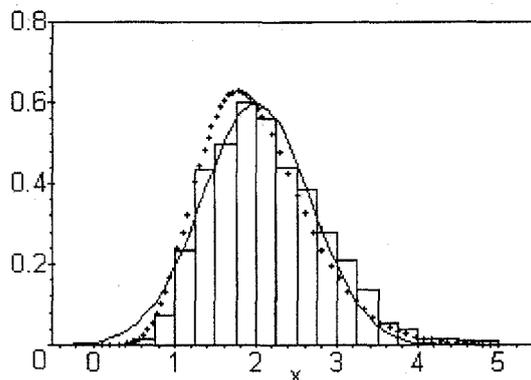


Figura 2.2

2.3 Estadísticos definidos mediante un funcional

Sea X_1, \dots, X_n una muestra *iid* con distribución desconocida F , y \mathcal{F} la clase convexa de todas las distribuciones que contiene a F y a todas las distribuciones degeneradas. Muchos parámetros de interés, θ , pueden ser definidos mediante un cierto funcional T definido sobre \mathcal{F} . Es decir θ es de la forma

$$\theta = T(F),$$

y que quiere decir que θ es una característica de la población generada por la distribución de interés F . Por ejemplo, la media de la población puede ser escrita como $\mu = T(F)$ con $T(G) = \int x dG(x)$ para toda distribución G en \mathcal{F} . Es natural, entonces, pensar en estimar la característica θ de la población generada a partir de F por la misma característica, definida por T , de la población generada por una distribución \hat{F} , siendo \hat{F} una estimación razonable de F . Es decir estimar θ por

$$\hat{\theta} = T(\hat{F}).$$

La relación que une el estadístico $T_n = \hat{\theta}$ a \hat{F} es la misma relación que une a θ con F y está definida por T . Por tanto si \hat{F} converge hacia F , y $T(\cdot)$ es continuo en algún sentido, se tiene que $\hat{\theta} = T(\hat{F})$ converge hacia $\theta = T(F)$, lo que significa que $\hat{\theta}$ así definido es consistente. Por ejemplo, si consideramos como estimación de F la distribución empírica $\hat{F} = F_n$, entonces podemos ver que la media muestral \bar{X}_n

viene definida por T definido anteriormente como sigue

$$\hat{\mu} = T(F_n) = \int x dF_n(x) = \sum_{i=1}^n x_i P\{X^* = x_i\} = \sum_{i=1}^n \frac{X_i}{n} = \bar{X}_n,$$

donde $X^* \sim F_n$. Este tipo de estimación generalmente se llama *Estimación por sustitución*.

Mas generalmente, sea $T_n = T_n(X_1, \dots, X_n)$ un estadístico simétrico en X_1, \dots, X_n , es decir que su valor no dependa del orden en que se considera la muestra. Esto significa que T_n sólo depende de los estadísticos ordenados $X_{(1)} \leq \dots \leq X_{(n)}$, o equivalentemente, que T_n sólo dependa de la distribución empírica F_n asociada a X_1, \dots, X_n . Así, T_n puede expresarse de la forma $T_n = T(F_n)$ con T un funcional definido sobre \mathcal{F} .

Notemos, sin embargo, que no todos los estadísticos usuales son de la forma $T_n = T(F_n)$. Por ejemplo, el estimador usual insesgado de la varianza $\hat{v} = (n-1)^{-1} \sum_{i=1}^n (X_i - \bar{X})^2$ no es exactamente de la forma $\hat{v} = T(F_n)$ sino $\hat{v} = nT(F_n)/(n-1)$ donde T es el funcional varianza. Y de la misma forma el estimador usual de la mediana no necesariamente coincide con $T(F_n) = F_n^{-1}(\frac{1}{2})$. No obstante, en estos casos podemos ver que $\hat{v} = t_n(F_n)$ con t_n un funcional que tiende a T cuando n tiende a infinito.

2.4 Método delta

2.4.1 Caso paramétrico

En situaciones paramétricas, a menudo, es posible representar un estimador T_n en función de estadísticos fundamentales U_1, \dots, U_n tales como la media muestral \bar{X}_n o la varianza muestral S_n^2 . Es decir, que es posible escribir $T_n = g(U_1, \dots, U_n)$. Consideremos el caso unidimensional $T_n = g(U)$ y supongamos que U es aproximadamente normal

$$U \sim N(\theta, n^{-1}\sigma^2(\theta)). \quad (2.6)$$

Podemos, entonces expresar U mediante las expresiones formales:

$$U = \theta + o_p(1) \text{ y } U = \theta + n^{-1/2}\sigma(\theta)Z + O_p(n^{-1}), \quad (2.7)$$

donde Z es la variable normal estándar. La primera expresión de (2.7) representa el carácter consistente de U , y la segunda declara su aproximación a la distribución normal además del orden de convergencia. Supongamos que $g(\cdot)$ es una función

regular (suave), es decir suficientemente derivable, y que $g(\theta) \neq 0$. Entonces se tiene que

$$\text{var}(T_n) \simeq g'^2(\theta)\text{var}(U). \quad (2.8)$$

En efecto, puesto que g es regular, entonces $T_n = g(U)$ es consistente para $g(\theta)$ ya que

$$g(U) = g(\theta + o_p(1)) = g(\theta) + o_p(1).$$

Además, aplicando el desarrollo de Taylor de primer orden a g en θ obtendremos

$$g(U) = g(\theta) + (U - \theta)g'(\theta) + o_p(n^{-1/2}),$$

ya que el resto del desarrollo es proporcional a $(U - \theta)^2$. Además, a partir de (2.7) podemos ver que

$$g(U) = g(\theta) + n^{-1/2}\sigma(\theta)g'(\theta)Z + o_p(n^{-1/2}). \quad (2.9)$$

Entonces, tendremos la aproximación

$$\text{var}(g(U)) \simeq n^{-1}\sigma^2(\theta)g'^2(\theta)$$

lo que prueba la aproximación en (2.8).

Consideremos, ahora, el desarrollo de Taylor de segundo orden

$$g(U) = g(\theta) + (U - \theta)g'(\theta) + \frac{1}{2}g''(\theta)(U - \theta)^2 + o_p(n^{-1}).$$

Ignorando el resto del desarrollo, tendremos la aproximación

$$E(g(U)) \simeq g(\theta) + g'(\theta)E(U - \theta) + \frac{1}{2}g''(\theta)E(U - \theta)^2.$$

Si suponemos que U es insesgado de θ , entonces obtendremos una aproximación del sesgo de T_n :

$$\text{sesg}(T_n) \simeq \frac{1}{2}g''(\theta)\text{var}(U) = \frac{1}{2n}g''(\theta)\sigma^2(\theta). \quad (2.10)$$

2.4.2 Caso no paramétrico

Consideramos, en este apartado, la aproximación de la varianza, de estadísticos definidos mediante un funcional, $T_n = T(\hat{F})$, con T definido sobre \mathcal{F} . En estas situaciones podemos extender el método delta visto en la sección anterior, para aproximar $\text{var}(T_n)$. Supondremos que el funcional T es regular en algún sentido.

Definimos la derivada Gateaux de T en F como

$$L_T(x, F) = \lim_{\varepsilon \rightarrow 0} \frac{T((1 - \varepsilon)F + \varepsilon\delta_x) - T(F)}{\varepsilon}, \quad (2.11)$$

donde δ_x es la distribución degenerada en x . Notemos que, si $h(\varepsilon) = T(F + \varepsilon(\delta_x - F))$, entonces $L_T(x, F)$ es simplemente la derivada habitual, $h'(0)$, de h en 0.

La forma lineal del desarrollo de $T(\cdot)$ es

$$T(G) \simeq T(F) + \int L_T(x, F)dG(x). \quad (2.12)$$

Si $G = F$ en esta última expresión, se puede ver que la derivada $L_T(x, F)$ satisface la ecuación $\int L_T(x, F)dF(x) = 0$. La función $L_T(x) = L_T(x, F)$ se llama la función de influencia de T . Notemos por $l(x) = L_T(x, \hat{F})$ la análoga empírica de $L_T(x)$. $l(y)$, llamada función de influencia empírica y sus valores $l_j = l(x_j)$, $j = 1, \dots, n$, son los valores de influencia empíricos. Sustituyendo G por F_n en (2.12) tendremos la aproximación lineal del estadístico $T_n = T(F_n)$

$$T(F_n) \simeq T(F) + \int L_T(x, F)dF_n(x) = T(F) + \frac{1}{n} \sum_{j=1}^n L_T(X_j, F). \quad (2.13)$$

Sea X una variable aleatoria con distribución F . Notemos por $v_L(F) = n^{-1}\text{var}(L_T(X))$. Aplicando el teorema central del límite a la suma en (2.13), tendremos la aproximación:

$$T_n \rightsquigarrow N(\theta, v_L(F)).$$

Y puesto que $E_F(L_T(X)) = \int L_T(x, F)dF(x) = 0$, $v_L(F)$ tiene la expresión

$$v_L(F) = \frac{1}{n} \int L_T^2(x, F)dF(x).$$

Esta expresión nos va a permitir estimar la varianza de T_n sustituyendo F por F_n , obteniendo

$$v_L(F_n) = \frac{1}{n^2} \sum_{j=1}^n l_j^2, \quad (2.14)$$

a la que nos referiremos como la *aproximación delta no paramétrica* de la varianza de T_n .

A partir de (2.13) podemos notar que los valores de influencia empíricos satisfacen

$$\frac{1}{n} \sum_{j=1}^n l_j = 0. \quad (2.15)$$

La función de influencia de los estadísticos lineales, cuya forma general es $T(F) = \int a(x)dF(x)$, viene dada por

$$L_T(x) = a(x) - E_F(a(X)).$$

Un ejemplo habitual, es la media muestral cuya función de influencia es $L_T(x) = x - \mu$, donde $\mu = T(F) = E_F(X)$.

Más generalmente, si $T(F) = a(T_1(F), \dots, T_m(F))$, entonces la función de influencia viene dada por

$$L_T(x) = \sum_{i=1}^m \frac{\partial a}{\partial T_i} L_{T_i}(x). \quad (2.16)$$

Ejemplo 2.1 Sea $X = (U, V)$ un vector aleatorio bidimensional y $\mu_{rs} = E(U^r V^s)$ son sus momentos, entonces el coeficiente de correlación entre U y V viene dado por

$$\rho(F) = \frac{\mu_{11} - \mu_{10}\mu_{01}}{[(\mu_{20} - \mu_{10}^2)(\mu_{02} - \mu_{01}^2)]^{1/2}}. \quad (2.17)$$

Puesto que los funcionales $\mu_{rs}(F) = \mu_{rs}$ son lineales en el sentido mencionado anteriormente, podemos probar fácilmente que la función de influencia es

$$L_{\mu_{rs}}(u, v) = u^s v^r - \mu_{rs}, \text{ con } r + s \leq 2.$$

Aplicando la fórmula (2.16) al funcional $\rho(\cdot)$ en (2.17), y derivando parcialmente con respecto a los funcionales momentos, obtendremos

$$L_\rho(u, v) = u_s v_s - \frac{1}{2}(u_s^2 + v_s^2),$$

donde $u_s = (u - \mu_{10})/(\mu_{20} - \mu_{10}^2)^{1/2}$ y $v_s = (v - \mu_{01})/(\mu_{02} - \mu_{01}^2)^{1/2}$.

Cuando el estimador está definido por una ecuación $\sum c(x_j, T) = 0$ de modo que $\int c(x, \theta)dF(x) = 0$, entonces la función de influencia del estadístico unidimensional T viene dada por

$$L_T(x) = \frac{c(x, \theta)}{E[-c'(X, \theta)]}, \text{ donde } c'(\cdot) = \frac{\partial c(\cdot)}{\partial \theta}.$$

Por lo que, los valores de influencia empíricos serían

$$l_j = \frac{-nc(x_j, T)}{\sum_{i=1}^n c'(x_i, T)}. \quad (2.18)$$

Entonces la estimación delta de la varianza de T , sería

$$v_L = \sum_{j=1}^n \frac{c(x_j, T)^2}{[\sum_{i=1}^n c(x_i, T)]^2}. \quad (2.19)$$

Consideremos el estadístico $\hat{\theta} = \bar{X}/\bar{U}$ usado en el ejemplo B del capítulo 1. $T_n = \hat{\theta}$ es la solución de la ecuación $\sum c(y, T) = 0$ con $c(y, \theta) = x - \theta u$ donde $y = (x, u)$. Entonces según (2.18) los valores de influencia empíricas son

$$l_j = \frac{x_j - \hat{\theta}u_j}{\bar{u}},$$

y según (2.19), la aproximación delta de la varianza de $\hat{\theta} = \bar{X}/\bar{U}$ es

$$\frac{1}{n^2} \sum_{j=1}^n \left(\frac{x_j - \hat{\theta}u_j}{\bar{u}} \right)^2.$$

Nota 1 Cuando es difícil evaluar teóricamente la derivada en (2.11), podemos usar la aproximación numérica

$$L_T(x, F) \simeq \frac{T(F + \varepsilon_0(\delta_x - F)) - T(F)}{\varepsilon_0},$$

con ε_0 elegido bastante pequeño, por ejemplo, $\varepsilon_0 = (100n)^{-1}$. La misma aproximación puede aplicarse a $L_T(X_j, F_n)$ con el fin de aproximar los valores de influencia empíricas l_j necesarios para la aproximación de la varianza.

En algunas situaciones, es más conveniente considerar la aproximación cuadrática de $T_n = T(F_n)$,

$$T(F_n) = T(F) + \int L_T(x, F) dF_n(x) + \frac{1}{2} \int \int Q_T(x, y, F) dF_n(x) dF_n(y),$$

donde $Q_T(x, y; F)$ es la segunda derivada del funcional T en F , definida por

$$Q_T(x, y; F) = \left. \frac{\partial^2 T[(1 - \varepsilon_1 - \varepsilon_2)F + \varepsilon_1\delta_x + \varepsilon_2\delta_y]}{\partial \varepsilon_1 \partial \varepsilon_2} \right|_{\varepsilon_1 = \varepsilon_2 = 0}.$$

Esta función satisface $\int Q_T(x, y; F) dF(x) = \int Q_T(x, y; F) dF(y) = 0$, aunque en general $\int Q_T(x, x; F) dF(x) \neq 0$. Los valores $q_{ij} = Q_T(x_i, x_j; F)$ son los valores de influencia análogos a l_j y correspondientes a la segunda derivada $Q_T(\cdot, \cdot; F)$. La aproximación delta basada en desarrollo cuadrático es mas precisa.

2.4.3 Desventajas de los métodos tradicionales

- A veces la obtención de fórmulas explícitas teóricas es muy difícil o incluso imposible.
- Las fórmulas teóricas o sus aproximaciones están basadas en el modelo de interés, el cuál es desconocido. Si el modelo es erróneo, los estimadores obtenidos pueden ser totalmente inexactos.
- La aplicación de los procedimientos tradicionales a diferentes problemas de estimación supone el cálculo de diferentes fórmulas para las medidas de precisión en los respectivos problema. El cálculo puede resultar muy difícil y tedioso.
- Los métodos tradicionales suponen que el analizador de datos tiene un conocimiento amplio en matemáticas y en la teoría de la estadística, para poder llevar a cabo la búsqueda de fórmulas o aproximaciones explícitas.
- A menudo, se necesita un tamaño de muestra n bastante grande con el fin de obtener estimaciones aceptables para las medidas de precisión. Desafortunadamente, en la práctica n no siempre puede ser suficientemente grande, por varios motivos como por ejemplo el elevado coste que eso supone en algunas situaciones.
- La varianza límite de los estadísticos que son funcionales de procesos empíricos es un funcional de procesos Gaussianos. Entonces la expresión de aquella varianza como una función simple de los parámetros del modelo es muy complicada.
- A veces las fórmulas teóricas resultarán muy complicadas como para ser útiles en la estimación de las medidas de precisión.

Capítulo 3

DESARROLLOS ORTOGONALES Y TRANSFORMADAS DE FOURIER

Es este capítulo presentamos algunas herramientas matemáticas que serán de gran utilidad en capítulos posteriores. En primer lugar veremos como una variable aleatoria, concretamente un estimador, puede expresarse en términos de un sistema ortogonal de variables aleatorias. Estos desarrollos nos permitirán obtener expresiones formales para el sesgo y la varianza. En segundo lugar, recordamos ciertos resultados sobre las transformadas de Fourier descritas y los algoritmos rápidos que nos permitirán obtener dichas transformadas a partir de una sucesión de números complejos dada.

3.1 Desarrollos ortogonales

3.1.1 Existencia

Sea X una variable aleatoria. Veremos que existe una sucesión de variables aleatorias ortogonales $\{X_k\}_{k \geq 0}$ y una sucesión de números escalares $\{a_k\}_{k \geq 0}$ tal que X puede expresarse como

$$X = \sum_{k \geq 0} a_k X_k. \quad (3.1)$$

Este desarrollo es muy útil ya que nos permite obtener una expresión de la varianza de X en términos de los a_k y las varianzas de X_k .

Teorema 2 Sea X una variable aleatoria absolutamente continua con soporte $\Omega \subseteq \mathbb{R}^d$ y densidad f respecto a una medida σ -finita ν , tal que $E(X) = 0$ y $\text{var}(X) < \infty$. Entonces, existe un sistema de variables aleatorias $\{X_k\}$ incorreladas y una sucesión de escalares $\{a_k\}$ tal que

$$X = \sum_{k \geq 0} a_k X_k$$

con

$$\text{var}(X) = \sum_{k \geq 0} a_k^2 \text{var}(X_k).$$

Además, la serie $\sum_{k \geq 0} a_k X_k$ converge en media cuadrática y casi seguro.

Demostración. Sin pérdida de generalidad supongamos que ν es la medida de Lebesgue. Consideremos los espacios $L^2(\Omega)$ y $L^2(f)$ de funciones medibles u sobre Ω tal que

$$\|u\|^2 = \int_I |u|^2 dx < \infty,$$

y

$$\|u\|_f^2 = \int_I |u|^2 f dx < \infty,$$

respectivamente. Los espacios $L^2(\Omega)$ y $L^2(f)$ son espacios de Hilbert separables (KINGMAN & TAYLOR (1966)) con normas cuadráticas $\|u\|$ y $\|u\|_f$ respectivamente. El funcional T_f definido sobre $L^2(\Omega)$ por

$$T_f u = f^{-\frac{1}{2}} u \quad (u \in L^2(\Omega))$$

es una isometría lineal de $L^2(\Omega)$ a $L^2(f)$ cuya inversa es $T_f^{-1} = T_{f^{-1}}$ es decir

$$T_f^{-1} v = T_{f^{-1}} v = f^{\frac{1}{2}} v \quad (v \in L^2(f)),$$

y

$$\|T_f u\| = \|u\|.$$

Sea $\{u_k\}_k$ una base ortonormal de $L^2(\Omega)$, entonces $\{v_k\}$ definida por

$$v_k = T_f u_k = f^{-\frac{1}{2}} u_k, \quad k \in \mathbb{N},$$

es una base ortonormal de $L^2(f)$. Consideremos, ahora, la función

$$h(x) = x, \quad x \in \mathbb{R}^d.$$

Puesto que $E(X) = 0$ y $\text{var}(X) < \infty$ entonces $h \in L^2(f)$. De hecho,

$$\|h\|_f^2 = \int_I x^2 f(x) dx = \text{var}(X) < \infty.$$

Entonces h tiene el desarrollo siguiente en $L^2(f)$

$$h = \sum_{k \geq 0} a_k v_k, \quad (3.2)$$

con

$$\|h\|_f^2 = \sum_{k \geq 0} a_k^2.$$

La serie en (3.2) converge en $L^2(f)$. Reemplazando x por X , y definiendo $X_k = v_k(X)$ obtendremos

$$X = \sum_{k \geq 0} a_k X_k \quad (3.3)$$

ya que $h(x) = x$. Esta última serie converge casi seguro puesto que la serie en (3.2) converge en $L^2(f)$. Sin pérdida de la generalidad, podemos elegir $v_0 = f^{1/2}$, entonces $v_0 = 1$ y para cualquier k ,

$$E(X_k) = \int_{\Omega} v_k(x) f(x) dx = \int_{\Omega} v_0(x) v_k(x) f(x) dx = \langle v_0, v_k \rangle_f = 0.$$

Entonces

$$\text{cov}(X_k, X_j) = E(X_k X_j) = \int_{\Omega} v_k(x) v_j(x) f(x) dx = a_k^2 \delta_{kj},$$

donde δ_{kj} es la delta de Kronecker, i.e., $\delta_{kj} = 0$ si $j \neq k$ y $\delta_{jj} = 1$ para todo j . Entonces si $k \neq j$ tendremos

$$\text{cov}(X_k, X_j) = 0.$$

Además

$$\begin{aligned} \text{var}(X) &= \|h\|_f^2 = \left\| \sum_{k \geq 0} a_k v_k \right\|_f^2 = \sum_{k \geq 0} a_k^2 \|v_k\|_f^2 \\ &= \sum_{k \geq 0} a_k^2 \text{var}(X_k). \end{aligned}$$

3.1.2 Expresión de un estimador

Sea X_1, \dots, X_n una muestra aleatoria procedente de una familia paramétrica F_θ con $\theta \in \Theta \subseteq \mathbb{R}^d$ un vector de parámetros desconocido. Sea $S_n = S(X_1, \dots, X_n)$ un estadístico con distribución $F_{n,\theta}(s)$ y densidad $f_{n,\theta}(s)$ respecto a una cierta medida σ -finita ν . Supongamos que queremos estimar una función $\xi = h(\theta)$ del parámetro θ mediante un estimador $\hat{\xi} = T_n(S_n)$ que es función del estadístico S_n . Notemos que si $\hat{\xi}$ es insesgado y tiene varianza finita, y si además S_n es suficiente y completo para la familia $\{F_\theta\}_\theta$ entonces $\hat{\xi}$ será simplemente el *UMVUE* de $h(\theta)$. Supongamos que $\hat{\xi}$ tiene varianza finita. Para cada $\theta \in \Theta$, consideremos los espacios

$$L_{n,\theta}^2 = \left\{ T : \mathbb{R}^d \longrightarrow \mathbb{R} \text{ medible} \mid \int |T(s)|^2 f_{n,\theta}(s) d\nu(s) < \infty \right\} \quad (3.4)$$

y

$$L_n = \bigcap_{\theta \in \Theta} L_{n,\theta}^2.$$

Como es usual en $L_{n,\theta}^2$, dos funciones T_1 y T_2 serán consideradas iguales si son iguales en casi por todo respecto a la medida ν . Evidentemente, para cada $\theta \in \Theta$, el espacio $L_{n,\theta}^2$ es un espacio de *Hilbert* completo considerando el producto escalar

$$\langle T_1, T_2 \rangle_\theta = \int T_1(s) T_2(s) f_{n,\theta}(s) d\nu(s).$$

Además, considerando la norma inducida por el producto escalar, $\|T\|_\theta = \langle T, T \rangle_\theta^{1/2}$, el espacio $L_{n,\theta}^2$ es un espacio de *Banach*.

En $L_{n,\theta}^2$ es posible extraer un sistema ortogonal de funciones $\{\psi_{k,n}(s, \theta)\}_{k \geq 0}$ tal que cualquier función T en $L_{n,\theta}^2$ admite un desarrollo convergente (en la topología natural de $L_{n,\theta}^2$) de la forma

$$T(s) = \sum_k c_{k,n}(\theta) \psi_{k,n}(s, \theta), \quad (3.5)$$

con $c_{k,n}(\theta) = \langle T, \psi_{k,n} \rangle_\theta$ son los coeficientes de *Fourier* asociados a T . Sin pérdida de generalidad podemos elegir $\psi_{0,n} \equiv 1$. Por comodidad, notaremos $\psi_{k,n} = \psi_k$ y $c_{k,n} = c_k$ sin olvidar que aquellas cantidades dependen del tamaño muestral n .

La expresión del estimador como en (3.5) es muy útil para varios propósitos. Por ejemplo nos permite caracterizar el conjunto de las funciones *UMVU* estimables en familias para las cuales existe un estadístico suficiente y completo. En esta memoria, utilizaremos estos desarrollos para estimar el sesgo y la varianza del estimador $T(S_n)$ usando el método *bootstrap* como veremos en el *Capítulo 6*. Para muchas familias de distribuciones de interés práctico, el sistema de funciones ortogonales correspondiente, es perfectamente conocido.

3.1.3 Sistema ortogonal en familias NEF

Familias NEF

Sea ν una medida de Radon definida sobre los Borel de \mathbb{R}^d y que no esté concentrada en un hiperplano afín. Consideremos la transformada de Laplace de ν , que es la función $L_\nu : \theta \in \mathbb{R}^d \rightarrow \mathbb{R}$ definida por

$$L_\nu(\theta) = \int \exp(\langle \theta, x \rangle) d\nu(x). \quad (3.6)$$

Sea D_ν el conjunto de los puntos θ tal que $L_\nu(\theta)$ es finita, y Θ es el interior de D_ν , que supondremos no vacío. Denotaremos por $k_\nu(\theta) = \log(L_\nu(\theta))$, que está bien definida, puesto que $\exp(\langle \theta, x \rangle) > 0$ y ν no está concentrada en un hiperplano afín. La *Familia Exponencial natural (NEF)* se define como la familia de distribuciones, F_θ , con densidad respecto a ν de la forma, (BROWN (1986)),

$$dP(\theta, \nu) = \exp(\langle \theta, x \rangle - k_\nu(\theta)) d\nu, \quad \theta \in \Theta. \quad (3.7)$$

Denotaremos por $f_\theta(x)$ la derivada de Radon-Nikodim $dP/d\nu = \exp(\langle \theta, x \rangle - k_\nu(\theta))$. Por las propiedades de la transformada de Laplace la función $k_\nu(\theta)$ es analítica en todo Θ . Para la familia *NEF* se verifica que

$$\mu = E_\theta[\mathbf{X}] = k'_\nu(\theta) \text{ y } V(\theta) = k''_\nu(\theta),$$

donde $V(\theta)$ es la matriz de varianza covarianza del vector $\mathbf{X} = (X_1, \dots, X_d)$. Nótese que $V(\theta) = k''_\nu(\theta)$ es siempre una matriz semidefinida positiva. Es más, dado que $\exp(\langle \theta, x \rangle - k_\nu(\theta)) > 0, \forall \theta \in \Theta$, cada probabilidad P_θ no está concentrada en un hiperplano afín, por lo que el vector \mathbf{X} tendrá matriz de varianza-covarianza definida positiva.

En las condiciones anteriores la aplicación $k'_\nu : \mathbb{R}^d \rightarrow \mathbb{R}^d$ es una aplicación inyectiva. La prueba es inmediata ya que $k''_\nu(\cdot)$ es definida positiva. Entonces, si denotamos por $\mathbf{M}_F = k'_\nu(\Theta) \subset \mathbb{R}^d$, denominado el espacio de medias de la familia F_θ . La aplicación $k'_\nu : \Theta \rightarrow \mathbf{M}_F$ es un difeomorfismo entre Θ y \mathbf{M}_F lo que permite reparametrizar la familia por sus medias como sigue:

$$F_\mu = \{f_\nu(x, \mu) d\nu = \exp[\langle \psi_\nu(\mu), x \rangle - k_\nu(\psi_\nu(\mu))] d\nu, \mu \in \mathbf{M}_F\}. \quad (3.8)$$

con $\psi_\nu(\mu) = (k'_\nu)^{-1}(\mu)$, para cualquier $\mu \in \mathbf{M}_F$.

Para un vector aleatorio \mathbf{X} cuya distribución pertenece a esta última familia y cuya matriz de varianzas y covarianzas es $V(\mu)$, notaremos $\mathbf{X} \rightsquigarrow NEF(V(\mu), \mathbf{M}_F)$ o simplemente $\mathbf{X} \rightsquigarrow (V(\mu), \mathbf{M}_F)$. El par $(V(\mu), \mathbf{M}_F)$ se llama función de varianza la cual caracteriza una familia *NEF* entre todas las familias *NEF*.

Construcción de un sistema ortogonal

Dada una familia NEF generada por una medida ν , y con función de densidad respecto a ν

$$f(x, \mu) = f_\nu(x, \mu) = \exp[\langle \psi_\nu(\mu), x \rangle - k_\nu(\psi_\nu(\mu))], \quad (3.9)$$

definimos las funciones

$$P_n(x, \mu) = \left\{ \frac{\partial^{|n|}}{\partial \mu^n} f(x, \mu) \right\} / f(x, \mu), \quad n \in \mathbb{N}^d. \quad (3.10)$$

y $P_0(x, \mu) = 1$, entendiendo que

$$|n| = n_1 + \dots + n_d \text{ y } x^n = x_1^{n_1} \dots x_d^{n_d} \text{ para todo } n \in \mathbb{N}^d \text{ y } x \in \mathbb{R}^d.$$

Dado que $f(x, \mu)$ es analítica, tenemos para todo μ en un entorno de μ_0 , se verifica

$$f(x, \mu) = \sum_{n \in \mathbb{N}^d} \frac{(\mu - \mu_0)^n}{n!} P_n(x, \mu_0) f(x, \mu_0).$$

con $n! = n_1! \dots n_d!$. Las funciones $P_n = P_n(x, \mu)$ verifican la siguiente relación de recurrencia

$$\begin{cases} P_{e_i} = (V^{-1})_i(x - \mu) \text{ y} \\ P_{n+e_i} = \frac{\partial}{\partial \mu_i} P_n + P_n P_{e_i} \end{cases} \quad (3.11)$$

donde $(V^{-1})_i$ es la i -ésima fila de la matriz $V^{-1}(\mu)$. Nótese que

$$\psi'_\nu(\mu) = [(k'_\nu)^{-1}(\mu)]' = [k''_\nu(\psi_\nu(\mu))]^{-1} = V^{-1}(\mu).$$

Se dice que una familia NEF , F_μ , es una familia con función de varianza cuadrática (NEF - QVF) si cada componente de la matriz $V(\mu) = \{V_{ij}(\mu)\}_{i,j}$, es un polinomio en μ de grado 2 a lo sumo. Y se dice que una tal familia NEF tiene varianza cuadrática simple (NEF - $SQVF$) si existe $a \in \mathbb{R}$ tal que $V(\mu)$ es de la forma

$$V_{ij}(\mu) = a\mu_i\mu_j + \sum_{k=1}^d b_{i,jk}\mu_k + c_{i,j},$$

es decir, el único elemento de orden 2 en V_{ij} es $a\mu_i\mu_j$ con $a \in \mathbb{R}$. En el caso unidimensional esto es equivalente a decir que

$$V(\mu) = v_2\mu^2 + v_1\mu + v_0. \quad (3.12)$$

MORRIS (1982) estudio diversas propiedades en este tipo de familias de distribuciones univariantes, y las clasificó en seis grupos, salvo transformaciones lineales y potencias de convolución,

- **Normal.** La familia de distribuciones normal $N(\mu, \sigma^2)$ con $\sigma > 0$ conocido, donde $\mathbf{M}_F = \mathbb{R}$ y $V(\mu) = \sigma^2$.
- **Gamma.** La familia de distribuciones Gamma $Ga(\alpha, \frac{\alpha}{\mu})$ con $\alpha > 0$ conocido, donde $\mathbf{M}_F = (0, +\infty)$ y $V(\mu) = \mu^2/\alpha$.
- **Poisson.** La familia de distribuciones de Poisson $Po(\mu)$, donde $\mathbf{M}_F = (0, +\infty)$ y $V(\mu) = \mu$.
- **Binomial.** La familia de distribuciones binomiales $Bi(N, \mu/N)$, con $N \in \mathbb{N}$ conocido, donde $\mathbf{M}_F = (0, N)$ y $V(\mu) = -\mu^2/N + \mu$.
- **Binomial Negativa.** La familia de distribuciones binomiales negativas $BN(N, \mu/(\mu + N))$, con $N > 0$ conocido, donde $\mathbf{M}_F = (0, +\infty)$ y $V(\mu) = \mu^2/N + \mu$.
- **Secante Hiperbólica Generalizada.** La familia de distribuciones $GHS(r, \arctan(\mu/r))$, con $r \in \mathbb{N}$ conocido, y donde $\mathbf{M}_F = \mathbb{R}$ y $V(\mu) = \mu^2/r + r$.

CASALIS (1994) Clasificó las familias exponenciales de varianza cuadrática simple en \mathbb{R}^d . El resultado fué las siguientes $2d + 4$ familias

- **$d+1$ familias de Poisson.** Corresponde a la familia de distribuciones (X_1, \dots, X_d) con X_1, \dots, X_d independientes, las k primeras componentes siguen una ley de Poisson y las $d - k$ últimas siguen una ley normal de varianza λ . La función de varianza viene determinada por

$$\begin{aligned} V_F(\mu) &= \text{diag}(\mu_1, \dots, \mu_k, \lambda, \dots, \lambda), \\ \mathbf{M}_F &= (0, +\infty)^k \times \mathbb{R}^{d-k}, \quad k = 0, \dots, d \end{aligned}$$

Obsérvese que la familia Normal Multivariante de media μ y matriz de varianza I_d constante esta incluida en esta familia tomando $k = 0$.

- **$d+1$ tipo multinomial negativa-gamma.** Corresponde a la familia de distribuciones (X_1, \dots, X_d) donde las k primeras componentes siguen una Multinomial Negativa de parámetro λ , X_{k+1} condicionada a X_1, \dots, X_k es una Gamma de parámetro de la forma $\sum_{i=1}^k X_i + \lambda$ y X_{k+2}, \dots, X_d condicionados a X_1, \dots, X_{k+1} sigue una ley Normal de varianza X_{k+1} . La función de varianza viene dada por

$$\begin{aligned} V_F(\mu) &= \frac{1}{\lambda} \mu \mu^t + \text{diag}(\mu_1, \dots, \mu_k, 0, \mu_{k+1}, \dots, \mu_{k+1}), \\ \mathbf{M}_F &= (0, +\infty)^k \times \mathbb{R}^{d-k-1}, \quad k = 0, \dots, d. \end{aligned}$$

- **Tipo Multinomial.** Corresponde a la familia Multinomial de N experimentos, $N \in \mathbb{N} \setminus \{0\}$, con $d + 1$ posibles resultados, cada uno con probabilidad $p_i = \mu_i/N$, $i = 1, \dots, N$, y $p_{N+1} = (N - \sum_{i=1}^N \mu_i)/N$. La función de varianza viene dada por

$$V_F(\mu) = -\frac{1}{N}\mu\mu^t + \text{diag}(\mu_1, \dots, \mu_d),$$

$$\mathbf{M}_F = \left\{ \mu \in \mathbb{R}^d : \mu_i > 0 \text{ y } \sum_{i=1}^d \mu_i < N \right\}.$$

- **Tipo Hiperbólico.** Esta familia está compuesta de los vectores (X_1, \dots, X_d) donde (X_1, \dots, X_{d-1}) sigue una Multinomial-Negativa de parámetro λ y X_d condicionada a X_1, \dots, X_{d-1} sigue una ley de coseno hiperbólico de parámetro $\sum_{i=1}^{d-1} X_i + \lambda$. La función de varianza viene dada por

$$V_F(\mu) = \frac{1}{\lambda}\mu\mu^t + \text{diag} \left(\mu_1, \dots, \mu_{d-1}, \sum_{i=1}^{d-1} \mu_i + \lambda \right),$$

$$\mathbf{M}_F = (0, +\infty)^{d-1} \times \mathbb{R}.$$

Cuando F es una familia NEF-QVF, entonces cada función P_n definida en (3.10) es un polinomio multivariante en x de grado n . Además, si $\mu \in \mathbf{M}_F$ es tal que $V(\mu)$ es una matriz diagonal entonces cada polinomio es de la forma

$$P_n(x, \mu) = \alpha_n x^n + \sum_{|k| < |n|} \beta_k x^k,$$

con

$$\alpha_n = (\text{diag}(V(\mu)))^{-n} = V_{11}(\mu)^{-n_1} \dots V_{dd}(\mu)^{-n_d}.$$

Definición 3 Sea F una familia NEF y $m \in \mathbf{M}_F$. Diremos que una familia $\{P_n\}$ es μ -ortogonal si $\int P_k P_n f(x, \mu) d\nu(x) = 0$ para $k \neq n$, en otras palabras el sistema es ortogonal en el espacio de Hilbert inducido por la medida $f(x, \mu) d\nu(x)$. Por otra parte, diremos que $\{P_n\}$ es μ -pseudoortogonal si $\int P_k P_n f(x, \mu) d\nu(x) = 0$ para $|k| \neq |n|$.

Teorema 4 Sea $(V(\mu), \mathbf{M}_F)$ una familia NEF con varianza cuadrática y $\mu_0 \in \mathbf{M}_F$. Entonces los polinomios $\{P_n(x, \mu_0)\}_{n \in \mathbb{N}^d}$ construidos como en (3.10) forman una familia μ_0 -pseudoortogonal. Además, si la familia $(V(\mu), \mathbf{M}_F)$ tiene varianza cuadrática simple y μ_0 es tal que $V(\mu_0)$ es diagonal entonces dicho sistema de polinomios es μ_0 -ortogonal.

Teorema 5 Sea F una familia NEF-QVF simple, con $V_{ij}(\mu) = a\mu_i\mu_j + R_{ij}(\mu)$ y μ_0 tal que $V(\mu_0)$ sea diagonal, entonces

$$\|P_n(\cdot, \mu)\|_{\mu_0}^2 = n!a_n \text{Diag}(V(\mu_0))^{-n}, \quad (3.13)$$

con

$$a_n = \prod_{j=0}^{|n|-1} (1 + ja).$$

La demostración de estos resultados puede encontrarse en POMMERET (1996, 1997).

Notemos que para obtener un sistema ortogonal es necesario buscar un punto μ_0 que diagonalice la matriz de varianza-covarianza $V(\mu)$. Dicho punto, incluso puede ser que no exista. Para solventar este hecho, es posible hacer un cambio de base, o bien generalizar la definición de los polinomios P_n en (3.10) como vamos a ver más adelante.

Si X es un vector aleatorio con distribución NEF-QVF($V(\mu)$, \mathbf{M}_F) y $\mu_0 \in \mathbf{M}_F$ cualquiera, existe A una matriz tal que $AV(\mu_0)A^t$ sea diagonal, debido a que $V(\mu_0)$ es definida positiva. Así, en la familia AF , el punto $A\mu_0 \in \mathbf{M}_{AF}$ hace la matriz de varianza covarianza diagonal y su correspondiente familia de polinomios es $A\mu_0$ -ortogonal.

Otro método consiste en generalizar la definición de los polinomios P_n en (3.10) derivando respecto a un cierto vector $A\mu$ donde A es una matriz no singular. Es decir, consideramos

$$P_{A,n}(x, \mu) = \frac{\partial^{|n|}}{\partial(A\mu)^n} f_\nu(x, \mu) / f_\nu(x, \mu). \quad (3.14)$$

donde derivamos n_1 veces respecto a $A_1\mu$, n_2 veces respecto a $A_2\mu$, etc, con A_i representa la i -ésima fila de la matriz A .

Obsérvese que, si $g(\mu) = f_\nu(x, \mu)$, entonces

$$\frac{\partial^{|n|}}{\partial(A\mu)^n} f_\nu(x, \mu) = \frac{\partial^{|n|}}{\partial(A\mu)^n} g(\mu) = g^{(n)}(\mu)(A^{-1}e_1, \dots, A^{-1}e_d),$$

donde el último término de la ecuación representa derivar n_1 veces en la dirección $A^{-1}e_1$, n_2 veces en la dirección $A^{-1}e_2$, etc.

De las propiedades de la transformada lineal de una familia NEF (Ver LETAC & MORA (1990)) deducimos que

$$f_\nu(x, \mu) = f_{A\nu}(Ax, A\mu).$$

Así, notando por $\tilde{P}_n(x, \mu)$ los polinomios definidos normalmente, i.e como en (3.10), para la familia AF , obtendremos

$$P_{A,n}(x, \mu) = \frac{\partial^{|n|}}{\partial(A\mu)^n} f_{A\nu}(Ax, A\mu) / f_{A\nu}(Ax, A\mu) = \tilde{P}_n(Ax, A\mu). \quad (3.15)$$

Nuestro interés es el sistema $\{P_{A,n}(x, \mu_0)\}$ cuando $AV(\mu_0)A^t$ es una matriz diagonal. En resumen, se tiene el siguiente resultado (ver POMMERET (1996)):

Teorema 6 *Sea F una familia NEF-QVF y $\mu_0 \in \mathbf{M}_F$ cualquiera. Sea A un matriz $d \times d$ no singular tal que $AV(\mu_0)A^t$ sea diagonal, entonces, el sistema $\{P_{A,n}(x, \mu_0)\}_n$ es μ_0 -ortogonal en la familia F , y además*

$$\|P_{A,n}(\cdot, \mu_0)\|^2 = n! a_n \text{Diag}(AV(\mu_0)A^t)^{-n}.$$

Formas explícitas

Para las familias NEF el estadístico media muestral \bar{X} es suficiente y completo. En dichas familias, donde la media μ es desconocida, si consideramos la estimación de un parámetro $\xi = h(\mu)$ usando un estimador función de \bar{X} , digamos $\hat{\xi} = T(\bar{X})$ entonces según (3.5) sabemos que dicho estimador tiene una expresión de la forma

$$T(\bar{X}) = \sum_{k \geq 0} c_{k,n}(\mu) P_{k,n}(\bar{X}, \mu).$$

En lo que sigue daremos la expresión explícita de los polinomios $P_{k,n}$ en el caso unidimensional, $d = 1$, en cada una de las familias que componen la familia NEF - QVF .

Distribución Normal

Sea X_1, \dots, X_n una muestra aleatoria que sigue una distribución normal $N(\mu, \sigma^2)$, con $\mu \in \mathbb{R}$ y $\sigma > 0$ conocida,

$$f(x, \mu) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2} \left(\frac{x - \mu}{\sigma}\right)^2\right), \quad x \in \mathbb{R}.$$

En este caso $V(\mu) = \sigma^2$ y el estadístico suficiente y completo \bar{X}_n tiene la función de densidad dada por

$$f_n(s, \mu) = \frac{\sqrt{n}}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2} \left(\frac{s - \mu}{\sigma} \sqrt{n}\right)^2\right]. \quad (3.16)$$

El sistema de polinomios mónicos ortogonales respecto a (3.16) es

$$P_{k,n}(s, \mu) = \frac{\sigma^k}{(2n)^{k/2}} H_k \left(\sqrt{\frac{n}{2}} \frac{s - \mu}{\sigma} \right), \quad k \geq 0, \quad (3.17)$$

donde H_k son los polinomios de Hermite cuya expresión explícita viene dada por

$$H_k(x) = k! \sum_{j=0}^{\lfloor k/2 \rfloor} \frac{(-1)^j (2x)^{k-2j}}{(k-2j)! j!} \quad (3.18)$$

notando por $[s]$ la parte entera de s . La norma de los polinomios (3.17) en el espacio $L_{n,\mu}^2$ viene dada por

$$\|P_{k,n}\|_{\mu}^2 = \frac{k! \sigma^{2k}}{n^k}. \quad (3.19)$$

Distribución Gamma

Sea X_1, \dots, X_n una m.a. con distribución $G(\alpha, 1/\theta)$ donde $\theta > 0$ y $\alpha > 0$ conocido, y con función de densidad

$$f(x, \theta) = \frac{1}{\theta^\alpha \Gamma(\alpha)} x^{\alpha-1} \exp\left(-\frac{x}{\theta}\right), \quad x > 0.$$

En este caso $\mu = \alpha\theta$ y $V(\mu) = \mu^2/\alpha$, con $\mu \in (0, +\infty)$. La función de densidad del estadístico suficiente y completo \bar{X}_n viene dada por

$$f_n(s, \theta) = \frac{n^{n\alpha}}{\theta^{n\alpha} \Gamma(n\alpha)} s^{n\alpha-1} \exp\left(-\frac{ns}{\theta}\right), \quad s > 0. \quad (3.20)$$

El sistema de polinomios ortogonales respecto a (3.20) viene dado por

$$P_{k,n}(s, \mu) = \frac{(-1)^k k! \mu^k}{(n\alpha)^k} L_k^{(n\alpha-1)} \left(\frac{n\alpha s}{\mu} \right), \quad k \geq 0, \quad (3.21)$$

con $L_k^{(\gamma)}$ los polinomios de Lagerre generalizados, definidos anteriormente, y cuya expresión explícita viene dada por

$$L_k^{(\gamma)}(x) = \sum_{j=0}^k \binom{k+\gamma}{k-j} \frac{(-1)^j}{j!} x^j.$$

La norma de los polinomios (3.21) en $L_{n,\theta}^2$ viene dada por

$$\|P_{k,n}\|_{\mu}^2 = \frac{k! \Gamma(n\alpha + k) \mu^{2k}}{(n\alpha)^{2k} \Gamma(n\alpha)}. \quad (3.22)$$

Distribución de Poisson

Sea X_1, \dots, X_n una m.a.s. que sigue una distribución de Poisson $Po(\mu)$, $\mu > 0$, cuya función de probabilidad es

$$f(x, \mu) = e^{-\mu} \frac{\mu^x}{x!}, \quad x \in \mathbb{N}.$$

La función de probabilidad del estadístico suficiente y completo \bar{X}_n viene dada por

$$f_n(s, \mu) = e^{-n\mu} \frac{(n\mu)^{ns}}{(ns)!}, \quad s = 0, \frac{1}{n}, \frac{2}{n}, \dots \quad (3.23)$$

El sistema de polinomios ortogonales respecto a (3.23) viene dado por

$$P_{k,n}(s, \mu) = \frac{1}{n^k} C_k^{(n\mu)}(ns), \quad k \geq 0, \quad (3.24)$$

con $C_k^{(\alpha)}$ los polinomios de Charlier, dados por

$$C_k^{(\alpha)}(x) = \sum_{j=0}^k \binom{k}{j} \binom{s}{j} j! (-\alpha)^{k-j}.$$

La norma en $L^2_{n,\mu}$ de los polinomios en (3.24) es

$$\|P_{k,n}\|_{\mu}^2 = \frac{k! \mu^k}{n^k}. \quad (3.25)$$

Distribución Binomial

Sea X_1, \dots, X_n una m.a.s. que sigue una distribución $Bi(N, p)$, $p \in (0, 1)$ y $N \geq 1$ un entero conocido, y cuya función de probabilidad viene dada por

$$f(x, p) = \binom{N}{x} p^x q^{N-x}, \quad x = 0, 1, \dots, N \text{ y } q = 1 - p.$$

En este caso $V(\mu) = \mu - \mu^2/N$ y $\mu = Np$. La función de probabilidad del estadístico suficiente y completo \bar{X}_n viene dada por

$$f_n(s, p) = \binom{nN}{ns} p^{ns} q^{n(N-s)}, \quad s = 0, \frac{1}{n}, \frac{2}{n}, \dots \quad (3.26)$$

y el sistema de polinomios ortogonales respecto a (3.26) es

$$P_{k,n}(s, \mu) = \frac{k!}{n^k} K_{k, Nn, \mu/N}(ns), \quad k = 0, 1, \dots, nN, \quad (3.27)$$

con $K_{k,N,p}$ el k -ésimo polinomio de Krawtchouk,

$$K_{k,N,p}(x) = \sum_{j=0}^k \binom{x}{k-j} \binom{N-x}{j} (-1)^j p^j q^{k-j}.$$

La norma en $L_{n,\mu}^2$ de los polinomios (3.27) viene dada por

$$\|P_{k,n}\|_{\mu}^2 = \frac{k! \Gamma(nN+1) (pq)^k}{\Gamma(nN-k+1) n^{2k}}. \quad (3.28)$$

Distribución Binomial Negativa

Sea X_1, \dots, X_n una m.a.s. con distribución $BN(r, p)$, $p \in (0, 1)$, r un entero conocido, y cuya función de probabilidad es

$$f(x, p) = \binom{x+r-1}{x} p^r q^x, \quad x = 0, 1, \dots$$

con $q = 1-p$. En este caso, $\mu = rq/p$ y $V(\mu) = \mu + \mu^2/r$, y la función de probabilidad del estadístico suficiente y completo \bar{X}_n es

$$f_n(s, p) = \binom{nr+ns-1}{ns} p^{nr} q^{ns}, \quad s = 0, \frac{1}{n}, \frac{2}{n}, \dots \quad (3.29)$$

El sistema de polinomios ortogonales respecto a (3.29) viene dado por

$$P_{k,n}(s, p) = \frac{1}{n^k} \left(\frac{q}{p}\right)^k M_{k,nr,q}(ns), \quad k \geq 0, \quad (3.30)$$

con $M_{k,\alpha,c}$ es el k -ésimo polinomio de Meixner de primera especie,

$$M_{k,\alpha,c}(x) = (-1)^k k! \sum_{j=0}^k \binom{x}{j} \binom{-x-\alpha}{k-j} c^{-j}.$$

La norma en $L_{n,\mu}^2$ de los polinomios (3.30) es

$$\|P_{k,n}\|_{\mu}^2 = \frac{k! \Gamma(nr+k)}{\Gamma(nr)} \frac{q^k}{(np)^{2k}}. \quad (3.31)$$

Distribución Secante Hiperbólica Generalizada

Sea X_1, \dots, X_n una m.a.s. con distribución $GHS(r, \theta)$, $|\theta| < \frac{\pi}{2}$ y $r > 0$ un entero conocido. La función de densidad de dicha distribución viene dada por

$$f(x, r, \lambda) = (1 + \lambda^2)^{-r/2} \exp[x \tan^{-1}(\lambda)] f(x, r, 0), \quad x > 0,$$

donde

$$f(x, r, 0) = \frac{2^{r-2}}{\Gamma(r)} \prod_{j=0}^{\infty} \left(1 + \left(\frac{x}{r+2j} \right)^2 \right)^{-1},$$

y $\lambda = \tan(\theta)$. En este caso $\mu = r\lambda$ y $V(\mu) = \frac{\mu^2}{r} + r$ con $\mu \in \mathbb{R}$. La función de densidad del estadístico suficiente y completo \bar{X}_n , viene dada por

$$f_n(s, \lambda) = n(1 + \lambda^2)^{-nr/2} \exp[ns \tan^{-1}(\lambda)] f(ns, nr, 0), \quad (3.32)$$

y el sistema de polinomios ortogonales respecto a (3.32) viene dado por la relación de recurrencia

$$\begin{aligned} P_{0,n} &= 1, & P_{1,n} &= s - r\lambda, \\ P_{k+1,n} &= \left(P_{1,n} - \frac{2k\lambda}{n} \right) P_{k,n} - \frac{kr(1 + \lambda^2)}{n} \left(1 + \frac{k-1}{nr} \right) P_{k-1,n}, & k &\geq 1 \end{aligned} \quad (3.33)$$

La norma de estos polinomios en $L^2_{n,\mu}$ viene dada por

$$\|P_{k,n}\|_{\mu}^2 = \frac{k! \Gamma(nr + k)}{n^{2k} \Gamma(nr)} (1 + \lambda^2)^k. \quad (3.34)$$

Para el caso $d = 2$, podemos encontrar la expresión explícita de los polinomios $P_{A,n}(x, \mu)$ en CASALIS (1994) o POMMERET (1996) para cada tipo de familias NEF-SQVF multivariantes.

3.1.4 Polinomios ortogonales en otras familias

Sea X_1, \dots, X_n una muestra aleatoria simple con distribución F no necesariamente paramétrica, y sea $S_n = S(X_1, \dots, X_n)$ un estadístico con densidad $f_n(s)$ respecto a una medida σ -finita ν . Supongamos que S_n tiene momentos finitos de todos los ordenes $m_{k,n}$, $k \geq 0$, y que la distribución de S_n es únicamente determinada por los $\{m_{k,n}\}$. En este caso el conjunto de funciones

$$\{g_k(s) = s^k, k \geq 0\}, \quad (3.35)$$

es completo en el espacio

$$L_n^2 = \left\{ T : \mathbb{R} \rightarrow \mathbb{R} \mid \int_{\mathbb{R}} T^2(s) f_n(s) d\nu(s) < \infty \right\}.$$

Entonces el método de Gram Schmidt nos permite construir un sistema ortogonal a partir de (3.35). Otra alternativa, es el método basado sobre el determinante de la matriz de los momentos, como explicaremos a continuación (Para más detalles ver CHIHARA (1978)). Consideremos la matriz de momentos

$$\mathcal{M}_{k,n} = [m_{i+j,n}]_{i,j=0,\dots,k}, \quad k \geq 0,$$

y sea $d_n = \dim(L_n^2)$. Entonces una condición necesaria y suficiente para la existencia de un sistema ortogonal de polinomios es que

$$\Delta_{k,n} = \det(\mathcal{M}_{k,n}) \neq 0, \quad k = 0, \dots, d_n - 1.$$

En este caso el k -ésimo polinomio mónico ortogonal es

$$P_{k,n}(s) = \Delta_{k-1,n}^{-1} Q_{k,n}(s), \quad k = 0, \dots, d_n - 1, \quad (3.36)$$

donde

$$Q_{k,n}(s) = \det \begin{bmatrix} m_{0,n} & m_{1,n} & \dots & m_{k,n} \\ m_{1,n} & m_{2,n} & \dots & m_{k+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{k-1,n} & m_{k,n} & \dots & m_{2k-1,n} \\ 1 & s & \dots & s^k \end{bmatrix}.$$

La norma en L_n^2 de los polinomios $P_{k,n}$ en (3.36) es, ver CHIHARA (1978),

$$\|P_{k,n}\|^2 = \Delta_{k-1,n}^{-1} \Delta_k. \quad (3.37)$$

3.2 Transformadas de Fourier

En el *Capítulo 8* veremos un método basado sobre las fórmulas de inversión, cuales son directamente relacionados a las transformadas de Fourier, para aproximar la densidad y la distribución de la media muestral. Para ello, en esta sección vamos a presentar un resumen sobre las transformadas de Fourier.

3.2.1 Operador discreto de Fourier

Sea n un entero positivo y consideremos los espacios

$$\Pi_n = \{x = \{x_k\}_{-\infty}^{\infty}, x_k \in \mathbb{C} \mid x_{k+n} = x_k, \forall k\},$$

de sucesiones bilaterales periódicas con periodo n . Entonces una sucesión x de Π_n está totalmente determinada por los elementos de un periodo, por ejemplo, x_0, \dots, x_{n-1} . Definimos la suma de dos sucesiones $x = \{x_k\}$ e $y = \{y_k\}$ de Π_n como

$$(x + y)_k = x_k + y_k,$$

y la multiplicación de una sucesión por un escalar c como

$$(cx)_k = cx_k.$$

Entonces, es evidente, que Π_n es un espacio vectorial. Cualquier elemento de Π_n puede expresarse en términos de las n sucesiones de Π_n , $e^{(m)} = \{e_k^{(m)}\}$, $m = 0, 1, \dots, n-1$, donde

$$e_k^{(m)} = \begin{cases} 1, & \text{si } k \equiv m \pmod{n} \\ 0, & \text{si } k \not\equiv m \pmod{n}. \end{cases}$$

Y puesto que los $e^{(m)}$ son linealmente independientes, el espacio Π_n es de dimensión n . Además, si definimos la norma

$$\|x\| = (|x_0|^2 + |x_1|^2 + \dots + |x_{n-1}|^2)^{1/2}$$

dicho espacio será un espacio vectorial normado completo, i.e. un espacio de Banach, ya que es isomorfo a \mathbb{C}^n .

Definimos el operador lineal \mathcal{F}_n de Π_n a Π_n por

$$\mathcal{F}_n x = y \in \Pi_n,$$

donde

$$(\mathcal{F}_n x)_k = y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j w_n^{-jk}, \quad (3.38)$$

con

$$w_n = w = \exp\left(\frac{2\pi i}{n}\right).$$

El operador \mathcal{F}_n se llama *Operador discreto de Fourier*. \mathcal{F}_n es una isometría de Π_n a Π_n , es decir de un espacio de dimensión finita a sí mismo, por lo que \mathcal{F}_n^{-1} existe y es el operador adjunto $\bar{\mathcal{F}}_n$ a \mathcal{F}_n . Es decir, para $y \in \Pi_n$

$$(\mathcal{F}_n^{-1}y)_k = (\bar{\mathcal{F}}_ny)_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} y_j w_n^{jk}. \quad (3.39)$$

La suma en (3.38) y (3.39) respecto a j puede ser extendida a cualquier conjunto de enteros diferentes módulo n , por ejemplo

$$(\mathcal{F}_nx)_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_{-j} w_n^{jk}.$$

3.2.2 Transformadas Rápidas de Fourier, FFT

Para $x \in \Pi_n$ consideremos el problema de cálculo de la transformada $y = \mathcal{F}_nx$. Evidentemente, basta con calcular los elementos de un período completo, por ejemplo, y_0, \dots, y_{n-1} , puesto que $y \in \Pi_n$. Supongamos que las potencias de w_n en (3.38) ya esta formadas, entonces para cada $k = 0, 1, \dots, n-1$, el cálculo y_k mediante la fórmula (3.38) necesita $n-1$ multiplicaciones complejas (hemos excluido las multiplicaciones triviales correspondientes a $j=0$). Por lo que el cálculo de una transformada completa necesita $n(n-1)$ multiplicaciones complejas. En práctica, como por ejemplo en el análisis de series temporales, se necesita n bastante elevado como $n = 2^{14}$, entonces necesitaremos $n(n-1) \simeq 2.68 \cdot 10^8$ multiplicaciones complejas. Por lo que el tiempo requerido para formar una única transformada parece ser bastante elevado, y eso puede impedir la aplicabilidad del método de Fourier.

El objetivo de esta sección es la descripción del método de transformadas rápidas de Fourier *FFT* del ingles (*Fast Fourier Transformation*). Este método nos permite calcular una transformada de Fourier $y = \mathcal{F}_nx$ con una reducción drástica en el tiempo respecto al uso directo de (3.38), sobre todo para n que admiten la descomposición en varios factores como por ejemplo $n = 2^l$.

Supongamos que $n = qp$ con $q, p \in \mathbb{N}$ positivos. Para $x \in \Pi_n$ consideremos las subsucesiones $x^{(j)}$ cuyo índice es $\equiv a \pmod{p}$

$$x^{(j)} = \{x_k^{(j)}\} = \{x_{j+kp}\}, \quad j = 0, 1, \dots, p-1.$$

Evidentemente

$$x^{(j)} \in \Pi_q, \quad j = 0, \dots, p-1.$$

Supongamos que las p transformaciones

$$y^{(j)} = \{y_k^{(j)}\} = \mathcal{F}_q x^{(j)}, \quad j = 0, 1, \dots, p-1,$$

ya están formadas, i.e. conocidas. Nuestro objetivo, es intentar expresar a $y = \mathcal{F}_n x$ en términos de los $y^{(j)}$. Para todo $m = 0, 1, \dots, n-1$, tenemos

$$y_m = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} x_k w_n^{-mk} = \frac{1}{\sqrt{p}} \sum_{j=0}^{p-1} \frac{1}{\sqrt{q}} \sum_{h=0}^{q-1} x_{j+hp} w_n^{-m(j+hp)}.$$

Y puesto que $w_n^p = w_q$ se tiene que

$$\begin{aligned} y_m &= \frac{1}{\sqrt{p}} \sum_{j=0}^{p-1} \left[\frac{1}{\sqrt{q}} \sum_{h=0}^{q-1} x_{j+hp} w_q^{-mh} \right] w_n^{-mj} \\ &= \frac{1}{\sqrt{p}} \sum_{j=0}^{p-1} y_m^{(j)} w_n^{-mj}. \end{aligned} \quad (3.40)$$

Puesto que $y_m^{(j)}$ son conocidas, el cálculo de una transformada $y = \mathcal{F}_n x$ en base de (3.40) sólo necesita $n(p-1)$ multiplicaciones.

Supongamos, ahora, que n puede ser compuesto en l factores

$$n = n_1 n_2 \dots n_l,$$

entonces podemos usar la fórmula (3.40) de forma recursiva. Es decir, para calcular $\mathcal{F}_n x$, necesitamos n_1 transformaciones de periodo $q = n_2 \dots n_l$, y para calcular estos últimos, se necesita $n_1 n_2$ transformaciones de periodo $q = n_3 \dots n_l$, y así sucesivamente hasta llegar a $n_1 \dots n_l = n$ transformaciones de periodo 1, $\mathcal{F}_1 x$, las cuales son triviales ya que $\mathcal{F}_1 x = x$. El número total de multiplicaciones necesario para hacer dicho cálculo es

$$n(n_1 - 1) + n_1 \frac{n}{n_1} (n_2 - 1) + \dots + (n_1 \dots n_{l-1}) \frac{n}{n_1 \dots n_{l-1}} (n_l - 1) = n \sum_{r=1}^l (n_r - 1).$$

Si por ejemplo $n = 2^l$, el número de multiplicaciones necesarias sería

$$nl = n \log_2 n,$$

que es una reducción considerable en el número de multiplicaciones respecto al uso de (3.38) directamente. Para más detalles se puede consultar a HINRICI (1986).

3.2.3 Coeficientes de Fourier

Consideremos el espacio Π de todas las funciones definidas sobre \mathbb{R} y a valor complejo, que son periódicas con período 1,

$$\Pi = \{f : \mathbb{R} \rightarrow \mathbb{C} \mid f(t+1) = f(t), \forall t \in \mathbb{R}\}.$$

Si $f \in \Pi$ es integrable, entonces sus coeficientes de Fourier complejos son definidos por

$$a_k = \int_0^1 f(t)e^{-2\pi ikt} dt, \quad k = 0, \pm 1, \pm 2, \dots \quad (3.41)$$

La serie de Fourier es la serie definida mediante los coeficientes a_k por

$$s(t) = \sum_{k=-\infty}^{\infty} a_k e^{2\pi ikt}.$$

Bajo ciertas condiciones, esta serie converge hacia $f(t)$, para algunos o todos los valores de t .

El objetivo de esta sección es la descripción del uso de la fórmula de trapecios para calcular el coeficiente a_k en (3.41). En diversas aplicaciones, no es posible obtener una expresión explícita cerrada para el cálculo de la integral en (3.41). Esto ocurre, particularmente, si la función f es conocida sólo empíricamente, o bien si f solo puede ser evaluada en un conjunto discreto de puntos t . Dichos puntos son conocidos como *Puntos Muestrales*. Generalmente, estos puntos se escogen equidistantes. Supongamos, entonces, que los puntos muestrales son

$$t_k = kh, \quad h = \frac{1}{n},$$

donde $k \in \mathbb{Z}$ y $n \geq 1$ es un entero fijo. Los valores

$$x_k = f(t_k), \quad k \in \mathbb{Z}$$

son llamados *Valores Muestrales* de f . Notemos que estos valores forman una sucesión en Π_n

$$x = \{x_k\}_{k=-\infty}^{\infty} \in \Pi_n. \quad (3.42)$$

Ahora veremos la fórmula del trapecio para calcular la integral de una función sobre un intervalo. Sea g una función definida sobre el intervalo acotado $[\alpha, \beta]$. Dividimos

dicho intervalo en n puntos equidistantes $t_k = \alpha + kh$, $k = 0, 1, \dots, n$ con $h = (\beta - \alpha)/n$ de modo que $t_0 = \alpha$ y $t_n = \beta$. Entonces la fórmula de trapecios viene dada por

$$\int_{\alpha}^{\beta} g(t) dt \simeq \sum_{k=1}^{n-1} hg(t_k) + \frac{h}{2}[g(t_0) + g(t_n)]. \quad (3.43)$$

Si aplicamos esta fórmula a (3.41), obtendremos, para $m \in \mathbb{Z}$,

$$a_m \simeq \frac{1}{n} \sum_{k=1}^{n-1} x_k w_n^{-km} + \frac{1}{2n}[x_0 + x_n],$$

ya que $h = 1/n$, y $w_n^{-km} = 1$ para $k = 0$ y $k = n$. Usando, el hecho de que f es periódica en $[0, 1]$, tendremos

$$x_0 = f(0) = f(1) = x_n.$$

Entonces

$$a_m \simeq \frac{1}{n} \sum_{k=0}^{n-1} x_k w_n^{-km} = \frac{1}{\sqrt{n}} (\mathcal{F}_n x)_m.$$

Podemos, entonces tratar a

$$\hat{a}_m = \frac{1}{n} \sum_{k=0}^{n-1} x_k w_n^{-km} \quad (3.44)$$

como una aproximación al coeficiente de Fourier a_m en (3.41).

Notemos entonces que la sucesión

$$\hat{a} = \{\hat{a}_m\}$$

es simplemente la transformada, por el operador de Fourier discreto \mathcal{F}_n , de la sucesión x definida en (3.42) multiplicada por $1/\sqrt{n}$. Es decir

$$\hat{a} = \frac{1}{\sqrt{n}} \mathcal{F}_n x,$$

y entonces puede ser evaluada usando el algoritmo de transformadas rápidas de Fourier, *FFT*. Se puede consultar HINRICI (1986) para más detalles.

Capítulo 4

ESTIMACIÓN BOOTSTRAP

INTRODUCCIÓN: En el *Capítulo 1* hemos hablado del método bootstrap en general sin entrar en detalles. En este capítulo estudiaremos las técnicas de estimación propuestas por el método bootstrap. Supongamos que estamos ante la situación siguiente: Dada una muestra aleatoria $\mathcal{X}_n = (X_1, X_2, \dots, X_n)$ de tamaño n i.i.d. extraída a partir de una distribución desconocida F , supongamos que queremos estimar un parámetro $\theta = t(F)$ por el estadístico $\hat{\theta} = T_n(\mathcal{X}_n)$ (que puede ser la estimación por substitución $\hat{\theta} = t(\hat{F})$, con \hat{F} una estimación razonable de F). El objetivo es aproximar algunas medidas de precisión de dicho estadístico, tal como su varianza, su error cuadrático medio o su error estándar, su sesgo o su distribución. En general las medidas de precisión citadas no son fáciles de evaluar mediante los métodos clásicos puesto que en general para casi todos los estadísticos no existen fórmulas que sean función de los datos, tal como en (1.6), que permitan calcular dichas medidas. El bootstrap fue introducido por BRADLEY EFRON en 1979 particularmente para resolver este tipo de problemas.

Consideremos una variable aleatoria $R_n = R_n(\mathcal{X}_n, F)$ que sea función de la muestra \mathcal{X}_n y posiblemente de la distribución desconocida F . Por ejemplo, cuando el objetivo es la construcción de un intervalo de confianza para θ , la variable aleatoria R_n puede ser la diferencia $T_n - \theta$, o la versión estudentizada $(T_n - \theta)/S_n$ donde S_n es un estimador de la desviación estándar del estadístico T_n . Cuando simplemente se necesita estimar el sesgo o la varianza de T_n , la variable aleatoria R_n es simplemente el estadístico T_n .

Por simplicidad y con objeto de ilustrar el procedimiento bootstrap, supongamos que la variable aleatoria de interés es $R_n = T_n(\mathcal{X}_n)$. El método bootstrap implementa una técnica artificial que imita la realidad con el fin de obtener aproximaciones de las medidas objetos del análisis. Para entenderlo bien, vamos a visualizar paralelamente

dos problemas, uno real y otro artificial.

El problema real: está representado por la función de distribución desconocida F , a partir de la cual la única información que tenemos son los datos observados $\mathcal{X}_n = (X_1, \dots, X_n)$, y por el parámetro $\theta = t(F)$ el cual queremos estimar basándonos en dicha información, digamos por el estadístico $\hat{\theta} = T_n(\mathcal{X}_n)$.

El problema artificial (bootstrap): es un problema que imita al problema real y está representado por una distribución \hat{F} que es una estimación consistente de F , y el parámetro θ reemplazado por el parámetro $\theta^* = t(\hat{F})$. La condición básica sobre los parámetros θ y θ^* de los dos problemas es que estén bien definidos mediante un mismo funcional t de las distribuciones, es decir, en el problema real $\theta = t(F)$, y en el problema artificial $\theta^* = t(\hat{F})$ de modo que la relación que hay entre θ y F es la misma relación que hay entre θ^* y \hat{F} . La elección de la distribución \hat{F} depende generalmente de la naturaleza de F . Cuando se sabe que F pertenece a una clase paramétrica de distribuciones, es decir, que $F = F_\xi$ con $\xi \in \Omega$ es un parámetro o vector de parámetros desconocidos entonces se utiliza $\hat{F} = F_{\hat{\xi}}$ donde $\hat{\xi}$ es un estimador consistente de ξ . Este procedimiento se llama *Bootstrap paramétrico*. Por otro lado, cuando no hay ningún modelo paramétrico evidente para F , se escoge entonces $\hat{F} = F_n$ la función de distribución empírica, FDE, asociada a los datos. Este último procedimiento se llama *Bootstrap no paramétrico*.

Notemos aquí que el parámetro bootstrap θ^* es entonces la estimación con sustitución de θ . Esto significa que θ^* coincide con el estadístico de interés $\hat{\theta} = T_n$ si $T_n(\mathcal{X}_n) = t(\hat{F})$ es la estimación con sustitución. Aunque en general θ^* y $\hat{\theta}$ no tienen por que ser iguales, como muestra el ejemplo siguiente

Ejemplo 4.1 *Sea una muestra de n observaciones independientes, extraídas a partir de una distribución continua $F(x)$. Definimos el parámetro θ como la mediana de F (aquí t es el funcional mediana de F), y supongamos que $F(x) = \frac{1}{2}$ tiene una única solución θ . Sea $X_{(1)}, \dots, X_{(n)}$ la muestra ordenada. Supongamos, por conveniencia, que n es impar. El estimador más natural de θ es la mediana de los datos*

$$\hat{\theta} = X_{(\frac{n+1}{2})}$$

Si en el problema bootstrap F se reemplaza por la distribución empírica F_n , el parámetro θ^ que es por definición la mediana de F_n es*

$$\theta^* = t(F_n) = X_{(\frac{n+1}{2})}$$

En este caso tenemos evidentemente que $\theta^ = \hat{\theta}$.*

Supongamos, ahora, que se sabe que F es simétrica y que su esperanza existe. Es bien sabido que en este caso la media y la mediana de F coinciden. Es entonces

más cómodo estimar la mediana θ de F por la media muestral,

$$\hat{\theta} = \bar{X},$$

pero el parámetro bootstrap θ^* que es siempre la mediana de la distribución empírica, queda siempre, $\theta^* = t(F_n) = x_{(\frac{n+1}{2})}$, por lo que en este caso $\theta^* \neq \hat{\theta}$.

4.1 Estimación Bootstrap del Sesgo y de la Varianza

Supongamos en esta sección y en la próxima que la variable aleatoria de interés es el propio estimador de θ , $R_n = T_n = \hat{\theta}$. En la inferencia estadística el objetivo es el estudio de ciertas medidas de precisión de un estimador $\hat{\theta}$ del parámetro θ . Estas medidas de precisión cuantifican diferentes aspectos del comportamiento de $\hat{\theta}$. Entre estas medidas está el sesgo y la varianza del estadístico $\hat{\theta}$. Sea $\mathcal{X}_n = (X_1, \dots, X_n)$ una muestra aleatoria extraída a partir de una distribución de probabilidad desconocida F . Supongamos que se pretende estimar el parámetro $\theta = t(F)$ mediante el estadístico $\hat{\theta} = T_n(\mathcal{X}_n)$ que puede ser la estimación con sustitución $t(\hat{F})$ de θ o no, y donde \hat{F} es una estimación consistente de F que puede ser la función de distribución empírica o una estimación paramétrica.

Definición 7 El sesgo de $\hat{\theta} = T_n(\mathcal{X}_n)$ como estimador de θ es definido como la diferencia entre el valor esperado de $\hat{\theta}$ y el parámetro θ :

$$s = \text{sesg}_F(\hat{\theta}) = E_F(T_n(\mathcal{X}_n)) - t(F), \quad (4.1)$$

y su varianza

$$v = \text{Var}_F(\hat{\theta}) = E_F \{ [T_n(\mathcal{X}_n) - E_F(T_n(\mathcal{X}_n))]^2 \}. \quad (4.2)$$

Para un estimador $\hat{\theta}$ un sesgo o una varianza grandes no son deseables. Suele preferirse una estimación insesgada, $E_F(\hat{\theta}) = \theta$. La estimación por sustitución $\hat{\theta} = t(\hat{F})$ no es necesariamente insesgada, pero el sesgo tiende a ser pequeño comparado con la magnitud de su error estándar. Ésta es una buena propiedad de la estimación por sustitución.

Definición 8 La estimación bootstrap teórica del sesgo del estimador $\hat{\theta} = T_n(\mathcal{X}_n)$ del parámetro θ es definida como la estimación $\text{sesg}_{\hat{F}}$ donde hemos sustituido F por su estimación \hat{F} en la fórmula (4.1),

$$s^* = \text{sesg}_{\hat{F}}(T_n(\mathcal{X}_n^*)) = E_{\hat{F}}(T_n(\mathcal{X}_n^*)) - t(\hat{F}), \quad (4.3)$$

y la varianza bootstrap teórica

$$v^* = \text{Var}_{\hat{F}}(T_n(\mathcal{X}_n^*)) = E_{\hat{F}} \{ [T_n(\mathcal{X}_n^*) - E_{\hat{F}}(T_n(\mathcal{X}_n^*))]^2 \}, \quad (4.4)$$

y donde $\mathcal{X}_n^* = (X_1^*, \dots, X_n^*)$ es una muestra i.i.d con distribución \hat{F} .

Definición 9 La estimación bootstrap teórica del error estandar de un estimador $\hat{\theta} = T_n(\mathcal{X}_n)$ es simplemente la raíz cuadrada de la estimación bootstrap de su varianza :

$$se^* = se_{\hat{F}}(\hat{\theta}^*) = \left(\text{var}_{\hat{F}}^*(\hat{\theta}^*) \right)^{1/2}. \quad (4.5)$$

En (4.3), $t(\hat{F}) = \theta^*$ es la estimación por substitución de θ que representa el parámetro bootstrap que puede ser diferente de $\hat{\theta}$. La estimación bootstrap del sesgo $\text{sesg}_{\hat{F}}$ es la estimación por substitución de sesg_F , donde $\hat{\theta}$ puede ser la estimación por substitución de θ o no, como muestra el siguiente ejemplo:

Ejemplo 4.2 Sea F simétrica y θ su mediana. Supongamos que el tamaño de la muestra n es impar. Podemos estimar θ por la media muestral $\hat{\theta} = \bar{X}$. En este caso, el parámetro bootstrap es $\theta^* = X_{(\frac{n+1}{2})}$ como se explicó en el Ejemplo 4.1. Se tiene entonces que

$$s = E_F(\hat{\theta}) - \theta,$$

y

$$s^* = E_{\hat{F}}(\hat{\theta}^*) - \theta^* = E_{\hat{F}}(\hat{\theta}^*) - X_{(\frac{n+1}{2})},$$

y no

$$s^* = E_{\hat{F}}(\hat{\theta}^*) - \hat{\theta} = E_{\hat{F}}(\hat{\theta}^*) - \bar{X}.$$

Nota 10 Supongamos por ejemplo que $\hat{F} = F_n$ es la FDE. Entonces extraer una muestra aleatoria simple de tamaño n , X_1^*, \dots, X_n^* i.i.d a partir de F_n es equivalente a extraer n observaciones sucesivamente con reemplazamiento y de forma uniforme a partir del conjunto formado por los elementos de la muestra original $\{X_1, \dots, X_n\}$. Es decir, $\Pr\{X_j^* = X_i\} = 1/n$, por todo $i = 1, \dots, n$ y $j = 1, \dots, n$. Este procedimiento se lleva a cabo por ayuda del ordenador utilizando el procedimiento de generación de números aleatorios "random(.)" para seleccionar un conjunto $\{i_1, i_2, \dots, i_n\}$ de números enteros, cada uno con probabilidad $\frac{1}{n}$, a partir del conjunto $\{1, 2, \dots, n\}$. Después, se toma la muestra bootstrap \mathcal{X}_n^* a partir de la muestra original, de forma que

$$X_1^* = X_{i_1}, X_2^* = X_{i_2}, \dots, X_n^* = X_{i_n}. \quad (4.6)$$

Las fórmulas (4.3) y (4.4) son las fórmulas teóricas de la estimación bootstrap. Lamentablemente, para la mayoría de los estimadores que figuran en la literatura, dichas fórmulas al igual que sus correspondientes clásicas en (4.1) y (4.2) no tienen siempre una expresión explícita en función de los datos. La diferencia entre las fórmulas (4.3), (4.4) y sus correspondientes clásicas es que aquí la distribución \hat{F} es conocida. No obstante, cuando el estadístico $T_n(\mathcal{X}_n)$ tiene una expresión complicada, aún siendo la distribución \hat{F} conocida es a menudo imposible de evaluar s^* y v^* exactamente. La ventaja de que \hat{F} sea conocida solamente nos permite usar las simulaciones de Monte Carlo cuando sea difícil evaluar s^* y v^* por la fórmula (4.3) y (4.4) respectivamente.

En la práctica las fórmulas (4.3) y (4.4) no se utilizan directamente. Se procede en general como sigue:

i.) Si en las fórmulas (4.3) y (4.4), $s^* = \text{Sesg}_{\hat{F}}(T_n(\mathcal{X}_n^*))$ y $v^* = \text{var}_{\hat{F}}(T_n(\mathcal{X}_n^*))$ son unas funciones explícitas de los datos, se toma entonces como estimadores bootstrap, respectivamente, del sesgo y de la varianza de $\hat{\theta} = T_n$.

ii.) Si no, como \hat{F} es conocida se utiliza el método de simulaciones de Monte Carlo (el método 2 de Efron (ver EFRON (1979)) para dar una aproximación a s^* y a v^* . Entonces se extraen B muestras bootstrap $\mathcal{X}_n^{*1}, \dots, \mathcal{X}_n^{*B}$ cada una de tamaño n a partir de \hat{F} repitiendo el proceso descrito en (4.6) si $\hat{F} = F_n$ la FDE. Después se calculan las replicas del estadístico T_n correspondientes a cada muestra:

$$\hat{\theta}_b^* = T_n(\mathcal{X}_n^{*b}) = T_n^{*b} \quad b = 1, \dots, B. \quad (4.7)$$

Este proceso nos lleva a obtener una muestra de tamaño B del estadístico T_n^* . Ahora, la aproximación del sesgo ideal s^* se hace sustituyendo la esperanza bootstrap $E_{\hat{F}}[T_n(\mathcal{X}_n^*)]$ en (4.3) mediante su análoga muestral, es decir, mediante la media de las replicas

$$\bar{T}_n^* = \frac{1}{B} \sum_{b=1}^B T_n^{*b} = \frac{1}{B} \sum_{b=1}^B T_n(\mathcal{X}_n^{*b}). \quad (4.8)$$

Entonces, obtendremos la aproximación $\hat{s}_{MC,B}^*$ del sesgo bootstrap ideal basado sobre las B replicas ,

$$\hat{s}_{MC,B}^* = \bar{T}_n^* - t(\hat{F}). \quad (4.9)$$

De la misma forma, se toma la varianza empírica correspondiente a la muestra de réplicas de T_n en (4.7) como aproximación a $v^* = \text{Var}_{\hat{F}}(\hat{\theta}^*)$,

$$\hat{v}_{MC,B}^* = \frac{1}{B-1} \sum_{b=1}^B (T_n^{*b} - \bar{T}_n^*)^2. \quad (4.10)$$

Así, la aproximación del error estandar del estimador $\hat{\theta}$ sería

$$\hat{se}_{MC,B}^* = \left\{ \frac{1}{B-1} \sum_{b=1}^B (T_n^{*b} - \bar{T}_n)^2 \right\}^{\frac{1}{2}}. \quad (4.11)$$

Notemos que las fórmula en (4.9), (4.10) y (4.11) no necesitan cálculo teórico alguno, y siempre son fáciles de obtener por muy complicado que sea el estadístico T_n .

Las aproximaciones bootstrap al sesgo $\hat{s}_{MC,B}^*$ en (4.9) y a la varianza $\hat{v}_{MC,B}^*$ en (4.10) son simplemente las equivalentes muestrales de las estimaciones bootstrap teóricas s^* en (4.3) y v^* en (4.4) respectivamente. Por lo tanto, están justificadas por las leyes de los grandes números, como nos muestra la proposición siguiente:

Proposición 11 *La estimación bootstrap de la varianza, v_B^* , usando las simulaciones de Monte-Carlo tiende a la estimación bootstrap ideal v^* cuando el número de replicaciones $B \rightarrow \infty$, es decir,*

$$\lim_{B \rightarrow \infty} \hat{v}_{MC,B}^* = v^*.$$

Demostración. Puesto que $\mathcal{X}_n^{*1}, \dots, \mathcal{X}_n^{*B}$ son extraídos independientemente de manera aleatoria a partir de \hat{F} , la muestra de replicaciones T_n^{*b} $b = 1, \dots, B$ es una muestra aleatoria de variables i.i.d. Se tiene

$$\begin{aligned} \sum_{b=1}^B (T_n^{*b} - \bar{T}^*)^2 &= \sum_{b=1}^B (T_n^{*b} - E_{\hat{F}}(T_n^*))^2 + \sum_{b=1}^B (E_{\hat{F}}(T_n^*) - \bar{T}^*)^2 \\ &\quad + 2(E_{\hat{F}}(T_n^*) - \bar{T}^*) \sum_{b=1}^B (T_n^{*b} - E_{\hat{F}}(T_n^*)), \end{aligned}$$

entonces,

$$\frac{1}{B} \sum_{b=1}^B (T_n^{*b} - \bar{T}^*)^2 = \frac{1}{B} \sum_{b=1}^B (T_n^{*b} - E_{\hat{F}}(T_n^*))^2 - (E_{\hat{F}}(T_n^*) - \bar{T}^*)^2.$$

Aplicando la ley de los grandes números (se supone que T^* tiene la varianza finita), cuando $B \rightarrow \infty$, el primer término de la derecha tiende a $v^* = \text{var}_{\hat{F}}(T_n^{*1})$, y el segundo término tiende a 0. Por lo tanto,

$$\lim_{B \rightarrow \infty} \hat{v}_{MC,B}^* = \lim_{B \rightarrow \infty} \left[\frac{B}{B-1} \frac{1}{B} \sum_{b=1}^B (T_n^{*b} - \bar{T}^*)^2 \right] = \text{var}_{\hat{F}}(T_n^{*1}) = v^*.$$

Nota 12 En la literatura, tanto a v^* en la fórmula (4.4) (respectivamente a s^* en (4.3)) como a su aproximación $\hat{v}_{MC,B}^*$ en la fórmula (4.10) (respectivamente a $\hat{s}_{MC,B}^*$ en (4.9)) se le denomina estimación bootstrap de la varianza (respectivamente del sesgo) de $\hat{\theta}$. La primera estimación corresponde al método 1 de EFRON y la segunda al método 2 (ver EFRON (1979)). En la práctica a menudo se utilizan las fórmulas (4.10) y (4.9) para calcular la varianza y el sesgo respectivamente, aunque en la teoría a menudo se refiere a las fórmulas (4.4) (4.3) como estimación bootstrap. Sin embargo algunos autores llaman a las estimaciones teóricas v^* y s^* estimaciones bootstrap ideales.

4.2 Estimación Bootstrap de la Distribución y de los Cuantiles

La estimación bootstrap de la distribución y de los cuantiles de una función $h(\hat{\theta}_n) = R_n(\mathcal{X}_n, F)$ de un estimador $\hat{\theta}_n$, de un parámetro desconocido θ relacionado con una f.d.D. desconocida F , juega un papel muy importante en la construcción de un intervalo de confianza de dicho parámetro. Consideremos el problema de estimación de la función de distribución de R_n

$$G_{n,F}(x) = P_F \{R_n(\mathcal{X}_n, F) \leq x\}. \quad (4.12)$$

Por simplicidad escribiremos $G_F(x)$ (o a veces $G(x)$) en lugar de $G_{n,F}(x)$ sin olvidar que dicha distribución depende de n y F . Cuando no sea evidente la estimación de la distribución G_F mediante los métodos clásicos, como la aproximación normal o de student, el método bootstrap suele ser eficaz para ofrecer estimadores razonables a dicha distribución. Como se explicó en el Capítulo 2, en los métodos clásicos la estimación de la distribución de R_n , G_F , generalmente se obtiene haciendo suposiciones, como la de normalidad, o mediante una aproximación asintótica o reemplazando los valores desconocidos en la fórmula teórica de G_F . Lamentablemente, la obtención de una fórmula explícita teórica de G_F es casi siempre imposible de obtener. Sea \hat{F} una estimación adecuada de F , que dependiendo de la naturaleza de F , puede ser la FDE F_n u otra estimación como por ejemplo una estimación paramétrica $F_{\hat{\xi}}$.

Definición 13 Se define la estimación bootstrap teórica de la distribución G_F de R_n en (4.12) como sigue

$$G^*(x) = G_{n,\hat{F}}(x) = P^* \{R_n(\mathcal{X}_n^*, \hat{F}) \leq x\}, \quad (4.13)$$

donde \mathcal{X}_n^* es una m. a. s. con distribución \hat{F} y $P^* = P_{\hat{F}}$ es la probabilidad bajo la distribución \hat{F} . Es equivalente a la probabilidad condicional fijada la muestra original.

La definición de G^* en (4.13) representa la fórmula teórica del estimador bootstrap de la distribución $G_F(x)$. Desafortunadamente dicha fórmula, al igual que la fórmula en (4.12) no tiene siempre una expresión cerrada sencilla como función de los datos. Sin embargo, la diferencia capital entre (4.12) y (4.13) es que en la definición bootstrap (4.13) la distribución \hat{F} es conocida. Por ello, en el método bootstrap se procede en la práctica de la manera siguiente:

i.) Si es posible obtener una expresión explícita exacta para $G^*(x)$ en (4.13) en función de la muestra, entonces se toma como estimación bootstrap de $G_F(x)$ y será la estimación ideal.

ii.) Si el cálculo de $G^*(x)$ en (4.13) es complicado y no se puede obtener una expresión explícita en función de los datos, en este caso se usa el método 2 de Efron que se basa en las simulaciones de Monte Carlo para aproximar la estimación ideal $G^* = G_{\hat{F}}$. Para ello, repetimos el procedimiento en (4.6) varias veces, digamos B veces, para obtener B muestras bootstrap $\mathcal{X}_n^{*b} = (X_1^{*b}, \dots, X_n^{*b})$ de tamaño n , $b = 1, \dots, B$, independientes todas con distribución \hat{F} . Calculamos las B replicas $R_n^{*b} = R_n(\mathcal{X}_n^{*b}, \hat{F})$, $b = 1, \dots, B$, de la variable R_n correspondientes a cada muestra bootstrap y aproximamos a G^* por su análoga empírica, G_B^* asociada a $R_n^{*1}, \dots, R_n^{*B}$,

$$G_B^*(x) = \frac{1}{B} \sum_{b=1}^B I_{\{R_n^{*b} \leq x\}}. \quad (4.14)$$

Notemos que la fórmula (4.14) es independiente de la expresión de la variable aleatoria R_n , es decir que no necesita cálculo teórico y siempre es fácil de evaluar para cualquier variable de interés R_n por muy complicada que sea. La convergencia de G_B^* a la distribución exacta de R_n^* bajo remuestreo a partir de \hat{F} esta asegurada por el teorema de Glivenko-Kantelli. En muchas situaciones la estimación G_B^* da buenos resultados y sobre todo evita a hacer suposiciones tales como la de normalidad.

Para la estimación de los cuantiles el resultado principal usado es el siguiente: Si Y_1, \dots, Y_N es una muestra i.i.d con distribución D y si $Y_{(j)}$ es j -ésimo valor ordenado de la muestra, entonces se tiene la aproximación (ver DAVISON & HINKLEY (1997)),

$$E(Y_{(j)}) \simeq D^{-1}\left(\frac{j}{N+1}\right).$$

Este resultado implica que una estimación sensible de $D^{-1}(\alpha)$ es $Y_{((N+1)\alpha)}$, suponiendo, aquí, que $(N+1)\alpha$ es un entero.

Sea $q_\alpha = q_{R_n}^{(\alpha)}$ el cuantil de orden α de la distribución G_F de $R_n(\mathcal{X}_n, F)$.

Definición 14 Se define el estimador bootstrap ideal del cuantil α , q_α , de la distribución de R_n como sigue:

$$q_\alpha^* = G^{*-1}(\alpha) = \inf \{t, G^*(t) \geq \alpha\}. \quad (4.15)$$

Y se define el estimador bootstrap basado sobre el método 2 de Efron del cuantil de orden α , de la distribución G_n de R_n como sigue:

$$\hat{q}_{\alpha,B}^* = G_B^{*-1}(\alpha) = \inf \{t : G_B^*(t) \geq \alpha\}. \quad (4.16)$$

En otras palabras, el estimador bootstrap ideal del cuantil α de R_n es el cuantil α de la estimación bootstrap ideal de la distribución de R_n , y la estimación bootstrap basada sobre B simulaciones del cuantil α de R_n es el cuantil α de la estimación bootstrap basada sobre B simulaciones de la distribución de R_n . En la práctica, a menudo, se usa el estimador bootstrap (4.16). Así, si $R_n^{*1}, \dots, R_n^{*B}$ son las B repeticiones de la variable de interés, R_n , correspondientes a las B muestras bootstrap $\mathcal{X}_n^{*1}, \dots, \mathcal{X}_n^{*B}$ extraídas a partir de \hat{F} , y si $R_n^{*(1)}, \dots, R_n^{*(B)}$ son los estadísticos ordenados, entonces $\hat{q}_{\alpha,B}^*$ viene dado por

$$\hat{q}_{\alpha,B}^* = \{R_n^{*((B+1)\alpha)}\}, \quad (4.17)$$

donde $[(B+1)\alpha]$ representa la parte entera de la cantidad $(B+1)\alpha$, es decir, el número entero más grande $\leq (B+1)\alpha$. En general, es más cómodo elegir el entero B de modo que $(B+1)\alpha$ sea un número entero. Sin embargo si por alguna razón $(B+1)\alpha$ no puede ser entero entonces se puede usar interpolación sobre escala de los cuantiles normales (ver DAVISON & HINKLEY (1997)). Es decir si $(B+1)\alpha$ no es un entero, y si $k = [(B+1)\alpha]$ entonces definimos $R_n^{*((B+1)\alpha)}$ como

$$R_n^{*((B+1)\alpha)} = R_n^{*(k)} + \frac{\Phi^{-1}(\alpha) - \Phi^{-1}(\frac{k}{B+1})}{\Phi^{-1}(\frac{k+1}{B+1}) - \Phi^{-1}(\frac{k}{B+1})} (R_n^{*(k+1)} - R_n^{*(k)}), \quad (4.18)$$

siempre que $k \notin \{0, B, B+1\}$, ya que para estos valores no podemos usar (4.18). Generalmente, la estimación de los cuantiles necesita que el número de simulaciones B sea superior a 1000. La elección de $B = 999$ junto con los valores convenientes de α (por ejemplo $\alpha = 0.025, 0.05, 0.1, 0.95, 0.975$) hace que $(B+1)\alpha$ sea entero.

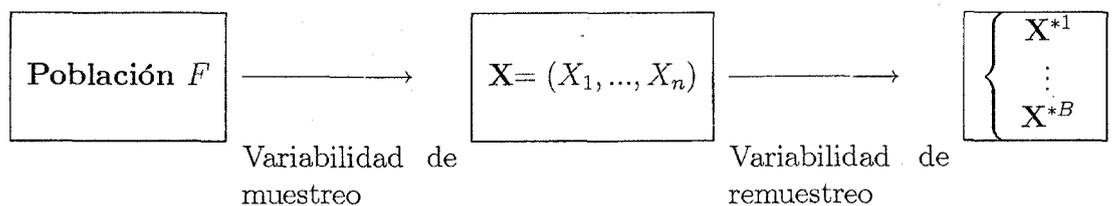
4.3 Error en la estimación Bootstrap

4.3.1 Introducción

Sea $\mathcal{X}_n = (X_1, \dots, X_n)$ una m.a.s. extraída a partir de una f.d.D F . Sea \hat{F} una estimación razonable de F (que puede ser paramétrica), y sean $\theta = t(F)$ y $\hat{\theta} = T(\mathcal{X}_n) =$

$t(\hat{F})$ el parámetro y el estadístico de interés respectivamente. La idea principal del método bootstrap es estimar una cierta característica $C(F)$ del estimador $\hat{\theta}$ como su varianza ó su sesgo mediante $C(\hat{F})$. Más bien, estimar una característica de una variable aleatoria $R_n(\mathcal{X}_n, F) = R(T_n, F)$ relacionada con T_n , y posiblemente dependiendo de otro estadístico, como $h(T_n) - h(\theta)$ o $(T_n - \theta) / \hat{v}_n$, donde \hat{v}_n es una estimación de la varianza de T_n . Cuando no sea posible calcular $C(\hat{F})$ con exactitud, la aproximamos por \hat{C}_B usando B simulaciones, y se tiene que $C(\hat{F}) = \lim_{B \rightarrow \infty} \hat{C}_B$. Es decir que \hat{C}_B tiende a la estimación bootstrap ideal $\hat{C}_\infty = C(\hat{F})$. Una de las ventajas de la estimación bootstrap es que es aproximadamente insesgada por construcción. Sin embargo, la estimación bootstrap puede tener una varianza considerable. Naturalmente, como cualquier método estadístico, el método de estimación bootstrap no es exacto, por ello tiene algunos errores inherentes. Cuando se utilizan las simulaciones, el error en el método bootstrap es una combinación de dos errores: un error estadístico y un error de simulación. En otras palabras la variabilidad en la estimación bootstrap basada en las simulaciones es una combinación de la *variabilidad de muestreo*, debida a que tenemos solamente una muestra de tamaño n y no toda la población, y la *variabilidad de remuestreo*, debida a que se simulan solamente un número finito B de muestras bootstrap y no un número infinito.

El objetivo de esta sección es estudiar como la varianza de \hat{C}_B depende de ambos: del tamaño muestral n y del número de replicaciones bootstrap B .



Estudiaremos también maneras de estimar dicha varianza a partir de las mismas muestras bootstrap. También discutiremos el número necesario de simulaciones B para efectuar algunas estimaciones bootstrap.

4.3.2 Error de Muestreo

El error estadístico o error de muestreo es debido a la diferencia que hay entre el modelo desconocido F y su estimación \hat{F} usada por el método bootstrap. Por lo que también es la diferencia entre $C(F)$ y $C(\hat{F})$. Este error generalmente puede ser reducido o incluso eliminado usando una variable aleatoria $R(T_n, F)$ adecuada. Por ejemplo, si $Q_n = R(T_n, F)$ es un estadístico pivote entonces la construcción

de un intervalo de confianza para θ basándonos en los cuantiles de Q_n puede ser más preciso que si nos basamos sobre los cuantiles de T_n directamente. En general, una variable aleatoria es un pivote si su distribución es la misma bajo F y \hat{F} . Si $F = F_\xi$ es una distribución paramétrica, entonces Q_n es un pivote si su distribución es independiente del parámetro ξ . Es decir, si estimamos F_ξ por $\hat{F} = F_\xi$ entonces Q_n y Q_n^* tendrán la misma distribución. Esto significa que los cuantiles bootstrap ideales, de Q_n^* , coincidirán con los verdaderos cuantiles de Q_n .

4.3.3 Error de Remuestreo

Este error resulta cuando se utilizan las simulaciones en la estimación bootstrap y es debido al uso de un número finito B de simulaciones para aproximar a $C(\hat{F})$ en lugar de un número infinito. En otras palabras, el error de simulación es debido al uso de estimaciones empíricas \hat{C}_B^* de las características deseadas, bajo remuestreo de \hat{F} , en lugar de considerar las características exactas $C(\hat{F})$. Por ejemplo, se aproxima la estimación bootstrap $C(\hat{F}) = E^*(T_n^*) - T_n$ del sesgo $C(F) = E_F(T_n) - \theta$ de T_n por la media $\hat{C}_B^* = B^{-1} \sum_{b=1}^B T_n^{*b} - T_n$ usando B replicas $T_n^{*1}, \dots, T_n^{*B}$ de T_n cada una basada en una muestra aleatoria simple extraída a partir de \hat{F} . El error de simulación es simplemente la diferencia entre \hat{C}_B^* y $\hat{C}_\infty^* = C(\hat{F})$. En consecuencia, la variabilidad en \hat{C}_B^* sólo puede ser eliminada usando un número infinito de simulaciones, lo que es imposible. Entonces surge la pregunta ¿Como de grande podemos elegir B para que nuestras aproximaciones sean lo suficientemente aceptables? Una respuesta general a esta pregunta es difícil de obtener, pero es posible dar una idea considerando la estimación del sesgo, del error estándar o de un cuantil en algunos casos concretos como veremos en una sección más adelante.

Estimación del Sesgo

Consideremos la estimación del sesgo $s = E(T_n) - \theta = C(F)$ mediante el uso de las simulaciones de Monte Carlo como se ha explicado en secciones anteriores de este capítulo. Entonces

$$\hat{s}_{MC,B}^* = \frac{1}{B} \sum_{b=1}^B T_n^{*b} - T_n = \hat{C}_B^*$$

La media y la varianza de este estimador respecto a todas las posibles simulaciones son

$$E^*(\hat{s}_{MC,B}^*) = \frac{1}{B} \sum_{b=1}^B E^*(T_n^{*b}) - T_n = E_{\hat{F}}(T_n^*) - T_n,$$

y

$$\text{var}^*(\hat{s}_{MC,B}^*) = \frac{1}{B^2} \sum_{b=1}^b \text{var}^*(T_n^{*b}) = \frac{\text{var}^*(T_n^*)}{B}.$$

Aplicamos el teorema de la varianza total a la varianza no condicionada de $\hat{s}_{MC,B}^*$ que toma en cuenta, también, la variabilidad de muestras obtenidas a partir de F , obtendremos

$$\text{var}(\hat{s}_{MC,B}^*) = \text{var}_F[E^*(\hat{s}_{MC,B}^*)] + E_F[\text{var}^*(\hat{s}_{MC,B}^*)]. \quad (4.19)$$

Supongamos que $\theta = \mu$ la media de la población y $T_n = \bar{X}$ la media muestral. En este caso se verifica

$$E^*(\hat{s}_{MC,B}^*) = 0 \text{ y } \text{var}^*(\hat{s}_{MC,B}^*) = \frac{\hat{\sigma}^2}{nB},$$

con $\hat{\sigma}^2 = n^{-1} \sum_{i=1}^n (X_i - \bar{X})^2$ la varianza muestral. Entonces la fórmula (4.19) da

$$\text{var}(\hat{s}_{MC,B}^*) = \text{var}_F(0) + E_F\left(\frac{\hat{\sigma}^2}{nB}\right) = \frac{\sigma^2}{nB} \frac{n-1}{n}.$$

Notemos que este resultado depende tanto de n como de B . Se obtiene un resultado similar si el estadístico T_n admite una aproximación lineal, como en (2.13), con un error de orden $O(n^{-2})$.

Nota 15 La varianza y la esperanza respectivamente en el primer y el segundo término de la parte derecha de (4.19) se refieren al mecanismo aleatorio de escoger las muestras \mathcal{X}_n a partir de F .

Estimación del Error Estándar

Estudiaremos en este apartado la variabilidad en la estimación bootstrap del error estandar se_B^* del estadístico $\hat{\theta} = T_n(\mathcal{X}_n)$. Es decir, aquí tenemos $\hat{C}_B^* = \text{se}_B^*$ tal que

$$\text{se}_{MC,B}^* = \left\{ \frac{1}{B} \sum_{b=1}^B (T(\mathcal{X}_n^{*b}) - \bar{T}^*)^2 \right\}^{\frac{1}{2}}, \quad (4.20)$$

donde $\bar{T}^* = \sum_1^B T(\mathcal{X}_n^{*b})/B$. Esta fórmula es la misma que la fórmula obtenida en (4.11) sólo que hemos dividido por B en lugar de por $B-1$ por conveniencia.

La cantidad se_B^* que mide la variabilidad del estadístico T_n tiene una varianza que consta de dos componentes, uno debido al efecto de muestreo y otro debido al efecto de remuestreo. Podemos probar que:

$$\text{var}(se_{MC,B}^*) = \frac{c_1}{n^2} + \frac{c_2}{nB}, \quad (4.21)$$

donde c_1 y c_2 son constantes dependientes de F y no de n ó B . En efecto, utilizando el teorema de Madow,

$$\text{var}(se_{MC,B}^*) = \text{var}_F [E^*(se_{MC,B}^*)] + E_F [\text{var}^*(se_{MC,B}^*)]. \quad (4.22)$$

Sea m_i^* el i -ésimo momento de la distribución bootstrap de $T(\mathcal{X}_n^*)$ y sea $\Delta^* = (m_4^*/m_2^{*2}) - 3$ su curtosis. Ambos m_i^* y Δ^* son funciones de los datos \mathcal{X}_n . Utilizando las fórmulas usuales de la media y de la varianza de la desviación estandar muestral (ver SERFLING (1980)), podemos escribir

$$\text{var}(se_{MC,B}^*) \simeq \text{var}(m_2^{*1/2}) + E \left(\frac{m_2^*}{4B} (\Delta^* + 2) \right). \quad (4.23)$$

Generalmente a partir de la fórmula (4.23) podemos obtener la fórmula (4.21). En lo que sigue, se verá para el caso más sencillo donde $T(\mathcal{X}_n) = \bar{X}$. Sea σ^2 la varianza de la población F , μ_4 su cuarto momento y κ la curtosis normalizada. Entonces se tiene $m_2^* = \hat{\sigma}^2/n$, $\Delta^* = \hat{\kappa}/n$. La fórmula (4.23) se escribe

$$\begin{aligned} \text{var}(se_{MC,B}^*) &\simeq \text{var}\left(\frac{\hat{\sigma}}{\sqrt{n}}\right) + E \left[\frac{\hat{\sigma}^2}{4B} \left(\frac{\hat{\kappa}}{n} + 2 \right) \right] \\ &\simeq \frac{(\mu_4/\mu_2) - \mu_2}{4n^2} + \frac{\sigma^2}{2nB} + \frac{\sigma^2 \kappa}{4n^2 B}. \end{aligned} \quad (4.24)$$

Ignorando el tercer término de la última expresión, se tiene (4.21) con $c_1 = \frac{1}{4}(\mu_4/\mu_2 - \mu_2)$ y $c_2 = \sigma^2/2$. Si F es la distribución normal, se tiene $\mu_4 = 3\sigma^4$ y $\kappa = 0$, lo que lleva a la simplificación de la fórmula (4.24) resultando

$$\text{var}(se_{MC,B}^*) \simeq \frac{\sigma^2}{2n^2} \left(1 + \frac{n}{B} \right).$$

El término c_1/n^2 en (4.21) representa la variación de muestreo, que tiende a cero cuando el tamaño muestral n tiende a infinito, y el término $c_2/(nB)$ representa la variación de remuestreo que tiende también a cero cuando el número de replicaciones bootstrap B tiende a infinito con n fijo.

Estimación de los cuantiles

Sea ahora $\hat{C}_B = \hat{q}_{\alpha,B}^* = \hat{G}_B^{-1}(\alpha)$ el estimador del cuantil α de la distribución bootstrap del estadístico $T_n(\mathcal{X}_n) - \theta$ basado en B simulaciones bootstrap. Entonces $\hat{q}_{\alpha,B}^* = T_n^{*((B+1)\alpha)} - \hat{\theta}$ donde $T_n^{*(1)}, \dots, T_n^{*(B)}$ son los estadístico ordenados y donde suponemos que $(B+1)\alpha$ es un entero. La varianza no condicionada del estimador $\hat{q}_{\alpha,B}^*$ también tiene la forma (4.21) pero con diferentes constantes c_1 y c_2 . Según la fórmula de la varianza total de $\hat{q}_{\alpha,B}^*$ se tiene

$$\text{var}(\hat{q}_{\alpha,B}^*) = \text{var}_F(E^*(\hat{q}_{\alpha,B}^*)) + E_F(\text{var}^*(\hat{q}_{\alpha,B}^*)). \quad (4.25)$$

En general, la varianza del estimador muestral, \hat{a}_α , del cuantil α , a_α , de un estadístico S_n cuya distribución H es continua y con densidad h , viene dada por

$$\text{var}(\hat{a}_\alpha) \simeq \frac{\alpha(1-\alpha)}{nh^2(a_\alpha)}. \quad (4.26)$$

Entonces, si hacemos la aproximación $E^*(\hat{q}_{\alpha,B}^*) = \hat{q}_\alpha$ que es el estimador bootstrap ideal del cuantil q_α de $T_n(\mathcal{X}_n) - \theta$, obtendremos a partir de (4.25) y (4.26),

$$\text{var}(\hat{q}_{\alpha,B}^*) \simeq \text{var}_F(\hat{q}_\alpha) + E_F\left(\frac{\alpha(1-\alpha)}{B\hat{g}^2(\hat{q}_\alpha)}\right), \quad (4.27)$$

suponiendo que la distribución bootstrap ideal $\hat{G} = G_{\hat{F}}$ de $T_n^* - \hat{\theta}$ es continua y con densidad $\hat{g} = g_{\hat{F}}$.

Para estadísticos generales T_n la obtención de propiedades de $\hat{q}_{\alpha,B}^*$ es complicado. Aquí vamos a suponer que los datos originales provienen de una distribución normal $N(\mu, \sigma^2)$ y consideremos $T_n = \bar{X}$ como estimador de $\theta = \mu$. Supongamos también que la aproximación normal a la distribución de \bar{X}^* es exacta, es decir, que la distribución de \bar{X}^* es exactamente $N(\bar{X}, n^{-1}\hat{\sigma}^2)$ con $\hat{\sigma}^2$ la varianza muestral. Esto es cierto si se usa el bootstrap paramétrico, aunque si se usa el bootstrap no paramétrico dicha distribución normal solo aproxima la distribución de \bar{X}^* .

Bajo nuestras hipótesis tendremos

$$E^*(\hat{q}_{\alpha,B}^*) \simeq \hat{q}_\alpha = n^{-1/2}\hat{\sigma}z_\alpha$$

y

$$\text{var}^*(\hat{q}_{\alpha,B}^*) \simeq \frac{\alpha(1-\alpha)}{B\hat{g}^2(\hat{q}_\alpha)} = \frac{2\pi\alpha(1-\alpha)\hat{\sigma}^2 \exp(z_\alpha^2)}{nB}$$

donde \hat{g} es la densidad de $\bar{X}^* - \bar{X}$ y que es la densidad de $N(0, n^{-1}\hat{\sigma}^2)$ y $z_\alpha = \Phi^{-1}(\alpha)$ es el cuantil α de la distribución normal estándar. Entonces a partir de ??) la varianza no condicionada de $\hat{q}_{\alpha, B}^*$ es aproximadamente

$$\text{var}(\hat{q}_{\alpha, B}^*) = \frac{\sigma^2 z_\alpha^2}{2n^2} + \frac{2\pi\alpha(1-\alpha)\sigma^2 \exp(z_\alpha^2)}{nB} \quad (4.28)$$

donde hemos usado la aproximación usual, ver SERFLING (1980), $\text{var}(\hat{\sigma}) \simeq (\mu_4/\mu_2 - \mu_2)/4n$ con $\mu_2 = \sigma^2$ y $\mu_4 = 3\sigma^4$ el segundo y el cuarto momento centrado de $X_1 \stackrel{d}{=} N(\mu, \sigma^2)$.

Notemos que (4.28) tiene la forma de (4.21), donde el primer término a la derecha de (4.28) representa la variabilidad de muestreo y el segundo representa la variabilidad de remuestreo.

Para un estudio del error de remuestreo en el caso de la estimación bootstrap de la distribución se puede consultar SHI, WU & CHEN (1990).

4.4 Número de simulaciones B

Hemos visto que la estimación bootstrap, digamos por ejemplo del error estandar se_B^* , usando el método de simulaciones de Monte Carlo es muy importante en la práctica. Esta estimación tiende a ser mejor al aumentar el número de replicaciones B como muestran las Figuras 1.1 y 1.2, en el sentido de que se acerca a la estimación bootstrap ideal $se_\infty^* = se_{\hat{F}}^*$ que se obtiene para $B = \infty$. Para el estadístico $\hat{\theta} = \bar{X}$, el error estandar se_∞^* se calcula sin simulación mediante la fórmula (1.6). Lamentablemente para la mayoría de los estadísticos no existe tal fórmula, lo que lleva a cabo la necesidad de efectuar simulaciones para aproximar $se_\infty^* = se_{\hat{F}}^*$ por una cierta se_B^* . Un aspecto fundamental de este método radica en la elección del número B de simulaciones. Si bien parece deseable un número de simulaciones lo mayor posible, el problema es que ello puede conllevar un tiempo de computación excesivo. Sin embargo en la mayoría de las simulaciones que figuran en la literatura, se observa que el aumento de B no tiene por qué conducir a mejoras sustanciales en la calidad de las estimaciones. Pero, ¿cómo de grande podemos escoger el número B para que la calidad de nuestra estimación sea adecuada, en el sentido de que nuestra estimación bootstrap tenga un sesgo y una desviación estandar muy cerca a la de la estimación bootstrap ideal? Una respuesta aproximada a esta cuestión se hace en términos del coeficiente de variación del estimador Bootstrap (ver EFRON (1987)). En esta sección veremos como se aproxima el número de simulaciones B necesario para obtener aproximaciones bootstrap aceptables. Veremos que para la estimación del error estándar se necesita en general 100 ó 200 simulaciones y para

la estimación de los cuantiles necesitamos un número más grande de simulaciones, B igual 1000 ó 2000.

4.4.1 Aproximación del error estandar

La variabilidad de $se_{MC,B}^*$ nos puede ayudar a determinar el número B de replicas suficiente para obtener una buena estimación $se_{MC,B}^*$ (como aproximación de la estimación bootstrap ideal del error estandar $se_{\hat{F}} = se_{\infty}^*$ de T_n). Cuando variamos n y B la cantidad $E(se_{MC,B}^*)$ también varía, entonces sería útil medir el tamaño de se_B^* relativo a $E(se_{MC,B}^*)$. Por ello, se considera el coeficiente de variación de se_B^* ,

$$CV(se_B^*) = \frac{\text{var}^{\frac{1}{2}}(se_{MC,B}^*)}{E(se_{MC,B}^*)}. \quad (4.29)$$

Dividimos la fórmula (4.23) por m_2^* y tomamos raíz cuadrada y entonces podemos escribir la fórmula (4.29) de la forma

$$CV(se_{MC,B}^*) \simeq \left\{ CV^2(se_{\infty}^*) + \frac{E(\Delta^*) + 2}{4B} \right\}^{1/2}, \quad (4.30)$$

donde $se_{\infty}^* = m_2^{*1/2}$ es la estimación bootstrap ideal del error estándar de θ .

La determinación del número B de replicas necesario para estimar el error estándar se hace comparando $CV(se_{\infty}^*)$ con $CV(se_{MC,B}^*)$ cuando B varía (ver EFRON (1987)). En la *Tabla 4.1* se comparan $CV(se_{\infty}^*)$ y $CV(se_{MC,B}^*)$ para varios valores de B suponiendo que $\Delta = 0$.

$CV(se_{\infty}^*)$ ↓	$B \rightarrow$	25	50	100	200	∞
0.25		0.29	0.27	0.26	0.25	0.25
0.20		0.24	0.22	0.21	0.21	0.20
0.15		0.21	0.18	0.17	0.16	0.15
0.10		0.17	0.14	0.12	0.11	0.10
0.05		0.15	0.11	0.09	0.07	0.05
0.00		0.14	0.10	0.07	0.05	0.00

Tabla 4.1

Observamos que $CV(se_{MC,B}^*)$ es muy cercano a $CV(se_{\infty}^*)$ para $B \geq 200$.

Consideremos, por ejemplo, el caso $T_n = \bar{X}$ donde la muestra original X_1, \dots, X_n es extraída a partir de una población normal. En este caso la fórmula (4.30) se simplifica a

$$CV(se_{MC,B}^*) \simeq \frac{1}{\sqrt{2}} \left(\frac{1}{n} + \frac{1}{B} \right)^{1/2}.$$

La representación gráfica de $CV(se_B^*)$ como función de B (n fijo) es similar al gráfico de la izquierda de la *Figura 4.1*. Se ve claramente según la curva que $CV(se_{MC,B}^*)$ es próximo a $CV(se_\infty^*)$ para $B \geq 200$. Por ejemplo para $n = 50$ se tiene el gráfico de la derecha de la *Figura 4.1*.

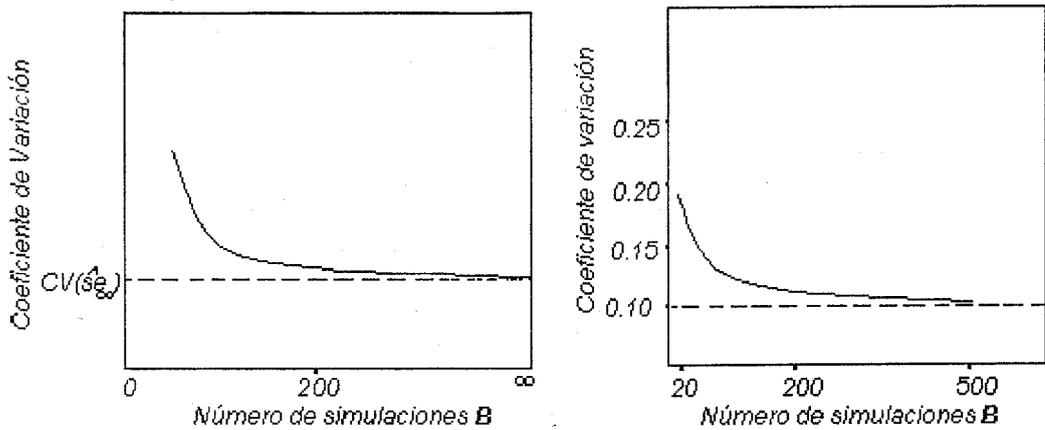


Figura 4.1

Nota 16 Usando las fórmulas usuales de los estimadores de los momentos (ver SERFLING (1980)), podemos escribir

$$E^*(se_{MC,B}^*) \simeq \sqrt{v^*} \text{ y } var^*(se_{MC,B}^*) \simeq \frac{\rho^* - v^{*2}}{4Bv^*},$$

donde $\rho^* = E^*[\hat{\theta}_n^* - E_*(\hat{\theta}_n^*)]^4$. Entonces la aproximación siguiente es válida:

$$CV^*(se_{MC,B}^*) = \frac{\sqrt{var^*(se_{MC,B}^*)}}{E_*(se_{MC,B}^*)} \simeq \sqrt{\frac{\hat{\Delta} + 2}{4B}} \tag{4.31}$$

con $\hat{\Delta}_n = \rho^*/v^{*2} - 3$. Según EFRON (1987), para determinar el número de simulaciones, B , necesario basado en (4.31), consideramos

$$CV^*(se_B^*) = \varepsilon_0, \tag{4.32}$$

donde ε_0 es un cierto nivel deseado. Puesto que $\hat{\Delta}_n \xrightarrow{p} 0$, podemos hacer la aproximación $\hat{\Delta}_n \approx 0$. Así, (4.32) implica $B = \frac{1}{2}\varepsilon_0^{-2}$. Por ejemplo, si $\varepsilon_0 = 0.05$, entonces $B = 200$, y si $\varepsilon_0 = 0.1$ entonces $B = 50$.

4.4.2 Aproximación de los cuantiles

Usando las aproximaciones anteriores para la esperanza y la varianza del cuantil muestral, obtendremos a partir de (4.27) la aproximación siguiente para el coeficiente de variación de $\hat{q}_{\alpha,B}^*$,

$$CV(\hat{q}_{\alpha,B}^*) \simeq \frac{1}{q_\alpha} \left[\frac{\alpha(1-\alpha)}{g(q_\alpha)^2} \left(\frac{1}{n} + \frac{1}{B} \right) \right]^{1/2},$$

donde g es la derivada de G . llevando a cabo los cálculos bajo las hipótesis de normalidad anteriores obtendremos

$$CV(\hat{q}_{\alpha,B}^*) = \left[\frac{2\pi\alpha(1-\alpha)\exp(z_\alpha^2)}{z_\alpha^2} \left(\frac{1}{n} + \frac{1}{B} \right) \right]^{1/2}.$$

La representación gráfica de $CV(\hat{q}_{\alpha,B}^*)$ como función de B con n fijo y con las observaciones originales extraídas de la población normal es similar a la *Figura 4.2*. Observemos que para $B \geq 1000$ el coeficiente de variación de $\hat{q}_{\alpha,B}^*$ es muy cercano al coeficiente de variación del estimador bootstrap ideal $\hat{q}_\infty^{(\alpha)} = \hat{q}^{(\alpha)}$ representado en el gráfico por la línea horizontal. Es decir que el número de replicaciones B necesario para obtener una buena aproximación $\hat{q}_{\alpha,B}^*$ de $\hat{q}_\infty^{(\alpha)}$ es al menos 1000.

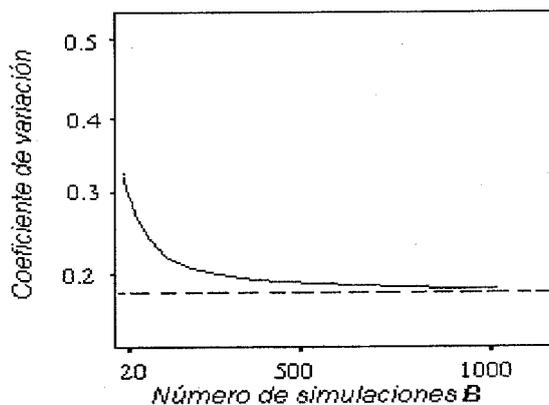


Figura 4.2

Otra manera para aproximar el número de simulaciones B necesario, es calculando la inflación en la varianza de $\hat{q}_{\alpha,B}^*$ debida a las simulaciones. A partir de la fórmula (4.28) podemos ver que dicha inflación es

$$I_{var} = \frac{4\pi n\alpha(1-\alpha)\exp(z_\alpha^2)}{z_\alpha^2 B} = \frac{n}{B}r(\alpha)$$

donde

$$r(\alpha) = \frac{4\pi\alpha(1-\alpha)\exp(z_\alpha^2)}{z_\alpha^2}.$$

Algunos valores de $r(\alpha)$ son

α ó $1-\alpha$	0.01	0.025	0.05	0.10	0.25
$r(\alpha)$	5.15	3.72	3.30	3.56	8.16

Tabla 4.2

Supongamos por ejemplo que $\alpha = 0.025$, entonces para imponer que el factor de inflación de la varianza es del 10% necesitaríamos $B \simeq 40n$.

Nota 17 *La estimación de los cuantiles necesita un número de simulaciones B más grande que el número de simulaciones necesario para estimar el error estándar. Esto es así porque los cuantiles de orden α , con α tendiendo a cero o a uno dependen de la cola de la distribución. Generalmente, si una característica \hat{C}_B del estimador $\hat{\theta}$ depende de la cola extrema de la distribución de $\hat{\theta}^*$, entonces \hat{C}_B necesita un número B más grande de replicaciones bootstrap para obtener una precisión aceptable de nuestra estimación.*

En la sección siguiente veremos un procedimiento que nos va a permitir aproximar el error de remuestreo sin necesidad de nuevas simulaciones.

4.5 Método Jakknife-after-Bootstrap

Este método figura en EFRON (1992). Supongamos que tenemos B muestras bootstrap a partir de los cuales hemos estimado una característica \hat{C}_B del estadístico $\hat{\theta}$. Nuestro objetivo es calcular una medida de incertidumbre de la estimación \hat{C}_B . Sin pérdida de generalidad, supongamos que $\hat{C}_B = \text{se}_B^*$ es la estimación bootstrap del error estándar de $\hat{\theta} = T_n(\mathcal{X}_n)$. El método *jackknife-after-bootstrap* ofrece una manera de estimar la varianza, $\text{var}(\text{se}_B^*)$, del estimador se_B^* usando solamente la información sobre las B muestras bootstrap que tengamos. Dicho método viene dado

por el algoritmo siguiente:

ALGORITMO

INICIO :

$$\mathcal{X}_n = (X_1, \dots, X_n), \hat{\theta} = T_n(\mathbf{X})$$

$$\mathcal{X}_n^{*b} = (X_1^{*b}, \dots, X_n^{*b}), b = 1, \dots, B$$

$$se_B^* = se_B^*(\mathcal{X}_n^{*1}, \dots, \mathcal{X}_n^{*B}) = \left\{ \sum_{b=1}^B [T(\mathcal{X}_n^{*b}) - \sum_{b=1}^B T(\mathcal{X}_n^{*b})/B]^2 \right\}^{1/2}$$

PARA $i = 1, \dots, n$

$$\mathcal{X}_{n(i)}^{*b} = \mathcal{X}_n^{*b} \text{ menos la observación } X_i, b = 1, \dots, B$$

$$\text{Calcular } se_{B(i)}^* = se_B^*(\mathcal{X}_{n(i)}^{*1}, \dots, \mathcal{X}_{n(i)}^{*B})$$

SIGUIENTE i

$$\text{Calcular } v_{\text{jack}} = \frac{1}{n(n-1)} \sum_{i=1}^n [se_{B(i)}^* - \sum_{i=1}^n se_{B(i)}^*/n]^2$$

v_{jack} es una estimación de $\text{var}(se_B^*)$.

La dificultad que se presenta aquí es el cálculo de $se_{B(i)}^*$, ya que, el cálculo de $se_{B(i)}^*$ necesita un conjunto completo de muestras bootstrap para cada i . Afortunadamente, podemos usar el lema siguiente para obviar este problema.

Lema 18 *Una muestra bootstrap extraída con reemplazamiento a partir de $X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n$. tiene la misma distribución que una muestra bootstrap extraída a partir de X_1, X_2, \dots, X_n en la que ninguna observación bootstrap es igual a X_i .*

Para cada i existen ciertas muestras bootstrap entre $\mathcal{X}_n^{*1}, \dots, \mathcal{X}_n^{*B}$ en las cuales la observación X_i no aparece. Entonces podemos estimar $se_{B(i)}^*$ mediante la muestra desviación estándar de $\hat{\theta}^*$ correspondiente a las muestras bootstrap \mathcal{X}_n^{*b} en las cuales no aparezca la observación X_i . En otras palabras, si $C_i = \{b \in \{1, \dots, B\} : X_i \notin \mathcal{X}_n^{*b}\}$ y si $B_i = \text{card } C_i$ entonces

$$se_{B(i)}^* = \left[\frac{1}{B_i} \sum_{b \in C_i} (T_n(\mathcal{X}_n^{*b}) - \bar{T}_n)^2 \right]^{1/2},$$

donde $\bar{T}_n = \frac{1}{B_i} \sum_{b \in C_i} T_n(\mathcal{X}_n^{*b})$.

Capítulo 5

◦ MÉTODOS COMPUTACIONALS

En la teoría bootstrap, a menudo, se habla de la estimación bootstrap ideal. Sin embargo, en la práctica cuando es difícil hallar una estimación ideal se efectúan las simulaciones de Monte Carlo (MC) para aproximarla. El problema es que en el caso de la utilización de las simulaciones de MC, además del error inherente debido al proceso de muestreo, se añade otro error debido al proceso de simulación (re-muestreo) como se vió en el capítulo anterior. Por ejemplo, la diferencia $\text{Var}_{\hat{F}}(\hat{\theta}_n) - \text{Var}_{MC}^*(\hat{\theta}_n) = \hat{v} - \hat{v}_{MC}^*$ es debida al proceso de simulación. Tales errores pueden reducirse aumentando B , pero es muy claro que ésta no es la mejor solución, sobre todo cuando $\hat{\theta}_n$ requiere un tiempo de computación considerable. Por ejemplo, para estimar un cuantil de una variable estudentizada basada en un L -estadístico, se necesitan al menos 1000 replicaciones de dicha variable, y para cada replicación se generan unas 100 muestras bootstrap para estimar la varianza del estadístico. Así, se tiene al menos un total de 10^5 conjuntos de datos de tamaño n . Este problema nos obliga a desarrollar ciertos métodos computacionales bootstrap de forma que sean más eficientes que el método usual de simulación Monte Carlo. Dichos métodos computacionales tienen por objetivo la reducción del número de simulaciones, B , necesario para obtener una determinada precisión, o equivalentemente, aumentar la precisión de una estimación basada en un número dado de simulaciones, B . Para ello, algunos autores proponen cambiar el proceso de simulación Monte Carlo por otros métodos de forma que en ciertas situaciones favorables se reduce sustancialmente el error de simulación. Otros proponen técnicas de cálculo que terminan reduciendo la varianza de la estimación de interés. Las técnicas usadas, generalmente, son más complicadas y difíciles de implementar que el método de Monte Carlo estándar,

a cambio se obtiene más precisión en la estimación. En este capítulo citaremos algunos de los métodos computacionales más conocidos en la literatura.

A lo largo de este capítulo consideraremos θ el parámetro de interés, y $\hat{\theta}_n$ el estadístico de interés basado en una muestra aleatoria $\mathcal{X}_n = (X_1, \dots, X_n)$ con distribución desconocida F . Sea F_n la f.d.D. empírica asociada a \mathcal{X}_n .

5.1 Bootstrap Balanceado

5.1.1 Introducción

Para ver el alcance de este método, consideremos la estimación del sesgo de un estimador simétrico en los datos, $\hat{\theta} = T(X_1, \dots, X_n)$. Supongamos que $\hat{\theta} = t(\hat{F})$ es el estimador con sustitución de $\theta = t(F)$ donde t es el funcional que define la relación que hay entre θ y F . Entonces el sesgo ideal sería

$$\hat{s} = E_{\hat{F}}(\hat{\theta}^*) - \hat{\theta} = \int T(x_1^*, \dots, x_n^*) f^*(x_1^*, \dots, x_n^*) dx_1^*, \dots, dx_n^*, \quad (5.1)$$

donde $\hat{\theta}^* = T(X_1^*, \dots, X_n^*)$ y X_1^*, \dots, X_n^* es una m. a. con distribución \hat{F} , y f^* es la densidad conjunta de X_1^*, \dots, X_n^* .

Consideremos el caso no paramétrico donde $\hat{F} = F_n$ es la distribución empírica asociada a X_1, \dots, X_n . En este caso f^* es igual a n^{-n} en cada elemento del conjunto $\mathcal{S} = \{X_1, \dots, X_n\}^n$, y la fórmula en (5.1) se escribe

$$\hat{s} = n^{-n} \sum_{\mathcal{S}} T(X_1^*, \dots, X_n^*) - \hat{\theta}, \quad (5.2)$$

donde la suma es sobre todas las n^n posibles muestras del conjunto \mathcal{S} . Puesto que $\hat{\theta}$ es simétrico en los datos, entonces para calcular \hat{s} en (5.2) sólo necesitaríamos calcular $\binom{2n-1}{n}$ diferentes réplicas de $\hat{\theta}$ para obtener una enumeración completa de los posibles valores de $\hat{\theta}^*$. Sin embargo, esto es impracticable incluso con n pequeño ya que, por ejemplo, si $n = 10$ necesitaríamos calcular $\binom{2n-1}{n} = 92378$ diferentes réplicas del estimador. Por lo que a menudo se recurre al método de Monte Carlo que usa solamente B muestras extraídas aleatoriamente del conjunto \mathcal{S} para aproximar la suma en (5.2) y luego obtener la estimación siguiente del sesgo

$$\hat{s}_{MC}^* = B^{-1} \sum_{b=1}^B T(X_1^{*b}, \dots, X_n^{*b}) - \hat{\theta}. \quad (5.3)$$

El principal defecto de este método es que las observaciones X_i no ocurren con igual frecuencias en el conjunto de las B muestras bootstrap extraídas aleatoriamente del conjunto \mathcal{S} , y esto significa que las B replicas calculadas no representan bien al conjunto total de los valores de $\hat{\theta}^*$. Como veremos mas adelante, la consecuencia de este hecho puede verse fácilmente con $\hat{\theta} = \bar{X}$ donde casi siempre se obtiene $\hat{s}_{MC}^* \neq 0$ que es pura aproximación de $\hat{s} = 0$.

Este error puede ser corregido tanto por el método bootstrap balanceado, debido a DAVISON ET AL. (1986), que proponen modificar el procedimiento usual de simulación, como por el método centrado, debido a EFRON (1990), que propone modificar (centrar) el estimador en (5.3) conservando el procedimiento de simulación de Monte Carlo.

5.1.2 Estimación bootstrap usual y balanceada

Sea $\mathcal{X}_n^{*b} = (X_1^{*b}, \dots, X_n^{*b})$, $b = 1, \dots, B$, B muestras bootstrap generadas mediante un cierto plan de muestreo. Para cada $i = 1, \dots, n$ y $b = 1, \dots, B$ sea

$$N_i^{*b} = \# \{j, X_j^{*b} = X_i\} \text{ y } P_i^{*b} = \frac{N_i^{*b}}{n}. \quad (5.4)$$

El vector $\mathbf{P}^{*b} = (P_1^{*b}, \dots, P_n^{*b})$ es el vector de remuestreo. Para cualquier b se verifica que $\sum_{i=1}^n P_i^{*b} = 1$. En el caso usual de simulación MC, los vectores $\mathbf{N}^{*b} = (N_1^{*b}, \dots, N_n^{*b})$ son i.i.d con distribución multinomial $Mult(n; \frac{1}{n}, \dots, \frac{1}{n})$.

Con el fin de ilustración, consideremos $\hat{\theta}_n = \bar{X}_n$. En este caso, sabemos que el sesgo bootstrap ideal $\hat{s} = E^*(\bar{X}_n^*) - \bar{X} = 0$ es nulo. Considerando el método de simulación MC para aproximar a \hat{s} , se obtiene

$$\hat{s}_{MC}^* = \frac{1}{B} \sum_{b=1}^B \bar{X}_n^{*b} - \bar{X}_n = \frac{1}{B} \sum_{b=1}^B \sum_{i=1}^n P_i^{*b} X_i - \bar{X}_n, \quad (5.5)$$

que es, en general, no nulo. Este error es debido al proceso de simulación MC. Con el fin de corregir este error, DAVISON, HINKLEY & SCHECHTMAN (1986) proponen añadir una condición adicional sobre las probabilidades de remuestreo en (5.4). Esta condición es

$$\sum_{b=1}^B P_i^{*b} = \frac{B}{n}, \quad b = 1, \dots, B \quad (5.6)$$

que es equivalente a $\sum_{b=1}^B N_i^{*b} = B$. En otras palabras, cada elemento de la muestra original debe aparecer exactamente B veces en el conjunto de las B muestras bootstrap. Este nuevo método de simulación se llama *balanceado*, y el método bootstrap

basado sobre dicho método se llama *Bootstrap Balanceado*. En este caso el vector $\mathbf{N}^{*b} = (N_1^{*b}, \dots, N_n^{*b})$ tiene una distribución hipergeométrica multivariante.

Nota 19 Para generar B muestras bootstrap usando el método balanceado, se copia la muestra original B veces en un vector de dimensión nB . A continuación se permutan aleatoriamente las componentes de este vector. Las n primeras componentes del vector permutado conforman la primera muestra bootstrap, \mathcal{X}_n^{*1} , las siguientes n componentes constituyen la segunda muestra bootstrap, \mathcal{X}_n^{*b} , y así sucesivamente hasta la B -ésima muestra bootstrap, \mathcal{X}_n^{*B} .

Usando el método balanceado para aproximar el sesgo ideal $\hat{s} = 0$ de $\hat{\theta}_n = \bar{X}_n$ se tiene que

$$\hat{s}_{Bal}^* = \frac{1}{B} \sum_{i=1}^n \sum_{b=1}^B P_i^{*b} X_i - \bar{X}_n = \frac{1}{nB} \sum_{i=1}^n \left(\sum_{b=1}^B N_i^{*b} \right) X_i - \bar{X}_n = 0.$$

Esto significa que no hay error de simulación. Entonces en este caso, el método *Bootstrap Balanceado (BB)* ha eliminado el error de simulación en \hat{s}_{MC}^* en (5.5).

Cabe entonces esperar que el método BB también mejore al método MC en el caso $R_n = \hat{\theta}_n - E\hat{\theta}_n$, donde $\hat{\theta}_n$ puede ser aproximado mediante un estadístico lineal.

Sea $\theta = T(F)$ y $\hat{\theta}_n = T_n = T(F_n)$, donde T es un funcional. Supongamos que T_n puede expresarse de la forma:

$$\hat{\theta}_n = \theta + \frac{1}{n} \sum_{i=1}^n L(X_i, F) + \frac{1}{2} \frac{1}{n^2} \sum_{j=1}^n \sum_{k=1}^n Q(X_j, X_k, F) + \dots, \quad (5.7)$$

con L y Q son la primera y la segunda derivada en el sentido de Von Mises (Ver Capítulo 2). Entonces la b -ésima replicación bootstrap de T_n se escribe

$$\hat{\theta}_n^{*b} = \hat{\theta}_n + \frac{1}{n} \sum_{i=1}^n N_i^{*b} l_i + \frac{1}{2} \frac{1}{n^2} \sum_{j=1}^n \sum_{k=1}^n N_j^{*b} N_k^{*b} q_{jk} + \dots, \quad (5.8)$$

con $l_i = L(X_i, F_n)$ y $q_{jk} = Q(X_j, X_k, F_n)$. Recordemos que se verifica $\sum_i l_i = 0$ y $\sum_i q_{ij} = \sum_j q_{ij} = 0$. DAVISON, HINKLEY & SCHECHTMAN (1986) se basaron en el lema siguiente para estudiar la expresión anterior,

Lema 20 . 1.) Si $(N_1^{*b}, \dots, N_n^{*b})$ sigue una distribución hipergeométrica multivariante (i.e. bajo el método BB), entonces se verifica la siguiente fórmula para los momentos factoriales,

$$E \left[\prod_{w=1}^m (N_{i_w}^{*b_w})^{(r_w)} \right] = \prod_u n^{(p_u)} \prod_v B^{(q_v)} / (nB)^{(\sum r_w)},$$

donde $n^{(r)} = n!/(n-r)!$, $p_u = \sum_{w: b_w=u} r_w$ y $q_v = \sum_{w: i_w=v} r_w$.

2.) Si $(N_1^{*b}, \dots, N_n^{*b})$ son i.i.d. con distribución multinomial de media $(1, \dots, 1)$, entonces

$$E \left[\prod_{w=1}^m (N_{i_w}^{*b_w})^{(r_w)} \right] = \frac{n^{(r)}}{n^r}, \text{ donde } r = \sum_{w=1}^m r_w.$$

Supongamos que los términos de orden 3 o más en la expresión (5.7) son nulos, i.e. T_n es una función cuadrática de los datos. Sea \hat{s}_B^* el estimador bootstrap del sesgo de $\hat{\theta}_n$ usando B simulaciones obtenidas mediante algún procedimiento. Es decir, \hat{s}_B^* puede ser el estimador usual \hat{s}_{MC}^* usando las simulaciones de Monte Carlo o el estimador bootstrap balanceado \hat{s}_{Bal}^* donde las simulaciones son balanceadas. Usando (5.8) para un estadístico cuadrático se obtiene

$$\begin{aligned} \hat{s}_B^* &= B^{-1} \sum_{b=1}^B \left(\hat{\theta} + \frac{1}{n} \sum_{j=1}^n N_j^{*b} l_j + \frac{1}{2n^2} \sum_{j=1}^n \sum_{k=1}^n N_j^{*b} N_k^{*b} q_{jk} \right) - \hat{\theta} \\ &= B^{-1} \sum_{b=1}^B \left(\frac{1}{n} \sum_{j=1}^n N_j^{*b} l_j + \frac{1}{2n^2} \sum_{j=1}^n \sum_{k=1}^n N_j^{*b} N_k^{*b} q_{jk} \right). \end{aligned} \quad (5.9)$$

A continuación estudiaremos algunas características de \hat{s}_B^* tanto en el caso cuando se utilizan las simulaciones de Monte Carlo como cuando se utilizan simulaciones balanceadas. Esto nos permitirá comparar los dos dichos métodos de simulación.

Simulaciones de Monte Carlo: En este caso los $(N_1^{*b}, \dots, N_n^{*b})$, $b = 1, \dots, B$, son realizaciones independientes de una ley Multinomial $Mult(n; \frac{1}{n}, \dots, \frac{1}{n})$, y $\hat{s}_B^* = \hat{s}_{MC}^*$. En este caso tendremos

$$E_{MC}^*(N_j^{*b}) = 1 \text{ y } Cov_{MC}^*(N_j^{*b}, N_k^{*s}) = \delta_{bs}(\delta_{jk} - n^{-1}).$$

A partir de (5.9) podemos calcular la media y la varianza de \hat{s}_{MC}^* , obteniendo

$$E_{MC}^*(\hat{s}_{MC}^*) = \frac{1}{2} \frac{1}{n^2} \sum_{j=1}^n q_{jj},$$

y

$$\begin{aligned} Var_{MC}^*(\hat{s}_{MC}^*) &= \frac{1}{n^2 B} \left\{ \sum_{j=1}^n l_j^2 + n^{-1} \sum_{j=1}^n l_j q_{jj} \right. \\ &\quad \left. + \frac{1}{4n^2} \left[\sum_{i=1}^n q_{ii}^2 - \frac{1}{n} \left(\sum_{i=1}^n q_{ii} \right)^2 - \frac{2(n-1)}{n} \sum_{i=1}^n \sum_{j=1}^n q_{ij}^2 \right] \right\} \end{aligned} \quad (5.10)$$

Simulaciones Balanceadas: En este caso la distribución conjunta de la tabla, de dimensión $B \times n$, de frecuencias

$$[N_j^{*b}] \quad \begin{array}{l} b = 1, \dots, B \\ j = 1, \dots, n \end{array}$$

es una distribución Hipergeometrica con filas suman n y columnas suman B . En esta situación $\hat{s}_B^* = \hat{s}_{Bal}^*$ y cuya expresión en (5.9) se simplifica a

$$\hat{s}_{Bal}^* = \frac{1}{2Bn^2} \sum_{j=1}^n \sum_{k=1}^n q_{jk} \sum_{b=1}^B N_j^{*b} N_k^{*b}, \quad (5.11)$$

ya que $\sum_j l_j = 0$ y $\sum_b N_j^{*b} = B$ para cada j . En este caso se tiene que

$$E_{Bal}^*(N_j^{*b}) = 1 \quad \text{y} \quad Cov_{Bal}^*(N_j^{*b}, N_k^{*s}) = \frac{(n\delta_{jk} - 1)(B\delta_{bs} - 1)}{nB - 1}.$$

A partir de (5.11) podemos calcular la media y la varianza de \hat{s}_{Bal}^* , bajo el bootstrap balanceado, se obtiene

$$E_{Bal}^*(\hat{s}_{Bal}^*) = \frac{1}{2} \frac{n(B-1)}{nB-1} \frac{1}{n^2} \sum_{i=1}^n q_{ii},$$

y

$$\begin{aligned} Var_{Bal}^*(\hat{s}_{Bal}^*) &= \frac{1}{4n^4 B} \left[-2B \sum_{j=1}^n q_{jj}^2 \right. \\ &\quad \left. + \frac{2}{n^2 B^2} \left(\sum_{j=1}^n q_{jj} \right)^2 + \frac{2(n-1)}{n} \sum_{j=1}^n \sum_{k=1}^n q_{jk}^2 \right]. \quad (5.12) \end{aligned}$$

Comparando las varianzas de los estimadores \hat{s}_{MC}^* y \hat{s}_{Bal}^* bajo el método bootstrap usual y el método balanceado podemos ver que el término líder en (5.10) es más grande que en (5.12) debido a que el término dependiente de los l_j en \hat{s}_B^* es eliminado por \hat{s}_{Bal}^* en (5.11). Sin embargo, las medias del estimador del sesgo bajo los dos métodos son aproximadamente iguales.

Nota 21 Las notaciones como E_{MC}^* (Var_{MC}^*) y E_{Bal}^* (Var_{Bal}^*) se refieren a la esperanza (varianza) condicional dada la muestra original bajo el método de simulación Monte Carlo y el método balanceado respectivamente.

Ejemplo 5.1 Consideremos la muestra de tamaño $n = 10$,

9.6, 10.4, 13, 15, 16.6, 17.2, 17.3, 21.8, 24, 33.8.

Sea $\hat{\theta}_n = \frac{1}{n} \sum_i (X_i - \bar{X}_n)^2$ que es exactamente una función cuadrática. Para $B = 19$ se ha obtenido:

$$\text{Var}_{MC}^*(\hat{s}_{MC}^*) = 14.31 \text{ y } \text{Var}_{Bal}^*(\hat{s}_{Bal}^*) = 1.98,$$

lo cual representa una gran reducción en el error de simulación para la estimación del sesgo. Aunque, en este caso es una situación muy favorable.

HALL (1989a) consideró el caso $\hat{\theta}_n = g(\bar{X}_n)$ con $g : \mathbf{R}^d \rightarrow \mathbf{R}$ una función regular. Bajo ciertas condiciones sobre los momentos de F , dicho autor probó que

$$ECM_{MC}^*(\hat{s}_{MC}^*) = O_p\left(\frac{1}{nB}\right) \text{ y } ECM_{Bal}^*(\hat{s}_{Bal}^*) = O_p\left(\frac{1}{n^2B}\right),$$

donde ECM es el error cuadrático medio. En conclusión, usando el método Balanceado, el orden del ECM se ha reducido de $(nB)^{-1}$ a $(n^2B)^{-1}$.

Para la estimación de la varianza de $\hat{\theta}_n = g(\bar{X}_n)$, HALL (1990) mostró, bajo ciertas condiciones de regularidad sobre g y la distribución de interés F , que

$$n^2B \cdot ECM_{MC}^*(\hat{v}_{MC}^*) \rightarrow^p 2[g'(\bar{X}_n)]^4[\text{var}(X_1)]^2,$$

y

$$n^2B \cdot ECM_{Bal}^*(\hat{v}_{Bal}^*) \rightarrow^p 2[g'(\bar{X}_n)]^4[\text{var}(X_1)]^2.$$

En conclusión, los dos métodos de simulación MC y BB son asintóticamente equivalentes en términos del error cuadrático medio en el caso de la aproximación de la varianza bootstrap.

Ejemplo 5.1 (Continuación) Con los mismos datos que en el ejemplo anterior, se tiene que,

$$\text{Var}_{MC}^*(\hat{v}_{MC}^*) = 2.46, \text{ y } \text{Var}_{Bal}^*(\hat{v}_{Bal}^*) = 2.17.$$

Esto significa que el método BB no mejora mucho el método MC en el caso de la aproximación de la varianza bootstrap.

5.2 El Método centrado

Como hemos visto el método bootstrap balanceado modifica el proceso de simulación Monte Carlo con el fin de reducir el error de simulación. EFRON (1990) introdujo un nuevo método que sigue usando el proceso habitual de simulación Monte Carlo, pero efectuando un ajuste sobre la estimación bootstrap usual resultante. Dicho método también reduce el error de simulación.

5.2.1 Estimación del sesgo

Supongamos que $\hat{\theta} = t(F_n)$ es la estimación con sustitución de θ , y por lo tanto es independiente del orden de la muestra. Sea $\mathcal{X}_n^* = (X_1^*, \dots, X_n^*)$ una muestra bootstrap extraída a partir de $\hat{F} = F_n$ (la distribución empírica asociada a la muestra original). Cualquier réplica bootstrap $\hat{\theta}^* = T_n(\mathcal{X}_n^*)$ puede ser expresada en función del vector de probabilidad \mathbf{P}^* asociado a \mathcal{X}_n^* como

$$\hat{\theta}^* = S_n(\mathbf{P}^*),$$

para cierta función $S_n(\cdot)$. Así, $T_n(\cdot)$ y $S_n(\cdot)$ representan el mismo estadístico, sólo que están definidos sobre diferentes espacios.

Sea \mathbf{P}^0 el vector de dimensión n , cuyas componentes son todas iguales a $\frac{1}{n}$. Muchos estadísticos pueden escribirse como

$$\hat{\theta}_n = S_n(\mathbf{P}^0),$$

con $S_n : \mathbf{R}^n \rightarrow \mathbf{R}$ una función dependiendo de la muestra $\mathcal{X}_n = (X_1, \dots, X_n)$. Por ejemplo $\bar{X}_n = \mathcal{X}_n \mathbf{P}^0 = S_n(\mathbf{P}^0)$, con $S_n(t_1, \dots, t_n) = \sum_{i=1}^n t_i X_i$.

Las B muestras bootstrap $\mathcal{X}_n^{*1}, \dots, \mathcal{X}_n^{*B}$ pueden ser representadas por sus correspondientes B vectores de probabilidad $\mathbf{P}^{*1}, \dots, \mathbf{P}^{*B}$. En términos de estos últimos vectores, el estimador bootstrap usual de Monte Carlo del sesgo de $\hat{\theta}$ se expresa como

$$\hat{s}_{MC}^* = \frac{1}{B} \sum_{b=1}^B S_n(\mathbf{P}^{*b}) - S_n(\mathbf{P}^0). \quad (5.13)$$

Bajo las simulaciones de Monte Carlo la esperanza teórica del vector \mathbf{P}^* es \mathbf{P}^0 y los vectores de remuestreo $\mathbf{P}^{*1}, \dots, \mathbf{P}^{*B}$ tienen la media

$$\bar{\mathbf{P}}^* = \frac{1}{B} \sum_{b=1}^B \mathbf{P}^{*b},$$

que es el centro geométrico de los B vectores. A menudo ocurre que $\bar{\mathbf{P}}^* \neq \mathbf{P}^0$.

Para corregir este hecho, EFRON (1990) propone modificar el estimador usual (5.13) al estimador centrado

$$\hat{s}_{Cent}^* = \frac{1}{B} \sum_{b=1}^B S_n(\mathbf{P}^{*b}) - S_n(\bar{\mathbf{P}}^*), \quad (5.14)$$

La estimación (5.14) es la aproximación bootstrap del sesgo mediante *el método centrado*.

Para ilustrar la ventaja que aporta \hat{s}_{Cent}^* sobre \hat{s}_{MC}^* consideremos, en principio, un estadístico lineal $\hat{\theta}$. Es decir que la función S_n es lineal, en el sentido:

$$S_n(\mathbf{P}^*) = S_0 + (\mathbf{P}^* - \mathbf{P}^0)' \mathbf{U}, \quad \mathbf{P}^* \in \mathbf{R}^n,$$

donde S_0 y los componentes del vector \mathbf{U} de dimensión n son funciones de los datos X_1, \dots, X_n . La última expresión es simplemente el desarrollo de Taylor de $S_n(\cdot)$ en un entorno de \mathbf{P}^0 y entonces $S_0 = S_n(\mathbf{P}^0) = \hat{\theta}_n$. La estimación bootstrap ideal del sesgo, en este caso, es entonces

$$\hat{s} = E^* (S_n(\mathbf{P}^*)) - \hat{\theta} = 0$$

La estimación usual de Monte Carlo en (5.13) da

$$\begin{aligned} \hat{s}_{MC}^* &= \frac{1}{B} \sum_{b=1}^B S_n(\mathbf{P}^{*b}) - \hat{\theta} = \frac{1}{B} \sum_{b=1}^B [\hat{\theta} + (\mathbf{P}^{*b} - \mathbf{P}^0)' \mathbf{U}] - \hat{\theta} \\ &= (\bar{\mathbf{P}}^* - \mathbf{P}^0)' \mathbf{U}, \end{aligned}$$

que es, en general, no nulo. Sin embargo, un cálculo similar nos muestra que $\hat{s}_{Cent}^* = 0 = \hat{s}$. Esto significa que \hat{s}_{Cent}^* aproxima al sesgo ideal \hat{s} mejor que \hat{s}_{MC}^* .

Supongamos, ahora, que $S_n(\cdot)$ es una función cuadrática, i.e.,

$$S_n(\mathbf{P}^*) = S_0 + (\mathbf{P}^* - \mathbf{P}^0)' \mathbf{U} + \frac{1}{2} (\mathbf{P}^* - \mathbf{P}^0)' \mathbf{V} (\mathbf{P}^* - \mathbf{P}^0), \quad (5.15)$$

donde \mathbf{U} es un vector de dimensión n satisfaciendo $\sum_{i=1}^n U_i = 0$, y \mathbf{V} es una matriz simétrica verificando $\sum_{i=1}^n V_{ij} = \sum_{j=1}^n V_{ij} = 0$. Para dicho estadístico podemos calcular teóricamente la estimación bootstrap ideal \hat{s} , obteniéndose

$$\hat{s} = E^* [S_n(\mathbf{P}^*)] - \hat{\theta} = \frac{1}{2} \text{tr} \mathbf{V} \Gamma = \frac{1}{2n} \text{tr} \frac{\mathbf{V}}{n}, \quad (5.16)$$

donde Γ es la matriz de covarianza del vector de remuestreo \mathbf{P}^* . Los cálculos multinomiales usuales dan

$$\Gamma = \frac{\mathbf{I}_n}{n^2} - \frac{\mathbf{1}_n \mathbf{1}_n'}{n^3},$$

donde \mathbf{I}_n es la matriz identidad, y $\mathbf{1}_n = (1, \dots, 1)'$. Definimos la matriz

$$\bar{\Gamma} = \frac{1}{B} \sum_{b=1}^B (\mathbf{P}^{*b} - \bar{\mathbf{P}}^*) (\mathbf{P}^{*b} - \bar{\mathbf{P}}^*)'.$$

Entonces, $[B/(B-1)]\bar{\Gamma}$ es la estimación insesgada usual de Γ . Podemos verificar que:

$$\begin{aligned}\hat{s}_{MC}^* &= \frac{1}{2}\text{tr}\mathbf{V}\bar{\Gamma} + (\bar{\mathbf{P}}^* - \mathbf{P}^0)' \mathbf{U} + \frac{1}{2}(\bar{\mathbf{P}}^* - \mathbf{P}^0) \mathbf{V}(\bar{\mathbf{P}}^* - \mathbf{P}^0), \text{ y} \\ \hat{s}_{Cent}^* &= \frac{1}{2}\text{tr}\mathbf{V}\bar{\Gamma}.\end{aligned}$$

Nótese que \hat{s}_{Cent}^* es una estimación razonable de \hat{s} en (5.16), sin embargo, \hat{s}_{MC}^* añade varios términos innecesarios que pueden aumentar la varianza de la estimación. Con $B = Cn$, y C una constante positiva, EFRON (1990) probó que, cuando $n \rightarrow \infty$,

$$\hat{s}_{Cent}^* - \hat{s} = O_p\left(\frac{1}{n^{3/2}}\right), \text{ y } \hat{s}_{MC}^* - \hat{s} = O_p\left(\frac{1}{n}\right).$$

Nota 22 Para la estimación del sesgo, el Método Balanceado y el Método Centrado actúan de la misma forma pero con diferentes procedimientos. El vector de remuestreo \mathbf{P}^* tiene esperanza teórica $E^*(\mathbf{P}^*) = \mathbf{P}^0$. La muestra de vectores, $\mathbf{P}^{*1}, \dots, \mathbf{P}^{*B}$, obtenida mediante las simulaciones de Monte Carlo, tiene media $\bar{\mathbf{P}}^*$, que es a menudo diferente de \mathbf{P}^0 . El método centrado de Efron considera el hecho de que $\bar{\mathbf{P}}^* \neq \mathbf{P}^0$. El método bootstrap balanceado usa otra técnica para corregir el hecho de que $\bar{\mathbf{P}}^* \neq \mathbf{P}^0$, que consiste en la elección de $\mathbf{P}^{*1}, \dots, \mathbf{P}^{*B}$ de modo que $\bar{\mathbf{P}}^* = \mathbf{P}^0$.

Para una función de la media muestral, HALL (1989a) probó que

$$\text{ECM}_{MC}^*(\hat{s}_{MC}^*) = O_p\left(\frac{1}{nB}\right), \text{ y } \text{ECM}_{MC}^*(\hat{s}_{Cent}^*) = O_p\left(\frac{1}{n^2B}\right).$$

En conclusión, el método centrado de Efron, al igual que el método balanceado, es mejor que el método bootstrap usual para la estimación del sesgo de un estimador.

5.2.2 Estimación de la varianza

EFRON (1990) también extendió el método centrado a la estimación bootstrap de la varianza y de los cuantiles. Dicha extensión no es evidente, y se basa en la descomposición ortogonal (ANOVA) de $S_n(\mathbf{P}^*)$, ver EFRON (1982),

$$S_n(\mathbf{P}^*) = \mu + \alpha(\mathbf{P}^*) + \beta(\mathbf{P}^*), \quad (5.17)$$

donde $\mu = E^*[S_n(\mathbf{P}^*)]$, $\alpha(\mathbf{P}^*) = \alpha' \mathbf{P}^*$, con $\alpha = (\alpha_1, \dots, \alpha_n)'$ donde

$$\alpha_i = n\{E^*[S(\mathbf{P}^*) \mid X_1^* = X_i] - \mu\},$$

y $\beta(\mathbf{P}^*)$ el término residual. La descomposición (5.17) tiene varias propiedades que no permiten expresar la varianza como

$$\hat{v} = \text{Var}^*[S_n(\mathbf{P}^*)] = \text{Var}^*[\alpha(\mathbf{P}^*)] + \text{Var}^*[\beta(\mathbf{P}^*)]. \quad (5.18)$$

Para más detalles sobre la descomposición ANOVA (5.17) se puede consultar a EFRON & STEIN (1981) y EFRON (1982).

A partir de (5.18) el estimador bootstrap ideal de la varianza se expresa como

$$\begin{aligned} \hat{v} &= \alpha' \text{Var}^*(\mathbf{P}^*) \alpha + V_\beta \\ &= \frac{\alpha' \alpha}{n^2} + V_\beta, \end{aligned} \quad (5.19)$$

con $V_\beta = \text{Var}^*[\beta(\mathbf{P}^*)]$.

El método centrado de Efron para la estimación de la varianza consiste en estimar el vector α y la varianza V_β mediante ciertos estimadores razonables, digamos, $\hat{\alpha}$ y \hat{V}_β y luego insertarlos en (5.19). Para ello, EFRON (1990) sugiere simular B muestras bootstrap $\mathcal{X}_n^{*1}, \dots, \mathcal{X}_n^{*B}$ y calcular sus correspondientes vectores de remuestreo $\mathbf{P}^{*1}, \dots, \mathbf{P}^{*B}$ y luego usar el método de mínimos cuadrados que consiste en minimizar

$$\sum_{b=1}^B (S^{*b} - \mu - \alpha' \mathbf{P}^{*b})^2 \quad (5.20)$$

donde $S^{*b} = S_n(\mathbf{P}^{*b})$, $b = 1, \dots, B$.

Sea $\bar{S} = B^{-1} \sum_{b=1}^B S_n(\mathbf{P}^{*b})$ y definimos

$$\mathbf{T} = (S^{*1} - \bar{S}, \dots, S^{*B} - \bar{S})' \quad \text{y} \quad \mathbf{Q} = (\mathbf{P}^{*1} - \bar{\mathbf{P}}^*, \dots, \mathbf{P}^{*B} - \bar{\mathbf{P}}^*).$$

Entonces \mathbf{T} es un vector de dimensión B y \mathbf{Q} es una matriz de dimensión $n \times B$. Se puede probar que la solución de mínimos cuadrados que minimiza (5.20) sujeta a $\alpha' \mathbf{P}^0 = 0$, es

$$\hat{\alpha}' = (\mathbf{Q}\mathbf{Q}' + \mathbf{1}_n \mathbf{1}_n')^{-1} \mathbf{Q}\mathbf{T}.$$

Con esta estimación de α , EFRON (1990) probó que la aproximación bootstrap usual de \hat{v} se expresa como

$$\hat{v}_{MC}^* = \hat{\alpha}' \hat{\Sigma} \alpha + \hat{V}_\beta, \quad (5.21)$$

y la aproximación mediante el método centrado

$$\hat{v}_{Cent}^* = \frac{\hat{\alpha}' \hat{\alpha}}{n^2} + \hat{V}_\beta,$$

donde $\hat{\Sigma} = \frac{1}{B} \mathbf{Q}\mathbf{Q}'$, y \hat{V}_β es la aproximación bootstrap de V_β dada por

$$\hat{V}_\beta = \frac{1}{B-1} \sum_{b=1}^B \hat{\beta}^{*b2},$$

donde

$$\hat{\beta}^{*b} = S^{*b} - \hat{\mu} - \hat{\alpha}' \mathbf{P}^{*b} \text{ y } \hat{\mu} = \bar{S}^*.$$

Nótese que $\hat{\Sigma}$ es una estimación de $\text{Var}^*(\mathbf{P}^*)$ que no es necesaria en (5.21), ya que, \hat{v} en (5.19) no depende de $\text{Var}^*(\mathbf{P}^*)$. Por lo que, intuitivamente, se ve que \hat{v}_{Cent}^* es más razonable que \hat{v}_{MC}^* para aproximar a la varianza ideal \hat{v} .

Definimos:

$$R^2 = \frac{\text{Var}^*[\alpha(\mathbf{P}^*)]}{\text{Var}^*[S_n(\mathbf{P}^*)]}, \text{ y } \rho = \frac{\text{Var}^*(\hat{v}_{MC}^*)}{\text{Var}^*(\hat{v}_{Cent}^*)}.$$

Mediante muchos experimentos, EFRON (1990) probó que, cuanto más se acerca R^2 a 1, más grande es ρ . En conclusión, si la función S_n es aproximadamente lineal, entonces el método centrado mejora sustancialmente al método bootstrap usual.

5.3 Bootstrap Lineal

El método lineal es un caso particular de un método más general llamado Método Control, del inglés "The Control Method". La idea de este último método es expresar al estimador de interés $\hat{\theta}^*$ como la suma de un estadístico \hat{L}^* altamente correlado con $\hat{\theta}^*$, y la diferencia, D^* , entre $\hat{\theta}^*$ y \hat{L}^* . Es decir

$$\hat{\theta}_n^* = \hat{L}^* + D^*, \text{ donde } D^* = \hat{\theta}^* - \hat{L}^*.$$

El estadístico \hat{L}^* se elige de modo que sus características pueden ser calculadas con exactitud. De esta manera la estimación de las propiedades de $\hat{\theta}^*$ necesita sólo la estimación de las de la diferencia D^* . Más concretamente, supongamos que $\hat{\mu}_L = E^*(\hat{L}^*)$ y $\hat{v}_L = \text{Var}^*(\hat{L}^*)$ son conocidas, entonces se tiene que

$$\hat{s} = E^*(\hat{\theta}_n^*) - \hat{\theta} = \hat{\mu}_L - \hat{\theta} + E^*(D^*), \quad (5.22)$$

y

$$\hat{v} = \text{Var}^*(\hat{\theta}^*) = \hat{v}_L + \text{Var}^*(D^*) + 2\text{Cov}^*(\hat{L}^*, D^*). \quad (5.23)$$

Por tanto, para aproximar el sesgo \hat{s} y la varianza \hat{v} de $\hat{\theta}^*$ basta aproximar las cantidades dependientes de D^* en (5.22) y (5.23). Dadas B muestras bootstrap

$\mathcal{X}_n^{*b} = (X_1^{*b}, \dots, X_n^{*b})$, $b = 1, \dots, B$, se calculan las B replicasiones $\hat{\theta}^{*b}$, \hat{L}^{*b} y D^{*b} , $b = 1, \dots, B$. Luego, se aproxima a \hat{s} con

$$\hat{s}_{Cont}^* = \hat{\mu}_L - \hat{\theta} + \bar{D}^*, \quad (5.24)$$

y a la varianza \hat{v} con

$$\hat{v}_{Cont}^* = \hat{v}_L + \frac{1}{B-1} \sum_{b=1}^B \left[(D^{*b} - \bar{D}^*)^2 + 2(D^{*b} - \bar{D}^*)(\hat{L}^{*b} - \bar{L}^*) \right], \quad (5.25)$$

donde $\bar{D}^* = B^{-1} \sum_{b=1}^B D^{*b}$ y $\bar{L}^* = B^{-1} \sum_{b=1}^B \hat{L}^{*b}$. El hecho de que \hat{L}^* sea altamente correlado con $\hat{\theta}^*$ significa que \hat{v}_L forma una gran parte de la varianza \hat{v} por lo que el término líder en (5.25) se ha obtenido sin error de simulación.

En la práctica, la elección más razonable de la variable control \hat{L}^* es la aproximación lineal de $\hat{\theta}^*$. Supongamos que el estimador de interés, $\hat{\theta}$, del parámetro θ , puede expresarse de la forma:

$$\hat{\theta}_n = \theta + \frac{1}{n} \sum_{i=1}^n L(X_i, F) + D, \quad (5.26)$$

con $L(\cdot, F)$ una función verificando $\int L(x, F) dF(x) = 0$ para cierta f.d.D. F , y D_n un término residual. La expresión (5.26) es la misma que (5.7) pero considerando solamente la aproximación lineal y entonces D_n incluye el término cuadrático y el resto del desarrollo en (5.7). Definimos

$$\hat{L} = \theta + \frac{1}{n} \sum_{i=1}^n L(X_i, F),$$

es la componente lineal de $\hat{\theta}$. Entonces $D = \hat{\theta} - \hat{L}$.

Supongamos que $L(x, F)$ está bien definida cuando se sustituye F por F_n . Sea, entonces,

$$\hat{L}^* = \hat{\theta} + \frac{1}{n} \sum_{i=1}^n L(X_i^*, F_n), \text{ y } D_n^* = \hat{\theta}^* - \hat{L}^*,$$

los análogos bootstrap de \hat{L} y D respectivamente. Expresando a \hat{L}^* en términos de los N_j^* , definidos en una sección anterior, se obtiene

$$\hat{L}^* = \hat{\theta} + \frac{1}{n} \sum_{j=1}^n N_j^* l_j.$$

con $l_i = L(X_i, F_n)$ (ver *Capítulo 2*). Y puesto que $n^{-1} \sum_{i=1}^n l_i = 0$,

$$\hat{\mu}_L = E^*[\hat{L}^*] = \hat{\theta} \text{ y } \hat{v}_L = \text{Var}^*[\hat{L}^*] = \frac{1}{n^2} \sum_{j=1}^n l_j^2.$$

Por lo tanto, a partir de (5.22) y (5.23) el sesgo y la varianza bootstrap ideales de $\hat{\theta}^*$ tienen la expresión

$$\hat{s} = E^*(D^*) \quad (5.27)$$

y

$$\hat{v} = \frac{1}{n^2} \sum_{i=1}^n l_i^2 + \text{Var}^*(D^*) + 2\text{Cov}^*(D^*, \hat{L}^*). \quad (5.28)$$

Y entonces, a partir de (5.24) y (5.25), obtendremos la siguiente aproximación correspondiente al Método *bootstrap lineal*

$$\hat{s}_{Lin}^* = \bar{D}^*,$$

y

$$\hat{v}_{Lin}^* = \frac{1}{n^2} \sum_{i=1}^n l_i^2 + \frac{1}{B-1} \sum_{b=1}^B \left[(D^{*b} - \bar{D}^*)^2 + 2(\hat{L}^{*b} - \bar{L}^*)(D^{*b} - \bar{D}^*) \right].$$

Notese que esta última aproximación es una mixtura del método bootstrap usual y del método delta. La aproximación \hat{v}_{Lin}^* es considerada como una mejoría al método delta (ver *Capítulo 2*).

Ejemplo 5.2 Sea X_1, \dots, X_n una m.a.s. con f.d.D. F de media μ y varianza σ^2 . Supongamos que $\hat{\theta}_n = g(\bar{X}_n)$, con g una función regular. Entonces, $\hat{\theta}_n$ puede escribirse

$$\hat{\theta}_n = \theta + g'(\mu)(\bar{X}_n - \mu) + R_n = \theta + \frac{1}{n} \sum_{i=1}^n g'(\mu)(X_i - \mu) + R_n.$$

Aquí $L(x, F) = g'(\mu)(x - \mu)$. La varianza bootstrap de la componente lineal de $\hat{\theta}_n$, \hat{L}_n , es

$$\hat{v}_L = \text{var}_{F_n}(\hat{L}_n^*) = g'^2(\mu) \frac{\hat{\sigma}^2}{n},$$

con $\hat{\sigma}^2$ la varianza muestral. Entonces \hat{v}_L se calcula sin simulaciones de Monte Carlo.

HALL (1989a), consideró el caso $\hat{\theta}_n = g(\bar{X}_n)$, y probó que el método BL es mejor que el método bootstrap usual de MC en términos del error cuadrático medio. Además el rendimiento del método BL puede ser mejorado suplementariamente usando el método bootstrap balanceado para aproximar los términos relacionados con D_n^* .

5.4 Método Bootstrap Antitético

Sean $\hat{\theta}_1$ y $\hat{\theta}_2$ dos estimadores de θ basados en una muestra de tamaño n . La idea de este método consiste en que si $\hat{\theta}_1$ y $\hat{\theta}_2$ tienen la misma varianza y están negativamente correladas, entonces la varianza del estimador

$$\hat{\theta} = \frac{1}{2}(\hat{\theta}_1 + \hat{\theta}_2),$$

es inferior a la mitad de la varianza de $\hat{\theta}_1$ y $\hat{\theta}_2$.

Con el fin de mejorar la aproximación bootstrap usual basada en el método de simulación de Monte Carlo, HALL (1989b) utiliza dicha técnica. Sea $\mathcal{X}^{*b} = (X_1^{*b}, \dots, X_n^{*b})$, $b = 1, \dots, B$, B muestras bootstrap extraídas a partir de F_n . Supongamos que queremos aproximar el estimador bootstrap (ideal) del sesgo, \hat{s} (ver (5.2)). Sea \hat{s}_{MC}^* la aproximación Monte Carlo de \hat{s} como en (5.3). Podemos usar las mismas B muestras bootstrap para encontrar otra aproximación de Monte Carlo, digamos \tilde{s}_{MC}^* , de \hat{s} que tiene la misma varianza que \hat{s}_{MC}^* pero negativamente correlada con \hat{s}_{MC}^* . Nótese que $X_i^{*b} = X_{u(b,i)}$, $i = 1, \dots, n$ con $\{u(b,i)\}_{i=1,n}$ son independientes uniformemente distribuidos sobre $\{1, \dots, n\}$ para cada $b = 1, \dots, B$. Definimos unas B nuevas muestras bootstrap como sigue:

$$\tilde{\mathcal{X}}^{*b} = \left(\tilde{X}_{\pi(u(b,1))}^{*b}, \dots, \tilde{X}_{\pi(u(b,n))}^{*b} \right), \quad b = 1, \dots, B.$$

donde π es una permutación del conjunto $\{1, \dots, n\}$. Notamos por \tilde{s}_{MC}^* la aproximación Monte Carlo de \hat{s} mediante $\tilde{\mathcal{X}}^{*1}, \dots, \tilde{\mathcal{X}}^{*B}$. Definimos el estimador

$$\hat{s}_{Antit}^* = \frac{1}{2}(\hat{s}_{MC}^* + \tilde{s}_{MC}^*).$$

La cuestión es como elegir la permutación π de modo que \hat{s}_{MC}^* y \tilde{s}_{MC}^* tengan la misma varianza pero negativamente correlados.

HALL (1989B), bajo algunas condiciones sobre el estadístico de interés $\hat{\theta}$, propuso un procedimiento para construir una permutación π verificando las condiciones citadas. Para más detalles sobre el método bootstrap antitético se puede consultar HALL (1989) y DO (1992).

Capítulo 6

MÉTODOS ORTOGONALES: BOOTSTRAP PARAMÉTRICO

En este capítulo y en el siguiente desarrollamos un método para aproximar la varianza y el sesgo bootstrap ideales de un cierto estimador T_n . Este método está basado en desarrollos ortogonales y técnicas de regresión múltiple. En la literatura existen métodos que se basan sobre expresiones ortogonales, como el método ofrecido por EFRON (1990), que se basa en la descomposición ANOVA del estimador T_n . Dicho método, descompone T_n en dos partes ortogonales $T_n = L_n + r_n$ donde L_n es la parte lineal del estadístico y r_n es el resto del desarrollo. En este caso si las varianzas de T_n , L_n y r_n son respectivamente v , γ y δ , entonces se verifica $v = \delta + \gamma$. Este método da buenos resultados si T_n es aproximadamente lineal. Es decir, cuando la parte lineal L_n es la que controla la varianza de T_n . Formalmente, si $R = \gamma/v$ entonces el método es favorable si R es muy próximo a 1.

Nuestro método emplea técnicas diferentes. La clave es la consideración de $T_n(\cdot)$ como una función de un espacio de Hilbert adecuado \mathcal{L}_n , en donde T_n admite un cierto desarrollo de la forma $T_n(\cdot) = \sum_k a_k \psi_k(\cdot)$ con $\{\psi_k(\cdot)\}_k$ una base de \mathcal{L}_n y $\{a_k\}_k$ los coeficientes de Fourier. En este caso se verifica que $v = \sum_k a_k^2 \|\psi_k\|_{\mathcal{L}}^2$ con $\|\cdot\|_{\mathcal{L}}$ la norma natural de \mathcal{L}_n . Este método tiene la ventaja de poder considerar no sólo la parte lineal o cuadrática del estimador, sino, que podemos considerar varios términos en el desarrollo hasta que $R = \gamma/v$ es próximo a 1, donde γ sea la varianza de los términos del desarrollo considerados.

6.1 Caso unidimensional

Consideremos una muestra aleatoria simple $\mathcal{X}_n = (X_1, \dots, X_n)$ de tamaño n procedente de una población con distribución $F = F_\theta(x)$ con parámetro desconocido $\theta \in \Theta \subset \mathbb{R}$, y densidad $f_\theta(x)$ respecto a una medida σ -finita ν (medida de Lebesgue o Cardinal). Sea $S_n = S_n(X_1, \dots, X_n)$ un estadístico para la familia F_θ . Para cada $\theta \in \Theta$, definimos los espacios

$$\mathcal{L}_{n,\theta}^2 = \{T : \mathbb{R} \rightarrow \mathbb{R} \text{ tal que } E_\theta[|T(S_n)|^2] < \infty\}, \quad (6.1)$$

en los cuales consideremos que dos funciones T_1 y T_2 son iguales si lo son casi por todo ν (i.e. si $\nu\{T_1 \neq T_2\} = 0$). Evidentemente, los espacios $\mathcal{L}_{n,\theta}^2$ son espacios de Banach con norma

$$\|T\|_\theta = E_\theta[|T(S_n)|^2]^{1/2}.$$

Esta norma está inducida por el producto escalar

$$\langle T_1, T_2 \rangle_\theta = E_\theta[T_1(S_n)T_2(S_n)].$$

ya que $(\mathcal{L}_{n,\theta}^2, \langle \cdot, \cdot \rangle_\theta)$ es también un espacio de Hilbert separable, ver KINGMAN & TAYLOR (1966). Entonces en $\mathcal{L}_{n,\theta}^2$ existe un sistema ortogonal de funciones $\{P_{k,n}(\cdot, \theta)\}_{k \geq 0}$. Sin pérdida de la generalidad, consideremos que $P_{0,n}(\cdot, \theta) \equiv 1$. Entonces cada función en $\mathcal{L}_{n,\theta}^2$ admite un desarrollo de la forma

$$T(s) = \sum_{k \geq 0} a_{k,n}(\theta) P_{k,n}(s, \theta), \quad (6.2)$$

con $a_{k,n}(\theta) = \langle T_n, P_{k,n} \rangle_\theta / \|P_{k,n}\|_\theta^2$. El desarrollo (6.2) es convergente en el sentido cuadrático.

Si $\hat{\xi}_n = T_n(S_n)$ es un estimador de una cierta función $\xi = h(\theta)$ del parámetro θ con $v = \text{Var}_\theta(T_n(S_n)) < \infty$, entonces T_n es un elemento de $\mathcal{L}_{n,\theta}^2$ y por lo tanto tiene un desarrollo como en (6.2).

Uno de los objetivos de la inferencia estadística es la estimación de la varianza $v = \text{Var}_\theta(\hat{\xi}_n)$ y el sesgo $s = \text{Sesgo}_\theta(\hat{\xi})$. A partir del desarrollo (6.2) podemos probar fácilmente que

$$v = \sum_{k \geq 1} a_{k,n}^2(\theta) \|P_{k,n}\|_\theta^2 \text{ y } s = a_{0,n}(\theta) - h(\theta). \quad (6.3)$$

Para una clase bastante amplia de familias paramétricas, el sistema de funciones ortogonales $\{P_{k,n}(\cdot, \theta)\}_k$ es bien conocido, o bien, puede ser obtenido ortogonalizando un sistema completo usando el método de Gram-Schmidt, por ejemplo. Sin embargo,

los coeficientes $\{a_{k,n}(\theta)\}_k$ son en general desconocidos y en muchas situaciones son muy difíciles de calcular. Como consecuencia, la varianza v y el sesgo s en (6.3) son difíciles de calcular también usando este método.

Ejemplo 6.1 Consideremos la familia exponencial cuya densidad es de la forma $f(x, \theta) = \exp\{x\theta - \psi(\theta)\}$ con respecto a la medida de Lebesgue o Cardinal.

Consideremos el estadístico $S_n = \bar{X}_n$ la media muestral, y $\mu = E_\theta(X)$ la media de la familia. Hemos visto en el Capítulo 3 que para la subclase de dicha familia cuya varianza es cuadrática, es decir, de la forma $v = v_2\mu^2 + v_1\mu + v_0$, el sistema de funciones ortogonales $\{P_{k,n}(\cdot, \mu)\}_k$ es bien conocido, y que dicha subclase contiene a la mayoría de las distribuciones unidimensionales usadas con frecuencia en la práctica. Entre ellas están las distribuciones Normal, Gamma, Poisson, Binomial y Binomial Negativa. Consideremos la estimación de una función $h(\mu)$ por un estimador insesgado $T_n(\bar{X}_n)$ cuya varianza es finita. En este caso, $T_n(\bar{X}_n)$ admite un desarrollo ortogonal como en (6.2) con

$$a_{k,n}(\mu) = \frac{h^{(k)}(\mu)}{v_{k,n}}, \text{ donde } v_{k,n} = k! \prod_{j=1}^{k-1} \left(1 + j \frac{v_2}{n}\right),$$

y $\{P_{k,n}(\cdot, \mu)\}_k$ es un sistema de polinomios mónicos ortogonales respecto a la densidad de \bar{X}_n y cuya forma explícita está dada en el Capítulo 3. Para más detalles se puede consultar MORRIS (1982, 1983). En el caso más general de familias exponenciales donde la varianza no es necesariamente cuadrática, los coeficientes de Fourier $a_{k,n}$ respecto a un sistema de polinomios ortogonales mónicos, son combinación lineal de las derivadas de h . Además, dicho sistema de polinomios ortogonales puede ser obtenido ortogonalizando el conjunto $\{1, s, s^2, \dots\}$. Para más detalles ver ABBY & DAVID (1970).

Si pudiéramos estimar los coeficientes $a_{k,n}(\theta)$ por $\widehat{a_{k,n}(\theta)}$ entonces una estimación natural de la varianza v y del sesgo s serían

$$\hat{v} = \sum_{k \geq 1} \widehat{a_{k,n}(\theta)}^2 \|P_{k,n}\|_{\hat{\theta}}^2 \text{ y } \hat{s} = \widehat{a_{0,n}(\theta)} - h(\hat{\theta}), \quad (6.4)$$

donde $\hat{\theta}$ es un estimador consistente de θ , por ejemplo, el estimador de Máxima Verosimilitud. Con el fin de estimar los coeficientes $a_{k,n}$ proponemos la construcción de un problema de regresión múltiple con parámetros desconocidos $\beta_0, \beta_1, \dots, \beta_m, \sigma^2$ y variables explicativas $P_{1,n}, \dots, P_{m,n}$ para cierto entero $m \geq 1$ como se explica más adelante. Luego, veremos que la estimación de mínimos cuadrados $\{\hat{\beta}_i\}_i$ es una buena estimación para $a_{i,n}(\theta)$, $i = 1, \dots, m$, y $\hat{\sigma}^2$ es una buena estimación de la varianza σ^2 del error en el desarrollo.

6.1.1 Estimación bootstrap

Sea $\mathcal{X}_n^* = (X_1^*, \dots, X_n^*)$ una muestra aleatoria simple procedente de la distribución estimada $\hat{F} = F_{\hat{\theta}}$, con $\hat{\theta}$ un estimador consistente de θ , por ejemplo el estimador de máxima verosimilitud, y sea $S_n^* = S_n(X_1^*, \dots, X_n^*)$. Al igual que en (6.2) se sigue verificando que

$$T_n(S_n^*) = \sum_{k \geq 0} a_{k,n}(\hat{\theta}) P_{k,n}(S_n^*, \hat{\theta}). \quad (6.5)$$

Entonces según (6.3), la estimación bootstrap ideal de v y s son respectivamente

$$\hat{v} = \sum_{k \geq 1} a_{k,n}^2(\hat{\theta}) \|P_{k,n}\|_{\hat{\theta}}^2 \text{ y } \hat{s} = a_{0,n}(\hat{\theta}) - h(\hat{\theta}). \quad (6.6)$$

Aunque hemos sustituido θ por $\hat{\theta}$ en (6.3) con el fin de obtener (6.6) nuestro problema sigue sin resolverse. Es decir que las estimaciones bootstrap \hat{v} y \hat{s} siguen desconocidas, y eso es debido a que la expresión explícita de los coeficientes $a_{k,n}(\cdot)$ no es conocida.

Nuestro objetivo en este capítulo es aproximar a los $a_{k,n}(\hat{\theta})$ con buena precisión, y como consecuencia obtendremos una buena aproximación a \hat{v} y \hat{s} . Veremos que nuestras aproximaciones son mucho más precisas que las aproximaciones bootstrap usuales.

Para $m \geq 1$ y $\beta = (\beta_0, \dots, \beta_m)$, tendremos el error

$$\begin{aligned} r_m^*(\beta) &= \|T_n(S_n^*) - \beta_0 - \sum_{k=1}^m \beta_k P_{k,n}(S_n^*, \hat{\theta})\|_{\hat{\theta}}^2 \\ &= \left\| \sum_{i=0}^m (a_{i,n}(\hat{\theta}) - \beta_i) P_{i,n}(S_n^*, \hat{\theta}) + \sum_{k>m} a_{k,n}(\hat{\theta}) P_{k,n}(S_n^*, \hat{\theta}) \right\|_{\hat{\theta}}^2 \\ &= \sum_{i=0}^m (a_{i,n}(\hat{\theta}) - \beta_i)^2 \|P_{i,n}\|_{\hat{\theta}}^2 + \sum_{j \geq m+1} a_{j,n}^2(\hat{\theta}) \|P_{j,n}\|_{\hat{\theta}}^2. \end{aligned} \quad (6.7)$$

Entonces a partir de (6.7) podemos ver que $r_m^*(\beta)$ es minimizado por la elección $\beta_i = a_{i,n}(\hat{\theta})$, $i = 0, \dots, m$. Intuitivamente, podemos pensar en aproximar a los $\beta_i = a_{i,n}(\hat{\theta})$ haciendo una regresión de $T_n(S_n^*)$ sobre $P_{1,n}(S_n^*, \hat{\theta}), \dots, P_{m,n}(S_n^*, \hat{\theta})$. Esto significa que la solución de mínimos cuadrados $\hat{\beta}$ es la mejor estimación del vector $\beta = (a_{0,n}(\hat{\theta}), \dots, a_{m,n}(\hat{\theta}))$ que minimiza el error $r_m^*(\beta)$.

Consideremos en lo que sigue los $m+1$ primeros términos del desarrollo (6.5). Por simplicidad denotaremos por $Y^* = T_n(S_n^*)$, $P_i^* = P_{i,n}(S_n^*, \hat{\theta})$, $i = 1, \dots, m$ y

$\beta_i = a_{i,n}(\hat{\theta})$, $i = 0, \dots, m$. Se trata entonces de aproximar a Y^* mediante una función lineal de los P_i^* , $i = 1, \dots, m$, de la forma $f(P_1^*, \dots, P_m^*) = \beta_0 + \sum_{i=1}^m \beta_i P_i^*$. Hacemos, ahora, las suposiciones usuales de la regresión y escribimos:

$$T_n(S_n^*) = \beta_0 + \sum_{k=1}^m \beta_k P_{k,n}(S_n^*, \hat{\theta}) + \varepsilon, \quad (6.8)$$

donde ε es una variable aleatoria no correlada con los $P_{i,n}(S_n^*, \hat{\theta})$, $i = 1, \dots, m$, con media $E_{\hat{F}}(\varepsilon) = 0$ y varianza $var_{\hat{F}}(\varepsilon) = \sigma^2$ finita. En efecto, estas propiedades pueden ser deducidas a partir de la ortogonalidad de los $P_{i,n}(\cdot, \hat{\theta})$ y la expresión (6.8). Entonces, a partir de (6.8) el sesgo y la varianza bootstrap ideales pueden ser expresados respectivamente como

$$\hat{s} = \beta_0 - h(\hat{\theta}) \text{ y } \hat{v} = \sum_{j=1}^m \beta_j^2 \|P_{j,n}\|_{\hat{\theta}}^2 + \sigma^2.$$

Y por lo tanto aproximaciones naturales para \hat{s} y \hat{v} serán respectivamente de la forma

$$\hat{s}_{Ort}^* = \hat{\beta}_0 - h(\hat{\theta}) \text{ y } \hat{v}_{Ort}^* = \sum_{j=1}^m \hat{\beta}_j^2 \|P_{j,n}\|_{\hat{\theta}}^2 + \hat{\sigma}^2,$$

donde $\hat{\beta}_j^*$ y $\hat{\sigma}^{*2}$ son aproximaciones de β_j y σ^2 respectivamente.

Consideremos, ahora, el método de Monte-Carlo. Extraemos B muestras bootstrap $\mathcal{X}_n^{*b} = (X_1^{*b}, \dots, X_n^{*b})$, $b = 1, \dots, B$ a partir de \hat{F} . Calculamos las B réplicas S_n^{*b} y después $\hat{\xi}_n^{*b} = T_n(S_n^{*b})$, de S_n y $\hat{\xi}_n$ respectivamente, correspondientes a cada muestra bootstrap \mathcal{X}_n^{*b} , $b = 1, \dots, B$. A+ partir de (6.8) tenemos para cada una de las B muestra bootstrap

$$T_n(S_n^{*b}) = \beta_0 + \sum_{k=1}^m \beta_k P_{k,n}(S_n^{*b}, \hat{\theta}) + \varepsilon_b, \text{ para } b = 1, \dots, B, \quad (6.9)$$

donde ε_b son realizaciones de la variable aleatoria ε . Puesto que el sistema de funciones ortogonales $\{P_{k,n}(\cdot, \cdot)\}_k$ es conocido, podemos calcular la matriz de diseño, \mathbf{X}_* , de dimensión $B \times (m + 1)$,

$$\mathbf{X}_* = \begin{bmatrix} 1 & P_{1,n}(S_n^{*1}, \hat{\theta}) & \cdots & P_{m,n}(S_n^{*1}, \hat{\theta}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & P_{1,n}(S_n^{*B}, \hat{\theta}) & \cdots & P_{m,n}(S_n^{*B}, \hat{\theta}) \end{bmatrix}. \quad (6.10)$$

Finalmente, (6.9) puede ser expresada de la forma usual

$$\mathbf{y}^* = \mathbf{X}_* \boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (6.11)$$

con $\mathbf{y}^* = (T_n(S_n^{*1}), \dots, T_n(S_n^{*B}))^t$, $\boldsymbol{\beta} = (\beta_0, \dots, \beta_m)^t$, y $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_B)^t$. Estamos frente a un problema de regresión múltiple de $Y^* = T_n(S_n^*)$ sobre $P_1^* = P_{1,n}(S_n^*, \hat{\theta})$, ..., $P_m^* = P_{m,n}(S_n^*, \hat{\theta})$, con parámetros desconocidos $\beta_0, \beta_1, \dots, \beta_m$ y σ^2 . Podemos entonces usar las técnicas usuales de mínimos cuadrados para estimar $\boldsymbol{\beta}$ y σ^2 . Sea $\hat{\boldsymbol{\beta}}^* = (\hat{\beta}_1^*, \dots, \hat{\beta}_m^*)^t$ y $\hat{\sigma}^{*2}$ la estimación de mínimos cuadrados. Es bien conocido que

$$\hat{\boldsymbol{\beta}}^* = (\mathbf{X}_*^t \mathbf{X}_*)^{-1} \mathbf{X}_*^t \mathbf{y}^* \text{ y } \hat{\sigma}^{*2} = \frac{\sum_{b=1}^B \left(y_b^* - \sum_{j=0}^m \hat{\beta}_j^* P_{j,n}(S_n^{*b}, \hat{\theta}) \right)^2}{B - m - 1}. \quad (6.12)$$

Consideremos los siguientes estimadores del sesgo y de la varianza respectivamente

$$\hat{s}_{O_{rt}}^* = \hat{\beta}_0^* - h(\hat{\theta}) \text{ y}, \quad (6.13)$$

$$\hat{v}_{O_{rt}}^* = \sum_{i=1}^m \hat{\beta}_i^{*2} \|P_{i,n}\|_{\hat{\theta}}^2 + \hat{\sigma}^{*2}. \quad (6.14)$$

Como veremos, numéricamente, más adelante, $\hat{s}_{O_{rt}}^*$ y $\hat{v}_{O_{rt}}^*$ aproximan con gran precisión a los estimadores bootstrap ideales \hat{s} y \hat{v} dados en (6.6).

Los estimadores en (6.13) y (6.14) son consistentes. En efecto, consideremos la ecuación normal:

$$\mathbf{X}_*^t \mathbf{y}^* = (\mathbf{X}_*^t \mathbf{X}_*) \hat{\boldsymbol{\beta}}^*. \quad (6.15)$$

Puesto que

$$(\mathbf{X}_*^t \mathbf{y}^*)_j = \sum_{b=1}^B P_{j,n}(S_n^{*b}, \hat{\theta}) T_n(S_n^{*b}) \text{ y } (\mathbf{X}_*^t \mathbf{X}_*)_{i,j} = \sum_{b=1}^B P_{i,n}(S_n^{*b}, \hat{\theta}) P_{j,n}(S_n^{*b}, \hat{\theta}),$$

entonces, según la ley de los grandes números (LGN), tendremos

$$\frac{(\mathbf{X}_*^t \mathbf{y}^*)_j}{B} \xrightarrow{B} E^* \left\{ P_{j,n}(S_n^{*b}, \hat{\theta}) T_n(S_n^{*b}) \right\} = a_{j,n}(\hat{\theta}) \|P_{j,n}\|_{\hat{\theta}}^2, \quad (6.16)$$

y

$$\frac{(\mathbf{X}_*^t \mathbf{X}_*)_{i,j}}{B} \xrightarrow{B} E^* \left\{ P_{i,n}(S_n^{*b}, \hat{\theta}) P_{j,n}(S_n^{*b}, \hat{\theta}) \right\} = \delta_{ij} \|P_{j,n}\|_{\hat{\theta}}^2 \quad (6.17)$$

cuando B tiende a infinito, en probabilidad condicional, donde δ_{ij} es la delta de Kronecker. Entonces la matriz $\mathbf{X}_*^t \mathbf{X}_* / B$ tiende a la matriz diagonal \mathbf{D} con $D_{j,j} = \|P_{j,n}\|_{\hat{\theta}}^2$, $j = 0, \dots, m$. Y según (6.15), (6.16) y (6.17), concluimos que

$$\hat{\beta}_j^* \longrightarrow a_{j,n}(\hat{\theta}), \quad B \longrightarrow \infty, \quad \text{para } j = 0, \dots, m. \quad (6.18)$$

Por tanto, $\hat{s}_{O_{rt}}^*$ tiende a $s^* = \hat{s}$ cuando B tiende a infinito. Además, puesto que $E^*(\varepsilon^*) = 0$ y $Var^*(\varepsilon^*)$ es finita, cuando B tiende a infinito, tendremos según la LGN,

$$\hat{\sigma}^{*2} = \frac{\sum_{b=1}^B \varepsilon_b^2}{B - m - 1} = \frac{B}{B - m - 1} \frac{\sum_{b=1}^B \varepsilon_b^2}{B} \longrightarrow E^*(\varepsilon^2) = \sigma^2, \quad (6.19)$$

en probabilidad condicional. En consecuencia, $\hat{v}_{O_{rt}}^*$ tiende a \hat{v} cuando B tiende a infinito. En conclusión, hemos probado la consistencia de $\hat{s}_{O_{rt}}^*$ y $\hat{v}_{O_{rt}}^*$ como estimadores del sesgo y de la varianza bootstrap ideales \hat{s} y \hat{v} respectivamente.

Estudiaremos, a continuación, ciertas propiedades de nuestros estimadores. Consideremos $\hat{s}_{O_{rt}}^*$, y veremos por qué este estimador es mejor que el estimador de Monte Carlo estándar s_{MC}^* y que es dado por

$$s_{MC}^* = \frac{1}{B} \sum_{b=1}^B T_n(S_n^{*b}) - h(\hat{\theta}).$$

Previamente, notaremos por E_c y Var_c la esperanza y la varianza condicionales siendo P_1^*, \dots, P_m^* (los regresores) fijos. Veremos que $\hat{s}_{O_{rt}}^*$ es, al igual que s_{MC}^* , insesgado de \hat{s} . De hecho, $E_c^*(\hat{s}_{O_{rt}}^*) = \beta_0 - h(\hat{\theta}) = \hat{s}$, y por el teorema de esperanza total

$$E^*(\hat{s}_{O_{rt}}^*) = E^*[E_c^*(\hat{s}_{O_{rt}}^*)] = E^*(\hat{s}) = \hat{s} \quad (6.20)$$

ya que \hat{s} es fija respecto a E^* . Entonces $\hat{s}_{O_{rt}}^*$ es condicionalmente insesgado de \hat{s} . Para s_{MC}^* es inmediato $E^*(s_{MC}^*) = \hat{s}$.

Podemos también calcular el sesgo no condicional considerando la variación de los datos originales: $E(\hat{s}_{O_{rt}}^*) = E[E^*(\hat{s}_{O_{rt}}^*)]$ y según (6.20) $E(\hat{s}_{O_{rt}}^*) = E(\hat{s})$. Entonces si \hat{s} es insesgado de s , $\hat{s}_{O_{rt}}^*$ también lo es. Esta propiedad también se verifica para s_{MC}^* .

Ahora vamos a calcular la varianza de $\hat{s}_{O_{rt}}^*$. Tenemos

$$Var^*(\hat{s}_{O_{rt}}^*) = Var^*(\hat{\beta}_0^*) = Var^*[E_c^*(\hat{\beta}_0^*)] + E^*[Var_c^*(\hat{\beta}_0^*)],$$

donde la última igualdad es debida al teorema de Madow, o de la varianza total. Y $E_c^*(\hat{\beta}_0^*) = \beta_0$ que es fija respecto al proceso de simulación, entonces $Var^*(\beta_0) = 0$. Además, según los resultados estándares de regresión

$$Var_c^*(\hat{\beta}^*) = \sigma^2 (\mathbf{X}_*^t \mathbf{X}_*)^{-1}, \quad \text{o, } var_c^*(\hat{\beta}_k^*) = \sigma^2 (\mathbf{X}_*^t \mathbf{X}_*)_{k,k}^{-1} \quad \text{para } k = 0, \dots, m. \quad (6.21)$$

Entonces,

$$Var^*(\hat{s}_{Ort}^*) = E^* [\sigma^2 (\mathbf{X}_*^t \mathbf{X}_*)_{0,0}^{-1}] = \frac{\sigma^2}{B} E^* [B (\mathbf{X}_*^t \mathbf{X}_*)_{0,0}^{-1}], \quad (6.22)$$

puesto que σ^2 es constante respecto al proceso de simulación. Según (6.17)

$$B (\mathbf{X}_*^t \mathbf{X}_*)_{k,k}^{-1} \longrightarrow \|P_{k,n}\|_{\hat{\theta}}^{-2} \text{ cuando } B \rightarrow \infty. \quad (6.23)$$

Entonces, para B bastante grande tenemos la aproximación

$$Var^*(\hat{s}_{Ort}^*) \simeq \frac{\sigma^2}{B}. \quad (6.24)$$

Para el estimador de s_{MC}^* tenemos

$$Var^*(s_{MC}^*) = \frac{\hat{v}}{B} = \frac{\sigma^2}{B} + \frac{1}{B} \sum_{j=1}^m \beta_j^2 \|P_{j,n}\|_{\hat{\theta}}^2. \quad (6.25)$$

A partir de (6.22) y (6.25) tendremos

$$\frac{Var^*(\hat{s}_{Ort}^*)}{Var^*(s_{MC}^*)} = \frac{E^* [B (\mathbf{X}_*^t \mathbf{X}_*)_{0,0}^{-1}]}{1 + \sigma^{-2} \sum_{j=1}^m \beta_j^2 \|P_{j,n}\|_{\hat{\theta}}^2}.$$

Y según (6.23),

$$\lim_{B \rightarrow +\infty} \frac{Var^*(\hat{s}_{Ort}^*)}{Var^*(s_{MC}^*)} = \frac{1}{1 + \sigma^{-2} \sum_{j=1}^m \beta_j^2 \|P_{j,n}\|_{\hat{\theta}}^2} < 1. \quad (6.26)$$

Podemos entonces ver que \hat{s}_{Ort}^* es más concentrado que el estimador de Monte Carlo estándar, para B bastante grande. Podemos también calcular la varianza incondicional que tiene en cuenta también la variabilidad en el proceso de sacar muestras originales. Tenemos, aplicando el teorema de Madow,

$$\begin{aligned} Var_c(\hat{s}_{Ort}^*) &= Var_c[E_c^*(\hat{s}_{Ort}^*)] + E_c[Var_c^*(\hat{s}_{Ort}^*)] \\ &= Var_c[\hat{s}] + E_c[\sigma^2 (\mathbf{X}_*^t \mathbf{X}_*)_{0,0}^{-1}] \\ &\simeq Var_c(\hat{s}) + E_c\left(\frac{\sigma^2}{B}\right) \text{ para } B \text{ suficientemente grande.} \end{aligned}$$

Entonces

$$\begin{aligned} Var(\hat{s}_{Ort}^*) &= Var[E_c(\hat{s}_{Ort}^*)] + E[Var_c(\hat{s}_{Ort}^*)], \text{ según el teorema de Madow,} \\ &\simeq Var[E_c(\hat{s})] + E[Var_c(\hat{s})] + E[E_c\left(\frac{\sigma^2}{B}\right)] \\ &= Var(\hat{s}) + \frac{1}{B} E[\sigma^2]. \end{aligned}$$

Para el estimador de Monte Carlo s_{MC}^* tenemos

$$\begin{aligned} \text{Var}(s_{MC}^*) &= \text{Var}[E^*(s_{MC}^*)] + E[\text{Var}^*(s_{MC}^*)] = \text{Var}(\hat{s}) + \frac{1}{B}E(\hat{v}) \\ &= \text{Var}(\hat{s}) + \frac{1}{B}E[\sigma^2] + \frac{1}{B}\sum_{i=1}^m E[a_{i,n}^2(\hat{\theta})\|P_{i,n}\|_{\hat{\theta}}^2]. \end{aligned}$$

Consideremos, ahora, el estimador de la varianza, \hat{v}_{Ort}^* . Tenemos $E_c^*[\hat{v}_{Ort}^*] = \sum_{k=1}^m \|P_{k,n}\|_{\hat{\theta}}^2 E_c^*[\hat{\beta}_k^{*2}] + E_c^*[\hat{\sigma}^{*2}]$ y $E_c^*[\hat{\sigma}^{*2}] = \sigma^2$. Además, para $k = 0, \dots, m$,

$$E_c^*[\hat{\beta}_k^{*2}] = \text{var}^*(\hat{\beta}_k^*) + \left(E_c^*(\hat{\beta}_k^*)\right)^2 = \text{var}^*(\hat{\beta}_k^*) + a_{k,n}^2(\hat{\theta}),$$

y según (6.21)

$$\begin{aligned} E_c^*(\hat{v}_{Ort}^*) &= \sum_{k=1}^m \|P_{k,n}\|_{\hat{\theta}}^2 \left\{ a_{k,n}^2(\hat{\theta}) + \sigma^2 (\mathbf{X}_*^t \mathbf{X}_*)_{k,k}^{-1} \right\} + \sigma^2 \\ &= \left(\sum_{k=1}^m a_{k,n}^2(\hat{\theta}) \|P_{k,n}\|_{\hat{\theta}}^2 + \sigma^2 \right) + \sigma^2 \sum_{k=1}^m (\mathbf{X}_*^t \mathbf{X}_*)_{k,k}^{-1} \|P_{k,n}\|_{\hat{\theta}}^2 \\ &= \hat{v} + \sigma^2 \sum_{k=1}^m (\mathbf{X}_*^t \mathbf{X}_*)_{k,k}^{-1} \|P_{k,n}\|_{\hat{\theta}}^2. \end{aligned} \quad (6.27)$$

Entonces concluimos que el sesgo condicional de \hat{v}_{Ort}^* como estimador de \hat{v} es

$$\text{sesg}_c^* = \sigma^2 \sum_{k=1}^m (\mathbf{X}_*^t \mathbf{X}_*)_{k,k}^{-1} \|P_{k,n}\|_{\hat{\theta}}^2. \quad (6.28)$$

Este sesgo condicional tiende a cero cuando B tiende a infinito. En efecto,

$$\text{sesg}_c^* = \frac{\sigma^2}{B} \sum_{k=1}^m B (\mathbf{X}_*^t \mathbf{X}_*)_{k,k}^{-1} \|P_{k,n}\|_{\hat{\theta}}^2.$$

Y puesto que σ^2 es finita y m en un entero fijo, según (6.23) concluimos que $\text{sesg}_B^* \rightarrow 0$, cuando $B \rightarrow \infty$.

Corrección del sesgo. Puesto que el sesgo condicional de \hat{v}_{Ort}^* como estimador de \hat{v} está dado por (6.28), podemos obtener un nuevo estimador a partir de \hat{v}_{Ort}^* eliminando su sesgo condicional. Según (6.27) tenemos

$$E_c^*(\hat{v}_{Ort}^*) = \hat{v} + E_c^*(\hat{\sigma}^{*2}) \sum_{k=1}^m (\mathbf{X}_*^t \mathbf{X}_*)_{k,k}^{-1} \|P_{k,n}\|_{\hat{\theta}}^2,$$

entonces

$$E_c^* \left\{ \hat{v}_{O_{rt}}^* - \hat{\sigma}^{*2} \sum_{k=1}^m (\mathbf{X}_*^t \mathbf{X}_*)_{k,k}^{-1} \|P_{k,n}\|_{\hat{\theta}}^2 \right\} = \hat{v}.$$

Consideremos el nuevo estimador

$$\tilde{v}_{O_{rt}}^* = \hat{v}_{O_{rt}}^* - \hat{\sigma}^{*2} \sum_{k=1}^m (\mathbf{X}_*^t \mathbf{X}_*)_{k,k}^{-1} \|P_{k,n}\|_{\hat{\theta}}^2 \quad (6.29)$$

$$= \sum_{i=1}^m \hat{\beta}_i^{*2} \|P_{i,n}\|_{\hat{\theta}}^2 + \hat{\sigma}^{*2} \left\{ 1 - \sum_{k=1}^m (\mathbf{X}^t \mathbf{X})_{k,k}^{-1} \|P_{k,n}\|_{\hat{\theta}}^2 \right\}, \quad (6.30)$$

entonces $\tilde{v}_{O_{rt}}^*$ es condicionalmente insesgado de \hat{v} . Sin embargo, $\text{sesg}_c^*(\hat{v}_{O_{rt}}^*)$ tiende a 0 cuando B tiende a infinito, los estimadores $\hat{v}_{O_{rt}}^*$ y $\tilde{v}_{O_{rt}}^*$ de \hat{v} son asintóticamente equivalentes. Los resultados numéricos muestran que aunque B sea pequeño $\hat{v}_{O_{rt}}^*$ y $\tilde{v}_{O_{rt}}^*$ son muy cercanos.

Nota 6.1

1. Puesto que $B^{-1}[\mathbf{X}_*^t \mathbf{X}_*]$ tiende a la matriz diagonal

$$\mathbf{D} = \text{diag}(\|P_{0,n}\|_{\hat{\theta}}^2, \dots, \|P_{m,n}\|_{\hat{\theta}}^2)$$

cuando B tiende a infinito, entonces $B[\mathbf{X}_*^t \mathbf{X}_*]^{-1}$ tiende a

$$\mathbf{D}^{-1} = \text{diag}(\|P_{0,n}\|_{\hat{\theta}}^{-2}, \dots, \|P_{m,n}\|_{\hat{\theta}}^{-2})$$

cuando B tiende a infinito. Entonces para B suficientemente grande tenemos,

$$[\mathbf{X}_*^t \mathbf{X}_*]_{kk}^{-1} \simeq B^{-1} \|P_{k,n}\|_{\hat{\theta}}^{-2}.$$

Y por tanto, según (6.29) obtenemos la aproximación,

$$\tilde{v}_{O_{rt}}^* \simeq \hat{v}_{O_{rt}}^* - \frac{m\hat{\sigma}^{*2}}{B} \simeq \hat{v}_{O_{rt}}^*, \quad (6.31)$$

ya que $\hat{\sigma}^{*2}$ es en general muy pequeña. Esto nos da una idea intuitiva que el sesgo condicional de $\hat{v}_{O_{rt}}^*$ tiende a 0 cuando B tiende a infinito.

2. Casi todo el software matemático usual como Maple, Splus, Statistica, Mathematica, etc, incluyen subrutinas estadísticas para hacer regresión o para generar muestras, además de un lenguaje de programación avanzado. Entonces el procedimiento descrito arriba puede ser fácilmente escrito en términos de un lenguaje de programación usando cualquiera de los software mencionados anteriormente. Los resultados numéricos en la sección siguiente son obtenidos mediante una rutina escrita en Maple.

6.1.2 Resultados numéricos

En esta sección consideremos tres ejemplos artificiales que nos van a mostrar con más claridad la precisión de nuestro procedimiento. En cada ejemplo consideramos estimadores no lineales en el estadístico S_n . En el primero y el tercero consideramos familias exponenciales continua y discreta respectivamente. En el primer ejemplo el estadístico de interés tiene un desarrollo finito (con 5 términos) en términos de los polinomios ortogonales, en cambio, en el tercer ejemplo el estadístico de interés tiene un desarrollo infinito. En el segundo ejemplo consideramos una familia que no es exponencial, es la familia Uniforme. Hemos elegido los ejemplos de modo que los cálculos teóricos pueden ser llevados a cabo con el fin de comparar nuestros resultados con los resultados exactos para así conocer la precisión que nuestro método puede ofrecer.

Ejemplo 6.2 Basándonos sobre una muestra de tamaño $n = 50$, procedente de la $N(\mu, 1)$, consideremos la estimación de una función paramétrica de la media $\xi = \mu^4 + 3\mu$. Consideremos el estadístico $S_n = \bar{X}$, que es la media muestral. Sea el estimador de ξ ,

$$\hat{\xi}_n = \bar{X}^4 - \frac{6}{n}\bar{X}^2 + 3\bar{X} + \frac{3}{n^2},$$

el cual es el *UMVUE* (estimador insesgado de mínima varianza) de ξ . La varianza exacta de $\hat{\xi}_n$ es

$$v = \frac{16}{n}\mu^6 + \frac{72}{n^2}\mu^4 + \frac{24}{n}\mu^3 + \frac{96}{n^3}\mu^2 + \frac{24}{n^4}, \quad (6.32)$$

y el *UMVUE* de v viene dado por

$$\hat{v}_{umvue} = \frac{16}{n}\bar{X}^6 - \frac{168}{n^2}\bar{X}^4 + \frac{24}{n}\bar{X}^3 + \frac{384}{n^3}\bar{X}^2 - \frac{72}{n^2}\bar{X} + \frac{9}{n} - \frac{96}{n^4}.$$

La familia de funciones ortogonales respecto a la densidad de \bar{X} que es la densidad de $N(\mu, 1/n)$ vienen dados por (ver Capítulo 3),

$$P_{k,n}(s, \mu) = \frac{1}{(2n)^{k/2}} H_k \left(\sqrt{\frac{n}{2}}(s - \mu) \right), \quad k \geq 0, \quad (6.33)$$

donde H_k son los polinomios de Hermite cuya expresión explícita esta dada en (3.18), ver CHIHARA (1978) para mas detalles. La norma de $P_{k,n}$ es

$$\|P_{k,n}\|_{\mu}^2 = \frac{k!}{n^k}.$$

Los polinomios $P_{k,n}$ en (6.33) pueden ser obtenidos por la formula de recurrencia

$$P_{k+1,n} = P_{1,n}P_{k,n} - \frac{k}{n}P_{k-1,n} \quad k \geq 1$$

$$P_{0,n} = 1, \quad P_{1,n} = s - \mu.$$

Con $\mu = 7$ hemos obtenido los resultados numéricos siguientes:

$$\bar{x} = 6.83283, \quad v = 37881.68643 \text{ y } \hat{v}_{umvue} = 32572.00191,$$

y la varianza bootstrap ideal obtenida sustituyendo μ por \bar{x} en (6.32),

$$\hat{v} = 32781.34526.$$

Consideremos, ahora, la aplicación del procedimiento explicado arriba con varios valores de B y m con el fin de aproximar \hat{v} . Primero fijamos $m = 4$, y variamos B , hemos obtenido las aproximaciones dadas en la Tabla 6.1:

B	\hat{v}_{Ort}^*	\tilde{v}_{Ort}^*	\hat{v}_{MC}^*	\hat{s}_{Ort}^*	\hat{s}_{MC}^*
6	32781.11613	32781.11609	11528.2103	.000858	30.23023
10	32781.37416	32781.37415	18698.7050	.000068	-38.8212
20	32781.34355	32781.34553	36169.3654	$-0.2 \cdot 10^{-5}$	-34.1568
40	32781.34510	32781.34511	32898.1225	0	-31.2732
60	32781.34525	32781.34525	32301.9975	$0.1 \cdot 10^{-5}$	-22.4187
80	32781.34526	32781.34526	33240.7893	$0.1 \cdot 10^{-5}$	-18.7976
100	32781.34522	32781.34522	35008.7642	$0.2 \cdot 10^{-5}$	-17.3197
120	32781.34525	32781.34525	35643.4437	$0.2 \cdot 10^{-5}$	-14.3102
140	32781.34521	32781.34521	32705.3355	$0.2 \cdot 10^{-5}$	-14.2850
160	32781.34522	32781.34522	32202.6725	0	-10.3236
180	32781.34516	32781.34516	33329.1621	0	-4.67841
200	32781.34516	32781.34516	32079.2066	$0.1 \cdot 10^{-5}$	-2.02081

Tabla 6.1. *variando B de 6 a 200, con m=4 (fijo).*

Los resultados de nuestro procedimiento se muestran junto a los obtenidos por las aproximaciones usuales \hat{v}_{MC}^* y \hat{s}_{MC}^* de Monte Carlo. Ahora, consideremos las mismas muestras bootstrap con las cuáles hemos obtenido los resultados correspondientes a $B = 200$ en la Tabla 1. Variando m (con $B = 200$ fijo) hemos obtenido los resultados

dados en la Tabla 6.2.

m	\hat{v}^*	\tilde{v}^*	\hat{s}^*
1	33140.27952	33140.27389	-0.153428
2	32774.49102	32774.49101	0.016204
3	32781.44662	32781.44662	-0.000083
4	32781.34516	32781.34516	$0.1 \cdot 10^{-5}$
5	32781.34512	32781.34512	0
6	32781.34514	32781.34514	0
7	32781.34293	32781.34293	$0.1 \cdot 10^{-5}$

Tabla 6-2. Variando m de 1 a 7, con $B=200$ (fijo).

Notemos a partir de la Tabla 6.1 que nuestros resultados no necesitan un número elevado, B , de muestras bootstrap. Por ejemplo con solo $B = 6$ en la Tabla 6.1, hemos obtenido una aproximación casi exacta $\hat{v}_{Ort}^* = 32781.11613$ a $\hat{v} = v^* = 32781.34526$. Comparemos este resultado con el obtenido por el estimador estándar de Monte Carlo $\hat{v}_{MC}^* = 11528.21037$ usando las mismas muestras bootstrap, se ve que este último es muy lejos del verdadero valor \hat{v} debido a que el número de simulaciones es muy pequeño. Observemos también que el valor de \hat{v}_{Ort}^* obtenido con $B = 6$ es mejor, en este caso, que el valor de $\hat{v}_{MC}^* = 32079.20661$ obtenido con $B = 200$.

Ejemplo 6.3 Consideremos la estimación de una función del parámetro θ en la familia $U(0, \theta)$. Sea por ejemplo la función $\xi = h(\theta) = \theta^5$, y consideremos el estimador

$$\hat{\xi}_n = T_n(S_n) = \frac{(5+n)S_n^5}{n},$$

donde $S_n = \max(X_1, \dots, X_n)$. La varianza exacta es,

$$v = \frac{25\theta^{10}}{n(n+10)},$$

cuyo estimador MVU es,

$$\hat{v}_{umvue} = \frac{25S_n^{10}}{n^2}.$$

En este caso, la familia de funciones ortogonales son polinomios y son dados por

$$P_{k,n}(s, \theta) = \frac{\Gamma(n+k)}{\Gamma(n+2k)} \sum_{j=0}^k (-1)^j \binom{k}{j} \frac{\Gamma(n+2k-j)}{\Gamma(n+k-j)} \left(\frac{s}{\theta}\right)^{k-j},$$

con norma

$$\|P_{k,n}\|_g^2 = \frac{n}{n+2k} \binom{2k+n-j}{k}^{-2}.$$

Estos polinomios pueden ser obtenidos fácilmente con una fórmula de recurrencia en términos de los polinomios de Jacobi (Ver CHIAHARA (1978)).

Hemos extraído una muestra de tamaño $n = 50$ a partir de la distribución $U(0, 3)$, y hemos obtenido las observaciones numéricas siguientes:

$$S_n = 2.9757, v = 492.075, \text{ and } \hat{v}_{umvue} = 544.3587.$$

La varianza bootstrap ideal es

$$\hat{v} = v^* = \frac{25S_n^{10}}{n(n+10)},$$

cuyo valor observado es

$$\hat{v} = 453.63226.$$

Con diferentes valores de B , hemos obtenido los resultados numéricos de la Tabla 6.3, con $m = 5$ fijo.

B	\hat{v}_{Ort}^*	\tilde{v}_{Ort}^*	\hat{v}_{MC}^*	\hat{s}_{Ort}^*	\hat{s}_{MC}^*
10	923.540032	925.528931	138.4798336	8.7223	8.7112
15	453.6540937	453.6540937	393.6765096	-0.000097	5.05236
20	453.8130338	453.8130336	378.9065329	-0.000801	1.94614
40	453.2868597	453.2868594	439.7110050	0.0015541	-0.2081
60	453.5550565	453.5550565	381.2745791	0.0003448	-0.7006
80	453.5805111	453.5805111	364.3566001	0.0002312	-0.7205
100	453.6317709	453.6317709	394.13188	$0.19 \cdot 10^{-5}$	0.21223
120	453.6340941	453.6340941	416.702215	$-0.75 \cdot 10^{-5}$	-0.1109
140	453.6332827	453.6332827	402.3877136	$-0.42 \cdot 10^{-5}$	0.53121
160	453.6325247	453.6325247	421.6946268	$-0.1 \cdot 10^{-5}$	1.48933
180	453.6315858	453.6315858	411.7689438	$0.27 \cdot 10^{-5}$	2.06458
200	453.6317885	453.6817885	399.2242577	$0.19 \cdot 10^{-5}$	2.24848

Tabla 6.3. Variando B de 10 a 200, con $m=5$ (fijo).

Variamos, ahora, el valor de m con $B = 200$ fijo, obteniendo los resultados de la

Tabla 6.4,

m	\hat{v}_{Ort}^*	\tilde{v}_{Ort}^*	\hat{s}_{Ort}^*
1	470.5124864	470.5124839	0.0048079
2	452.9149497	452.9149497	0.0048234
3	453.6733041	453.6733041	-0.0001406
4	453.6312473	453.6312473	$0.4 \cdot 10^{-5}$
5	453.6322000	453.6322000	$0.3 \cdot 10^{-6}$
6	453.5473425	453.5473425	0.0003267
7	466.7156426	4667149649	0.0360616

Tabla 6.4. Variando m de 1 a 7, con $B=200$ (fijo).

Ejemplo 6.4 Supongamos, en este caso, que X_1, \dots, X_n son v. a. procedentes de una distribución de Poisson $Po(\mu)$. Consideremos a estimar la función paramétrica $h(\mu) = e^{-\mu}$ la cual no es un polinomio como en los ejemplos anteriores. Consideremos el estimador de Máxima verosimilitud de ζ , $\hat{\zeta} = T(\bar{X}) = e^{-\bar{X}}$. En este ejemplo, $\hat{\zeta}$ no es un polinomio en \bar{X} , y por lo tanto tiene un desarrollo infinito en $\mathcal{L}_{\theta, n}^2$.

El sesgo exacto s y el sesgo bootstrap ideal \hat{s} de $\hat{\zeta}$ son respectivamente

$$s = \exp(n\mu(e^{-1/n} - 1)) - e^{-\mu} \text{ y } \hat{s} = \exp(n\bar{X}(e^{-1/n} - 1)) - e^{-\bar{X}}.$$

La varianza exacta v y la varianza bootstrap ideal v^* de $\hat{\zeta}$ son

$$v = \exp(n\mu(e^{-2/n} - 1)) - \exp(2n\mu(e^{-1/n} - 1)),$$

y

$$\hat{v} = \exp(n\bar{X}(e^{-2/n} - 1)) - \exp(2n\bar{X}(e^{-1/n} - 1)).$$

La familia de funciones ortogonales respecto a \bar{X} en este caso es

$$P_{k,n}(s, \mu) = \frac{1}{n^k} C_k^{(n\mu)}(ns), \quad k \geq 0, \quad (6.34)$$

donde $C_k^{(\alpha)}$ es el k -ésimo polinomio de Charlier [ver CHIHARA (1978)]:

$$C_k^{(\alpha)}(s) = \sum_{j=0}^k \binom{k}{j} \binom{s}{j} j! (-\alpha)^{k-j}.$$

La norma de estos polinomios viene dada por

$$\|P_{k,n}\|_{\mu}^2 = \frac{k! \mu^k}{n^k}.$$

Los polinomios en (6.34) pueden ser obtenidos recurrentemente por:

$$P_{k+1,n} = \left(P_{1,n} - \frac{k}{n}\right)P_{k,n} - \frac{k\mu}{n}P_{k-1,n}, \quad k \geq 1$$

$$P_{0,n} = 1, \quad P_{1,n} = s - \mu.$$

Hemos considerado una muestra de tamaño $n = 50$ a partir de la distribución de Poisson $Po(5)$ y hemos obtenido los resultados siguientes:

$$\hat{s} = 0.0003655962 \text{ y } \hat{v} = 0.59488138 \cdot 10^{-5}.$$

Consideremos, ahora, el procedimiento descrito en este capítulo con diferentes valores de B . Hemos obtenido los resultados numéricos de la Tabla 6.5, donde $m = 4$ es fijo:

B	\hat{v}_{Ort}^*	\tilde{v}_{Ort}^*	\hat{v}_{MC}^*	\hat{s}_{Ort}^*	\hat{s}_{MC}^*
20	0.58597601	0.58597595	0.277783	0.0003586	-0.0004023
40	0.59085937	0.59085936	0.461116	0.0003621	-0.6336 10^{-5}
60	0.59017551	0.59017549	0.451178	0.0003621	-0.0002263
80	0.59666088	0.59666081	0.584547	0.0003638	0.0000212
100	0.59633959	0.59633954	0.580620	0.0003638	0.0000478

Tabla 6.5 Variando B de 20 a 100, y donde $m=4$ (fijo). Las tres columnas correspondientes a la estimación de la varianza son multiplicados por 10^{-5} .

Variando ahora m , con $B = 100$ fijo, hemos obtenido los resultados de la tabla 6.6.

m	\hat{v}_{Ort}^*	\tilde{v}_{Ort}^*	\hat{s}_{Ort}^*
1	0.61808074 10^{-5}	0.61805657 10^{-5}	0.00036950
2	0.59019505 10^{-5}	0.59019286 10^{-5}	0.00038067
3	0.59633959 10^{-5}	0.59633954 10^{-5}	0.00036365
4	0.59475705 10^{-5}	0.59475704 10^{-5}	0.00036570
5	0.59490161 10^{-5}	0.59490161 10^{-5}	0.00036558
6	0.59487770 10^{-5}	0.59487770 10^{-5}	0.00036559
7	0.59488148 10^{-5}	0.59488148 10^{-5}	0.00036559
8	0.59488128 10^{-5}	0.59488128 10^{-5}	0.00036559

Table 6.6 Variando m de 1 a 8, con $B=100$.

6.2 Caso multiparamétrico

El procedimiento descrito en la sección anterior puede ser generalizado fácilmente al caso multidimensional, $d \geq 2$. La única dificultad, quizás, es que cuando las funciones ortogonales son polinomios, como es el caso para la mayoría de las familias usuales, por ser multidimensionales no hay un orden evidente que nos permita colocarlos de una forma muy clara en el desarrollo como en (6.2) donde el índice k es unidimensional y corresponde al polinomio de grado k . En el caso multidimensional, para cada entero $k_0 \geq 1$ existen varios polinomios de mismo grado k_0 en el desarrollo, por lo que usamos un índice de dimensión d . Este hecho puede causar dificultades cuando se construye la matriz de diseño \mathbf{X}_* como en (6.10) para familias de dimensión elevada, a la hora de colocar las columnas. Sin embargo, para una dimensión $d = 2$, o $d = 3$, esto puede hacerse fácilmente.

Supongamos que $\mathcal{X}_n = (X_1, \dots, X_n)$ es una muestra de tamaño n , (X_i son ahora vectores aleatorios), procedente de una población con distribución desconocida $F = F_\theta(x)$, $\theta \in \Theta \subset \mathbb{R}^d$, con $d \geq 2$, con densidad $f_\theta(x)$ respecto a una medida σ -finita ν (medida de Lebesgue o Cardinal). Sea $S_n = S_n(X_1, \dots, X_n)$ un estadístico para la familia F_θ . En este caso S_n es de dimensión d . Para cada $\theta \in \Theta$, definimos los espacios como en (6.1), i.e.,

$$\mathcal{L}_{n,\theta}^2 = \{T : \mathbb{R}^d \rightarrow \mathbb{R} \text{ tal que } E_\theta[|T(S_n)|^2] < \infty\}.$$

Como hemos dicho, los espacios $\mathcal{L}_{n,\theta}^2$ son espacios de Hilbert separables en los cuales cada función T admite un desarrollo de la forma (6.2). Si usamos un índice k unidimensional, entonces el procedimiento es exactamente igual que en el caso uniparamétrico. La diferencia surge cuando es necesario usar un índice multidimensional como ocurre cuando el sistema $\{P_{k,n}(s, \theta)\}_k$ son polinomios multidimensionales. Así, por simplificar, supongamos que el sistema ortogonal está formado por polinomios multivariantes. En lo que sigue notemos por $\mathbf{0}$ el vector cero d -dimensional, y por e_j el vector de dimensión d cuyas componentes son igual a cero excepto la j -ésima componente que es igual a 1. Para cada $k \in \mathbb{N}^d$, con $k = (k_1, \dots, k_d)$, notaremos por $|k| = k_1 + \dots + k_d$.

Si $\hat{\xi}_n = T_n(S_n)$ es un estimador de una cierta función real $\xi = h(\theta)$ del parámetro θ con $v = \text{Var}_\theta(T_n(S_n)) < \infty$, entonces T_n es un elemento de $\mathcal{L}_{n,\theta}^2$ y por lo tanto tiene un desarrollo en términos de las funciones ortogonales de la forma

$$T_n(S_n) = \sum_{k \in \mathbb{N}^d} a_{k,n}(\theta) P_{k,n}(S_n, \theta).$$

Dado que el desarrollo anterior es absolutamente convergente en media cuadrática, podemos reordenar la serie para obtener

$$T_n(S_n) = a_{0,n}(\theta) + \sum_{j \in \mathbb{N} \setminus \{0\}} \sum_{k \in \mathbb{N}^d, |k|=j} a_{k,n}(\theta) P_{k,n}(S_n, \theta), \quad (6.35)$$

ya que $P_{0,n} \equiv 1$.

La varianza $v = \text{Var}_\theta(\hat{\xi}_n)$ y el sesgo $s = \text{Sesgo}_\theta(\theta)$ son

$$v = \sum_{j \in \mathbb{N} \setminus \{0\}} \sum_{k \in \mathbb{N}^d, |k|=j} a_{k,n}^2(\theta) \|P_{k,n}\|_\theta^2 \text{ y } s = a_{0,n}(\theta) - h(\theta). \quad (6.36)$$

Truncamos el desarrollo (6.35) y consideremos, como en el caso uniparamétrico, el método bootstrap para estimar v y s .

Sea $\mathcal{X}_n^{*1}, \dots, \mathcal{X}_n^{*B}$ B muestras bootstrap cada una de tamaño n procedentes del estimador paramétrico $F_{\hat{\theta}}$, y consideremos el método de mínimos cuadrados para estimar los coeficientes de Fourier. Con el mismo procedimiento que en la sección anterior, tendremos

$$T_n(S_n^{*b}) = \beta_0 + \sum_{j=1}^m \sum_{k \in \mathbb{N}^d, |k|=j} \beta_k P_{k,n}(S_n^{*b}, \hat{\theta}) + \varepsilon^b, \quad b = 1, \dots, B, \quad (6.37)$$

donde el sesgo y la varianza bootstrap ideales serían

$$\hat{s} = \beta_0 - h(\hat{\theta}) \text{ y } \hat{v} = \sum_{j=1}^m \sum_{k \in \mathbb{N}^d, |k|=j} \beta_k^2 \|P_{k,n}\|_\theta^2 + \sigma^2.$$

Por ejemplo, supongamos que $d = 3$, y $m = 2$. Sea $T^{*b} = T_n(S_n^{*2})$ y $P_k^{*b} = P_{k,n}(S_n^{*b}, \hat{\theta})$. En este caso, para cada $b = 1, \dots, B$, el desarrollo (6.37) es

$$T^{*b} = \beta_0 + \sum_{j=1}^3 \beta_{e_j} P_{e_j}^{*b} + \sum_{j=1}^3 \beta_{2e_j} P_{2e_j}^{*b} + \sum_{i=1}^2 \sum_{j=i+1}^3 \beta_{e_i+e_j} P_{e_i+e_j}^{*b} + \varepsilon^b.$$

La matriz de diseño será entonces

$$\mathbf{X}_* = \begin{bmatrix} 1 & P_{e_1}^{*1} & P_{e_2}^{*1} & P_{e_3}^{*1} & P_{2e_1}^{*1} & P_{2e_2}^{*1} & P_{2e_3}^{*1} & P_{e_1+e_2}^{*1} & P_{e_1+e_3}^{*1} & P_{e_2+e_3}^{*1} \\ \vdots & \vdots \\ 1 & P_{e_1}^{*B} & P_{e_2}^{*B} & P_{e_3}^{*B} & P_{2e_1}^{*B} & P_{2e_2}^{*B} & P_{2e_3}^{*B} & P_{e_1+e_2}^{*B} & P_{e_1+e_3}^{*B} & P_{e_2+e_3}^{*B} \end{bmatrix}. \quad (6.38)$$

En general, nuestras aproximaciones de \hat{s} y \hat{v} serían simplemente

$$\begin{aligned}\hat{s}_{Ort}^* &= \hat{\beta}_0^* - h(\hat{\theta}) \text{ y} \\ \hat{v}_{Ort}^* &= \sum_{j=1}^m \sum_{k \in \mathbb{N}^d, |k|=j} \hat{\beta}_k^* \|P_{k,n}\|_{\hat{\theta}}^2 + \hat{\sigma}^{*2}.\end{aligned}$$

donde $\hat{\beta}_k^*$ y $\hat{\sigma}^{*2}$ son las estimaciones de mínimos cuadrados de β_k y σ^2 .

6.2.1 Resultados numericos

Con el fin de ver la mejora que nos pueden ofrecer los métodos ortogonales sobre los metodos bootstrap estándares, vamos a considerar una muestra de tamaño n a partir de una distribución normal bidimensional con media $\mu = (\mu_1, \mu_2)^t$ desconocida:

$$(X_1, Y_1), \dots, (X_n, Y_n) \rightsquigarrow \mathbf{F}_\mu \equiv \mathbf{N}(\mu, \Sigma).$$

Vamos a considerar la estimación de dos parámetros de interes $\xi = \mu_1 \mu_2$ el producto de las medias y $\delta = \mu_2 / \mu_1$ su razón. Con $\mu = (5, 8)^t$ y $\Sigma = \mathbf{I}_2$, hemos sacado una muestra de tamaño $n = 20$.

a.) Consideremos, en primer lugar, la estimación de $\xi = h(\mu_1, \mu_2) = \mu_1 \mu_2$, usando el estimador de interés

$$\hat{\xi} = T(\bar{X}, \bar{Y}) = \bar{X}\bar{Y}.$$

Para este estimador simple podemos calcular su sesgo \hat{s} y su varianza \hat{v} bootstrap exactos sin necesidad de simulación. Este hecho nos permitira comparar los resultados de simulación a las estimaciones ideales. Un cálculo elemental nos lleva a

$$s = 0 \text{ y } v = \left(\mu_1^2 + \frac{\sigma_1^2}{n}\right) \left(\mu_2^2 + \frac{\sigma_2^2}{n}\right) - \mu_1 \mu_2.$$

Y puesto que $\sigma_1 = \sigma_2 = 1$ por hipótesis, reemplazando μ_1 y μ_2 por \bar{X} y \bar{Y} respectivamente en la expresión anterior obtendremos las estimaciones bootstrap ideales

$$\hat{s} = 0 \text{ y } \hat{v} = \left(\bar{X}^2 + \frac{1}{n}\right) \left(\bar{Y}^2 + \frac{1}{n}\right) - \bar{X}\bar{Y}.$$

Con una muestra de tamaño $n = 20$ hemos obtenido las realizaciones siguientes del sesgo y la varianza bootstrap ideales,

$$\hat{s} = 0 \text{ y } \hat{v} = 4.480263005.$$

Consideremos, ahora, el procedimiento bootstrap basado en simulación a partir de $F_{\hat{\mu}}$ con $\hat{\mu} = (\bar{X}, \bar{Y})^t$, para aproximar a \hat{s} y \hat{v} . Usaremos dos métodos para calcular aproximaciones para el sesgo y la varianza bootstrap ideales: el método bootstrap usual y el método ortogonal. En este último hemos considerado un desarrollo hasta el grado $m = 2$ para el estimador $\hat{\xi}$. Es decir, eliminando los argumentos,

$$\hat{\xi} = \beta_0 + \beta_1 \mathbf{P}_{e_1} + \beta_2 \mathbf{P}_{e_2} + \beta_3 \mathbf{P}_{2e_1} + \beta_4 \mathbf{P}_{2e_2} + \beta_5 \mathbf{P}_{e_1+e_2} + \varepsilon.$$

donde $e_1 = (1, 0)$ y $e_2 = (0, 1)$. En nuestro caso, puesto que X y Y son independientes, estos polinomios vienen dados por

$$\begin{aligned} \mathbf{P}_{e_1}(x, y) &= P_1(x), \quad \mathbf{P}_{e_2}(x, y) = P_1(y), \quad \mathbf{P}_{2e_1}(x, y) = P_2(x), \\ \mathbf{P}_{2e_2}(x, y) &= P_2(y) \quad \text{y} \quad \mathbf{P}_{e_1+e_2}(x, y) = P_1(x)P_1(y), \end{aligned}$$

donde P_1 y P_2 son los polinomios de grado 1 y de grado 2 ortogonales respecto a la distribución de \bar{X} , dados en (6.33). Entonces las aproximaciones al sesgo y a la varianza bootstrap ideales mediante el método ortogonal vienen dadas por

$$\begin{aligned} \hat{s}_{Ort}^* &= \hat{\beta}_0^* - \hat{\xi} \\ \hat{v}_{Ort}^* &= \hat{\beta}_1^{*2} \|\mathbf{P}_{e_1}\|^2 + \hat{\beta}_2^{*2} \|\mathbf{P}_{e_2}\|^2 + \hat{\beta}_3^{*2} \|\mathbf{P}_{2e_1}\|^2 + \hat{\beta}_4^{*2} \|\mathbf{P}_{2e_2}\|^2 + \hat{\beta}_5^{*2} \|\mathbf{P}_{e_1+e_2}\|^2 + \hat{\sigma}^{*2}. \end{aligned}$$

donde $\{\hat{\beta}_j^*\}_{j=0,\dots,5}$ y $\hat{\sigma}^{*2}$ es la estimación de mínimos cuadrados de los parámetros $\{\beta_j\}_{j=0,\dots,5}$ y la varianza σ^2 de ε . En nuestro caso las normas en la fórmula anterior de \hat{v}_{Ort}^* vienen dadas por

$$\begin{aligned} \|\mathbf{P}_{e_1}\|^2 &= \|\mathbf{P}_{e_2}\|^2 = \|P_1\|^2, \quad \|\mathbf{P}_{2e_1}\|^2 = \|\mathbf{P}_{2e_2}\|^2 = \|P_2\|^2 \quad \text{y} \\ \|\mathbf{P}_{e_1+e_2}\|^2 &= \|P_1\|^4. \end{aligned}$$

Con varios valores para el número de muestras bootstrap B , hemos calculado en cada caso los sesgos \hat{s}_{Ort}^* y \hat{s}_{MC}^* y las varianzas \hat{v}_{Ort}^* y \hat{v}_{MC}^* . Hemos obtenido los resultados de la Tabla 6.7. Los valores de la varianza \hat{v}_{Ort}^* se muestran hasta el noveno decimal para poder ver mejor la precisión del método ortogonal.

B	\hat{v}_{Ort}^*	\hat{v}_{MC}^*	\hat{s}_{Ort}^*	\hat{s}_{MC}^*
10	4.480263081	5.17825	$0.4 \cdot 10^{-7}$	-0.3175
20	4.480263152	4.0921	$0.1 \cdot 10^{-7}$	0.1737
50	4.480262963	5.1512	$0.4 \cdot 10^{-7}$	-0.348
100	4.480263036	4.9329	$0.2 \cdot 10^{-7}$	-0.2378
200	4.480263006	4.5873	0	0.0081

Tabla 6.7

Tanto para el sesgo como para la varianza podemos ver la precisión del método ortogonal, incluso con un número de simulaciones B muy pequeño.

b.) Consideremos la estimación del parámetro $\delta = \mu_2/\mu_1$ mediante el estimador $\hat{\delta} = T(\bar{X}, \bar{Y}) = \bar{Y}/\bar{X}$. En este caso no es posible obtener una fórmula cerrada para el cálculo del sesgo o la varianza bootstrap ideales de $\hat{\delta}$ como es el caso del apartado anterior. En este tipo de problemas donde el método bootstrap resulta útil y puede ofrecer soluciones para aproximar la varianza o el sesgo, generalmente usando simulación. En la mayoría de los problemas en la literatura Bootstrap o Jackknife, se enfoca sobre las medidas de precisión del cociente $\hat{\delta}$.

En este caso, el procedimiento que vamos a seguir para comprobar la eficacia del método ortogonal es el siguiente: con una muestra original de tamaño $n = 20$ fija, sacamos $M = 30$ grupos de muestras bootstrap. Cada grupo está constituido por $B = 40$ muestras bootstrap extraídas a partir de F_{μ} . Para cada uno de los M grupos calculamos el sesgo y la varianza bootstrap usando los dos métodos: el método ortogonal y el método estándar. Obtendremos, entonces, M pares

$$(\hat{s}_{Ort,j}^*, \hat{s}_{MC,j}^*) \text{ y } (\hat{v}_{Ort,j}^*, \hat{v}_{MC,j}^*) \quad j = 1, \dots, M,$$

donde, para cada $j = 1, \dots, M$, $\hat{s}_{Ort,j}^*$ y $\hat{v}_{Ort,j}^*$ son aproximaciones del sesgo y la varianza bootstrap ideales, \hat{s} y \hat{v} , respectivamente mediante el método ortogonal, y $\hat{s}_{MC,j}^*$ y $\hat{v}_{MC,j}^*$ son los correspondientes aproximaciones usando el método de Monte Carlo. Puesto que las estimaciones bootstrap ideales no están disponibles en este caso, para poder comparar los resultados obtenidos a ellos, vamos a proceder así: primero consideremos la aproximación de la varianza y calculamos la varianza \hat{v}_{delta} del estimador $\hat{\delta}$ usando el método delta (Ver Capítulo 2). Hemos obtenido:

$$\hat{v}_{delta} = 5.2715.$$

Consideremos a \hat{v}_{delta} como si fuera la varianza bootstrap ideal. Los 30 valores $(\hat{v}_{Ort,j}^*, \hat{v}_{MC,j}^*) \quad j = 1, \dots, 30$ obtenidos son:

$$\begin{aligned} & (5.5654, 7.250), (5.5626, 7.802), (5.5586, 6.405), (5.5604, 7.203), (5.5405, 3.973), \\ & (5.5501, 7.523), (5.5574, 4.787), (5.5805, 7.125), (5.5492, 3.954), (5.5469, 4.786), \\ & (5.5407, 4.818), (5.5408, 4.640), (5.5805, 9.354), (5.5502, 9.026), (5.5681, 6.783), \\ & (5.5540, 6.380), (5.5257, 5.209), (5.5574, 6.507), (5.5358, 7.731), (5.5706, 5.737), \\ & (5.5446, 5.407), (5.5467, 4.840), (5.5367, 5.250), (5.5446, 4.731), (5.5706, 7.273), \\ & (5.5412, 6.604), (5.5734, 6.735), (5.5470, 5.610), (5.5430, 5.423), (5.5453, 4.624). \end{aligned}$$

En realidad hemos multiplicado los resultados por 10^3 para facilitar la lectura. Ahora vamos a calcular el error cuadrático medio cometido en cada uno de los dos métodos,

$$ECM_{delta}(\hat{v}_{Ort}^*) = \frac{1}{30} \sum_{j=1}^{30} (\hat{v}_{Ort,j}^* - \hat{v}_{delta})^2 = 0.0778,$$

y

$$ECM_{delta}(\hat{v}_{MC}^*) = \frac{1}{30} \sum_{j=1}^{30} (\hat{v}_{MC,j}^* - \hat{v}_{delta})^2 = 2.66.$$

Entonces el cociente de los errores da

$$\frac{ECM_{delta}(\hat{v}_{MC}^*)}{ECM_{delta}(\hat{v}_{Ort}^*)} = 34.2.$$

Puesto que la aproximación delta de la varianza del estimador $\hat{\delta} = \bar{Y}/\bar{X}$ identifica la varianza real de $\hat{\delta}$ con la varianza del estimador lineal, $\hat{\delta}_L$, más proximo a $\hat{\delta}$, en algunas situaciones puede resultar bastante lejana de la varianza correcta, concretamente cuando $Var(\hat{\delta}_L)/Var(\hat{\delta})$ no es muy proxima a 1.

Ahora, para calcular la varianza bootstrap ideal de $\hat{\delta}$, vamos a usar un número suficientemente grande de simulaciones bootstrap. Para ello, hemos usado $B = 5000$, luego $B = 50000$ muestras bootstrap para calcular \hat{v} . Hemos obtenido los resultados siguientes

$$\hat{v}_{5000} = 5.616 \text{ y } \hat{v}_{50000} = 5.550.$$

Calculando el error cuadrático medio en cada caso se obtiene, para $B = 5000$,

$$ECM_{5000}(\hat{v}_{Ort}^*) = \frac{1}{30} \sum_{j=1}^{30} (\hat{v}_{Ort,j}^* - \hat{v}_{5000})^2 = 0.45 \cdot 10^{-2},$$

y

$$ECM_{5000}(\hat{v}_{MC}^*) = \frac{1}{30} \sum_{j=1}^{30} (\hat{v}_{MC,j}^* - \hat{v}_{5000})^2 = 2.18,$$

con el cociente

$$\frac{ECM_{5000}(\hat{v}_{MC}^*)}{ECM_{5000}(\hat{v}_{Ort}^*)} = 488.9.$$

Para $B = 50000$, en este caso la aproximación \hat{v}_{50000} a \hat{v} es sin duda más precisa que en cualquiera de los casos anteriores. Calculando el error cuadrático medio de \hat{v}_{Ort}^*

y \hat{v}_{MC}^* respecto a \hat{v}_{50000} , se han obtenido los siguientes resultados

$$ECM_{50000}(\hat{v}^*) = \frac{1}{30} \sum_{j=1}^{30} (\hat{v}_{Ort,j}^* - \hat{v}_{50000})^2 = 0.17 \cdot 10^{-3},$$

y

$$ECM_{50000}(\hat{v}_{MC}^*) = \frac{1}{30} \sum_{j=1}^{30} (\hat{v}_{MC,j}^* - \hat{v}_{50000})^2 = 2.25,$$

con el cociente

$$\frac{ECM_{50000}(\hat{v}_{MC}^*)}{ECM_{50000}(\hat{v}_{Ort}^*)} = 13235.3.$$

Para el sesgo hemos obtenido resultados similares:

$$\hat{s}_{5000} = 0.0029 \text{ y } \hat{s}_{50000} = 0.0025.$$

y los cocientes de los errores cuadráticos medios en cada caso son

$$\frac{ECM_{5000}(\hat{s}_{MC}^*)}{ECM_{5000}(\hat{s}_{Ort}^*)} = 554.65 \text{ y } \frac{ECM_{50000}(\hat{s}_{MC}^*)}{ECM_{50000}(\hat{s}_{Ort}^*)} = 53470.58.$$

Capítulo 7

MÉTODOS ORTOGONALES: BOOTSTRAP NO PARAMÉTRICO

En este capítulo consideremos la estimación de la varianza y el sesgo mediante desarrollos ortogonales basándonos en procedimientos no paramétricos. Como es usual, en dichas situaciones, la distribución empírica es la clave. En el caso paramétrico, los estimadores a menudo son funciones de un estadístico S_n . Es decir $\hat{\theta} = T(S_n)$, por lo que, el desarrollo de $T(\cdot)$ está en términos de funciones ortogonales respecto a la densidad de S_n . Entonces, cuando S_n es d dimensional, las funciones ortogonales dependen de d variables, i.e. que el número de argumentos es independiente de n .

En el caso no paramétrico, al contrario, los estimadores en general no son función de un estadístico específico, sino, son función de la muestra, $\hat{\theta} = T(X_1, \dots, X_n)$, entonces las funciones ortogonales en el desarrollo de $\hat{\theta}$, tienen n argumentos. En el caso unidimensional, puesto que F_n tiene momentos finitos, el sistema ortogonal respecto a F_n es un sistema de polinomios.

En este capítulo, consideremos dos procedimientos: el primero está basado en el sistema de los polinomios ortogonales asociados a la distribución empírica, y el segundo procedimiento está basado en el sistema de polinomios ortogonales respecto a la distribución multinomial.

7.1 Desarrollos basados en la fdD empírica

Sea $\mathcal{X}_n = (X_1, \dots, X_n)$ una muestra aleatoria simple procedente de una distribución desconocida F . Sea θ y $\hat{\theta}_n = T(\mathcal{X}_n)$ respectivamente el parámetro y el estadístico de interés. Supongamos que queremos estimar la varianza v y el sesgo s de $\hat{\theta}_n$ usando el método bootstrap no paramétrico, donde las simulaciones se efectúan a partir de la distribución empírica F_n . Sea entonces, \hat{v} y \hat{s} las estimaciones bootstrap ideales de v y s respectivamente. Cuando \hat{v} y \hat{s} son calculables en función de los datos \mathcal{X}_n , se toman entonces como estimaciones de v y s , sino, se recurre a una aproximación a \hat{v} y \hat{s} como se explicó en el *Capítulo 4*.

En este capítulo, supondremos que se necesita una aproximación a \hat{v} y \hat{s} . Con este fin, vamos a considerar el desarrollo de $\hat{\theta}_n^*$ en términos de los polinomios ortogonales respecto a la distribución empírica F_n . Para ello, consideremos el espacio de funciones

$$\mathcal{L}^2 = \{T : E^*[T^2(X^*)] < \infty\},$$

donde, como es habitual, la esperanza $E^*(\cdot) = E_{F_n}(\cdot)$ es la esperanza bootstrap, y X^* es una variable aleatoria con distribución F_n . Dos funciones T_1 y T_2 de \mathcal{L}^2 son consideradas iguales si lo son casi por todo. El espacio \mathcal{L}^2 es un espacio de Hilbert con producto escalar

$$\langle T_1, T_2 \rangle = E^*[T_1(X^*)T_2(X^*)].$$

7.1.1 Polinomios ortogonales asociados a F_n

Sea $m_0 = 1$, y $m_k = \frac{1}{n} \sum_{i=1}^n X_i^k$ el k -ésimo momento empírico, y consideremos la matriz de momentos $\mathcal{M}_k = [m_{i+j}]_{i,j=0,\dots,k}$. Puesto que F_n tiene soporte finito (de cardinal n) entonces sus momentos k son finitos. Supongamos que los determinantes $\Delta_k = \det \mathcal{M}_k$, $k = 0, \dots, n-1$, son no nulos. Entonces un sistema de polinomios ortogonales respecto a F_n existe y viene dado por (ver CHIHARA (1978))

$$P_k(s) = \Delta_{k-1}^{-1} D_k(s), \quad k = 0, \dots, n-1, \quad (7.1)$$

donde

$$D_k(s) = \det \begin{bmatrix} m_0 & m_1 & \dots & m_k \\ m_1 & m_2 & \dots & m_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ m_{k-1} & m_k & \dots & m_{2k-1} \\ 1 & s & \dots & s^k \end{bmatrix}.$$

La norma de los polinomios en (7.1) viene dada por

$$\|P_k\|_{\mathcal{L}^2}^2 = \|P_k\|^2 = E^*[P_k^2(X^*)] = \Delta_{k-1}^{-1}\Delta_k, \quad k = 0, \dots, n-1,$$

con el convenio $\Delta_{-1} = 1$. Los polinomios $\{P_k(s)\}_{k \geq 0}$ verifican la relación de recurrencia

$$\begin{aligned} P_{k+1}(s) &= (s - c_{k+1})P_k(s) - \lambda_{k+1}P_{k-1}(s), \\ P_0(s) &= 1, \text{ y } P_1(s) = s - m_1; \end{aligned}$$

con

$$\lambda_{k+1} = \frac{\Delta_{k-2}\Delta_k}{\Delta_{k-1}^2} \text{ y } c_{k+1} = \frac{\langle s, P_k^2(s) \rangle}{\|P_k\|_2^2}.$$

Sea $\mathcal{X}_n^* = (X_1^*, \dots, X_n^*)$ una muestra bootstrap de tamaño n . Consideremos los espacios de Hilbert

$$\mathcal{L}_n^2 = \{T : \mathbb{R}^n \rightarrow \mathbb{R}, E_{\mathcal{X}}^*[T^2(\mathcal{X}_n^*)]\},$$

donde $E_{\mathcal{X}}^*$ es la esperanza respecto a la distribución conjunta de $\mathcal{X}_n^* = (X_1^*, \dots, X_n^*)$. Si $\hat{\theta}^* = T_n(\mathcal{X}_n^*)$ es un estimador con varianza finita, entonces $T_n(\cdot)$ es un elemento de \mathcal{L}_n^2 y por tanto tiene un desarrollo de la forma

$$T_n(\mathcal{X}_n^*) = \sum_{k \in \mathbb{N}^n} a_k \mathbf{P}_k(\mathcal{X}_n^*). \quad (7.2)$$

donde $\{\mathbf{P}_k(\cdot)\}_{k \in \mathbb{N}^n}$ son polinomios multivariantes ortogonales respecto a la distribución conjunta de \mathcal{X}_n^* , donde $\mathbf{P}_0 \equiv 1$, y $a_k = \langle T_n, \mathbf{P}_k \rangle_{\mathcal{L}_n^2} / \|\mathbf{P}_k\|_{\mathcal{L}_n^2}^2$. El sesgo y la varianza ideales de $\hat{\theta}_n^* = T_n(\mathcal{X}_n^*)$ vienen dados por

$$\hat{s} = a_0 - t(F_n) \text{ y } \hat{v} = \sum_{k \in \mathbb{N}^n \setminus \{0\}} a_k^2 \|\mathbf{P}_k\|^2. \quad (7.3)$$

donde $t(\cdot)$ es el funcional que define la relación que hay entre la distribución de interés F desconocida y el parámetro de interés θ , i.e. $\theta = t(F)$. Esto significa que $t(F_n)$ en (7.3) puede ser reemplazada por $\hat{\theta}_n$ cuando este último viene dado por la misma relación, i.e. cuando $\hat{\theta}_n = t(F_n)$. Puesto que las componentes de \mathcal{X}_n^* son independientes, entonces la distribución conjunta de \mathcal{X}_n^* es (F_n, \dots, F_n) de dimensión n . Así, el sistema

$$\mathbf{P}_k(\mathcal{X}_n^*) = \prod_{j=1}^n P_{k_j}(X_j^*). \quad (7.4)$$

es un sistema ortogonal en \mathcal{L}_n^2 , donde, $k = (k_1, \dots, k_n)$ y para cada $j = 1, \dots, n$, P_{k_j} es el polinomio de grado k_j ortogonal respecto a F_n . La norma en \mathcal{L}_n^2 de los \mathbf{P}_k en (7.3) es

$$\|\mathbf{P}_k\|_{\mathcal{L}_n^2} = \prod_{j=1}^n \|P_{k_j}\|_{\mathcal{L}^2}.$$

En particular, tenemos

$$\begin{aligned} \mathbf{P}_0(s_1, \dots, s_n) &= 1, \\ \mathbf{P}_{e_j}(s_1, \dots, s_n) &= P_1(s_j), \quad j = 1, \dots, n, \\ \mathbf{P}_{2e_j}(s_1, \dots, s_n) &= P_2(s_j), \quad \mathbf{P}_{e_i+e_j}(s_1, \dots, s_n) = P_1(s_i)P_1(s_j). \end{aligned}$$

con normas

$$\begin{aligned} \|\mathbf{P}_0\| &= 1, \quad \|\mathbf{P}_{e_j}\| = \|P_1\|, \quad j = 1, \dots, n, \\ \|\mathbf{P}_{2e_j}\| &= \|P_2\|, \quad j = 1, \dots, n \\ \|\mathbf{P}_{e_i+e_j}\| &= \|\mathbf{P}_1\|^2, \quad \forall i, j \text{ con } i \neq j. \end{aligned}$$

7.1.2 Estimación bootstrap

En la expresión del sesgo y la varianza ideales en (7.3) las cantidades desconocidas son los coeficientes a_k . Ahora, vamos a seguir el mismo procedimiento, que en el caso paramétrico, para estimarlos consistentemente. Sea $\mathcal{X}_n^{*1}, \dots, \mathcal{X}_n^{*B}$, B muestras bootstrap cada una de tamaño n extraídas a partir de F_n . Escribimos a $\hat{\theta}_n^{*b}$, como

$$\hat{\theta}_n^{*b} = \beta_0 + \sum_{j=1}^m \sum_{|k|=j} \beta_k \mathbf{P}_k(\mathcal{X}_n^{*b}) + \varepsilon^b, \quad b = 1, \dots, B \quad (7.5)$$

donde ε es una variable aleatoria con media 0, varianza σ^2 e incorrelada con los \mathbf{P}_k . Tratamos, entonces, de aproximar los parámetros $\{\beta_k\}_k$ y σ^2 mediante el método de mínimos cuadrados.

Sea $\hat{\beta}_k^*$ y $\hat{\sigma}^{*2}$ la solución de mínimos cuadrados, dada por (6.12) con la matriz de diseño correspondiente a los polinomios \mathbf{P}_k , $|k| \leq m$, y que tiene un aspecto como en (6.38) cuando $n = 3$ y $m = 2$.

La aproximación natural a \hat{s} y \hat{v} vienen entonces dados, como en el caso paramétrico, por

$$\hat{s}_{Ort}^* = \hat{\beta}_0^* - t(F_n), \quad \text{y} \quad \hat{v}_{Ort}^* = \sum_{j=1}^m \sum_{|k|=j} \hat{\beta}_k^{*2} \|\mathbf{P}_k\|^2 + \hat{\sigma}^{*2}.$$

Por las mismas razones que en el caso paramétrico, \hat{s}^* y \hat{v}^* , mejoran sustancialmente a los estimadores bootstrap usuales. Para estimadores que tiene una aproximación lineal, un desarrollo hasta $m = 1$ basta para dar resultados aceptables. Mas formalmente, si $\hat{\theta}_n^* = L_n^* + R_n^*$ donde L_n^* es la parte lineal del desarrollo de $\hat{\theta}_n^*$. Entonces, si $var(L_n^*)/\hat{v}$ es próxima a uno, la elección de $m = 1$ en el desarrollo (7.5) basta para tener aproximaciones satisfactorias. Sin embargo, podemos siempre elegir un nivel m en el desarrollo dependiendo del estimador de interés usado. En muchas situaciones, la elección de $m=2$ es satisfactoria, como muestran los resultados numéricos.

7.1.3 Resultados numéricos

Con el fin de verificar numéricamente el procedimiento bootstrap ortogonal descrito arriba, hemos considerado una muestra, $\mathcal{X}_n = (X_1, \dots, X_n)$, de tamaño $n = 10$ procedente de una distribución $Exp(\alpha)$ con $\alpha = 1$. Vamos a aplicar el procedimiento sobre cuatro estadísticos simples para los cuales es posible calcular con exactitud el sesgo y la varianza bootstrap ideales. Este hecho nos permitirá ver con claridad la precisión del método bootstrap ortogonal comparando los resultados obtenidos con los resultados exactos y con los resultados obtenidos usando el método bootstrap estándar.

Los estadísticos considerados son $\hat{\mu} = T_n(X_1, \dots, X_n) = \bar{X}$ la media muestral como estimador de la media de la población $\mu = E(X_1)$, el estadístico $\hat{\sigma}^2 = T_n(X_1, \dots, X_n) = S_n^2$ la varianza muestral como estimador de la varianza de la población $\sigma^2 = Var(X_1)$, el estadístico $\hat{\theta} = T_n(X_1, \dots, X_n) = \bar{X}^2$ como estimador del parámetro $\theta = h(\mu) = \mu^2$ y el estadístico $\hat{\xi} = T_n(X_1, \dots, X_n) = \exp(-\bar{X})$ como estimador del parámetro $\xi = \exp(-\mu)$.

Hemos elegido 3 tipos de estadísticos: un estadístico lineal en los datos, \bar{X} , un estadístico cuadrático en los datos, S_n^2 o \bar{X}^2 , y un estadístico, $\exp(-\bar{X})$, con un desarrollo, (7.2), infinito en términos de polinomios ortogonales.

Consideremos el método bootstrap no paramétrico, donde las muestras bootstrap son extraídas de la función de distribución empírica F_n que pone probabilidad $\frac{1}{n}$ en cada observación original X_i , $i = 1, \dots, n$. Sea $\mathcal{X}_n^* = (X_1^*, \dots, X_n^*)$ una muestra bootstrap, entonces para cada $i = 1, \dots, n$,

$$P^* \{X_i^* = X_j\} = \frac{1}{n}, \quad j = 1, \dots, n.$$

Para cada uno de los estadísticos citados consideremos los desarrollos siguientes

$$\begin{aligned}
 (L) T_n(\mathcal{X}_n^*) &= \beta_0 + \sum_{j=1}^n \beta_{e_j} \mathbf{P}_{e_j}(\mathcal{X}_n^*) + \varepsilon, \\
 (Q1) T_n(\mathcal{X}_n^*) &= \beta_0 + \sum_{j=1}^n \beta_{e_j} \mathbf{P}_{e_j}(\mathcal{X}_n^*) + \sum_{j=1}^n \beta_{2e_j} \mathbf{P}_{2e_j}(\mathcal{X}_n^*) + \varepsilon \quad y \\
 (Q2) T_n(\mathcal{X}_n^*) &= \beta_0 + \sum_{j=1}^n \beta_{e_j} \mathbf{P}_{e_j}(\mathcal{X}_n^*) + \sum_{j=1}^n \beta_{2e_j} \mathbf{P}_{2e_j}(\mathcal{X}_n^*) + \\
 &\quad \sum_{i=1}^{n-1} \sum_{j=i+1}^n \beta_{e_i+e_j} \mathbf{P}_{e_i+e_j}(\mathcal{X}_n^*) + \varepsilon.
 \end{aligned}$$

donde e_j es el j -ésimo vector canónico de \mathbb{N}^n . El desarrollo (L) es el desarrollo lineal del estadístico T_n , el desarrollo (Q1) es el desarrollo cuadrático incompleto y (Q2) es el desarrollo cuadrático completo de T_n . La expresión de los polinomios \mathbf{P}_k en el desarrollo está expresada en la sección anterior, (7.4), en términos de los polinomios ortogonales respecto a F_n . Ahora, vamos a aplicar el método bootstrap ortogonal con un número de muestras bootstrap $B = 70$.

a.) Sea $T_n(X_1^*, \dots, X_n^*) = \bar{X}_n^* = n^{-1} \sum_{i=1}^n X_i^*$, en este caso el sesgo y la varianza bootstrap ideales vienen dados, sencillamente, por

$$\hat{s} = 0 \quad y \quad \hat{v} = \frac{\hat{\sigma}^2}{n},$$

cuyo resultado observado a partir de la muestra es

$$\hat{s} = 0 \quad y \quad \hat{v} = 0.02908653113$$

b.) Sea $T_n(X_1^*, \dots, X_n^*) = \bar{X}_n^{*2}$. El sesgo y la varianza bootstrap ideales vienen dados por

$$\hat{s} = \frac{\hat{\sigma}^2}{n} \quad y \quad \hat{v} = \frac{4\bar{X}^2 \hat{\sigma}^2}{n} + \frac{4\bar{X} \hat{\mu}_3 + 2\hat{\sigma}^4}{n^2} + \frac{\hat{\mu}_4 - 3\hat{\sigma}^4}{n^3}.$$

donde $\hat{\mu}_i$ es el i -ésimo momento muestral centrado

$$\hat{\mu}_i = \frac{1}{n} \sum_{j=1}^n (X_j - \bar{X})^i, \quad i = 3, 4.$$

Los valores observados son

$$\hat{s} = 0.02908653113 \quad y \quad \hat{v} = 0.07622274$$

c.) Y para $T_n(X_1^*, \dots, X_n^*) = \exp(-\bar{X}_n^*)$, el sesgo y la varianza bootstrap ideales vienen dados por

$$\hat{s} = \left(\frac{1}{n} \sum_{j=1}^n \exp\left(-\frac{X_j}{n}\right) \right)^n - \exp(-\bar{X}), \quad y$$

$$\hat{v} = \left(\frac{1}{n} \sum_{j=1}^n \exp\left(-\frac{2X_j}{n}\right) \right)^n - \left(\frac{1}{n} \sum_{j=1}^n \exp\left(-\frac{X_j}{n}\right) \right)^{2n}.$$

Los valores observados son

$$\hat{s} = 0.0066 \quad y \quad \hat{v} = 0.0061$$

d.) Sea, ahora, $T_n(X_1^*, \dots, X_n^*) = S_n^{*2} = n^{-1} \sum_{i=1}^n (X_i^* - \bar{X}^*)^2$. El sesgo y la varianza bootstrap ideales vienen dados, en este caso, por

$$\hat{s} = -\frac{\hat{\sigma}^2}{n} \quad y \quad \hat{v} = \left(\frac{n-1}{n} \right)^2 \left(\frac{\hat{\mu}_4}{n} - \frac{n-3}{n(n-1)} \hat{\sigma}^4 \right).$$

Los valores observados son

$$\hat{s} = -0.029086 \quad y \quad \hat{v} = 0.00766$$

En la Tabla 7-1 resumimos los resultados en a.), b.), c.) y d.).

	Varianza Bootstrap Ideal	Sesgo Bootstrap Ideal
X	0.02908653113	0
X^2	0.07622274	0.02908653113
$\exp(-X)$	0.0061	0.0066
S_n^2	0.00766	-0.029086

Tabla 7.1

Los resultados de simulación para obtener una aproximación bootstrap a \hat{s} y a \hat{v} , correspondientes al método bootstrap ortogonal y al método bootstrap estandar para cada uno de los cuatro estadísticos considerados están resumidos en la Tabla

7.2.

		\hat{v}_{ort}^*	\hat{v}_{MC}^*	\hat{s}_{ort}^*	\hat{s}_{MC}^*
\bar{X}	(L)	0.029086531	0.0251	$0.3 \cdot 10^{-9}$	-0.023
	(Q1)	0.029086530		$0.19 \cdot 10^{-8}$	
	(Q2)	0.029086530		$0.17 \cdot 10^{-7}$	
\bar{X}^2	(L)	0.06563	0.06116	0.02763	-0.009
	(Q1)	0.06752		0.0289	
	(Q2)	0.07622273		0.0290865	
$\exp(-\bar{X})$	(L)	0.006825	0.00667	0.00642	0.0167
	(Q1)	0.0064		0.00674	
	(Q2)	0.006114		0.00643	
S_n^2	(L)	0.0103	0.00898	-0.02586	-0.023
	(Q1)	0.0087		-0.02896	
	(Q2)	0.007663		-0.02908653	

Tabla 7.2

En realidad, los resultados anteriores corresponden al grupo 1 de $M = 25$ grupos de $B = 70$ muestras bootstrap de tamaño n . Para cada grupo j de los $M = 25$ grupos hemos calculado \hat{s}_{ort}^{*j} , \hat{v}_{ort}^{*j} , \hat{s}_{MC}^{*j} y \hat{v}_{MC}^{*j} , $j = 1, \dots, 25$, para cada uno de los cuatro dichos estadísticos.

$$X_1, \dots, X_n \rightarrow \begin{cases} \left\{ \begin{array}{l} X_1^{*1,1}, \dots, X_n^{*1,1}, \\ \vdots \\ X_1^{*1,B}, \dots, X_n^{*1,B} \end{array} \right. & \rightarrow \hat{s}_{ort}^{*1}, \hat{v}_{ort}^{*1}, \hat{s}_{MC}^{*1} \text{ y } \hat{v}_{MC}^{*1} \\ \vdots & \vdots \\ \left\{ \begin{array}{l} X_1^{*M,1}, \dots, X_n^{*M,1}, \\ \vdots \\ X_1^{*M,B}, \dots, X_n^{*M,B} \end{array} \right. & \rightarrow \hat{s}_{ort}^{*M}, \hat{v}_{ort}^{*M}, \hat{s}_{MC}^{*M} \text{ y } \hat{v}_{MC}^{*M} \end{cases}$$

Después hemos calculado el error estandar correspondiente a cada una de las cuatro aproximaciones, es decir

$$ECM(\hat{s}_{ort}^*) = \frac{1}{M} \sum_{j=1}^M (\hat{s}_{ort}^{*j} - \hat{s})^2 \text{ y } ECM(\hat{v}_{ort}^*) = \frac{1}{M} \sum_{j=1}^M (\hat{v}_{ort}^{*j} - \hat{v})^2,$$

y

$$ECM(\hat{s}_{MC}^*) = \frac{1}{M} \sum_{j=1}^M (\hat{s}_{MC}^{*j} - \hat{s})^2 \text{ y } ECM(\hat{v}_{MC}^*) = \frac{1}{M} \sum_{j=1}^M (\hat{v}_{MC}^{*j} - \hat{v})^2.$$

También hemos calculado las razones:

$$Ratio_{sesg}^* = \frac{ECM(\hat{s}_{MC}^*)}{ECM(\hat{s}_{ort}^*)} \text{ y } Ratio_{var}^* = \frac{ECM(\hat{v}_{MC}^*)}{ECM(\hat{v}_{ort}^*)}.$$

Los resultados obtenidos son los de la Tabla 7.3.

		ECM(\hat{v}_{ort}^*)	ECM(\hat{v}_{MC}^*)	ECM(\hat{s}_{ort}^*)	ECM(\hat{s}_{MC}^*)	Ratio $_{var}^*$	Ratio $_{sesg}^*$
X	(L)	0.59 10 ⁻¹⁹	0.84 10 ⁻⁵	0.11 10 ⁻¹⁷	0.31 10 ⁻³	14.2 10 ¹³	28.2 10 ¹³
	(Q1)	0.92 10 ⁻¹⁹		0.38 10 ⁻¹⁷		91.3 10 ¹²	81.5 10 ¹²
	(Q2)	0.56 10 ⁻¹⁶		0.32 10 ⁻¹⁴		15 10 ¹⁰	9.6 10 ¹⁰
\bar{X}^2	(L)	0.2 10 ⁻⁴	1.2 10 ⁻⁴	0.15 10 ⁻⁴	7.8 10 ⁻⁴	6	52
	(Q1)	0.17 10 ⁻⁴		0.14 10 ⁻⁴		7.05	55.7
	(Q2)	10 ⁻¹⁶		0.2 10 ⁻¹⁴		12 10 ¹¹	39 10 ¹⁰
$e^{-\bar{X}}$	(L)	1.6 10 ⁻⁵	5.3 10 ⁻⁵	7.6 10 ⁻⁵	6.7 10 ⁻³	3.3	88.1
	(Q1)	1.5 10 ⁻⁵		7.2 10 ⁻⁵		3.5	93
	(Q2)	8.6 10 ⁻⁷		2.75 10 ⁻⁵		61.6	243.6
S_n^2	(L)	0.31 10 ⁻⁵	0.22 10 ⁻⁵	0.16 10 ⁻³	0.14 10 ⁻³	0.7	0.87
	(Q1)	10 ⁻⁶		0.23 10 ⁻⁴		2.2	6.08
	(Q2)	3.25 10 ⁻¹⁶		0.35 10 ⁻¹⁵		67.6 10 ¹⁰	4 10 ¹¹

Tabla 7.3

Para comprobar el procedimiento ortogonal, 100 muestras originales suplementarias de tamaño 10 fueron extraídas. Hemos llevado a cabo todos los cálculos hechos anteriormente para cada una de las 100 muestras originales. Es decir que hemos ejecutado el algoritmo para cada una de las 100 muestras originales. Esto nos ha llevado a obtener una sucesión de 100 razones del error estandar para el sesgo y para la varianza:

$$Ratio_{sesg}^{*1}, \dots, Ratio_{sesg}^{*100} \text{ y } Ratio_{var}^{*1}, \dots, Ratio_{var}^{*100}. \tag{7.6}$$

Hemos calculado las medias,

$$RatioMedio_{sesg}^* = \frac{1}{100} \sum_{j=1}^{100} Ratio_{sesg}^{*j} \text{ y } RatioMedio_{var}^* = \frac{1}{100} \sum_{j=1}^{100} Ratio_{var}^{*j}$$

correspondientes a cada uno de los cuatro estadísticos considerados. Los resultados

obtenidos son los registrados en la Tabla 7.4.

		$RatioMedio_{var}^*$	$RatioMedio_{sesq}^*$
\bar{X}	(L)	$2.3 \cdot 10^{14}$	$2 \cdot 10^{14}$
	(Q1)	$10.6 \cdot 10^{13}$	$4.4 \cdot 10^{13}$
	(Q2)	$8.3 \cdot 10^{11}$	$2.5 \cdot 10^{11}$
\bar{X}^2	(L)	4.5	41.4
	(Q1)	5.6	44
	(Q2)	$92 \cdot 10^{10}$	$34 \cdot 10^{10}$
$\exp(-\bar{X})$	(L)	4.2	42.9
	(Q1)	5.1	43.5
	(Q2)	82	233
S_n^2	(L)	0.8	1.2
	(Q1)	3.5	8.7
	(Q2)	$71 \cdot 10^{10}$	$52 \cdot 10^{10}$

Tabla 7.4

7.2 Desarrollos basados en la distribución Multinomial

Se trata del mismo procedimiento, sólo cambiamos de espacio en el que trabajamos, por lo que, se cambia el sistema ortogonal utilizado. Consideremos una muestra bootstrap $\mathcal{X}_n^* = (X_1^*, \dots, X_n^*)$ extraída de la distribución empírica F_n . Entonces cada observación X_j^* puede ser una de las observaciones originales X_i con probabilidad $1/n$. Por lo que definimos

$$N_i^* = \text{Card}\{j : X_j^* = X_i\},$$

el número de apariciones de la observación X_i en la muestra bootstrap. Entonces podemos ver claramente que el vector $\mathbf{N}^* = (N_1^*, \dots, N_{n-1}^*)^T$ sigue una ley multinomial

$$\mathbf{N}^* \rightsquigarrow \text{Mult}_{n-1} \left(n; \frac{1}{n}, \dots, \frac{1}{n} \right). \quad (7.7)$$

Notemos que hemos considerado \mathbf{N}^* con $n-1$ componentes, ya que, una vez conocidos los N_1^*, \dots, N_{n-1}^* , entonces $N_n^* = n - \sum_{i=1}^{n-1} N_i^*$. Entonces a cada muestra bootstrap \mathcal{X}_n^* se le puede asignar un vector multinomial \mathbf{N}^* de dimensión $n-1$. Sea φ esta

transformación, es decir, $\mathcal{X}_n^* = \varphi(\mathbf{N}^*)$. Entonces $\hat{\theta}_n^* = T_n(\mathcal{X}_n^*) = T_n \circ \varphi(\mathbf{N}^*)$, y notando por $S_n = T_n \circ \varphi$ podemos entonces considerar, a partir de ahora, a $\hat{\theta}_n^* = S_n(\mathbf{N}^*)$ como función de \mathbf{N}^* .

Ejemplo 7.1 Consideremos el estadístico $\hat{\theta}_n = \bar{X}_n$. Entonces podemos ver fácilmente que la replica $\hat{\theta}_n^*$, de $\hat{\theta}_n$, correspondiente a una muestra bootstrap $X_n^* = (X_1^*, \dots, X_n^*)$, puede expresarse como

$$\hat{\theta}_n^* = \frac{1}{n} \sum_{i=1}^n N_i^* X_i = \frac{1}{n} \sum_{i=1}^{n-1} N_i^* X_i + (n - \sum_{j=1}^{n-1} N_j^*) X_n.$$

De otra forma,

$$\hat{\theta}_n^* = \mathcal{X}_{n-1}^T \mathbf{N}^* + (n - \mathbf{1}^T \mathbf{N}^*) X_n, \quad (7.8)$$

donde $\mathbf{1}$ es el vector de dimensión $n - 1$ cuyos componentes son todos igual a 1. Notemos que, en (7.8), la única cantidad aleatoria es \mathbf{N}^* , donde X_n es fija. En este caso, la transformación $S_n(\cdot)$ viene dada por

$$S_n(\mathbf{U}) = \mathcal{X}_{n-1}^T \mathbf{U} + (n - \mathbf{1}^T \mathbf{U}) X_n,$$

para cada vector de probabilidad \mathbf{U} de dimensión $n - 1$, y se verifica que

$$\hat{\theta}_n^* = S_n(\mathbf{N}^*) = T_n(\mathcal{X}_n^*),$$

para cada muestra bootstrap X_n^* . Notemos que puesto que $T_n(\cdot)$ es lineal en los datos X_n^* , entonces $S_n(\cdot)$ también lo es en \mathbf{N}^* .

Consideremos el estadístico, $\hat{\theta}_n = \frac{1}{n} \sum_{j=1}^n (X_j - \bar{X})^2$, la varianza muestral, que es cuadrático en los datos. Un calculo elemental da, para \mathbf{N}^* con dimensión n ,

$$\hat{\theta}_n^* = S_n(\mathbf{N}^*) = \frac{n-1}{n^2} \sum_{i=1}^n X_i^2 N_i^{*2} - \frac{2}{n^2} \sum_{i=1}^{n-1} \sum_{j=i+1}^n X_i X_j N_i^* N_j^*. \quad (7.9)$$

Para expresar a $\hat{\theta}_n^*$ solo en términos de N_1^*, \dots, N_{n-1}^* , basta reemplazar N_n^* por $n - \sum_{i=1}^{n-1} N_i^*$ en la expresión (7.9). Notemos también que, $T_n(\cdot)$ es cuadrático en los datos, y $S_n(\cdot)$ es cuadrático en \mathbf{N}^* .

Consideremos el espacios de Hilbert definido por

$$\mathcal{L}_{Mult_{n-1}}^2 = \left\{ T : E_{Mult} [T^2(\mathbf{N})] < \infty \text{ donde } \mathbf{N} \rightsquigarrow Mult_{n-1} \left(n, \frac{1}{n}, \dots, \frac{1}{n} \right) \right\}.$$

donde $E_{Mult}(\cdot)$ es la esperanza respecto a la distribución multinomial mencionada. Entonces $\mathcal{L}_{Mult_{n-1}}^2$ es simplemente el espacio de funciones con cuadrado integrable respecto a la función de probabilidad multinomial de \mathbf{N}^* en (7.7). Entonces en $\mathcal{L}_{Mult_{n-1}}^2$ existe un sistema de polinomios multivariantes ortogonales $\{\mathbf{Q}_k(s)\}_{k \in \mathbf{N}^{n-1}}$, donde cada \mathbf{Q}_k es de grado $|k|$ a lo sumo, sin pérdida de generalidad podemos escoger $\mathbf{Q}_0(\cdot) \equiv 1$.

Si el estimador $\hat{\theta}_n^* = S_n(\mathbf{N}^*)$ tiene varianza finita entonces $S_n(\cdot)$ es un elemento de $\mathcal{L}_{Mult_{n-1}}^2$, y se verifica

$$\hat{\theta}^* = a_0 + \sum_{k \in \mathbf{N}^{n-1} \setminus \{0\}} a_k \mathbf{Q}_k(\mathbf{N}^*), \quad (7.10)$$

con $a_k = \langle S_n, \mathbf{Q}_k \rangle / \|\mathbf{Q}_k\|^2$, donde la norma y el producto escalar son definidos como es usual en $\mathcal{L}_{Mult_{n-1}}^2$. A partir de (7.10), el sesgo \hat{s} y la varianza \hat{v} bootstrap ideales tienen la expresión

$$\hat{s} = a_0 - t(F_n) \text{ y } \hat{v} = \sum_{k \in \mathbf{N}^{n-1} \setminus \{0\}} a_k^2 \|\mathbf{Q}_k\|^2, \quad (7.11)$$

donde las cantidades desconocidas son los coeficientes de Fourier a_k . El funcional $t(\cdot)$ es el funcional que define la relación que hay entre θ y F , es decir, $\theta = t(F)$. Una aproximación natural de \hat{s} y \hat{v} sería sustituir los coeficientes a_k por sus estimaciones en (7.11). Para ello, recurrimos al procedimiento descrito en las secciones anteriores, donde se considera el método de mínimos cuadrados. Primero, truncamos el desarrollo en (7.10) y escribimos

$$\hat{\theta}_n^{*b} = \beta_0 + \sum_{j=1}^m \sum_{|k|=j} \beta_k \mathbf{Q}_k(\mathbf{N}^{*b}) + \varepsilon^b, \quad (7.12)$$

para cada muestra bootstrap \mathcal{X}_n^{*b} , $b = 1, \dots, B$, donde, como antes, ε^b son realizaciones de una variable aleatoria ε cuya media 0 y varianza σ^2 , e incorrelada con los \mathbf{Q}_k , $|k| \leq m$, siendo m un entero elegido adecuadamente. En términos de (7.12), el sesgo \hat{s} y la varianza \hat{v} bootstrap ideales tienen la expresión

$$\hat{s} = \beta_0 - t(F_n) \text{ y } \hat{v} = \sum_{j=1}^m \sum_{|k|=j} \beta_k^2 \|\mathbf{Q}_k\|^2 + \sigma^2.$$

Sea $\hat{\beta}_k^*$ y $\hat{\sigma}^{*2}$ la estimación de mínimos cuadrados dada en (6.12), pero con matriz de diseño formada por $\mathbf{Q}_k(\mathbf{N}^{*b})$, entonces una aproximación a \hat{s} y \hat{v} es

$$\hat{s}^* = \hat{\beta}_0^* - t(F_n) \text{ y } \hat{v}^* = \sum_{j=1}^m \sum_{|k|=j} \hat{\beta}_k^{*2} \|\mathbf{Q}_k\|^2 + \hat{\sigma}^{*2}.$$

Por las mismas razones que en la primera sección del capítulo anterior (ver por ejemplo (6.26)), las aproximaciones \hat{s}^* y \hat{v}^* mejoran sustancialmente a los estimadores usuales de Monte Carlo. Esto, también lo comprobaremos mediante resultados numéricos más abajo. Antes, vamos a ver cómo podemos obtener el sistema ortogonal $\{Q_k\}_k$ en el espacio $\mathcal{L}_{Mult_{n-1}}^2$.

7.2.1 Sistema ortogonal en la familia Multinomial

El método que vamos a seguir ahora para construir un sistema ortogonal, es aplicable a las familias exponenciales multivariantes donde la familia *Multinomial es una de ellas*. Para más detalles sobre este procedimiento se puede consultar POMMERET (1996). Notemos por $f_X(\mathbf{x}, \mathbf{m})$ la función de probabilidad de una variable aleatoria X con distribución multinomial de dimensión d , $Mult_d(n, m_1, \dots, m_d)$, parametrizada por su media \mathbf{m} . Entonces $f_X(\mathbf{x}, \mathbf{m})$ viene dada por

$$f_X(\mathbf{x}, \mathbf{m}) = \frac{n!n^n}{x_1! \dots x_d! \left(n - \sum_{i=1}^d x_i\right)!} m_1^{x_1} \dots m_d^{x_d-1} \left(n - \sum_{i=1}^d m_i\right)^{n - \sum_{i=1}^d x_i}. \quad (7.13)$$

Sea $V(\mathbf{m})$ la varianza de X , entonces $V(\mathbf{m})$ viene dada por

$$V(\mathbf{m}) = -\frac{1}{n} \mathbf{m} \mathbf{m}^T + \text{diag}(m_1, \dots, m_d).$$

Sea \mathbf{m}_0 un punto del espacio de medias de X , por ser $V(\mathbf{m}_0)$ definida positiva, existe una matriz A cuadrada no singular tal que $AV(\mathbf{m}_0)A^T = D(\mathbf{m}_0)$ sea diagonal. Consideremos la variable aleatoria $Y = AX$, entonces Y tiene la media $\mathbf{u} = A\mathbf{m}$ y la matriz de varianza $W(\mathbf{u}) = D(A^{-1}\mathbf{u})$ diagonal en el punto $\mathbf{u}_0 = A\mathbf{m}_0$. La función de probabilidad de Y viene dada por

$$g_Y(\mathbf{y}, \mathbf{u}) = f_X(A^{-1}\mathbf{y}, A^{-1}\mathbf{u}). \quad (7.14)$$

Para $k \in \mathbb{N}^d$, consideremos las funciones

$$P_k(\mathbf{y}, \mathbf{u}_0) = \frac{\partial^{|k|}}{\partial \mathbf{u}^k} g_Y(\mathbf{y}, \mathbf{u}) \Big|_{\mathbf{u}=\mathbf{u}_0} / g_Y(\mathbf{y}, \mathbf{u}_0), \quad (7.15)$$

donde la notación $\mathbf{u}^k = u_1^{k_1} \dots u_d^{k_d}$. Entonces

$$Q_k(\mathbf{x}) = P_k(\mathbf{y}, \mathbf{u}_0) = P_k(A\mathbf{x}, \mathbf{u}_0) \quad (7.16)$$

es un polinomio de grado $|k|$ en \mathbf{x} , ortogonal respecto a $f_X(\mathbf{x}, \mathbf{m}_0)$. En particular, estos polinomios verifican la relación de recurrencia siguiente

$$\mathbf{P}_{k+e_j}(\mathbf{y}, \mathbf{u}) = \frac{\partial}{\partial u_j} \mathbf{P}_k(\mathbf{y}, \mathbf{u}) + \mathbf{P}_k(\mathbf{y}, \mathbf{u}) \mathbf{P}_{e_j}(\mathbf{y}, \mathbf{u}), \quad (7.17)$$

$$\mathbf{P}_{e_j}(\mathbf{y}, \mathbf{u}) = [W^{-1}]_j(\mathbf{y} - \mathbf{u}),$$

donde $[W^{-1}]_j$ es la j -ésima fila de la matriz $[W(\mathbf{u})]^{-1}$. La norma del polinomio \mathbf{Q}_k viene dada por

$$\|\mathbf{Q}_k\|^2 = \|\mathbf{P}_k(\cdot, \mathbf{u}_0)\|^2 = k! c_k (\text{diag}(AV(\mathbf{m}_0)A^T))^{-k}, \quad (7.18)$$

para cada $k \in \mathbb{N}^d$, donde $k! = k_1! \dots k_d!$, y

$$c_k = \prod_{j=0}^{|k|-1} \left(1 - \frac{j}{n}\right).$$

Nota 7.1 Puesto que en la definición de $f_X(\mathbf{x}, \mathbf{m})$ en (7.13) el término

$$\frac{n!n^n}{x_1! \dots x_d! \left(n - \sum_{i=1}^d x_i\right)!}$$

es independiente de \mathbf{m} , entonces, si definimos

$$f_X^0(\mathbf{x}, \mathbf{m}) = m_1^{x_1} \dots m_d^{x_d} \left(n - \sum_{i=1}^d m_i\right)^{n - \sum_{i=1}^d x_i}, \quad y$$

$$g_Y^0(\mathbf{y}, \mathbf{u}) = f_X^0(A^{-1}\mathbf{y}, A^{-1}\mathbf{u}),$$

entonces, la definición de $\mathbf{P}_k(\mathbf{y}, \mathbf{u})$ en (7.15) no cambia si sustituimos $g_Y(\mathbf{y}, \mathbf{u})$ por $g_Y^0(\mathbf{y}, \mathbf{u})$. Es decir,

$$\mathbf{P}_k(\mathbf{y}, \mathbf{u}_0) = \frac{\partial^{|\mathbf{k}|}}{\partial \mathbf{u}^{\mathbf{k}}} g_Y^0(\mathbf{y}, \mathbf{u}) \Big|_{\mathbf{u}=\mathbf{u}_0} / g_Y^0(\mathbf{y}, \mathbf{u}_0). \quad (7.19)$$

En el caso de nuestra variable *Multinomial* \mathbf{N}^* , donde $d = n - 1$, la media \mathbf{m}_0 es el vector de dimensión $n - 1$ cuyos componentes son todos iguales a 1, ya que, $E^* N_j^* = 1$. La matriz de varianza covarianza es la matriz $(n - 1) \times (n - 1)$ dada por

$$V = \begin{bmatrix} \frac{n-1}{n} & -\frac{1}{n} & \dots & -\frac{1}{n} \\ -\frac{1}{n} & \frac{n-1}{n} & \dots & -\frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{n} & -\frac{1}{n} & \dots & \frac{n-1}{n} \end{bmatrix} = \mathbf{I}_{n-1} - \frac{1}{n} \mathbf{1}\mathbf{1}^T.$$

Para encontrar un sistema ortogonal respecto a la función de probabilidad de \mathbf{N}^* , necesitamos encontrar una matriz A tal que AVA^T sea diagonal. Para ello, podemos considerar, por ejemplo, el método de Cholesky para buscar una matriz B tal que $V = BB^T$, que siempre existe puesto que V es simétrica y definida positiva. Luego escogeremos $A = B^{-1}$, y con esta elección se verifica

$$AVA^T = \mathbf{I}_{n-1}.$$

Ejemplo 7.2 Con el fin de ver este procedimiento, supongamos que $n = 3$, con esto tendremos $m_0 = (1, 1)$, y

$$V = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} \end{bmatrix}.$$

La matriz A dada por

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

verifica que

$$AVA^T = \begin{bmatrix} \frac{2}{3} & 0 \\ 0 & 2 \end{bmatrix}.$$

consideremos el vector aleatorio $Y = AX$, entonces, $u_0 = (2, 0)$, y

$$g_Y^0(\mathbf{y}, \mathbf{u}) = \left(\frac{u_1 + u_2}{2}\right)^{\frac{1}{2}[y_1 + y_2]} \left(\frac{u_1 - u_2}{2}\right)^{\frac{1}{2}[y_1 - y_2]} (3 - u_1)^{[3 - y_1]}.$$

Entonces, tendremos

$$\mathbf{P}_{e_1}(\mathbf{y}, \mathbf{u}) = \frac{y_1 + y_2}{4} \left(\frac{u_1 + u_2}{2}\right)^{-1} + \frac{y_1 - y_2}{4} \left(\frac{u_1 - u_2}{2}\right)^{-1} - (3 - y_1)(3 - u_1)^{-1},$$

y

$$\mathbf{P}_{e_2}(\mathbf{y}, \mathbf{u}) = \frac{y_1 + y_2}{4} \left(\frac{u_1 + u_2}{2}\right)^{-1} - \frac{y_1 - y_2}{4} \left(\frac{u_1 - u_2}{2}\right)^{-1}.$$

Por lo tanto

$$\mathbf{P}_{e_1}(\mathbf{y}, \mathbf{u}_0) = \frac{3}{2}y_1 - 3, \text{ y } \mathbf{P}_{e_2}(\mathbf{y}, \mathbf{u}_0) = \frac{1}{2}y_2.$$

Y a partir de la fórmula de recurrencia en (7.17) podemos calcular los polinomios de grado más alto. Tenemos

$$\begin{aligned}\frac{\partial}{\partial u_1} \mathbf{P}_{e_1}(\mathbf{y}, \mathbf{u}) &= -\frac{y_1 + y_2}{8} \left(\frac{u_1 + u_2}{2} \right)^{-2} - \frac{y_1 - y_2}{8} \left(\frac{u_1 - u_2}{2} \right)^{-2} - \\ &\quad (3 - y_1)(3 - u_1)^{-2}, \\ \frac{\partial}{\partial u_1} \mathbf{P}_{e_2}(\mathbf{y}, \mathbf{u}) &= -\frac{y_1 + y_2}{8} \left(\frac{u_1 + u_2}{2} \right)^{-2} + \frac{y_1 - y_2}{8} \left(\frac{u_1 - u_2}{2} \right)^{-2}, \text{ y} \\ \frac{\partial}{\partial u_2} \mathbf{P}_{e_2}(\mathbf{y}, \mathbf{u}) &= -\frac{y_1 + y_2}{8} \left(\frac{u_1 + u_2}{2} \right)^{-2} - \frac{y_1 - y_2}{8} \left(\frac{u_1 - u_2}{2} \right)^{-2}.\end{aligned}$$

Entonces, según (7.17),

$$\begin{aligned}\mathbf{P}_{e_1+e_2}(\mathbf{y}, \mathbf{u}_0) &= \frac{3}{4}y_1y_2 - \frac{7}{4}y_2, \\ \mathbf{P}_{2e_1}(\mathbf{y}, \mathbf{u}_0) &= \frac{9}{4}y_1^2 - \frac{33}{4}y_1 + 6, \text{ y} \\ \mathbf{P}_{2e_2}(\mathbf{y}, \mathbf{u}_0) &= \frac{1}{2}y_2^2 - \frac{1}{4}y_1.\end{aligned}$$

Usando la fórmula (7.18), obtendremos las normas

$$\|\mathbf{P}_{e_1}\|^2 = \frac{3}{2}, \quad \|\mathbf{P}_{e_2}\|^2 = \frac{1}{2}, \quad \|\mathbf{P}_{2e_1}\|^2 = 3, \quad \|\mathbf{P}_{2e_2}\|^2 = \frac{1}{3} \text{ y } \|\mathbf{P}_{e_1+e_2}\|^2 = \frac{1}{2}.$$

Notemos que, para expresar los $\mathbf{P}_k(\mathbf{y}, \mathbf{u}_0)$ en términos de la variable \mathbf{x} , con el fin de obtener $\mathbf{Q}_k(\mathbf{x})$ basta sustituir \mathbf{y} por $A\mathbf{x}$, (ya que $\mathbf{y} = A\mathbf{x}$). En el caso del ejemplo anterior, tenemos que sustituir y_1 por $x_1 + x_2$ y y_2 por $x_1 - x_2$. Sin embargo, en la aplicación del procedimiento, la sustitución no es necesaria, podemos trabajar directamente con los datos transformados $\mathbf{y} = A\mathbf{x}$.

7.2.2 Resultados numéricos

A.) La primera parte de esta subsección es la continuación de los resultados numéricos anteriores, donde los datos proceden de una distribución exponencial $Exp(\alpha)$. En esta sección, usaremos las mismas muestras originales y las mismas muestras bootstrap que las que han dado lugar a los resultados numéricos registrados en las Tablas de la sección anterior. Lo único que hemos cambiado es el sistema de polinomios ortogonales, donde usamos ahora el sistema ortogonal respecto a la Multinomial

en lugar del sistema ortogonal respecto a la distribución empírica. Los resultados de simulación que corresponden al primer grupo de muestras bootstrap, usando la primera muestra original, están registrados en la Tabla 7.5, que es una versión de la tabla 7.2

		\hat{v}_{ort}^*	\hat{v}_{MC}^*	\hat{s}_{ort}^*	\hat{s}_{MC}^*
\bar{X}	(L)	0.029086531	0.0251	10^{-10}	-0.023
	(Q1)	0.029086531		$0.4 \cdot 10^{-9}$	
	(Q2)	0.029086533		$0.6 \cdot 10^{-8}$	
\bar{X}^2	(L)	0.0663	0.06116	0.02782	-0.009
	(Q1)	0.0728		0.0289	
	(Q2)	0.07622274		0.0290865	
$\exp(-\bar{X})$	(L)	0.00677	0.00667	0.00606	0.0167
	(Q1)	0.0063		0.0061	
	(Q2)	0.00605		0.0065	
S_n^2	(L)	0.0101	0.00898	-0.0262	-0.023
	(Q1)	0.0086		-0.028	
	(Q2)	0.007662		-0.02908652	

Tabla 7.5

La tabla de los errores cuadráticos medios correspondiente a la primera muestra original, que es la analoga de la Tabla 7.3, es la Tabla 7.6 siguiente

		$ECM(\hat{v}_{ort}^*)$	$ECM(\hat{v}_{MC}^*)$	$ECM(\hat{s}_{ort}^*)$	$ECM(\hat{s}_{MC}^*)$	$Ratio_{var}^*$	$Ratio_{sesg}^*$
X	(L)	$0.85 \cdot 10^{-21}$	$0.84 \cdot 10^{-5}$	$0.36 \cdot 10^{-19}$	$0.31 \cdot 10^{-3}$	$98 \cdot 10^{14}$	$86 \cdot 10^{14}$
	(Q1)	$0.127 \cdot 10^{-19}$		$0.2 \cdot 10^{-17}$		$66 \cdot 10^{13}$	$15.5 \cdot 10^{13}$
	(Q2)	$0.19 \cdot 10^{-17}$		$0.58 \cdot 10^{-16}$		$15 \cdot 10^{10}$	$9.6 \cdot 10^{10}$
\bar{X}^2	(L)	$0.18 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$	$0.146 \cdot 10^{-4}$	$7.8 \cdot 10^{-4}$	6.66	53.2
	(Q1)	$0.11 \cdot 10^{-4}$		$0.13 \cdot 10^{-4}$		10.9	60
	(Q2)	$0.41 \cdot 10^{-17}$		$0.38 \cdot 10^{-16}$		$29.2 \cdot 10^{12}$	$2 \cdot 10^{13}$
$e^{-\bar{X}}$	(L)	$0.15 \cdot 10^{-5}$	$5.3 \cdot 10^{-5}$	$7.55 \cdot 10^{-5}$	$6.7 \cdot 10^{-3}$	3.5	88.7
	(Q1)	$0.71 \cdot 10^{-7}$		$7.1 \cdot 10^{-5}$		7.46	94.36
	(Q2)	$0.29 \cdot 10^{-8}$		$7.3 \cdot 10^{-6}$		182.7	917
S_n^2	(L)	$0.15 \cdot 10^{-5}$	$0.22 \cdot 10^{-5}$	$0.146 \cdot 10^{-4}$	$0.14 \cdot 10^{-3}$	1.46	9.6
	(Q1)	$0.72 \cdot 10^{-6}$		$0.13 \cdot 10^{-4}$		3.05	10.77
	(Q2)	$0.53 \cdot 10^{-19}$		$0.65 \cdot 10^{-15}$		$41.5 \cdot 10^{12}$	215 10^{11}

Tabla 7.6

La aplicación del procedimiento a 100 muestras originales exponenciales nos ha llevado a obtener 100 ratios correspondientes a la estimación de la varianza y del sesgo como en (7.6) para cada uno de los desarrollos (L), (Q1) y (Q2). La Tabla 7.7

muestra las medias de dichos ratios para cada caso.

		$RatioMedio^*_{var}$	$RatioMedio^*_{sesg}$
\bar{X}	(L)	$16.4 \cdot 10^{15}$	$7.1 \cdot 10^{15}$
	(Q1)	$14 \cdot 10^{14}$	$5 \cdot 10^{14}$
	(Q2)	$9.6 \cdot 10^{12}$	$6.3 \cdot 10^{12}$
\bar{X}^2	(L)	5.1	38.4
	(Q1)	14.7	52.6
	(Q2)	$37.5 \cdot 10^{12}$	$41.3 \cdot 10^{12}$
$\exp(-\bar{X})$	(L)	4.6	45.5
	(Q1)	10.4	67.1
	(Q2)	221.6	1076.8
S_n^2	(L)	2.2	15.7
	(Q1)	5.4	30.1
	(Q2)	$36.2 \cdot 10^{12}$	$29.4 \cdot 10^{12}$

Tabla 7.7

B.) Ocho sujetos han usado parches medicinales con el fin de infundir cierta hormona natural en la corriente sanguínea. Se ha medido el nivel de sangre de cada uno de los ocho sujetos después de usar tres diferentes parches: Parche P0 que no contiene ninguna hormona, Parche PA que fue fabricado según un plan antiguo, y Parche PN que ha sido fabricado recientemente según un nuevo plan. Las tres primeras columnas de la Tabla 4.1 muestran las medidas de nivel de sangre de los ocho sujetos.

Sujeto	P0	PA	PN	$z=PA-P0$	$y=PN-PA$
1	9243	17649	16449	8406	-1200
2	9671	12013	14614	2342	2601
3	11792	19979	17274	8187	-2705
4	13357	21816	23798	8459	1982
5	9055	13850	12560	4795	-1290
6	6290	9806	10157	3516	351
7	12412	17208	16570	4796	-638
8	18806	29044	26325	10238	-2719
Media				6342	-452.3

Tabla 7.8

El objetivo del experimento es mostrar la bioequivalencia entre PA y PN. El parche fabricado según el plan antiguo fue aprobado para la venta por la FDA "Food

and Drug Administration". El parche fabricado según el nuevo proyecto no necesita una nueva investigación de la FDA. Sería aprobado para la venta si se pudiera probar que es bioequivalente al antiguo. El criterio de bioequivalencia para la FDA es:

$$\frac{|E(PN) - E(PA)|}{E(PA) - E(P0)} \leq 0.20.$$

Sea entonces θ el parámetro definido por

$$\theta = \frac{E(PN) - E(PA)}{E(PA) - E(P0)}. \quad (7.20)$$

En este ejemplo consideremos la estimación del sesgo y el error estándar del estimador por substitución $\hat{\theta}$ de θ . Nos interesamos en dos estadísticos Z e Y obtenidos para cada uno de los ocho sujetos de la forma:

$$Z = \text{medición bajo } PA - \text{medición bajo } P0,$$

y

$$Y = \text{medición bajo } PN - \text{medición bajo } PA.$$

Supongamos que el par $X_i = (Z_i, Y_i)$ es obtenido por muestreo aleatorio a partir de una distribución bivalente desconocida F , y que Y e Z son independientes entre sí. Entonces la muestra de interés \mathcal{X}_n consiste en $n = 8$ observaciones, y θ en (7.20) viene definido como

$$\theta = t(F) = \frac{E_F(Y)}{E_F(Z)}.$$

En este caso, $t(\cdot)$ es el funcional que a cada distribución de probabilidad bivalente (Z, Y) le hace corresponder el cociente de las esperanzas. La estimación por substitución de θ es entonces:

$$\hat{\theta} = t(\hat{F}) = \frac{\bar{Y}}{\bar{Z}} = \frac{\sum_1^8 Y_i/8}{\sum_1^8 Z_i/8} = T(\mathcal{X}_n).$$

Las dos ultimas columnas de la Tabla anterior muestran las Z_i e Y_i para cada uno de los ocho sujetos. El valor esperado de θ es entonces:

$$\hat{\theta} = \frac{-452.3}{6342} = -0.0713.$$

Observemos que $|\hat{\theta}|$ es considerablemente inferior a 0.2, entonces hay, posibilidad de satisfacer la condición de bioequivalencia para la FDA. Queda saber la precisión de nuestra estimación.

Para ello, se ejecutan $B = 400$ simulaciones de Monte Carlo y se calculan las $B = 400$ repeticiones del estadístico $\hat{\theta}$ correspondientes a cada muestra bootstrap. Dichas 400 repeticiones han dado una estimación del error estándar $\widehat{se}_{MC,400}^* = 0.105$ y la media de las repeticiones $\bar{\theta}^* = -0.067$. Las estimaciones bootstrap del sesgo y del error estándar basada en las fórmulas (4.9) y (4.11) respectivamente, han dado

$$\hat{s}_{MC,400}^* = 0.0043 \quad \text{y} \quad \widehat{se}_{MC,400}^* = 0.105.$$

El cociente entre la estimación del sesgo y del error estándar $\hat{s}_{MC,400}^*/\widehat{se}_{MC,400}^* = 0.041$ es pequeño, y según la regla (ver EFRON & TIBSHIRANI (1993)), un sesgo menor que 0.25 por error estándar puede ser ignorado. Esto indica que no tenemos que preocuparnos sobre el sesgo de $\hat{\theta}$.

Para ver como son los resultados obtenidos respecto a la realidad, es decir, como son respecto al sesgo y la varianza bootstrap ideales, vamos a ejecutar el método de Monte Carlo con un número bastante grande de simulaciones $B = 10000$. Hemos obtenido los siguiente resultados que son considerados como estimaciones ideales:

$$\hat{s} \simeq \hat{s}_{MC,10000}^* = 0.0084,$$

y

$$\hat{v} \simeq \hat{v}_{MC,10000}^* = 0.01042.$$

Por lo tanto, el error estándar,

$$\widehat{se} \simeq \widehat{se}_{MC,10000}^* = 0.102.$$

Notemos que el sesgo ideal \hat{s} es considerablemente superior a la estimación obtenida con $B = 400$, $\hat{s}_{MC,400}^* = 0.0043$.

Por otra parte, los cálculos basados sobre el método centralizado de EFRON (1990) que mejora considerablemente la estimación del sesgo para los estimadores que tienen una aproximación lineal como es en nuestro caso, han dado la estimación siguiente del sesgo con $B = 400$:

$$\hat{s}_{Cent}^* = 0.008.$$

Esta última estimación está más cerca de la estimación ideal que la estimación de Monte Carlo estándar con $B = 400$, $\hat{s}_{MC,400}^* = 0.0043$. La estimación delta de la varianza nos ha dado

$$\hat{v}_{delta} = 0.095,$$

y por lo tanto

$$\widehat{se}_{delta} = 0.0974.$$

que es sensiblemente inferior a la estimación ideal.

Con el fin de comparar los resultados anteriores a los resultados obtenidos mediante nuestro método ortogonal hemos simulado 25 grupos, donde cada grupo está formado por $B = 400$ muestras bootstrap. Hemos calculado la varianza y el sesgo de $\hat{\theta}$ correspondientes a cada grupo usando el método de Monte Carlo y el método ortogonal basado en cada uno de los desarrollos (L), (Q1) y (Q2). La Tabla 7.9 recoge los resultados correspondientes al primer grupo.

		\hat{v}_{ort}^*	\hat{v}_{MC}^*	\hat{s}_{ort}^*	\hat{s}_{MC}^*
$\frac{\bar{Y}}{\bar{Z}}$	(L)	0.01011	0.012	0.00781	0.00376
	(Q1)	0.01015		0.00788	
	(Q2)	0.01028		0.00811	

Tabla 7.9

Siguiendo el razonamiento de los apartados anteriores, hemos calculado el error cuadrático medio correspondiente a las 25 estimaciones obtenidas para cada uno de los métodos mencionados. Los resultados obtenidos son los registrados en la Tabla 7.10

		$ECM(\hat{v}_{ort}^*)$	$ECM(\hat{v}_{MC}^*)$	$ECM(\hat{s}_{ort}^*)$	$ECM(\hat{s}_{MC}^*)$	$Ratio_{var}^*$	$Ratio_{sesg}^*$
$\frac{\bar{Y}}{\bar{Z}}$	(L)	$0.108 \cdot 10^{-6}$	$0.803 \cdot 10^{-6}$	$0.3 \cdot 10^{-5}$	$0.23 \cdot 10^{-4}$	7.43	7.76
	(Q1)	$0.75 \cdot 10^{-7}$		$0.25 \cdot 10^{-5}$		10.7	9.32
	(Q2)	$0.131 \cdot 10^{-7}$		$0.14 \cdot 10^{-5}$		61.7	16.6

Tabla 7.10

Comentario:

Sea $\hat{\theta}_n^*$ un estadístico cualquiera que admite un desarrollo de la forma

$$\hat{\theta}_n^* = a_0 + \sum_{k \in \mathbb{N}^n} a_k \mathbf{P}_k(\mathcal{X}_n^*) \circ \hat{\theta}_n^* = a_0 + \sum_{k \in \mathbb{N}^{n-1}} a_k \mathbf{Q}_k(\mathbf{N}^*), \quad (7.21)$$

donde $\mathcal{X}_n^* = (X_1^*, \dots, X_n^*)$ y $\{\mathbf{P}_k\}_{k \in \mathbb{N}^n}$ es un sistema ortogonal en \mathcal{L}_n^2 , y $\mathbf{N}^* = (N_1^*, \dots, N_{n-1}^*)$ como en (7.7) y $\{\mathbf{Q}_k\}_{k \in \mathbb{N}^{n-1}}$ es un sistema ortogonal en $\mathcal{L}_{Mult_{n-1}}^2$.

El estadístico $\hat{\theta}_n^* = \bar{X}_n^*$ es una función lineal en los datos \mathcal{X}_n^* por lo que tiene un desarrollo finito hasta el grado 1. Es decir que $a_k = 0$, en los desarrollos en (7.21), para $|k| \geq 2$. A partir de las Tablas 7.3 y 7.6 podemos ver que para dicho estadístico, la consideración del desarrollo (L) es mejor que si se considera cualquier

otro desarrollo, para estimar el sesgo y la varianza bootstrap. En otras palabras, el error cuadrático medio cometido para aproximar el sesgo y la varianza bootstrap de \bar{X}^* es menor cuando se usa el desarrollo (L) que cuando se usa (Q1), (Q2) o cualquier otro desarrollo de grado superior para dicho estadístico.

Para los estadísticos $\hat{\theta}^* = \bar{X}^{*2}$ o $\hat{\theta}^* = S_n^{*2}$, que son cuadrático en los datos, podemos ver a partir de las mismas Tablas 7.3 y 7.6 que la estimación bootstrap ortogonal es muy precisa usando el desarrollo (Q2). También, se puede apreciar que el uso del desarrollo (Q1) mejora la estimación obtenida usando el desarrollo (L) que a su vez es mejor que la estimación obtenida usando el método estándar de Monte-Carlo.

Para el estadístico $\hat{\theta}_n^* = e^{\bar{X}^*}$ que tiene un desarrollo infinito (7.21) en términos de polinomios ortogonales, se ve a partir de dichas tablas que la estimación bootstrap ortogonal considerando el desarrollo (Q2) mejora la estimación que si se considera el desarrollo (Q1), y que esta última mejora la estimación que si se considera el desarrollo (L1). También habrá de esperar que la consideración de un desarrollo superior a (Q2) mejore la estimación bootstrap ortogonal, aunque se puede ver que la consideración de (Q2) es suficiente para obtener estimaciones satisfactorias.

Consideremos, por ejemplo, la aproximación del sesgo bootstrap ideal $\hat{s} = \text{sesg}^*(\hat{\theta}_n^*)$. Sabemos que el estimador habitual de Monte Carlo \hat{s}_{MC}^* , usando B simulaciones, es insesgado de \hat{s} , por lo que $ECM^*(\hat{s}_{MC}^*) = Var^*(\hat{s}_{MC}^*)$. Entonces, se puede ver fácilmente que

$$ECM^*(\hat{s}_{MC}^*) = \frac{c}{B},$$

donde

$$c = Var^*(\hat{\theta}^*).$$

Sea

$$R = \frac{ECM^*(\hat{s}_{MC}^*)}{ECM^*(\hat{s}_{ort}^*)}$$

la eficiencia del estimador bootstrap ortogonal \hat{s}_{ort}^* respecto al estimador de Monte Carlo estándar \hat{s}_{MC}^* . Entonces, tendremos

$$B \cdot ECM^*(\hat{s}_{MC}^*) = R \cdot B \cdot ECM^*(\hat{s}_{ort}^*).$$

Esto significa que la precisión en la aproximación de \hat{s} mediante \hat{s}_{ort}^* usando B simulaciones, es la misma precisión en la aproximación de \hat{s} mediante \hat{s}_{MC}^* pero usando BR simulaciones. Por ejemplo, a partir de la Tabla 7.6 podemos ver que la precisión que hemos obtenido en la aproximación del sesgo de la media muestral usando \hat{s}_{ort}^* con $B = 70$ simulaciones necesitaría $BR = 70 \times 86 \cdot 10^{14} = 602 \cdot 10^{15}$

simulaciones si usamos el estimador de Monte Carlo estandar \hat{s}_{MC}^* . En este caso particular el numero de simulaciones necesario para que \hat{s}_{MC}^* sea igual de preciso que \hat{s}_{ort}^* con $B = 70$, es demasiado grande de modo que en la práctica es imposible usar \hat{s}_{MC}^* con el fin de obtener la precisión que ofrece \hat{s}_{ort}^* con solo 70 simulaciones.

Capítulo 8

ESTIMACIÓN BOOTSTRAP DE LA DENSIDAD Y DE LA DISTRIBUCIÓN VÍA FÓRMULAS DE INVERSIÓN

8.1 Introducción

En este capítulo consideremos la aproximación bootstrap de la densidad y la distribución de la media muestral basándonos sobre un procedimiento libre de simulaciones. Dicho procedimiento está basado sobre las fórmulas de inversión que relacionan la distribución y la densidad de una variable aleatoria con su función característica.

Sea X_1, \dots, X_n una muestra aleatoria simple idénticamente distribuida con distribución F continua y con media $\mu = EX_1$ y varianza $\sigma^2 = \text{var}(X_1)$. Sea f y ψ respectivamente la función de densidad y la función característica de X_1 . Consideremos la variable aleatoria

$$Z_n = \frac{M_n - A\mu}{\sigma\sqrt{C}},$$

donde $M_n = \sum_{i=1}^n a_i X_i$ con $\{a_i\}_{i=1, \dots, n}$ es una sucesión de números reales, $A = \sum_{i=1}^n a_i$ y $C = \sum_{i=1}^n a_i^2$. Sea g_n y ψ_n respectivamente la función de densidad y la función características de Z_n . El objetivo de este capítulo es estimar g_n . En general cuando n es suficientemente grande y μ y σ son finitas, bajo determinadas condiciones para las $\{a_i\}_{i=1, \dots, n}$, el teorema central del límite nos da la solución, ya que, afirma que la variable Z_n tiende a la variable normal estándar. En otras palabras que g_n puede ser aproximada cuando n es bastante grande mediante la densidad normal

estándar. Sin embargo, esta aproximación es correcta solo asintóticamente, es decir cuando n tiende a infinito. En práctica el número de datos n suele ser moderado o no es suficientemente grande o puede ser bastante pequeño como para confiar en la aproximación ofrecida por el teorema central del límite. Hay que mencionar aquí, por ejemplo, que la densidad normal es simétrica respecto al origen, sin embargo, la densidad de Z_n no lo es en general. Por lo que, si n no es suficientemente grande, la aproximación normal puede resultar incorrecta. Una alternativa en tal situación es la aproximación bootstrap g_n^* a g_n , que es la densidad de la variable aleatoria

$$Z_n^* = \frac{M_n^* - A\bar{X}}{S_n\sqrt{C}},$$

que es la réplica de Z_n con muestra X_1^*, \dots, X_n^* extraída a de una distribución estimada de F , digamos \hat{F} , que puede ser por ejemplo la distribución empírica F_n o un estimador paramétrico $F_{\hat{\theta}}$ cuando $F = F_{\theta}$ pertenece a una familia paramétrica. \bar{X} y S_n^2 son la media y la varianza empíricas asociadas a la muestra original. En general g_n^* es al igual que g_n difícil de calcular exactamente. Pero la ventaja que tenemos en el caso bootstrap es que la distribución \hat{F} es conocida y por lo tanto podemos sacar tantas muestras como queramos con el fin de aproximar a g_n^* . Como hemos visto en capítulos anteriores, el método de Monte Carlo consiste en sacar B muestras bootstrap $X_1^{*b}, \dots, X_n^{*b}$ a partir de \hat{F} , $b = 1, \dots, B$. Después calcular las B repeticiones Z_n^{*b} , $b = 1, \dots, B$ de Z_n correspondientes a cada muestra bootstrap. finalmente aproximar a g_n con el histograma correspondiente a

$$Z_n^{*1}, Z_n^{*2}, \dots, Z_n^{*B}.$$

Para la aproximación de la densidad, en general, se necesita un número B bastante elevado de simulaciones, por ejemplo 2000, con el fin de obtener resultados aceptables. Y esto es el inconveniente del método de Monte Carlo.

8.2 Fórmulas de inversión

Recordemos en esta sección los resultados principales que constituyen las fórmulas de inversión. Dichas fórmulas establecen una relación entre la función de distribución o la función de densidad de una variable aleatoria y su función característica.

Teorema 23 Sea $F(x)$ la función de distribución de una variable aleatoria cuya función característica es $\psi(t)$. Sea a y b pertenecientes a \mathbb{R} . Entonces

$$\lim_{T \rightarrow +\infty} \frac{1}{2\pi} \int_{-T}^T \frac{e^{-ita} - e^{-itb}}{it} \psi(t) dt = \frac{1}{2} [F(b+) + F(b-)] - \frac{1}{2} [F(a+) + F(a-)].$$

Corolario 24 Si a y b son dos puntos de continuidad de F entonces

$$\lim_{T \rightarrow +\infty} \frac{1}{2\pi} \int_{-T}^T \frac{e^{-ita} - e^{-itb}}{it} \psi(t) dt = F(b) - F(a). \quad (8.1)$$

Corolario 25 Si $\psi(t)$ es absolutamente integrable, es decir si $\int_{-\infty}^{+\infty} |\psi(t)| dt < +\infty$, entonces $F(x)$ es derivable en \mathbb{R} y se verifica que

$$F'(x) = f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-itx} \psi(t) dt. \quad (8.2)$$

Para la demostración de estos últimos resultados se puede consultar, por ejemplo, a SHIRYAYEV A.N. (1984).

8.3 Aproximación de la densidad

Sea $\psi_n^*(t)$ la función característica de Z_n^* , sabemos que si

$$\int_{-\infty}^{+\infty} |\psi_n^*(t)| dt < +\infty, \quad (8.3)$$

entonces la densidad de g_n^* de Z_n^* viene dada por

$$g_n^*(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-itx} \psi_n^*(t) dt = \int_{-\infty}^{+\infty} e^{-2\pi itx} \psi_n^*(2\pi t) dt. \quad (8.4)$$

Entonces puesto que $\psi_n^*(t)$ puede ser evaluada, si podemos calcular o aproximar el integral de Fourier (8.4) para ciertos valores x de interés, podemos entonces obtener valores para la densidad g_n^* . Veremos en lo que sigue un procedimiento basado sobre la fórmula de trapecios y las transformadas discretas de Fourier para calcular el integral (8.4) en una sucesión de puntos x_j diferentes y así obtener una sucesión de valores bastante precisos para la densidad g_n^* .

Sea $\varphi^*(t)$ la función característica de la variable aleatoria X^* cuya función de distribución es \hat{F} . Entonces se tiene que

$$\psi_n^*(t) = \exp\left(-\frac{iA\bar{X}}{S_n\sqrt{C}}t\right) \prod_{j=1}^n \varphi^*\left(\frac{a_j t}{S_n\sqrt{C}}\right). \quad (8.5)$$

En particular si $a_j = n^{-1}$, $j = 1, \dots, n$ entonces

$$\psi_n^*(t) = \exp\left(-\frac{i\sqrt{n}\bar{X}}{S_n}t\right) \varphi^{*n}\left(\frac{t}{S_n\sqrt{n}}\right),$$

donde \bar{X} y S_n^2 son respectivamente la media y la varianza muestrales asociadas a la muestra original. Cuando \hat{F} es la distribución empírica como es usual en el caso del bootstrap no paramétrico entonces

$$\varphi^*(t) = E^*(e^{itX^*}) = \frac{1}{n} \sum_{j=1}^n e^{itX_j},$$

que es una expresión conocida. De la misma manera, si $F = F_\theta$ es miembro de una familia paramétrica con función característica $\varphi_\theta(t)$, y si se decide usar el bootstrap paramétrico entonces $\hat{F} = F_{\hat{\theta}}$ y $\varphi^*(t) = \varphi_{\hat{\theta}}(t)$ con $\hat{\theta}$ es un estimador consistente de θ , generalmente se utiliza el estimador de máxima verosimilitud. Por lo tanto la función $\psi_n^*(t)$ en (8.5) es totalmente conocida.

La condición (8.3) implica que la función $\exp(-2\pi itx)\psi_n^*(t)$ es integrable ya que $|\exp(-2\pi itx)| = 1$. Entonces, con la elección de un número T bastante grande podemos hacer la aproximación

$$g_n^*(x) \simeq \int_{-T}^{+T} e^{-2\pi itx} \psi_n^*(2\pi t) dt. \quad (8.6)$$

Para calcular este integral vamos a aplicar la formula de trapecios. Se tiene que

$$g_n^*(x) = h \sum_{k=-m+1}^{m-1} e^{-2\pi ikhx} \psi_n^*(2\pi kh) + \frac{h}{2} [e^{2\pi imhx} \psi_n^*(-2\pi mh) + e^{-2\pi imhx} \psi_n^*(2\pi mh)] \quad (8.7)$$

donde $h = T/m$ y m es un entero positivo. Esta ultima expresión es simplemente la fórmula de trapecios. Dicha fórmula también puede ser escrita como

$$g_n^*(x) = \frac{T}{m} \sum_{k=-m+1}^{m-1} e^{-\frac{2\pi ikTx}{m}} \psi_n^*\left(\frac{2\pi kT}{m}\right) + \frac{T}{2m} [e^{2\pi iTx} \psi_n^*(-2\pi T) + e^{-2\pi iTx} \psi_n^*(2\pi T)].$$

Entonces si evaluamos $g_n^*(x)$ en los puntos

$$x_j = \frac{j}{2T}, \quad j = -m, \dots, m,$$

obtendremos para todo j ,

$$g_n^*(x_j) = \frac{T}{m} \sum_{k=-m+1}^{m-1} e^{-\frac{2\pi ikj}{2m}} \psi_n^*\left(\frac{2\pi kT}{m}\right) + \frac{T}{2m} [e^{\pi ij} \psi_n^*(-2\pi T) + e^{-\pi ij} \psi_n^*(2\pi T)].$$

Si notamos por $w_n = \exp(2\pi i/n)$ entonces se obtiene

$$g_n^*(x_j) = \frac{T}{m} \sum_{k=-m+1}^{m-1} \psi_n^* \left(\frac{2\pi kT}{m} \right) w_{2m}^{-jk} + \frac{T}{2m} [\psi_n^*(-2\pi T) + \alpha \psi_n^*(2\pi T)] w_{2m}^{-jm},$$

con $\alpha = \exp(-2\pi i j) = \cos(-2\pi j) + i \sin(-2\pi j) = 1$, para cualquier entero j . Finalmente, podemos escribir, para cada $j = -m, \dots, m$,

$$f_{n,j}^* = g_n^*(x_j) = \frac{T}{m} \sum_{k=-m}^{m-1} \psi_{n,k}^* w_{2m}^{-jk}, \quad (8.8)$$

donde la sucesión $\psi_{n,k}^*$ viene dada por

$$\begin{cases} \psi_{n,-m}^* = \frac{1}{2} [\psi_n^*(-2\pi T) + \psi_n^*(2\pi T)], \\ \psi_{n,k}^* = \psi_n^* \left(\frac{2\pi kT}{m} \right), \end{cases} \quad k = -m+1, \dots, m-1.$$

Notando por $f_n^* = \{f_{n,j}^*\}_j$ y $\psi_n^* = \{\psi_{n,j}^*\}_j$, podemos observar que

$$f_n^* = \frac{2T}{\sqrt{2m}} \mathcal{F} \psi_n^*, \quad (8.9)$$

con \mathcal{F} la transformada discreta de Fourier. Así, la sucesión f_n^* es simplemente la transformada de Fourier multiplicada por $2T/\sqrt{2m} = T\sqrt{2/m}$ de la sucesión ψ_n^* . En este lugar podemos usar el algoritmo de la transformada de Fourier (FFT), Ver el Capítulo 3, para calcular de golpe las $2m$ elementos $f_{n,j}^*$, $j = -m, \dots, m-1$. Por consecuencia, obtendremos $2m$ puntos

$$P_j = (x_j, f_{n,j}^*), \quad j = -m, \dots, m-1, \quad (8.10)$$

correspondientes al gráfico de la función de densidad g_n^* de la variable aleatoria Z_n^* .

Para algún x cualquiera podemos obtener una aproximación $f_{n,x}^*$ a $g_n^*(x)$, basada sobre los puntos en (8.13) ya calculados. La aproximación de la densidad obtenida sera, para cualquier x ,

$$g_n^*(x) \simeq \begin{cases} 0 & \text{si } x < \frac{-m}{2T} \text{ o } x > \frac{m-1}{2T} \\ f_{n,x}^* = (2Tx - j)(f_{n,j+1}^* - f_{n,j}^*) + f_{n,j}^* & \text{si } \frac{j}{2T} \leq x < \frac{j+1}{2T} \end{cases} \quad (8.11)$$

La motivación de esta última aproximación y los detalles sobre el cálculo de $f_{n,x}^*$ seran dados en una sección más abajo cuando daremos una aproximación de la distribución G_n^* de Z_n^* , (Ver *Figura 8-6*). Veremos mediante los resultados numéricos en la siguiente subsección como de preciso es este método, comparándolo con la aproximación normal o el método de Monte Carlo.

8.3.1 Resultados numéricos

Nuestro objetivo en esta sección es comparar los resultados obtenidos mediante la aplicación del procedimiento de inversión explicado anteriormente con los resultados obtenidos aplicando el método bootstrap estandar (Simulación de Monte Carlo) o aplicando la aproximación normal. Para ello, hemos sacado una muestra X_1, \dots, X_n de tamaño $n = 20$ a partir de la distribución $F_\theta \equiv \text{Gamma}(\alpha, \theta)$ con $\alpha = 1$ y $\theta = 1$. Supongamos que queremos estimar la densidad g_n de Z_n donde $a_j = n^{-1}$ por todo $j = 1, \dots, n$. Sea \hat{F} la distribución estimada de F a partir de la muestra X_1, \dots, X_{20} . Sea X_1^*, \dots, X_{20}^* una muestra extraída a partir de \hat{F} y g_n^* la densidad de la replica Z_n^* de Z_n correspondiente a dicha muestra. g_n^* es la densidad bootstrap ideal que es la mejor aproximación, en el sentido bootstrap, para g_n . Vamos a considerar tanto el procedimiento paramétrico como el procedimiento no paramétrico.

Caso paramétrico: En este caso $\hat{F} = F_{\hat{\beta}}$ con $\hat{\beta} = (\hat{\alpha}, \hat{\theta})$ una estimación consistente de $\beta = (\alpha, \theta)$. En este ejemplo, escogeremos el estimador de máxima verosimilitud $(\hat{\alpha}, \hat{\theta}) = (\bar{X}^2/S_n^2, S_n^2/\bar{X})$. La densidad bootstrap ideal es exactamente

$$g_n^*(x) = \frac{n^{\frac{1}{2}} \hat{\alpha}^{\frac{n\alpha}{2}}}{\Gamma(n\hat{\alpha})} \left(\sqrt{n}x + n\sqrt{\hat{\alpha}} \right)^{n\alpha-1} \exp[-(\sqrt{n}\hat{\alpha}x + n\hat{\alpha})]. \quad (8.12)$$

Si suponemos que $\alpha = 1$ fuera conocido entonces la densidad ideal sería

$$g_n^*(x) = \frac{\sqrt{n}}{\Gamma(n)} (\sqrt{n}x + n)^{n-1} \exp(-\sqrt{n}x - n),$$

y es igual en este caso a la densidad $g_n(x)$ de Z_n . Generalmente, una expresión explícita exacta para g_n^* como en (8.12) es imposible de obtener. En la *Figura 8-1* podemos ver la representación de la densidad exacta g_n^* representada por la curva continua, la aproximación normal a g_n^* , representada por la curva discontinua y la aproximación bootstrap de g_n^* vía fórmulas de inversión, representada por los círculos. Por otro lado, el histograma representa la aproximación bootstrap estandar usando $B=2000$ simulaciones de Monte Carlo a partir de $F_{\hat{\beta}}$. Destacamos en dicha figura el espectacular cambio, en el aspecto del histograma al cambiar la anchura de la ventana $h = 0.25$ en la parte A. de la *Figura 8.1* a $h = 0.1$ en la parte B de la misma figura. Dicho cambio, es una de las desventajas del uso de los histogramas ya que una mala elección de la ventana h puede dar lugar a aproximación errónea de la densidad bootstrap.

En la *Figura 8.1* se representan las diferencias entre la densidad bootstrap ideal g_n^* (a la que estamos tratando de aproximar) y las tres aproximaciones de la misma: la aproximación bootstrap estandar (cuadros), la aproximación Normal (círculos)

y la aproximación via fórmulas de inversión (cruces). Se puede apreciar claramente que la aproximación via fórmula de inversión es mejor tanto de la aproximación bootstrap como de la aproximación Normal.

Usando una muestra diferente que a partir de la cual hemos obtenido los gráficos, hemos calculado aproximaciones para g_n^* , en once puntos, correspondientes a tres procedimientos diferentes: la aproximación normal, el método de Monte Carlo y el procedimiento objeto de este capítulo. En el caso de la aproximación bootstrap usando el método de Monte Carlo hemos usado 2000 simulaciones bootstrap y con el mismo conjunto de datos hemos dado dos aproximaciones para la densidad, considerando primero un histograma con ventana $h = 0.5$ luego $h = 0.25$. Los resultados numéricos están registrados en la *Tabla 8.1*. Notemos, entonces, como varían los valores de la densidad con la variación de la ventana h elegida para representar el histograma, lo que constituye un inconveniente de dicho procedimiento.

x	Ideal	Inversión	Bootstrap $h = 0.5$	Bootstrap $h = 0.25$	Normal
-5/3	0.0973	0.09751	0.0960	0.1360	0.0994
-4/3	0.1851	0.1856	0.2200	0.1760	0.1640
-1	0.2837	0.2848	0.3040	0.2770	0.2419
-2/3	0.3646	0.3658	0.3040	0.3360	0.3194
-1/2	0.3907	0.3916	0.3880	0.3920	0.3520
0	0.3972	0.3972	0.3160	0.3360	0.3989
1/2	0.3180	0.3173	0.2680	0.3200	0.3520
2/3	0.2824	0.2816	0.2680	0.3200	0.3194
1	0.2099	0.2092	0.1760	0.1680	0.2419
4/3	0.1454	0.1450	0.1760	0.1840	0.1640
5/3	0.0945	0.0944	0.1120	0.1360	0.0994

Tabla 8.1

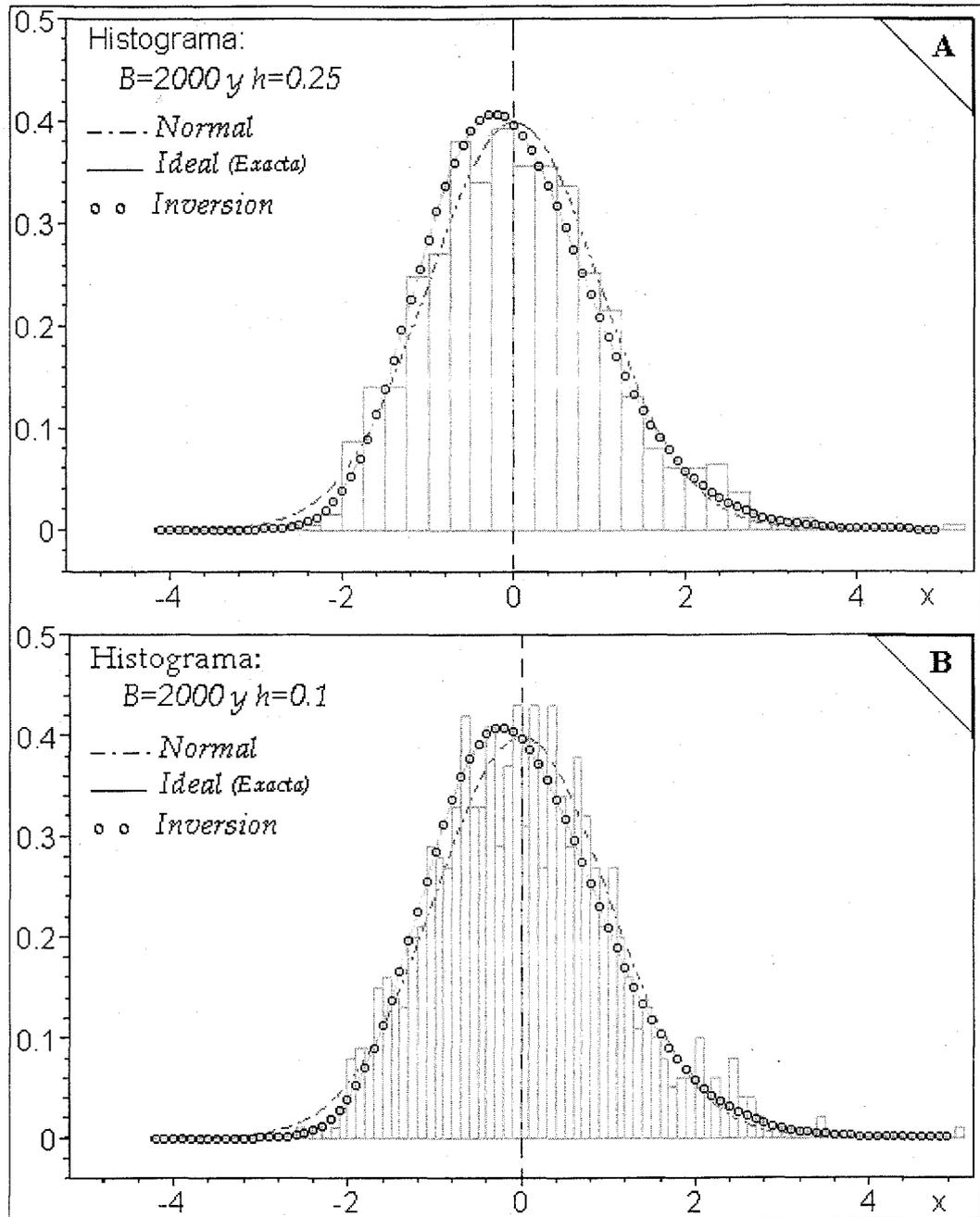


Figura 8-1

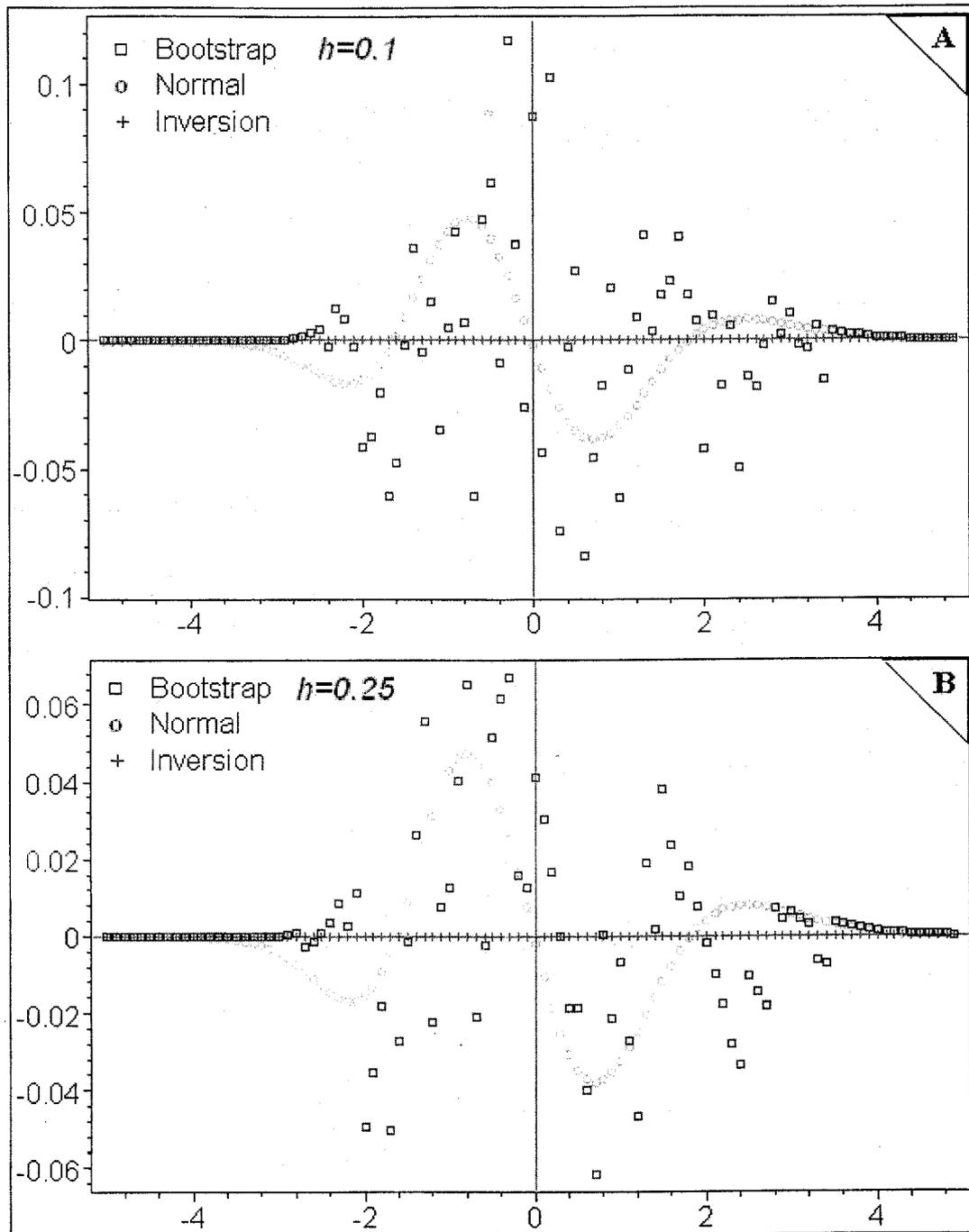


Figura 8-2

Caso no paramétrico: Este caso es el más importante puesto que no supone ningún conocimiento previo sobre el aspecto de densidad de Z_n . En este caso $\hat{F} = F_n$ es la distribución empírica asociada a la muestra original. Es decir que en el caso no paramétrico nos comportamos como si no sabíamos que los datos provienen de una familia paramétrica como es el caso de nuestro actual ejemplo. En esta situación no podemos llegar a una expresión exacta para la densidad ideal como en (8.12). Puesto que la densidad ideal corresponde a la estimación bootstrap considerando un número de simulaciones igual a $B = \infty$, vamos a considerar en este ejemplo que la estimación ideal es la obtenida con $B = 30000$ simulaciones a partir de la distribución empírica F_n . Por otro lado, la estimación bootstrap estándar sería la obtenida con $B = 2000$ simulaciones a partir de F_n . En las Figuras 8.3 y 8.4 se representan cuatro estimaciones para la función de densidad g_n^* de Z_n^* . La densidad exacta obtenida mediante el procedimiento paramétrico y representada por la curva continua, la aproximación normal representada por la curva discontinua, la aproximación vía fórmulas de inversión representada por los círculos. Dichas tres curvas están representadas junto con el histograma bootstrap ideal ($B = 30000$) en la Figura 8.3 y junto al histograma bootstrap estándar ($B = 2000$) en la Figura 8.4. Para cada valor de B hemos dibujado el histograma bootstrap considerando dos valores para la ventana $h = 0.25$, en la parte A de la figura, y $h = 0.1$, en la parte B, notando, así, como cambia el histograma con el cambio de h .

Por otro lado, en la Figura 8.5 representamos las diferencias entre la estimación ideal (que normalmente estamos aproximando) de la densidad g_n^* y las demás estimaciones citadas en este ejemplo: la normal, la bootstrap estándar y la obtenida vía fórmulas de inversión. En la parte A de la Figura 8.5 consideremos las estimaciones bootstrap de la densidad obtenidas con una ventana $h = 0.25$ y en la parte B de la misma figura las estimaciones bootstrap son obtenidas con una ventana $h = 0.1$.

Considerando el histograma obtenido con $B = 30000$ como la estimación ideal para la distribución bootstrap, podemos apreciar que la aproximación de la densidad usando las fórmulas de inversión es igual de precisa que la densidad ideal. Sin embargo, la aproximación bootstrap vía fórmulas de inversión necesita muchísimo menos esfuerzo y tiempo de computación que la aproximación bootstrap estándar que se basa sobre simulación.

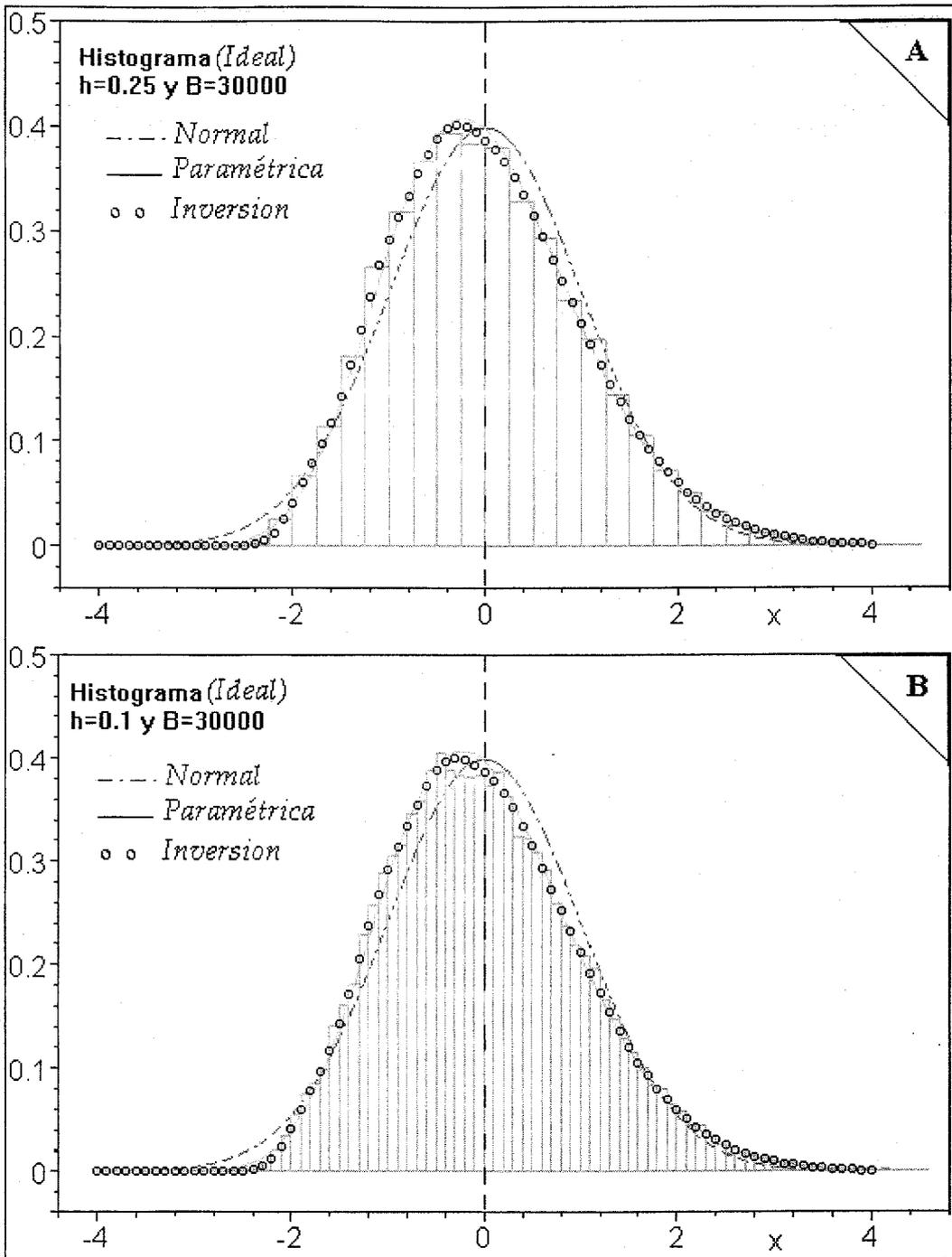


Figura 8-3

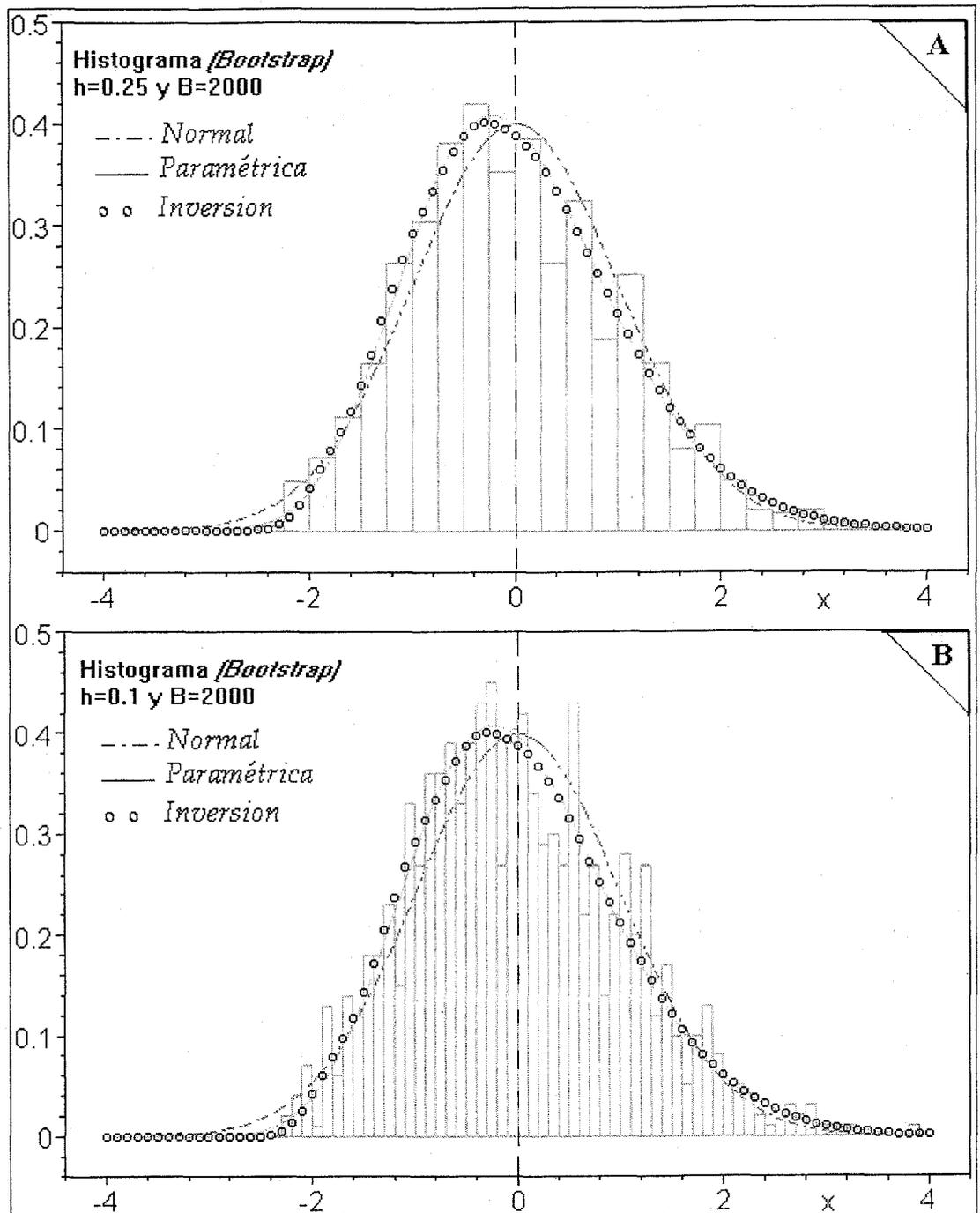


Figura 8-4

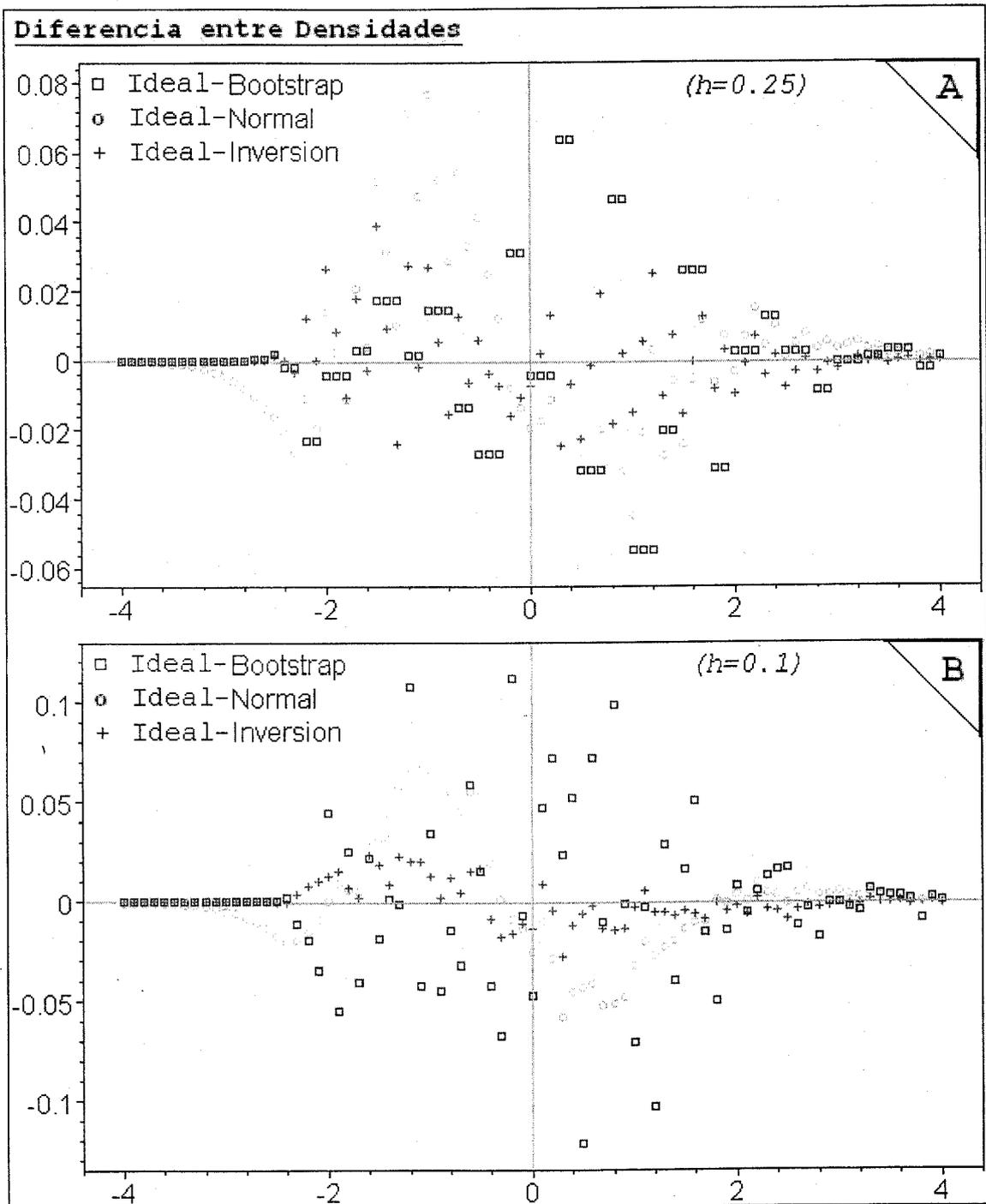


Figura 8-5

8.4 Aproximación bootstrap de la función de distribución

8.4.1 Cálculo de la distribución a partir de la aproximación de la densidad

A partir de los puntos P_j en (8.10) podemos también aproximar la función de distribución $G_n^*(x)$ de la variable aleatoria Z_n^* . Supongamos que m es bastante grande de modo que $m/2T$ es suficientemente grande para poder hacer la aproximación

$$P^* \left\{ Z_n^* \leq -\frac{m}{2T} \right\} \simeq 0 \text{ y } P^* \left\{ Z_n^* \geq \frac{m}{2T} \right\} \simeq 0.$$

Esta condición implica también que $g_n^*(r/2T) \simeq 0$ para cualquier entero r tal que $|r| > m$. Sea $x \in \mathbb{R}$, entonces si $x \leq -m/2T$ tendremos $G_n^*(x) \simeq 0$, y si $x \geq m/2T$ tendremos $G_n^*(x) \simeq 1$. En otro caso, vamos a coger como aproximación a $G_n^*(x)$ la suma de los áreas de los trapecios limitados por los puntos $(x_i, 0)$, $(x_{i+1}, 0)$, P_{i+1} y P_i donde los x_i son inferiores o igual a x , mas el área del trapecio limitado por los puntos $(x_j, 0)$, $(x, 0)$, (x, f_x^*) y P_j donde x_j es máximo de los puntos x_i que son inferiores o igual a x , y donde $f_x^* = g_n^*(x)$ a la cual daremos una aproximación más adelante, (ver *Figura 8.6*). Notaremos por A_x el área total de todos los citados trapecios.

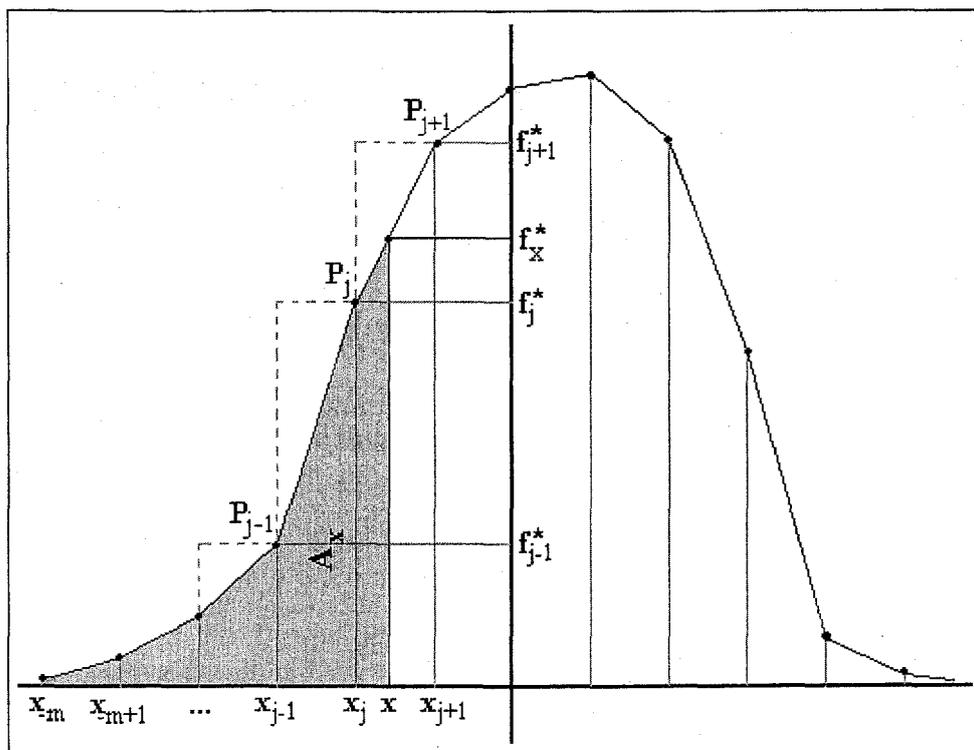


Figura 8-6

Sea $x \in \mathbb{R}$ tal que para algún j , $\frac{j}{2T} < x \leq \frac{j+1}{2T}$ con $-m \leq j \leq m-2$, y sea $f_x^* = g_n^*(x)$. Se tiene entonces que

$$G_n^*(x) \simeq A_x = \frac{1}{4T} \sum_{k=-m}^{j-1} (f_{n,k}^* + f_{n,k+1}^*) + \frac{1}{2} \left(x - \frac{j}{2T}\right) (f_{n,j}^* + f_x^*).$$

Nos falta ahora calcular f_x^* . Para ello, notemos que el punto (x, f_x^*) pertenece a la recta que pasa por los puntos P_j y P_{j+1} . Podemos entonces calcular a f_x^* en términos de $f_{n,j}^*$ y $f_{n,j+1}^*$, obtendremos fácilmente

$$f_x^* = (2Tx - j)(f_{n,j+1}^* - f_{n,j}^*) + f_{n,j}^*.$$

Por lo tanto

$$G_n^*(x) \simeq \frac{1}{4T} \sum_{k=-m}^{j-1} (f_{n,k}^* + f_{n,k+1}^*) + \frac{1}{2} \left(x - \frac{j}{2T}\right) [(2Tx - j)(f_{n,j+1}^* - f_{n,j}^*) + 2f_{n,j}^*]. \quad (8.13)$$

En particular si $x = \frac{j}{2T}$, $-m + 1 \leq j \leq m - 1$, entonces

$$G_n^*(x) \simeq \frac{1}{4T} \sum_{k=-m}^{j-1} (f_{n,k}^* + f_{n,k+1}^*),$$

y

$$G_n^*\left(\frac{-m}{2T}\right) \simeq 0.$$

Nota 26 Si suponemos que hemos calculado la densidad g_n^* en los puntos $x_j = j/2T$, $-m \leq j \leq m - 1$, entonces podemos considerar el polinomio interpolado, digamos $p(x)$, que pasa por todos los puntos P_j obtenidos. Por consecuencia, para todo x , tal que $-m/2T \leq x \leq (m - 1)/2T$ tendremos $g_n^*(x) \simeq p(x)$. Además si se supone que la densidad esta concentrada entre los números x_{-m} y x_{m-1} , entonces tendremos también la aproximación de la distribución

$$G_n^*(x) = \int_{x_{-m}}^x p(u) du.$$

8.4.2 Cálculo directo de la distribución

En esta sección procuraremos de aproximar la distribución bootstrap de Z_n^* directamente usando la fórmula de inversión (8.1). Sea a y b dos puntos de continuidad de $G_n^*(x)$, entonces se verifica

$$G_n^*(b) - G_n^*(a) = VP \int_{-\infty}^{\infty} \frac{e^{-2\pi ita} - e^{-2\pi itb}}{2\pi it} \psi_n^*(2\pi t) dt.$$

Notemos por $\Psi(t)$ la función

$$\Psi(t) = \frac{e^{-2\pi ita} - e^{-2\pi itb}}{2\pi it} \psi_n^*(2\pi t),$$

y sea T un numero bastante grande de modo que podemos hacer la aproximación

$$G_n^*(b) - G_n^*(a) \simeq \int_{-T}^T \Psi(t) dt. \quad (8.14)$$

Ahora vamos a aplicar la fórmula de trapecios para calcular el integral en (8.14). Sea x y a dos números reales que sean puntos de continuidad de G_n^* , y sea $c = G_n^*(a)$. Sea h un número real tal que $m = T/h$ sea un número entero bastante grande, entonces

$$G_n^*(x) - G_n^*(a) = \frac{T}{m} \sum_{k=-m+1}^{m-1} \Psi(kh) + \frac{T}{2m} [\Psi(-T) + \Psi(T)].$$

De otra forma,

$$G_n^*(x) = \frac{T}{m} \sum_{k=-m+1}^{m-1} \Psi(kh) + \frac{T}{2m} [\Psi(-T) + \Psi(T)] + c. \quad (8.15)$$

A la hora de programar el cálculo de la suma en (8.15) se nos presenta un problema para $k = 0$, ya que se divide por cero. Para solventar dicho problema, hay que reemplazar a $\Psi(0)$ por $(x - a)$ ya que es el límite de $\Psi(t)$ cuando t tiende a 0. También, podemos aproximarnos a $\Psi(0)$ con $\Psi(\varepsilon)$ donde ε tiende a 0. Si suponemos que $c = G_n^*(a)$ es conocida para cierta a , obtendremos para cualquier x punto de continuidad de G_n^* ,

$$G_n^*(x) = \frac{T}{m} \sum_{\substack{k=-m+1 \\ k \neq 0}}^{m-1} \Psi(kh) + \frac{T(x-a)}{m} + \frac{T}{2m} [\Psi(-T) + \Psi(T)] + c.$$

Podemos siempre considerar a tendiendo a $-\infty$ para que sea $c \simeq 0$. En la sección siguiente veremos que los resultados numéricos para $G_n^*(x)$ obtenidos mediante el uso de (8.15) son muy similares a los obtenidos usando (8.13), ver *Tabla 8.2*.

8.4.3 Resultados numéricos

Consideremos los mismos datos del ejemplo anterior. En esta sección propongamos aproximar la función de distribución $G_n^*(x)$ usando el procedimiento de la sección anterior y luego comparar los resultados tanto a la aproximación bootstrap estandar como a la aproximación Normal de $G_n^*(x)$. A partir de los puntos $P_j = (x_j, f_{n,j}^*)$ de la curva de la densidad calculados via fórmula de inversión podemos calcular la función de distribución $G_n^*(x)$ en cualquier punto x usando el procedimiento visto en la sección anterior, ver la fórmula (8.13). En la *Tabla 8.2* podemos ver los resultados numéricos correspondientes a la aproximación de G_n^* usando diferentes procedimientos. Notemos que los resultados obtenidos mediante la fórmula (8.13) basada sobre la estimación de la densidad (Método de inversión) y los resultados obtenidos usando la fórmula (8.15) basada sobre la fórmula de inversión directa de

la distribución (inversión directa) son muy próximos.

x	Ideal	Inversión	Inversión directa	Bootstrap	Normal
-2	0.0066	0.0066	0.0064	0.0130	0.0227
-1.5	0.0522	0.0511	0.0509	0.0590	0.0668
-1	0.1639	0.1612	0.1610	0.1660	0.1586
-0.5	0.3351	0.3325	0.3324	0.3370	0.3085
0	0.5292	0.5303	0.5304	0.5300	0.5000
0.5	0.7060	0.7086	0.7087	0.6920	0.6914
1	0.8377	0.8402	0.8403	0.8200	0.8413
1.5	0.9229	0.9223	0.9224	0.9240	0.9331
2	0.9675	0.9664	0.9664	0.9700	0.9772
2.5	0.9882	0.9870	0.9870	0.9870	0.9937
3	0.9955	0.9955	0.9955	0.9960	0.9986
3.5	0.9987	0.9986	0.9986	0.9990	0.9997
4	0.9996	0.9996	0.9996	1	0.9999

Tabla 8.2

Podemos apreciar la precisión de la aproximación bootstrap vía fórmulas de inversión frente al método bootstrap estandar o la aproximación normal. Recordemos que en el caso de la aproximación Normal, la función de densidad es siempre simétrica respecto a cero lo que no es cierto en el caso de nuestro ejemplo donde la verdadera densidad g_n^* no es simétrica respecto a cero.

En la *Figura 8.7* podemos ver una comparación grafica entre la distribución bootstrap ideal (obtenida con $B = 30000$ simulaciones) tanto a la densidad exacta (arriba) como a la aproximación via inversión (abajo). De igual forma, en la *Figura 8.8* se representa una comparación de la densidad ideal tanto a la aproximación bootstrap estandar (obtenida con $B = 2000$), arriba, como a la aproximación Normal, abajo.

Por otra parte, la *Figura 8.9* nos muestra la representación grafica de las diferencias entre la distribución Bootstrap ideal (que es el objetivo de nuestra aproximación) y la aproximación via inversión, la aproximación bootstrap estandar y la aproximación normal.

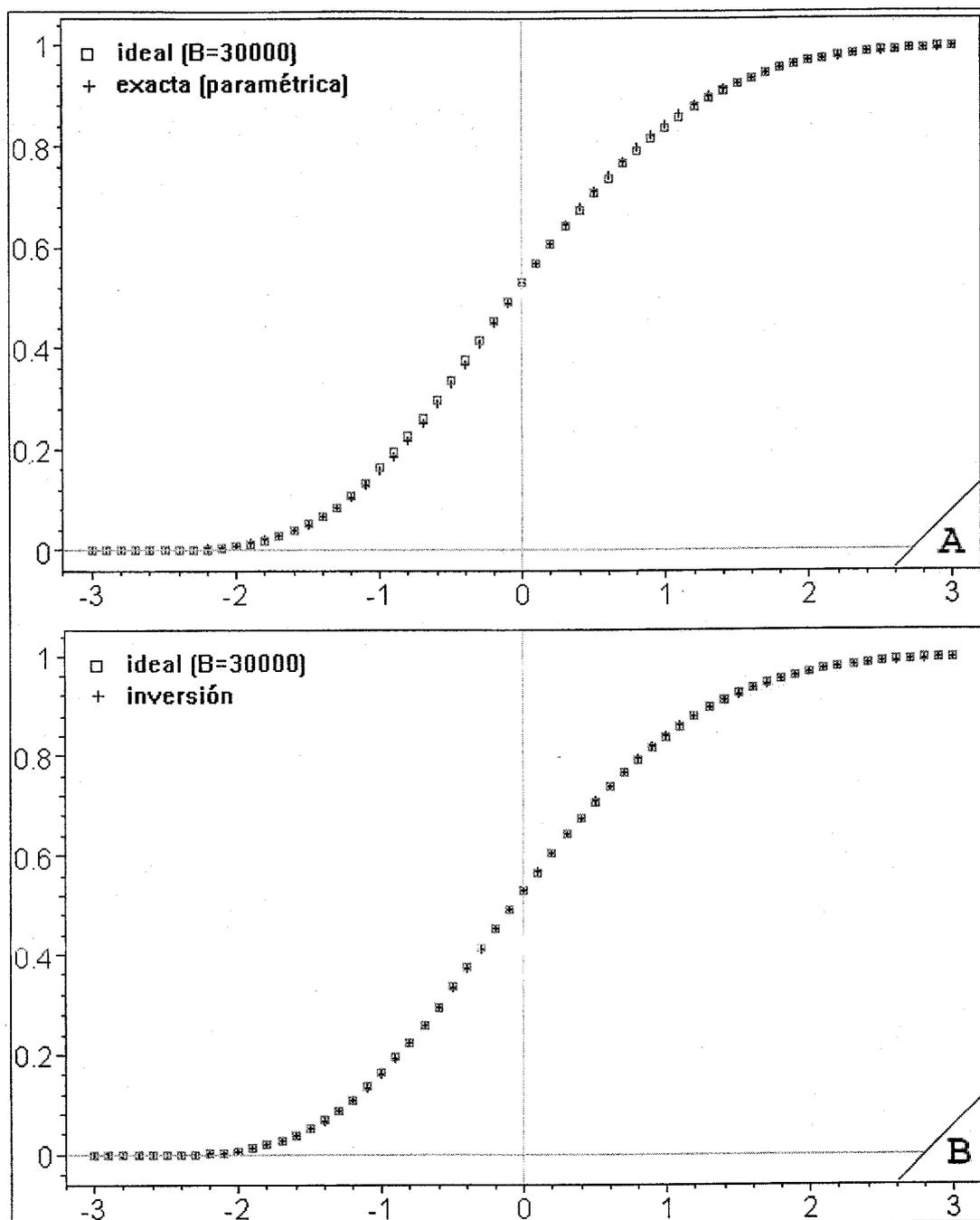


Figura 8-7

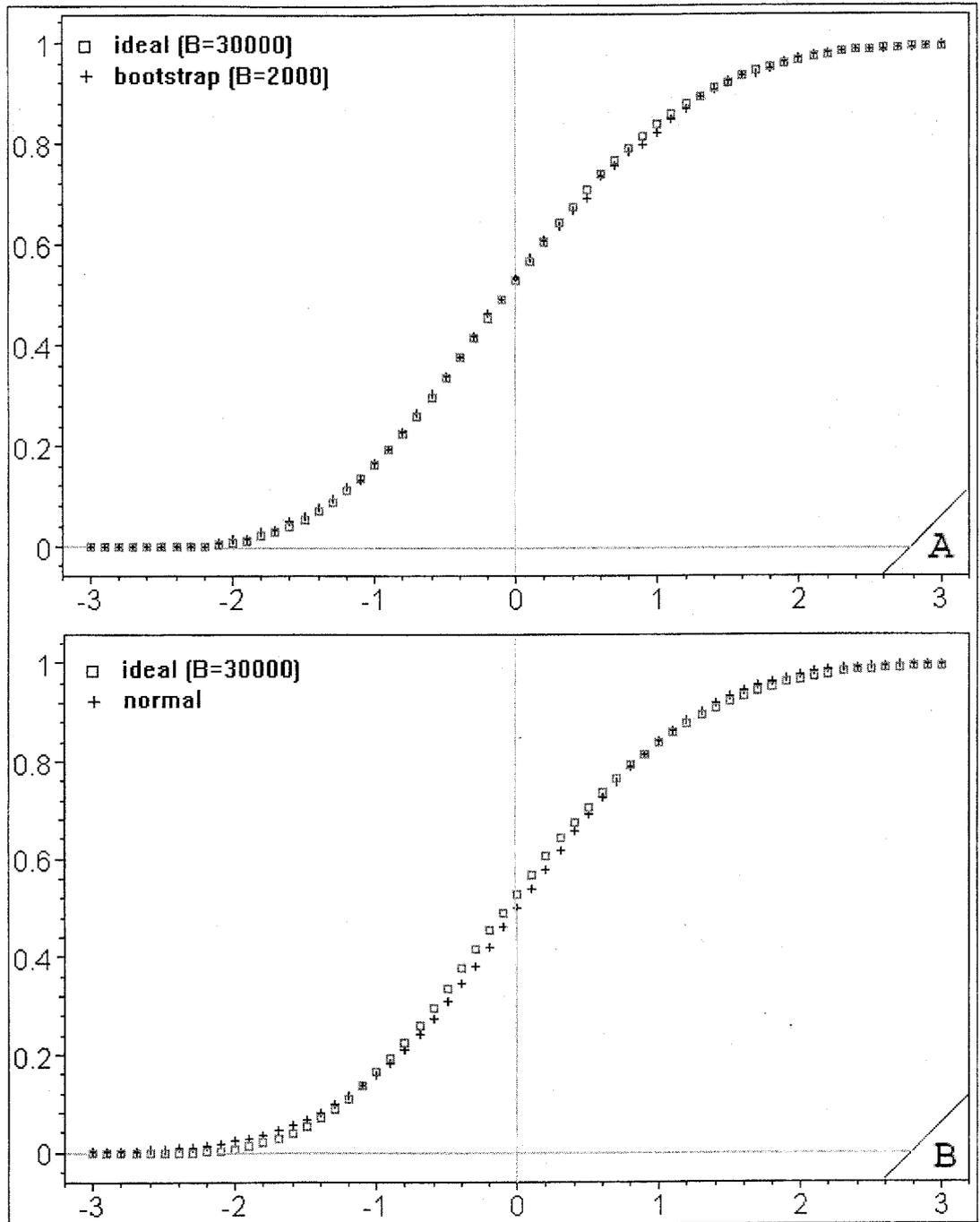


Figura 8-8

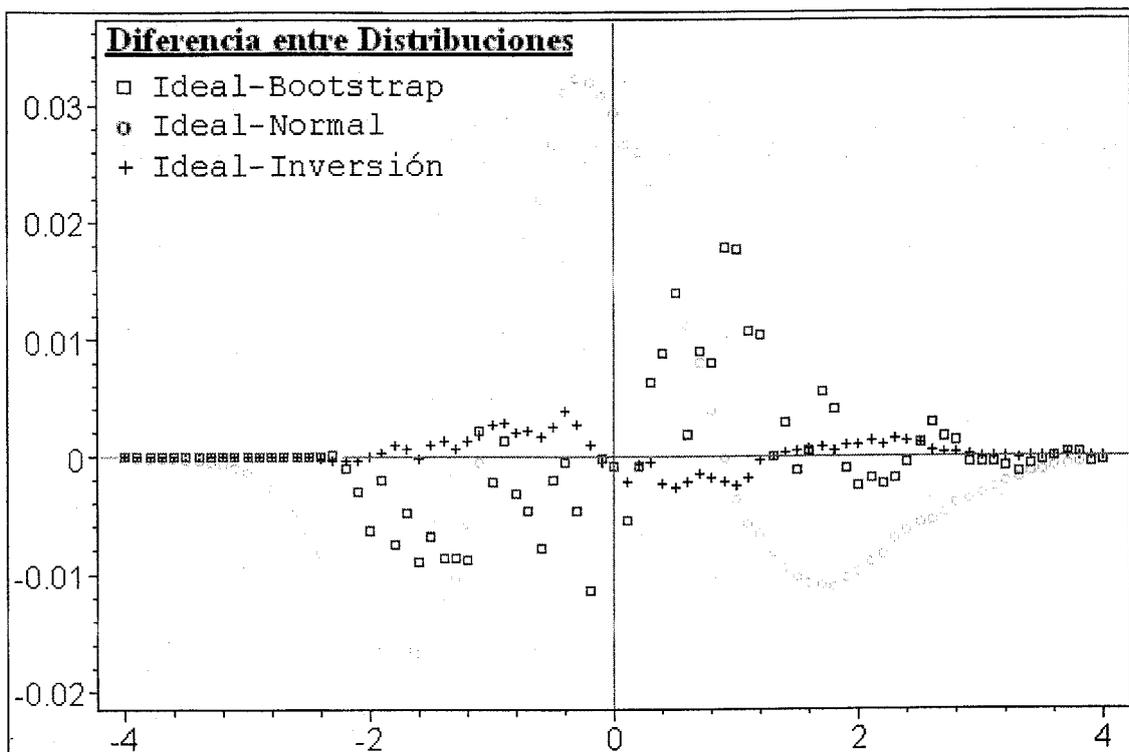


Figura 8-9

Conclusión. Viendo las comparaciones gráficas sobre todo (las Figuras 8.2, 8.5 y 8.9) las que representan las diferencias entre la densidad objeto de la aproximación y las aproximaciones citadas en este capítulo y que son la aproximación normal, la aproximación bootstrap y la aproximación via fórmulas de inversión, podemos concluir que esta última es mejor que las otras dos. Es conocido que en general la aproximación bootstrap es mejor en muchas situaciones que la aproximación normal sobre todo cuando el tamaño muestral no es suficientemente grande. Sin embargo, la aproximación bootstrap necesita un esfuerzo computacional alto tanto para simular muestras bootstrap como para calcular replicas del estadístico de interés (en nuestro caso la variable Z_n en (8.4)). Además, como hemos mencionado anteriormente en este capítulo, en el caso de la estimación de la densidad, el histograma bootstrap puede variar espectacularmente al variar la amplitud de la ventana h elegida para dibujar el histograma. Esto significa que una elección inadecuada de h puede afectar mucho la calidad de la aproximación. Al usar el algoritmo de las transformadas rápidas de Fourier, *FFT*, el cálculo de la densidad usando la fórmula de inversión necesita mucho menos esfuerzo computacional y mucho menos tiempo de computación que el método bootstrap estándar con $B = 2000$, sin embargo, la

Capítulo 9

IMPLEMENTACIÓN EN EL ORDINADOR

Esta bien claro que sin la existencia del Ordenador, cualquier método bootstrap no hubiera tenido éxito alguno. Es más, en algunas situaciones, es necesario un ordenador lo suficientemente potente para poder procesar el gran conjunto de datos que requieren dichos métodos. Esto justifica la época en la que apareció el método bootstrap y que coincide con la revolución informática. Al principio no cualquier método bootstrap que existen hoy hubiera sido práctico y fácil de implementar en cualquier ordenador por motivos de coste de computación o por el largo tiempo que puede durar la computación debido a la poca potencia en los procesadores y la pequeña dimensión de las memorias RAM de aquellos tiempos.

Afortunadamente, hoy en día, incluso los ordenadores de uso doméstico son capaces de llevar a cabo, para no decir todas, casi cualquier tipo de computación por más complicada que sea. Por otra parte, la dramática bajada de los precios de dichos ordenadores. Esto justifica el desarrollo de métodos bootstrap computacionales que se basan sobre el uso intensivo del computador. El método bootstrap original de EFRON (1979) que usa las simulaciones de Monte Carlo directas y al que se refiere a menudo como método bootstrap estándar o método de Monte Carlo estándar, es fácil de programar y puede que no necesita un procesador muy potente pero sí necesita una memoria RAM bastante grande cuando se usa un número alto de simulaciones. Pero su eficacia es puramente relacionada con el número de simulaciones que se usan. Cuanto más mayor es el número de simulaciones más eficaz es el método. Sin embargo, los métodos mejorados a las que referiremos como métodos computacionales, rebajan sustancialmente el número de simulaciones requerido para obtener cierta eficacia pero a cambio son más difíciles de programar y necesitan una computa-

cion extra y por consecuencia son mucho mas eficaces que el método estandar sobre todo en situaciones favorables. Se puede notar que los métodos computacionales se han ido desarrollando a medida que se desarrolla el mundo de las computadoras, concretamente, a partir de la segunda mitad de la decada de los ochenta.

En el presente capítulo, vamos a ver como se puede implementar el método ortogonal en el ordenador. Presentaremos varias programas cuyo codigo esta escrito en language *Maple V* cuya estructura es similar al popular language *C*. Mostraremos mediante ejemplos como funcionan la mayoría de las programas.

El conjunto de las programas esta clasificado en varios grupos o paquetes cuyo manejo es similar al de cualquier paquete de *Maple V*. Cada paquete incluye varias programas donde cada programa puede ser ejecutado de forma independiente o como subprograma de otros que los necesitan. Los principales paquetes son **Estima** y que consiste en un grupo de programas relacionados con la estimación paramétrica, el paquete **NPestima** que es un grupo de programas relacionados con la estimación no paramétrica y el paquete **Repita** que consiste en programas cuyo fin es automatizar ciertas tareas como es la reejecución un número determinado de veces de un programa del paquete **Estima** o **NPestima** mediante diversos parámetros.

Como cualquier paquete de *Maple V*, los paquetes **Estima**, **NPestima** o **Repita** pueden ser activados usando el comando `with(.)`. Tambien se puede activar solamente un cierto programa dentro del paquete mediante el mismo comando `with(..)` pero usando dos argumentos, donde el primer argumento es el nombre del paquete y el segundo es el nombre del programa perteneciente a dicho paquete, o bien, se puede usar la sentencia

```
>with(Paquete[Programa]);
```

Una vez activado un paquete se puede usar cualquiera de los programas que incluye usando sus argumentos adecuados. A cada programa esta asociado una hoja de ayuda explicando el objetivo del programa y el uso de los argumentos requeridos por el programa. Para obtener dicha ayuda se puede usar el comando `ayuda` de *Maple* como si se trata de cualquier programa propio del mismo:

```
>help(Programa);
```

O también:

>? Programa;

De igual manera, una vez activado un Paquete, se puede ver el código fuente de cualquier programa en dicho Paquete. Asimismo, se puede ver el código entero de un paquete una vez activado. Para ello, hay que ejecutar el comando `eval(.)` de Maple como sigue:

>eval(Programa); Para ver el código del programa "Programa".

>eval(Paquete); Para ver el código entero (de todos los programas) del paquete "Paquete".

El código entero de los programas más importantes será incluido en esta memoria al final del presente capítulo.

Nota: Como argumento de los comandos anteriores la palabra "Paquete" debe ser sustituido por el nombre del paquete usado, asimismo, la palabra "Programa" debe ser sustituida por el nombre del programa del paquete en cuestión. El símbolo ">" en las sentencias anteriores, y más adelante, indica que estamos en una sesión de Maple.

Cualquier programa escrito en el lenguaje *Maple* es una función de ciertos argumentos que pueden ser números, listas, funciones, etc. Esto nos permite usar, a veces, el término *función* en lugar de *programa*. Para más información sobre los comandos de *Maple* se puede consultar las referencias siguientes: MANUEL J. SOTO & JOSE LUIS VICENTE (1996), "*Matemáticas con Maple*", ADDISON-WESLEY IBEROAMERICANA o AGUSTIN CARRILLO DE ALBORNOZ & INMACULADA LLAMAS (1995), "*MAPLE V, Aplicaciones matemáticas para PC*", RA-MA.

9.1 El paquete Estima

El paquete *Estima* incluye un grupo de programas donde cada uno de ellos puede ser ejecutado de forma independiente. También, algunos programas principales usan a otros como subprogramas. Por ejemplo, si solamente queremos obtener cualquier polinomio de Charlier ejecutamos el programa `Charlier(.)`, y si lo que queremos es estimar el sesgo o la varianza de un estimador mediante el método ortogonal ejecutaremos el programa `Aproximar(.)` que a su vez llama al programa `Charlier(.)` como subprograma cuando necesita los polinomios de Charlier para llevar a cabo la estimación. En lo que sigue explicaremos los programas más importantes que incluye el

paquete *Estima*. Para ver el contenido de dicho paquete, ejecutaremos el comando `with(.)` con el argumento *Estima*. Es decir

```
>with(Estima);
  [Aproximar, Bin, Charlier, Hermite, Krawt, Lag, MC, Max-
  imo, Meix, Muestra, Norma, Pcharl, Pdat, Pperm, Pjac, Pkrawt, Plag,
  Pmeix, Suprimir, Umvue, Ydata ]
```

Los programas `Charlier(.)`, `Hermite(.)`, `Krawt(.)`, `Lag(.)` y `Meix(.)` ejecutados con sus debidos argumentos son simples funciones o programas que calculan los polinomios estándares de Charlier, Hermite, Krawtchouk, Laguerre y Meixner respectivamente (ver CHIHARA T. S. (1978)). Consideremos por ejemplo el cálculo del polinomio de Hermite. Entonces `Hermite(x,k)` nos da el polinomio de grado k de Hermite con argumento x .

```
>Hermite(x,k);
```

$$k! \left[\sum_{j=0}^{\lfloor \frac{1}{2}k \rfloor} \frac{(-1)^j (2x)^{k-2j}}{(k-2j)! j!} \right]$$

donde $\text{floor}(z) = [z]$ es la Parte Entera del número real z .

```
>Hermite(x,5);
```

$$32x^5 - 160x^3 + 120.$$

```
>Hermite(1/2,5);
```

$$41.$$

Los programas `Pcharl(.)`, `Pperm(.)`, `Pjac(.)`, `Pkrawt(.)`, `Plag(.)` y `Pmeix(.)` calculan el sistema de los polinomios ortogonales respecto a la densidad de la media muestral \bar{X} cuando la muestra original proviene de la distribución de *Poisson*, *Normal*, *Uniforme*, *Binomial*, *Gamma* y *Binomial Negativa* respectivamente. Dichos polinomios estan intimamente relacionados con los polinomios de Charlier, Hermite, Jakob, Krawtchouk, Laguerre y Meixner respectivamente (ver CHIHARA (1978)). Para cualquier programa se puede solicitar la ayuda de igual manera como si fuera una función propia de *Maple*. Consideremos, por ejemplo, los polinomios `Pperm(.)`

>?Pherm

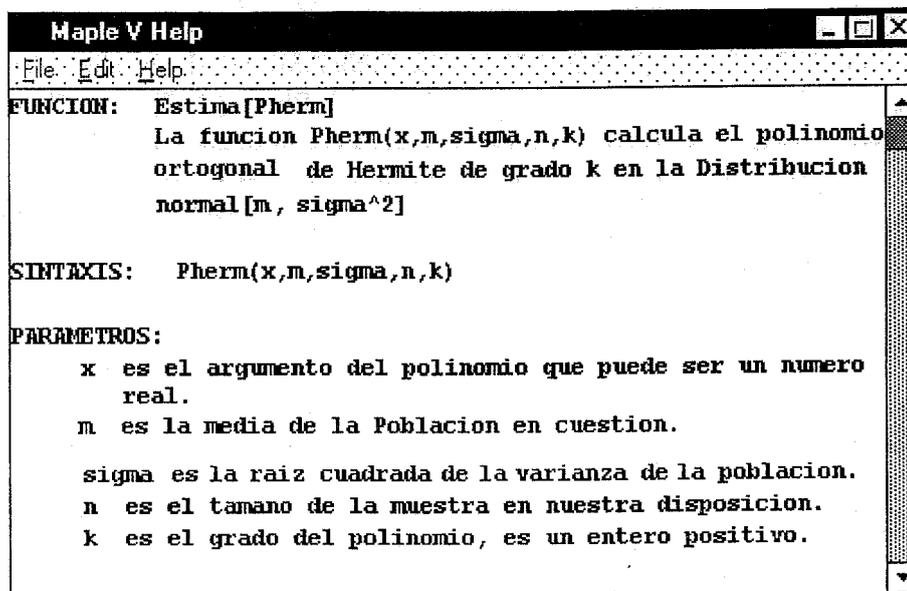


Figura 9.1

>Pherm(x,mu,sigma,n,k);

$$k! \left[\sum_{j=0}^{\lfloor \frac{1}{2}k \rfloor} \frac{(-1)^j (x - \mu)^{k-2j} \sigma^{2j}}{2^j n^j (k-2j)! j!} \right]$$

>Pherm(x,5,1,10,3);

$$x^3 - 15x^2 + \frac{747}{10}x - \frac{247}{2}$$

Notemos que, además de los argumentos comunes x , n y k , los polinomios anteriores tienen como argumentos otros parámetros que corresponden a los parámetros de la familia de distribuciones correspondiente a cada sistema de polinomios. Dichos parámetros vienen en el orden usual en el que figuran en las notaciones habituales de cada familia de distribuciones. Por ejemplo, los polinomios $P_{\text{charl}}(\cdot)$ que

corresponden a la distribución de *Poisson* $Po(\mu)$ tienen los argumentos siguientes $Pcharl(x, \mu, n, k)$ y los polinomios $Pherm(.)$ y que corresponden a la familia *Normal* $N(\mu, \sigma)$ tienen los argumentos siguientes $Pherm(x, \mu, \sigma, n, k)$.

Otro programa relacionado directamente con los anteriores es la función $Norma(.)$ que calcula la norma de los polinomios anteriores en el espacio correspondiente. Puesto que cada uno de los polinomios $Pcharl(.)$, $Pherm(.)$, $Pjac(.)$, $Pkrawt(.)$, $Plag(.)$ y $Pmeix(.)$ tiene sus propios parámetros entonces el primer argumento de la función $Norma(.)$ es el nombre del polinomio indexado con sus parámetros particulares en el orden habitual. El segundo argumento es n el tamaño de la muestra original y el tercero es el grado del polinomio para el que se quiere calcular la norma. Por ejemplo:

```
>Norma(Pcharl[mu],n,k);
```

$$\sqrt{\frac{\mu^k k!}{n^k}}$$

```
>Norma(Pherm[mu,sigma],n,k);
```

$$\sqrt{\frac{\sigma^{2k} k!}{n^k}}$$

Otros programas más sencillos son $Maximo(.)$ y $Suprimir(.)$. El primero es similar al comando $max(.)$ de Maple que devuelve el máximo de una sucesión de números, la única diferencia es que el argumento de $Maximo(.)$ debe ser una lista de números y no una sucesión. Eso porque en Maple usaremos a las muestras de datos como listas. El programa $Suprimir(.)$ nos permite quitar elementos de una lista o quitar listas de una lista de listas. Esta función será útil cuando trabajamos con muestras bootstrap. Por ejemplo cuando se requiere tener una buena aproximación a cierta estimación ideal se necesita simular un número bastante elevado, B , de muestras bootstrap, (por ejemplo $B = 10000$), luego para estudiar el efecto de la reducción de B sobre nuestra aproximación, podemos ir quitando muestras bootstrap del conjunto ya simulado, evitando así volver a simular de nuevo.

```
>Maximo([2,1,5,7,20,1,6]);
```

20.

>Suprimir(3,[2,1,5,7,20,1,6]);

[2, 1, 5, 7].

>Suprimir(2,[[1,a,6], [7,3,m], [40,1,0], [1,4,3], [a,b,c]]);

[[1, a, 6], [7, 3, m], [40, 1, 0]].

El programa Muestra(.) nos permite simular una o varias muestras al mismo tiempo. La simulación se efectúa a partir de cierta distribución como la distribución normal, Gamma, Uniforme, binomial etc. Dicho programa necesita al menos dos argumentos para ser ejecutado. También, admite un tercer argumento como opcional como vamos a ver. El primer argumento es de la forma *distribución/parámetros* y es el nombre de la distribución a partir de la cual deseamos simular una o varias muestras. Dicho nombre va acompañado con los parámetros característicos de la distribución en su orden usual, por ejemplo, *normal*[μ, σ], *binomial*[N, p], *poisson*[μ] etc. El segundo argumento es el tamaño n de la muestra requerida. Opcionalmente, se puede añadir un tercer argumento que sería el número B de las muestras a simular. Un ejemplo de ejecución de Muestra(.) es

>Muestra(normal[7,1],10);

[8.175, 6.436, 7.235, 5.557, 5.920, 6.978, 4.414, 6.556, 5.996, 6.972].

>Muestra[poisson[5],10,3);

[3, 4, 7, 1, 2, 9, 7, 5, 6, 8], [9, 3, 3, 4, 5, 4, 3, 6, 7, 6], [4, 7, 6, 6, 4, 0, 2, 6, 3, 6].

El programa MC(.): Supongamos, por ejemplo, que hemos estimado la función paramétrica $\xi = h(\theta)$ mediante el estimador $\hat{\xi}_n = T_n(S_n)$ que es función de un estadístico S_n de la familia de interés F_θ . Nuestro objetivo, ahora, es aproximar el sesgo o la varianza de $\hat{\xi}_n$. Mediante argumentos necesarios, el programa MC(.) calcula el sesgo y la varianza bootstrap de $\hat{\xi}_n$ usando el método estándar de Monte Carlo, donde las simulaciones bootstrap se efectúan desde el estimador paramétrico F_θ . Los argumentos de MC(.) tienen la forma general siguiente

MC(*distribución*[$\xi, \hat{\xi}_n, \text{objetivo}$], *muestra_original*, *muestras_bootstrap* o B).

La palabra *distribución* en el primer argumento es el nombre de la familia de distribuciones de interés que puede ser, por ejemplo, *normal*, *uniform*, *etc.* La palabra *objetivo* puede ser *var* si queremos aproximar la varianza o *bias* si queremos aproximar el sesgo. Si la palabra *objetivo* es eliminada el programa calcula tanto el sesgo como la varianza de $\hat{\xi}_n$. El segundo argumento *muestra_original* es una lista de numeros que corresponde a la muestra original \mathcal{X}_n que proviene de la distribución desconocida F_θ . El tercer argumento puede ser, *muestras_bootstrap*, es una lista de listas correspondientes a las muestras bootstrap simuladas anteriormente y que van a ser usadas para efectuar las aproximaciones o un numero entero B . Cuando este ultimo argumento es un numero entero B , entonces el programa $MC(\cdot)$ llama al programa $Muestra(\cdot)$ para efectuar B simulaciones a partir de F_θ y luego llevar a cabo el cálculo. Supongamos que $\xi = \mu^2$ y $\hat{\xi} = \bar{X}^2 - \sigma^2/n$, y donde la muestra original proviene de $N(\mu, \sigma)$ con μ desconocida. Supongamos que $\sigma = 1$. En este caso $\hat{\xi}$ es el *UMVUE* de ξ por lo que $s = sesgo(\hat{\xi}) = 0$. En seguida vemos un ejemplo de ejecución de $MC(\cdot)$:

```
>n:=10:B:=20:
>mues_orig:=Muestra(normal[7,1],n);

[8.175, 6.436, 7.235, 5.557, 5.920, 6.978, 4.414, 6.556, 5.996, 6.972].

>media:=sum(mues_orig[i],i=1..n)
>mues_boot:=[Muestra(normal[media,1],n,B)];
>MC(normal[mu^2,x^2-1/n,bias], mues_orig, mues_boot);

-0.335
```

La variable x en el primer argumento de $MC(\cdot)$ representa el estadístico \bar{X} .

```
>MC(normal[mu^2,x^2-1/n,var], mues_orig, mues_boot);

17.656
```

El sesgo y la varianza bootstrap ideales son respectivamente 0 y 19.62.

Los programas $Pdata(\cdot)$ y $Ydata(\cdot)$ tienen por objetivo la adaptación del conjunto de los datos bootstrap al problema de regresión múltiple. Cualquiera de los dos programas puede ser ejecutado como programa independiente o como subprograma

del programa principal `Aproximar(.)` del que hablaremos mas abajo. La función `Ydata(.)` toma como argumentos, entre otros, al conjunto de las muestras bootstrap y al estimador $T_n(S_n)$ como función del estadístico suficiente y completo S_n . El resultado es un vector de dimensión B (el numero de muestras bootstrap)

$$[T_n(S_n^{*1}), \dots, T_n(S_n^{*B})],$$

y que forma la variable a explicar por la regresión. Por otra parte la función `Pdata(.)` es la encargada de formar la matriz de regresión \mathcal{X}_* a partir de los datos bootstrap.

El programa `Umvue(.)` usa como subprogramas a la mayoría de las programas vistas anteriormente. Dicho programa tiene por objetivo el cálculo, simbólicamente o numéricamente, del *Estimador de Mínima Varianza* de una función del parámetro natural en varias familias de distribuciones de uso frecuente. Esto resulta muy útil para buscar ejemplos y verificar los resultados de un cálculo teórico o numerico. Para más detalles sobre el programa `Umvue(.)` se puede consultar la oja de ayuda correspondiente a la misma. Para ver como funciona el programa `Umvue(.)` supogamos que queremos calcular el UMVUE de la función paramétrica $h(\mu) = \xi = \mu^4 + 3\mu$ donde se dispone de una muestra procedente de la familia $F_\mu \equiv N(\mu, \sigma)$ con μ desconocido y σ conocida. Entonces el estimador de mínima varianza de ξ puede obtenerse simplemente ejecutando dicho programa de la forma siguiente

`>Umvue(mu^4+3*mu,normal[mu,sigma]);`

$$\left[x^4 - \frac{6\sigma^2}{n}x^2 + 3x + \frac{3\sigma^4}{n^2}, \right. \quad (9.1)$$

$$\left. \frac{16\sigma^2}{n}\mu^6 + \frac{72\sigma^4}{n^2}\mu^4 + \frac{24\sigma^2}{n}\mu^3 + \frac{96\sigma^6}{n^3}\mu^2 + \frac{24\sigma^8}{n^4} \right]$$

donde el primer termino de la lista en (9.1) es el estimador UMVU de ξ con x representando el estadístico suficiente y completo que es, en nuestro caso la media muestral \bar{X} . El segundo término es la varianza $v = v(\mu)$ de $\hat{\xi}$. Es decir

$$\hat{\xi}_n = \bar{X}^4 - \frac{6\sigma^2}{n}\bar{X}^2 + 3\bar{X} + \frac{3\sigma^4}{n^2}, \quad (9.2)$$

y

$$Var_\mu(\hat{\xi}_n) = v(\mu) = \frac{16\sigma^2}{n}\mu^6 + \frac{72\sigma^4}{n^2}\mu^4 + \frac{24\sigma^2}{n}\mu^3 + \frac{96\sigma^6}{n^3}\mu^2 + \frac{24\sigma^8}{n^4}.$$

La función $Umvue(.)$ toma dos argumentos. El primero es la función paramétrica a estimar y el segundo es la familia de distribuciones en cuestión, por ejemplo, $normal[\mu, \sigma]$, $poisson[\mu]$, $uniform[\theta]$, etc. Puesto que $v(\mu)$ es una función paramétrica, del mismo modo, podemos calcular su estimador $UMVU$. Para ello basta sustituir la función $h(\mu)$ por $v(\mu)$ en la sesión de Maple. Supongamos que $\sigma = 1$, entonces

`>Umvue(mu^4+3*mu, normal[mu,1]);`

$$\left[x^4 - \frac{6}{n}x^2 + 3x + \frac{3}{n^2}, \frac{16}{n}\mu^6 + \frac{72}{n^2}\mu^4 + \frac{24}{n}\mu^3 + \frac{96}{n^3}\mu^2 + \frac{24}{n^4} \right].$$

`>v:=unapply(op(2,"), mu);`

$$v := \mu \longrightarrow \frac{16}{n}\mu^6 + \frac{72}{n^2}\mu^4 + \frac{24}{n}\mu^3 + \frac{96}{n^3}\mu^2 + \frac{24}{n^4}.$$

`>Umvue(v(mu), normal[mu,1]);`

$$\left[\frac{16}{n}x^6 - \frac{168}{n^2}x^4 + \frac{24}{n}x^3 + \frac{384}{n^3}x^2 - \frac{72}{n^2}x + \frac{9}{n} - \frac{96}{n^4}, \frac{9216}{n^3}\mu^{10} + \frac{170496}{n^4}\mu^8 \right. \\ \left. + \frac{13824}{n^3}\mu^7 + \frac{1148928}{n^5}\mu^6 + \frac{110592}{n^4}\mu^5 + \left(\frac{3064320}{n^6} + \frac{5184}{n^3} \right) \mu^4 + \frac{244224}{n^5}\mu^5 \right. \\ \left. \left(\frac{2635776}{n^7} + \frac{10368}{n^4} \right) \mu^2 + \frac{110592}{n^6}\mu + \frac{327168}{n^8} + \frac{3456}{n^5} \right].$$

Entonces, el estimador $UMVU$ de v es

$$\hat{v}_{umvue} = \frac{16}{n}\bar{X}^6 - \frac{168}{n^2}\bar{X}^4 + \frac{24}{n}\bar{X}^3 + \frac{384}{n^3}\bar{X}^2 - \frac{72}{n^2}\bar{X} + \frac{9}{n} - \frac{96}{n^4}. \quad (9.3)$$

Nota 27 Las funciones $unapply(.)$ y $op(.)$ son comandos propios de Maple. Para mas detalles sobre ellos se puede consultar la ayuda de Maple o las siguientes referencias MANUEL J. SOTO & JOSE LUIS VICENTE (1996), o AGUSTIN CARRILLO DE ALBORNOZ & INMACULADA LLAMAS (1995).

La función $Aproximar(.)$ es el programa de mas importancia del paquete *Estima*. Dicha función llama practicamente a todos los programas vistos hasta ahora. Como su nombre indica, el objetivo de $Aproximar(.)$ es aproximar la varianza y el sesgo

ideales de un estimador dado en familias paramétricas usando el método bootstrap ortogonal.

Consideremos el ejemplo 1 del capítulo 7, en el que tratamos de aproximar el sesgo y la varianza bootstrap ideales del estimador $\hat{\xi}_n$ en (9.2). Para $\sigma = 1$ y $\mu = 7$ hemos considerado una muestra de tamaño $n = 50$ a partir de $N(7, 1)$.

```
>muestra_normal:=Muestra(normal[7,1],50):
```

El programa `Muestra(.)` acaba de simular una muestra normal de tamaño $n = 50$ y la muestra se guarda con el nombre `muestra_normal`.

Los resultados obtenidos a partir de dicha muestra son los siguientes:

$$v = 37881.68643, \hat{v}_{umvue} = 32572.00191 \text{ y } \hat{v} = 32781.34526.$$

Por otra parte, el sesgo exacto s y el sesgo ideal \hat{s} valen 0:

$$s = \hat{s} = 0.$$

Para comparar los resultados obtenidos mediante el uso del método de Monte Carlo estandar y el método bootstrap ortogonal vamos a ejecutar los programas `MC(.)` y `Aproximar(.)`. En primer lugar vamos a considerar $B = 20$ simulaciones:

```
>with(Estima):
```

```
>muestra_normal:=Muestra(normal[7,1],20):
```

```
>media:=stats[describe,mean](muestra_normal);
```

$$media := 6.83283$$

Esta última instrucción calcula la media muestral $\bar{x} = media$. Para más detalles sobre los comandos de Maple se puede consultar la ayuda de maple o las siguientes referencias: MANUEL J. SOTO & JOSE LUIS VICENTE (1996), o AGUSTIN CARRILLO DE ALBORNOZ & INMACULADA LLAMAS (1995).

```
>h:=mu->mu^4+3*mu;
```

$$h := \mu \longrightarrow \mu^4 + 3\mu.$$

```
>Umvue( mu^4+3*mu, normal[mu,1] );
```

$$\left[x^4 - \frac{6}{n}x^2 + 3x + \frac{3}{n^2}, \frac{16}{n}\mu^6 + \frac{72}{n^2}\mu^4 + \frac{24}{n}\mu^3 + \frac{96}{n^3}\mu^2 + \frac{24}{n^4} \right].$$

```
>Tn:=unapply( op(1,""); x );
```

$$Tn := x \longrightarrow x^4 - \frac{6}{n}x^2 + 3x + \frac{3}{n^2}.$$

```
>muestras_boot:=Muestra(normal[media,1],50,20):
```

Ahora el nombre muestra_boot guarda 20 muestras bootstrap de tamaño 50 cada una.

```
>MC([h(mu),Tn(x)],muestra_normal,var,muestra_boot);
```

36169.3654

Este comando ha calculado una aproximación a la varianza bootstrap ideal $\hat{v} = 32781.34526$ mediante el método de Monte Carlo estandar.

```
>MC([h(mu),Tn(x)],muestra_normal,segs,muestra_boot);
```

-34.1568

Este comando ha calculado una aproximación al sesgo bootstrap ideal $\hat{s} = 0$ mediante el método de Monte Carlo estandar.

```
>Aproximar([h(mu),Tn(x)],normal[mu,1],muestra_normal,20,4,muestra_boot);
```

32781.34553, 0.2 10⁻⁵.

Este comando ha calculado, simultaneamente, una aproximación a la varianza bootstrap ideal $\hat{v} = 32781.34526$ (el primer numero 32781.34553) y una aproximación al sesgo bootstrap ideal $\hat{s} = 0$ (el segundo numero 0.2 10⁻⁵) mediante el método bootstrap ortogonal usando un desarrollo de orden $m = 4$. Notemos que la aproximación ortogonal es casi exacta. Sin embargo, la aproximación de Monte Carlo es bastante encorrecita debido al pequeño numero de simulaciones $B = 20$.

Consideremos ahora, un número de simulaciones mas grande $B = 200$ para aproximar la varianza y el sesgo bootstrap ideales anteriores.

```

>muestras_boot:=Muestra(normal[media,1],50,200):
  Ahora el nombre muestra_boot guarda 200 muestras bootstrap de
  tamaño 50 cada una.
>MC([h(mu),Tn(x)],muestra_normal,var,muestra_boot);
      32079.2066
>MC([h(mu),Tn(x)],muestra_normal,sesg,muestra_boot);
      -2.02081
>Aproximar([h(mu),Tn(x)],normal[mu,1],muestra_normal,200,4,muestra_boot);
      32781.34516,    10-6.

```

9.2 El Paquete NPEstima

Practicamente, el paquete *NPEstima* es homólogo al paquete *Estima*, solo que los programas del paquete *NPEstima* usan los procedimientos estadísticos no paramétricos para efectuar la estimación y llevar a cabo las aproximaciones. El contenido de *NPEstima* es lo siguiente:

```
>with(NPEstima);
```

[*Boot*, *ECmedio*, *Esp*, *Frecboot*, *MonteCarlo*, *Multinomaprox*, *NPaproximar*, *NPdata*, *NYdata*, *Nspo*, *PolMultinom*, *Spo*, *Vboot2d*].

Empezamos por el programa *Boot*(.). Este programa es el homólogo de *Muestra*(.) visto en la sección anterior y nos permite simular muestras a partir de la función de distribución empírica F_n asociada a una muestra de tamaño n . Dicho programa se ejecuta con dos argumentos. El primero es una lista de datos que constituye la muestra a partir de la cual se van a simular otras muestras independientes del mismo tamaño n . Y el segundo argumento es un número entero, B , que representa el número de muestras que deseamos simular. Supongamos, por ejemplo, que queremos simular $B = 5$ muestras independientes a partir de la f.d.D. empírica asociada a la muestra $\mathcal{X}_n = (1, 2, 3, 4, 5, 6, 7)$ de tamaño $n = 7$. Un ejemplo de ejecución es:

```
>muestra:=[ 1, 2, 3, 4, 5, 6, 7 ]:
>Boot(muestra, 5);
```

[3,1,6,7,3,2,5], [1,6,7,2,2,4,4], [4,1,3,6,5,4,5], [2,7,3,3,3,4,3], [1,5,1,5,1,6,7].

El programa `Esp(.)` nos permite calcular la esperanza de una función, $h(X^*)$, de una variable aleatoria X^* , respecto a la función de distribución empírica F_n asociada a una muestra \mathcal{X}_n de tamaño n . Dicho programa se ejecuta con dos argumentos. El primero es la función $h(x)$ y el segundo es la muestra \mathcal{X}_n . Por ejemplo, para calcular el momento centrado de orden 4 de X^* , ejecutamos `Esp(.)` con $h(x) = (x - \bar{x})^4$. Consideremos los datos del **Ejemplo A** del Capítulo 1 que consisten en una muestra de tamaño $n = 12$:

```
>muestra_orig:=[135,20,7,86,420,5,45,103,230,10,94,100];
muestra_orig:=[135, 20, 7, 86, 420, 5, 45, 103, 230, 10, 94, 100].
>media:=Esp( x, muestra_orig );
media := 104.583
>M4:=Esp( (x-media)^4, muestra_orig );
M4 := 87.325323 109.
```

El programa `MonteCarlo(.)` nos permite calcular el sesgo y la varianza de un estimador $\hat{\xi}_n$ usando las simulaciones de Monte Carlo. Las simulaciones se efectúan usando el estimador no paramétrico, F_n , de la distribución de interés. `MonteCarlo(.)` necesita 3 argumentos para ser ejecutado, aunque admite un cuarto argumento como opcional. El primer argumento consiste en el estimador $\hat{\xi}_n = T_n(X_1, \dots, X_n)$ como función de los datos $\mathcal{X}_n = (X_1, \dots, X_n)$. El segundo argumento es una lista de datos que representan la muestra original \mathcal{X}_n , y el tercero es el número, B , de simulaciones requeridas para efectuar la aproximación. Si el cuarto argumento es eliminado entonces el programa `MonteCarlo(.)` llama al programa `Boot(.)` para simular B muestras

a partir de la muestra original dada como segundo argumento. Sin embargo, si el número de argumentos dado a `MonteCarlo(.)` es 4 entonces el cuarto argumento debe ser una lista de muestras bootstrap simuladas anteriormente y que dicho programa usara para efectuar la aproximación sin necesidad de simular datos. Para ver un ejemplo de ejecución, consideremos la muestra anterior y supongamos que nuestro estimador de interés es simplemente la varianza muestral $\hat{\xi}_n = n^{-1} \sum_{i=1}^n (X_i - \bar{X})^2 = \hat{\sigma}^2$. En este caso, notando por $\hat{\mu}_j$ el j -ésimo momento muestral centrado, el sesgo y la varianza bootstrap ideales vienen dados por

$$\hat{s} = -\frac{\hat{\sigma}^2}{n} \text{ y } \hat{v} = \left(\frac{n-1}{n}\right)^2 \left(\frac{\hat{\mu}_4}{n} - \frac{n-3}{n(n-1)} \hat{\sigma}^4\right).$$

cuyos valores numéricos has dado

$$\hat{s} = -1081.339697 \text{ y } \hat{v} = 0.5150125213 \cdot 10^8 \quad (9.4)$$

```
>muestra_orig:=[135,20,7,86,420,5,45,103,230,10,94,100];
>Tn:=sum( (x[i]-sum(x[j],j=1..n)/n)^2, i=1..n )/n;
```

$$Tn := \frac{1}{n} \sum_{i=1}^n \left(x_i - \frac{1}{n} \sum_{j=1}^n x_j \right)^2.$$

```
>n:=12;
>muest_boot:= [Boot(muestra_orig,100)];
>MonteCarlo(Tn, muestra, 100, muest_boot);
```

$$sesgo := -1339.54, \text{ varianza} := 0.5814306706 \cdot 10^8.$$

El programa `Spo(.)` calcula el sistema de polinomios ortogonales respecto a la función de empírica F_n asociada a la muestra \mathcal{X}_n . Para tal fin, `Spo(.)` emplea el método basado sobre los momentos de F_n explicado en el capítulo 3. Los argumentos necesarios para su ejecución son tres. El primero es un número real o un símbolo, por ejemplo x y es el argumento del polinomio. El segundo argumento es una lista de números reales que representa la muestra. El tercer argumento es un número entero k que es el grado del polinomio que queremos calcular. Por otra parte, el programa `Nspo(.)` calcula la norma en el espacio de Banach generado por la distribución F_n , de los polinomios ortogonales calculados usando `Spo(.)`. `Nspo(.)` necesita dos

argumentos: la muestra como primer argumento y el grado del polinomio al que queremos calcular la norma, como segundo argumento. En seguida vemos un ejemplo de ejecución: consideremos los datos anteriores y vamos a calcular el polinomio de grado 3 ortogonal respecto a la f.d.D empírica F_n asociada a \mathcal{X}_n :

```
>muestra:=[135,20,7,86,420,5,45,103,230,10,94,100];
>Spo(x, muestra, 3);
       $x^3 - 578.4683384x^2 + 71536.48907x - 1346121.838.$ 
>Spo(230, muestra, 3);
      -3326704.453
>Nspo(muestra, 3);
      1262445.544.
```

Los programas NPdata(.) y NYdata(.) son las versiones no paramétricas de Pdata(.) y Ydata(.) respectivamente, y su uso es similar a estos últimos. Dichos programas juegan más importancia como subprogramas del programa principal NPaproximar(.) que a su vez es la versión no paramétrica de Aproximar(.) visto en la sección anterior. NPaproximar(.) emplea el desarrollo ortogonal de un estimador en términos de los polinomios ortogonales respecto a la distribución empírica, como se explica en el *Capítulo 7*, para aproximar su sesgo y su varianza bootstrap ideales con mejor precisión que el método de Monte Carlo estándar. El programa NPaproximar(.) se ejecuta con 4 argumentos obligatorios y acepta un quinto argumento como opcional. Su expresión general es de la siguiente forma

NPaproximar (Estimador, muestra_original, B, m, muestras_bootstrap).

El primer argumento consiste en la expresión del estimador como función de los datos $\hat{\xi}_n(X_1, \dots, X_n)$. El segundo argumento será la muestra de interés \mathcal{X}_n . El tercer argumento es un número entero, B , que será el número de muestras bootstrap que el programa debe simular para llevar a cabo la aproximación siempre que el quinto argumento es eliminado. El cuarto argumento es un número entero, m , que representa el grado del desarrollo ortogonal del estimador (Ver *Capítulo 6 y 7*) utilizado para la aproximación del sesgo y la varianza bootstrap ideales. Cuando ya disponemos

de un conjunto de muestras bootstrap simuladas anteriormente, podemos darlas a NPaproxima(.) como quinto argumento. De esta forma el programa ignora el tercer argumento y no simula nuevas muestras bootstrap. Para ver un ejemplo de ejecución consideremos los datos anteriores:

```
>muestra_orig:=[135,20,7,86,420,5,45,103,230,10,94,100];
>Tn:=sum( (x[i]-sum(x[j],j=1..n)/n)^2, i=1..n )/n;
```

$$Tn := \frac{1}{n} \sum_{i=1}^n \left(x_i - \frac{1}{n} \sum_{j=1}^n x_j \right)^2.$$

```
>n:=12:
>muest_boot:= [Boot(muestra_orig,100)]:
>MonteCarlo(Tn, muestra, 100, muest_boot);
```

sesgo := -1339.54, *varianza* := 0.5814306706 10⁸.

```
>NPaproximar(Sn, muestra_orig, 100, 2,muest_boot);
```

sesgo := -1081.33717, *varianza* := 0.5150121171 10⁸

Comparamos las aproximaciones obtenidas mediante el método de Monte Carlo estandar MonteCarlo(.) y las aproximaciones obtenidas mediante el método ortogonal NPaproximar(.) a los valores exactos en (9.4) podemos ver con toda claridad que el método bootstrap ortogonal es mucho mejor que el método bootstrap estandar.

Consideremos el **Ejemplo A** del Capítulo 1 donde el estimador de interes es la media muestral $\hat{\xi}_n = \bar{X}_n$. Como hemos visto, los valores exactos del sesgo y la varianza bootstrap ideales de $\hat{\xi}_n$ aplicando el método no paramétrico son $\hat{s} = 0$ y $\hat{v} = 1081.339697$ respectivamente. Las aproximaciones a dichos valores usando el método tradicional y el método ortogonal se han obtenido ejecutando las siguientes instrucciones:

```
>with(NPestima):
>muestra_orig:=[135,20,7,86,420,5,45,103,230,10,94,100];
```

```
>Xn:=sum(x[i], i=1..n )/n;
```

$$X_n := \frac{1}{n} \sum_{i=1}^n x_i.$$

```
>n:=12:
```

```
>muest_boot:= [Boot(muestra_orig,20)]:
```

```
>MonteCarlo(Xn, muestra, 20, muest_boot);
```

```
sesgo := 6.4708334,    varianza := 1137.077.
```

```
>NPaproximar(Sn, muestra_orig, 20, 1,muest_boot);
```

```
sesgo := -0.14 10-5,    varianza := 1081.339757.
```

Notemos la diferencia de precisión entre los dos métodos. Si bien, el número de muestras bootstrap, $B = 20$, en este caso es bastante pequeño, por eso parece que el método bootstrap estandar es impreciso. Sin embargo, exactamente con las mismas muestras bootstrap el método bootstrap ortogonal es practicamente exacto.

El programa `Frecboot(.)` nos permite transformar muestras bootstrap simuladas de manera no paramétrica a vectores de frecuencias bootstrap. Es decir, si $\mathcal{X}_n^* = (X_1^*, \dots, X_n^*)$ es una muestra bootstrap extraída de la f.d.D empírica F_n asociada a cierta muestra $\mathcal{X}_n = (X_1, \dots, X_n)$ entonces `Frecboot(.)` transforma \mathcal{X}_n^* al vector $\mathbf{N}^* = (N_1^*, \dots, N_n^*)$ donde N_j^* es el número de apariciones de la observación X_j en la muestra bootstrap \mathcal{X}_n^* . Tambien nos permite transformar de golpe un conjunto de muestras bootstrap $\mathcal{X}_n^{*1}, \dots, \mathcal{X}_n^{*B}$ al conjunto de vectores $\mathbf{N}^{*1}, \dots, \mathbf{N}^{*B}$ como acabamos de explicar. `Frecboot(.)` necesita dos argumentos: el primero es una o un conjunto de muestras bootstrap y el segundo es la muestra original a partir de la cual fueron simuladas dichas muestras bootstrap. Para ver un ejemplo de ejecución consideremos la muestra anterior:

```
>muestra_orig:= [135,20,7,86,420,5,45,103,230,10,94,100]:
```

```
>muest_boot:= [Boot(muestra_orig, 3)]:
```

```
>muest_boot;
```

```
[ [94,230,100,420,10,7,420,45,10,103,10], [7,94,20,86,86,10,86,230,7,135,103],  
[135,420,100,45,94,45,5,420,7,86,94,5] ].
```

```
>Frecboot(muest_boot, muestra_orig);
```

[[0,0,1,1,2,0,1,1,1,3,1,1], [1,1,2,3,0,0,1,1,1,1,0], [1,0,1,1,2,2,2,0,0,2,1]].

Sea \mathbf{N} la variable aleatoria *Multinomial* de dimensión n y con media $\mathbf{m}_0 = \mathbf{1}_n$, el vector de dimensión n cuyos componentes son todos igual a 1. Notemos por V la matriz de varianza de \mathbf{N} . Sea A una matriz tal que AVA^t es diagonal, cual existe siempre (Ver el capítulo 6). Consideremos la transformada de \mathbf{N} dada por $\mathbf{M}=\mathbf{A}\mathbf{N}$. El programa PolMultinom(.) calcula los polinomios ortogonales respecto a la distribución de \mathbf{M} . Ver el capítulo 6 para mas detalles del procedimiento y sus objetivos. PolMultinom(.) admite dos argumentos: el primero es la dimensión n de \mathbf{N} , y el segundo un entero r que es el grado maximo de los polinomios ortogonales que queramos calcular. Por ejemplo, supongamos que $r = 2$, entonces si ejecutamos Polmultinom($n,2$) el programa calcula todos los polinomios de primer grado $P_{e_i}(\cdot)$, $i = 1, \dots, n$ y todos los de segundo grado $P_{2e_i}(\cdot)$, $i = 1, \dots, n$ y $P_{e_i+e_j}(\cdot)$, $i = 1, \dots, n-1$ y $j = i+1, \dots, n$, y almacena el resultado en las variables globales $Pm[s, i]$ y $Qm[i, j]$. La variable $Pm[s, i]$ almacenara el polinomio $P_{se_i}(\cdot)$, $i = 1, \dots, n$ y $s = 1, 2$. Por otro lado, la variable $Qm[i, j]$ almacenara el polinomio $P_{e_i+e_j}(\cdot)$, $i = 1, \dots, n-1$ y $j = i+1, \dots, n$. Recordamos que aquí se trata de polinomios multivariantes de dimensión n . Ejemplo de ejecución, para $n = 10$:

```
>PolMultinom(10,2);
>xx:=seq(x[i],i=1..10);
```

$$xx := x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}$$

```
>Pm[1,5](xx);
```

$$x_5 - \frac{\sqrt{30}}{3}.$$

```
>Pm[2,3](xx);
```

$$x_3^2 - \frac{13\sqrt{14}}{14}x_3 + \frac{\sqrt{2}}{12}x_2 + \frac{\sqrt{10}}{30}x_1 + \frac{45}{28}.$$

```
>Qm[4,9](xx);
```

$$x_4x_9 - 5\sqrt{2}x_4 - \frac{3\sqrt{42}}{14}x_9 + \frac{15\sqrt{21}}{7}.$$

Tando `Frecboot(.)` como `PolMultinom(.)` además de que son programas independientes, también son subprogramas isenciales del programa principal `Mutinomaprox(.)` cuyo objetivo es similar a `NPaproximar(.)` visto antes. `Mutinomaprox(.)` aproxima el sesgo y la varianza bootstrap ideales de un estimador $\hat{\xi}_n^*$ usando su desarrollo en términos de polinomios ortogonales respecto a la distribución *Multinomial* como se ha explicado en el capítulo 6. Los argumentos admitidos por `Mutinomaprox(.)` son los mismos admitidos por `NPaproximar(.)`. Para ver un ejemplo de ejecución consideremos los datos del ejemplo de los Parches hormonales (Ver Tabla 15-6 en el Capítulo 6), donde disponemos de $n = 8$ observaciones de una variable aleatoria bidimensional $X = (Y, Z)$. El estadístico de interés es $\hat{\xi}_n = \bar{Y}/\bar{Z}$ y los datos son $(-1200, 8406)$, $(2601, 2342)$, $(-2705, 8187)$, $(1982, 8489)$, $(-1290, 4795)$, $(351, 3516)$, $(-638, 4796)$, $(-2719, 10238)$. El sesgo \hat{s} y error estandar \hat{se} ideales de $\hat{\xi}_n$ (obtenidos con $B = 10000$ simulaciones) son:

$$\hat{s} = 0.0084 \text{ y } \hat{se} = 0.102.$$

La estimación del sesgo y del error estandar obtenidos en EFRON (1993) basados sobre el método bootstrap estandar con $B = 400$ simulaciones son

$$\hat{s}_{MC}^* = 0.0043 \text{ y } \hat{se}_{MC}^* = 0.105.$$

Aplicamos el método bootstrap ortogonal, donde se usa el desarrollo ortogonal multinomial de $\hat{\xi}_n^*$, ejecutando `Mutinomaprox(.)`, hemos obtenido los siguientes resultados en la hoja de Maple:

```
>muestra_orig:=[ [-1200,8406],[2601,2342],[-2705,8187],[1982,8489],[-
1290,4795],[351,3516],
[-638,4796],[-2719,10238] ];
>muestras_boot:=[Boot(muestra_orig,400)];
>Tn:=sum(x[i],i=1..n)/sum(y[i],i=1..n);

$$Tn := \frac{\sum_{i=1}^n x_i}{\sum_{i=1}^n y_i}$$

>Multinomaprox(Tn,muestra_orig, 400, 2, muestras_boot);
sesgo := 0.00811,   varianza := 0.01028.
>error_estandar:=sqrt(0.01028);
error_estandar := 0.1014.
```

Por lo que la aproximación del error estandar de $\hat{\xi}_n$ mediante el método bootstrap ortogonal es $\hat{s}e_{Ort}^* = 0.1014$ cual se ve que es mas precisa como aproximación de $\hat{s}e$ que $\hat{s}e_{MC}^*$.

9.3 El Paquete Repita

Sea $\hat{\xi}_n$ un estadístico al que queremos aproximar el sesgo \hat{s} o la varianza \hat{v} ideales mediante algún método bootstrap, estandar o ortogonal. Supongamos que el estimador admite un desarrollo en terminos de algún sistema ortogonal $\{\psi_k\}_k$. Como hemos visto en el Capítulo 7, los cálculos se hacen considerando el desarrollo hasta algún orden m como sigue

$$\hat{\xi}_n^* = \sum_{j=0}^m a_j \psi_j^* + \varepsilon. \quad (9.5)$$

Podemos siempre escoger un grado m adecuado en (9.5) para efectuar las aproximaciones (del sesgo y de la varianza). Sea $s_{B,m}^{*Ort}$ y $v_{B,m}^{*Ort}$ respectivamente las aproximaciones del sesgo y la varianza de $\hat{\xi}_n^*$ mediante el método bootstrap ortogonal basado sobre un desarrollo (9.5) hasta el grado m y usando B simulaciones bootstrap. Consideremos la estimación del sesgo y supongamos, por ejemplo, que queremos calcular las r realizaciones $s_{B,1}^{*Ort}, s_{B,1}^{*Ort}, \dots, s_{B,r}^{*Ort}$ del sesgo. Esto se puede realizarse ejecutando r veces seguidamente el programa `Aproximar(.)` aumentando su quinto argumento una unidad en cada ejecución. O supongamos, por ejemplo, que queremos calcular $s_{20,r}^{*Ort}, s_{30,r}^{*Ort}, \dots, s_{100,r}^{*Ort}$ donde variamos el número de simulaciones B de 20 hasta 100 con salto de 10. Esto tambien puede hacerse ejecutando `Aproximar(.)` 9 veces variando su cuarto argumento y posiblemente su sexto argumento como se explico en la sección 1 del actual capítulo. El paquete `Repita` incluye programas que nos permiten realizar lo dicho de forma automatica sin tener que ejecutar muchas veces algun programa como `NPestima(.)` o `Estima(.)` con el fin de obtener lo que queremos. Para entender mejor la idea, consideremos la *Tabla 6-1* donde se recopilan varias aproximaciones del sesgo \hat{s} y la varianza \hat{v} variando el número de simulaciones B de 6 hasta 200. Dicha table fue obtenida de una vez simplemente ejecutando un programa del paquete `Repita` como vamos a ver. El contenido de `Repita` es el siguiente:

```
>with(Repita);
```

```
[IdealRep, PMCrepetir, Recortar, Repetir, Resimular].
```

Empezamos con el programa `Recortar(.)`. Este programa nos permite repetir la ejecución del programa `Aproximar(.)` variando su quinto argumento un número determinado de veces. Un ejemplo es la *Tabla 7-2* que es el resultado de la ejecución de `Recortar(.)` donde este programa llama 7 veces al programa `Aproximar(.)` y donde en cada ejecución el quinto argumento es aumentado una unidad manteniendo fijas las muestras bootstrap. Otro programa parecido es `Resimular(.)` cuya función es reejecutar el programa `Aproximar(.)` variando su cuarto argumento. El resultado de ejecución, por ejemplo, es la segunda, la tercera y la quinta columna de la *Tabla 6-1* que se obtienen de golpe. El programa `PMCrepetir(.)` reejecuta el programa `MC(.)` del paquete `Estima` variando el número de muestras bootstrap B . El resultado, por ejemplo, es la cuarta y la sexta columna de la *Tabla 6-1* que se obtienen de una vez. En seguida vemos un ejemplo de ejecución de dichas programas en una sección de *Maple*. Para ello, consideremos la *Tabla 6-1* pero con un salto de 40 entre los valores de B , y la *Tabla 6-2* sin las dos últimas líneas:

```
>with(Estima):
>with(Repita):
>muestra_orig:=Muestra(normal[7,1],50):
>media:=stats[describe,mean](muestra):
>muestras_boot:=[Muestra(normal[media,1],50,200):
>h:=mu^4+3*mu:
>Tn:=x^4-6*x^2/n+3*x+3/n^2:
>Resimular([h,Tn],normal[mu,1],muestra_orig,200,3,4,40,muestras_boot);
```

B	$var1$	$var2$	$sego$
200	32781.34516	32781.34516	10^{-6}
160	32781.34522	32781.34522	0
120	32781.34525	32781.34525	$0.2 \cdot 10^{-5}$
80	32781.34526	32781.34526	10^{-6}
40	32781.34510	32781.34511	0
10	32781.37416	32781.37415	$0.68 \cdot 10^{-4}$

```
>Recortar([h,Tn],normal[mu,1],muestra_orig,200,1,4,mboot);
```

$grado\ m$	$var1$	$var2$	$sesgo$
1	33140.27952	33140.27389	-0.153428
2	32774.49102	32774.49101	0.016204
3	32781.44662	32781.44662	$-0.83 \cdot 10^{-4}$
4	32781.34516	32781.34516	10^{-6}
5	32781.34512	32781.34512	0

```
>PMCrepetir([h,Tn],muestra_orig,muestras_boot,5,40);
```

B	varianza	sesgo
200	32079.2066	-2.0208
160	32202.6725	-10.3236
120	35643.4437	-14.3102
80	33240.7893	-18.7976
40	32898.1225	-31.2732
10	11528.2103	30.23023

El programa Repetir(.) nos permite reejecutar algunas programas como NPaproximar(.), Multinomaprox(.) y MonteCarlo(.) del paquete NPestima. Sea $\hat{\theta}_n$ un cierto estadístico y supongamos que disponemos de N muestras originales $\mathcal{X}_n^j = (X_1^j, \dots, X_n^j)$, $j = 1, \dots, N$ donde cada muestra es de tamaño n . Supogamos que para cada muestra \mathcal{X}_n^j , necesitamos obtener M realizaciones bootstrap, digamos C_m^j $m = 1, \dots, M$, de una cierta característica C del estadístico $\hat{\theta}$ donde C puede ser el sesgo s o la varianza v de $\hat{\theta}$. El cálculo de repetidas realizaciones de C nos permite comparar dos métodos de aproximación bootstrap calculando el error cuadrático medio de las realizaciones. Con el fin de realizar lo dicho, necesitaremos por ejemplo ejecutar M veces alguna de las programas NPaproximar(.), Multinomaprox(.) o MonteCarlo(.) para cada una de las N muestras originales. Esto significa que necesitaremos un total de MN ejecuciones. El programa Repetir(.) ejecutado con adecuados argumentos hace que dichas MN ejecuciones se haces de forma automática. Por ejemplo, los resultados numéricos de la subsección "Resultados númricos" de la primera sección del Capítulo 6 son obtenedos usando el programa Repetir(.). La forma general de los argumentos de Repetir(.) es la siguiente

```
Repetir(Estimador, Muestras_originales, M, B, Grado, Programa);
```

donde Estimador representa la expresión del estadístico $\hat{\theta}(X_1, \dots, X_n)$ como función de los datos, Muestras_originales es una lista de N muestras originales o simplemente una muestra original si desponemos solamente de una. M sería el número de realizaciones de la característica C de $\hat{\theta}$ que se requieran y B el número de muestras bootstrap usadas para el calculo de cada realización C . El argumento Grado sería el grado del desarrollo ortogonal de $\hat{\theta}$ que va a ser usado para efectuar los cálculos. Programa represente el nombre del programa que queremos usar para obtener las realizaciones C mediante el método ortogonal. Entonces Programa puede ser NPaproximar(.) o Multinomaprox(.). Además de dichos argumentos, Repetir(.) admite un septimo argumento que sera una lista de N listas de M grupos de muestras, donde cada uno de las M grupos esta formado de B muestras bootstrap.

9.4 Otros Programas

9.4.1 Familia NEF

En primer lugar veremos el paquete *NEF* que incluye programas relacionados con la familia de distribuciones *Exponencial Natural*. Dicho paquete esta formado por cuatro programas:

```
>with(NEF);
```

[*Cumulante, Momento, MomentoCentrado, Resumen*].

Empezamos con el programa *Resumen(.)* cuyo objetivo es dar un pequeño resumen sobre cualquier miembro de la familia *NEF* pasado al programa como argumento. Por ejemplo, podemos estar trabajando con una familia determinada y en momento concreto necesitamos consultar algo sobre la familia como por ejemplo la expresión explícita de su distribución, su media, su función característica, etc. El programa *Resumen(.)* nos proporciona dicha información. Consideremos por ejemplo, la familia *Secante Hiperbólica Generalizada, SHG(r, θ)* :

```
>Resumen(shg);
```

Sec. Hiperb. Generalizada	$SHG(r, \theta), \theta < \frac{\pi}{2}$ y $\rho = \tan(\theta), r > 0$.
Función de Densidad	$f(x, r, \rho) = (1 + \rho^2)^{-r/2} \exp(x \arctan(\rho)) f(x, r, 0)$ con $f(x, r, 0) = \frac{2^{r-2}}{\Gamma(r)} \prod_{j=0}^{\infty} \left[1 + \frac{x^2}{(r-2j)^2} \right]^{-1}$.
Media	$\mu = r\rho$
Función de Varianza	$V(\mu) = r^2(1 + \rho^2) = r + \frac{\mu^2}{r}$.
Estad. suf. y completo	La media muestral \bar{x} .
Densidad de \bar{x}	$f_n(s, r, \rho) = n(1 + \rho^2)^{-nr/2} \exp(ns \arctan(\rho)) f(ns, nr, 0)$.
Func. Gener. de cumulantes	$\psi(t) = -r \ln(\cos(t) - \rho \sin(t))$.

Por otra parte, los programas *Cumulante(.)*, *Momento(.)* y *MomentoCentrado(.)* como sus nombres indican, nos permiten calcular numéricamente o simbólicamente los cumulantes, los momentos y los momentos centrados de una variable aleatoria

perteneiente a una familia *NEF*. Dichos programas utilizan los mismos argumentos y se ejecutan usando la siguiente expresión

Programa(Orden, Distribución[Paramétros]);

donde Programa representa cualquiera de los tres: Cumulante, Momento o Momento-Centrado. La palabra Orden es un entero positivo que sera el orden del cumulante, del momento o del momento centrado requeridos. Distribución representa el nombra de la distribución miembro de la familia *NEF*, y Paramétros son los paramétros de dicha distribución. Consideremos, por ejemplo, la sesión Maple siguiente:

>Cumulante(6, binomial[N, p]);

$$Np - 31Np^2 + 180Np^3 - 390Np^4 + 360Np^5 + 24Np^5.$$

>Momento(4, binomial[10, p]);

$$5040p^4 + 4320p^3 + 630p^2 + 10p.$$

>Momento(5, poisson[μ]);

$$\mu + 15\mu^2 + 25\mu^3 + 10\mu^4 + \mu^5.$$

>MomentoCentrado(7, gamma[α, θ]);

$$1854\theta^7\alpha.$$

9.4.2 Estimación de la Densidad y la Distribución

Para la aproximación de la densidad y de la distribución mediante las fórmulas de inversión, como se ha explicado en el capítulo anterior, disponemos de un paquete de cuatro programas. Dos de ellos relacionados con la estimación de la densidad y otros dos con la estimación de la distribución. El paquete de programas le hemos llamado FFTestima, y contiene los siguientes programas:

>with(FFTestima);

[DensidadPuntos, DensidadEstima, DistribucionEstima, InversionDirectaDistib].

Antes supongamos que las notaciones del capítulo anterior (como Z_n , T , m , P_j , ...) tienen el mismo significado y denotaremos por *muestra_orig* una lista de números que constituyen la muestra original \mathcal{X}_n . El programa `DensidadPuntos()` nos permite obtener una aproximación a $2m$ puntos P_j , $j = -m, \dots, m-1$, pertenecientes al gráfico de la densidad (Ver Figura 8-6). Dicho programa se ejecuta con tres argumentos *muestra_orig*, m y T y un cuarto *func_caract* como opcional. Este último argumento representa la expresión de la función característica de X_1 en el caso paramétrico cuando se supone que es conocida. Si el programa `DensidadPuntos()` se ejecuta solamente con tres argumentos entonces se utiliza el método de estimación no paramétrica (Ver el capítulo anterior). Sea *2mPuntos* la lista de los $2m$ puntos obtenidos como resultado de la ejecución de `DensidadPuntos()`.

Los programas `DensidadEstima()` y `DistibucionEstima()` se ejecutan siempre después de `DensidadPuntos()` puesto que suponen que ya disponemos de los $2m$ puntos de la densidad. Dichos programas nos permiten aproximar a la densidad $g_n^*(x)$ y a la distribución $G_n^*(x)$ respectivamente y en cualquier punto x . Las aproximaciones se basan sobre los $2m$ puntos obtenidos usando las fórmulas (8.11) y (8.13) del capítulo anterior. Cada una de los dos programas `DensidadEstima()` y `DistibucionEstima()` se ejecuta con dos argumentos x y *2mPuntos* donde x es el punto en el que se desea una aproximación para la densidad $g_n^*(x)$ o la distribución $G_n^*(x)$.

El programa `InversionDirectaDistrib(.)`, al igual que `DistibucionEstima()`, nos permite aproximar a $G_n^*(x)$ en cualquier punto real x , pero usando la fórmula de inversión correspondiente a la distribución directamente sin necesidad de estimar la densidad. La aproximación se basa sobre el método de los trapecios explicado en el capítulo anterior (Ver (8.15)). Dicho programa se ejecuta con seis argumentos x , *muestra_orig*, m , T , a y Fa . Los cuatro primeros son igual que como se ha explicado antes, a es un punto real en el que se supone que conocemos a $Fa = G_n^*(a)$.

En una sesión de *Maple* el sintaxis de los programas anteriores se ve como sigue:

```
>2mPuntos :=DensidadPuntos( muestra_orig, m, T );
>2mPuntos :=DensidadPuntos( muestra_orig, m, T, func_caract);
>DensidadEstima(x, 2mPuntos);
>DistibucionEstima(x, 2mPuntos);
>InversionDirectaDistrib(x, muestra_orig, m, T, a, Fa);
```

9.5 Códigos fuente de los programas

9.5.1 Estima.

El código del paquete Estima:

```
Estima := table([

Hermite = (proc(x, k)
  k!*sum((-1)^j*(2*x)^(k - 2*j)/((k - 2*j)!*j!),
  j = 0 .. floor(1/2*k))
end)

Lag = (proc(x, g, k)
  sum((k + g)!*(-x)^j/(j!*(g + j)!*(k - j)!), j = 0 .. k)
end)

Krawt = (proc(x, p, N, k)
  sum(x!*(N - x)!*(-1)^j*p^j*(1 - p)^(k - j)/(
  j!*(k - j)!*(x - k + j)!*(N - x - j)!), j = 0 .. k)
end)

Meix = (proc(x, alpha, cc, k)
  (-1)^(2*k)*k!*sum(x!*(-x - alpha)!*cc^(-j)/(
  j!*(x - j)!*(k - j)!*(-x - alpha - k + j)!), j = 0 .. k)
end)

Charlier = (proc(x, e, k)
  expand(sum(k!*x!*(-e)^(k - j)/(j!*(k - j)!*(x - j)!),
  j = 0 .. k))
end)

Pherm = (proc(x, mu, sigma, n, k)
  expand(
  sigma^k*Estima['Hermite'](sqrt(1/2*n)*(x - mu)/sigma, k)/
  (2*n)^(1/2*k))
end)
```

```
Pjac = (proc(x, theta, n, k)
  expand(GAMMA(n + k)*sum(
    (-1)^i*k!*GAMMA(n + 2*k - i)*(x/theta)^(k - i)/(
    i!*(k - i)!*GAMMA(n + k - i)), i = 0 .. k)/GAMMA(n + 2*k))
end)
```

```
Plag = (proc(x, mu, alpha, n, k)
  if nargs = 5 then sort(expand((-1)^k*k!*mu^k*
    Estima['Lag'](n*alpha*x/mu, n*alpha - 1, k)/
    (n*alpha)^k), x)
  else (-1)^k*k!*mu^k*
    Estima['Lag'](n*alpha*x/mu, n*alpha - 1, k)/
    (n*alpha)^k
  fi
end)
```

```
Pcharl = (proc(x, mu, n, k)
  expand(Estima['Charlier'](n*x, n*mu, k)/n^k)
end)
```

```
Pkrawt = (proc(x, mu, N, n, k)
  expand(k!*Estima['Krawt'](n*x, mu/N, n*N, k)/n^k)
end)
```

```
Pmeix = (proc(x, mu, r, n, k)
  (mu/r)^k*Estima[Meix](n*x, n*r, mu/(r + mu), k)/n^k
end)
```

```
Norma = (proc(x, n, k)
  local mu, theta, sigma, alpha, N, p, r;
  if op(0, x) = 'Pperm' then
    sigma := op(2, x); RETURN(sqrt(k!*sigma^(2*k)/n^k))
  elif op(0, x) = 'Pcharl' then
    RETURN(sqrt(k!*op(1, x)^k/n^k))
  elif op(0, x) = 'Pkrawt' then
    N := op(1, x);
    p := op(2, x);
    RETURN(sqrt(k!*GAMMA(n*N + 1)*(p*(1 - p))^k/((
```

```

        GAMMA(n*N - k + 1)*n^(2*k))))
elif op(0, x) = 'Plag' then RETURN(sqrt(
    k!*GAMMA(n*op(1, x) + k)*op(2, x)^(2*k)/(
    n^(2*k)*GAMMA(n*op(1, x))))))
elif op(0, x) = 'Pmeix' then
    r := op(1, x);
    p := op(2, x);
    RETURN(sqrt(k!*GAMMA(n*r + k)*(1 - p)^k/(
        GAMMA(n*r)*(n*p)^(2*k))))))
elif op(0, x) = 'Pjac' then RETURN(
    sqrt(n/(n + 2*k))*k!*(n + k - 1)!/(n + 2*k - 1)!)
fi
end)

Umvue = (proc(h::algebraic, c)
local d, T, v, a, aa, w, d1, sigma, N, r, alpha;
    d := degree(h, mu);
    if op(0, c) = 'normal' then
        sigma := op(2, c);
        T := h + sum(diff(h, mu $ i)*
            Estima['Pherm'](x, mu, sigma, n, i)/i!,
            i = 1 .. d);
        v := sum(diff(h, mu $ i)^2*
            Estima['Norma'](Pherm[mu, sigma], n, i)^2/(i!)^2,
            i = 1 .. d);
        RETURN([expand(T), collect(expand(v), mu)])
    elif op(0, c) = 'poisson' then
        T := h + sum(
            diff(h, mu $ i)*Estima['Pcharl'](x, mu, n, i)/i!,
            i = 1 .. d);
        v := sum(diff(h, mu $ i)^2*
            Estima['Norma']('Pcharl[mu]', n, i)^2/(i!)^2,
            i = 1 .. d);
        RETURN([expand(T), collect(expand(v), mu)])
    elif op(0, c) = 'binomial' then
        N := op(1, c);
        a :=
            (i, q) -> i!*product(1 - j/(n*q), j = 0 .. i - 1)

```

```

;
T := h + sum(
    diff(h, mu $ i)*Estima['Pkrawt'](x, mu, N, n, i)/
    a(i, N), i = 1 .. d);
v := subs(p = mu/N, sum(diff(h, mu $ i)^2*
    Estima['Norma'](Pkrawt[N, p], n, i)^2/a(i, N)^2,
    i = 1 .. d));
RETURN([expand(simplify(T)), collect(expand(v), mu)])
elif op(0, c) = 'gamma' then
aa := i!*product(1 + j/(op(1, c)*n), j = 0 .. i - 1);
a := unapply(aa, i);
T := h + sum(diff(h, mu $ i)*
    Estima['Plag'](x, mu, op(1, c), n, i)/a(i),
    i = 1 .. d);
v := sum(diff(h, mu $ i)^2*Estima['Norma'](
    Plag[op(1, c), mu/op(1, c)], n, i)^2/a(i)^2,
    i = 1 .. d);
RETURN([expand(simplify(T)), collect(expand(v), mu)])
elif op(0, c) = 'bnegat' then
r := op(1, c);
a :=
    (i, q) -> i!*product(1 + j/(n*q), j = 0 .. i - 1)
;
T := h + sum(
    diff(h, mu $ i)*Estima['Pmeix'](x, mu, r, n, i)/
    a(i, r), i = 1 .. d);
v := subs(p = r/(r + mu), sum(diff(h, mu $ i)^2*
    Estima['Norma'](Pmeix[r, p], n, i)^2/a(i, r)^2,
    i = 1 .. d));
RETURN([simplify(expand(T)),
    collect(expand(simplify(v)), mu)])
elif op(0, c) = 'uniform' then
d1 := degree(h, theta);
a := (x, n, k) -> ((k + n)*x^k)/n;
w := (theta, n, k) -> k^2*theta^(2*k)/(n*(n + 2*k));
T := sum(coeff(h, theta, i)*a(x, n, i), i = 0 .. d1);
v := sum(coeff(h, theta, i)*w(theta, n, i),
    i = 0 .. d1);

```

```

        RETURN([T, v])
    fi
end)

Maximo = (proc(l::list)
local j, z, n;
    n := nops(l);
    z := op(1, l);
    for j from 2 to n do
        if z < op(j, l) then z := op(j, l) fi
    od;
    RETURN(z)
end)

Suprimir = (proc(r::posint, l::list)
global datos;
    if nops(l) < r then
        ERROR('la lista tiene menos de '.r.' elementos.')
    else datos := [op(1 .. nops(l) - r, l)]
    fi
end)

Bin = (proc(N, p, n)
local K, k, t, i, v, u, x, j, r, mues;
    for r to n do
        x[r] := 0;
        K := 2*N;
        k := N;
        t := p;
        while k <= K and floor(1 + k*t) <> k + 1 do
            i := floor(1 + k*t);
            v := stats[random, beta[i, k + 1 - i]]();
            if t < v then t := t/v; k := i - 1
            else
                x[r] := x[r] + i;
                t := (t - v)/(1 - v);
                k := k - i
            fi
        fi
    end
end)

```

```

    od;
  for j to k do
    u := stats[random, uniform[0, 1]]();
    if u < p then x[r] := x[r] + 1 fi
  od
od;
RETURN([seq(x[r], r = 1 .. n)])
end)

Muestra = (proc(c, n::posint)
local i, B;
global mnor, mgam, mpois, mbin, mbneg, mbet, mchi, munif;
  mnor := 'mnor';
  mgam := 'mgam';
  mpois := 'mpois';
  mbneg := 'mbneg';
  mbet := 'mbet';
  mchi := 'mchi';
  munif := 'munif';
  mbin := 'mbin';
  if op(0, c) = normal then
    if nargs = 2 then mnor :=
      [stats[random, normald[op(1, c), op(2, c)]](n)]
    elif nargs = 3 then
      B := args[3];
      for i to B do mnor[i] := [
        stats[random, normald[op(1, c), op(2, c)]](n)
      ]
    od
  fi
  elif op(0, c) = binomial then
    if nargs = 2 then
      mbin := Estima[Bin](op(1, c), op(2, c), n)
    elif nargs = 3 then
      B := args[3];
      for i to B do
        mbin[i] := Estima[Bin](op(1, c), op(2, c), n)
      od
    end
  end
end)

```

```
fi
elif op(0, c) = gamma then
  if nargs = 2 then mgam :=
    [stats[random, gamma[op(1, c), op(2, c)]](n)]
  elif nargs = 3 then
    B := args[3];
    for i to B do mgam[i] :=
      [stats[random, gamma[op(1, c), op(2, c)]](n)]
    od
  fi
elif op(0, c) = poisson then
  if nargs = 2 then
    mpois := [stats[random, poisson[op(1, c)]](n)]
  elif nargs = 3 then
    B := args[3];
    for i to B do mpois[i] :=
      [stats[random, poisson[op(1, c)]](n)]
    od
  fi
elif op(0, c) = bnegat then
  if nargs = 2 then mbneg := [stats[
    random, negativebinomial[op(1, c), op(2, c)]](n)]
  elif nargs = 3 then
    B := args[3];
    for i to B do mbneg[i] := [stats[
      random, negativebinomial[op(1, c), op(2, c)]](
      n)]
    od
  fi
elif op(0, c) = beta then
  if nargs = 2 then mbet :=
    [stats[random, beta[op(1, c), op(2, c)]](n)]
  elif nargs = 3 then
    B := args[3];
    for i to B do mbet[i] :=
      [stats[random, beta[op(1, c), op(2, c)]](n)]
    od
  fi
fi
```

```

elif op(0, c) = uniform then
  if nargs = 2 then munif :=
    [stats[random, uniform[op(1, c), op(2, c)]]](n)
  elif nargs = 3 then
    B := args[3];
    for i to B do munif[i] := [
      stats[random, uniform[op(1, c), op(2, c)]](n)
    ]
  od
fi
elif op(0, c) = chisquare then
  if nargs = 2 then
    mchi := [stats[random, chisquare[op(1, c)]]](n)
  elif nargs = 3 then
    B := args[3];
    for i to B do mchi[i] :=
      [stats[random, chisquare[op(1, p)]]](n)
    od
  fi
fi
end)

```

```

Pdata = (proc(c, n, b, m)
local N, alpha, sigma, j, i, p;
global Pn, s, Pdat;
  if op(0, c) = normal then
    sigma := op(2, c);
    p := op(1, c);
    for i to b do s[i] := stats[describe, mean](mnorb[i])
    od;
    for j to m do Pn[j] := evalf(
      [seq(Pherm(s[i], p, sigma, n, j), i = 1 .. b)])
    od;
    Pdat := [seq(Pn[j], j = 1 .. m)]
  elif op(0, c) = gamma then
    alpha := op(1, c);
    p := op(1, c)*op(2, c);
    for i to b do s[i] := stats[describe, mean](mgamb[i])

```

```

od;
for j to m do Pn[j] := evalf(
  [seq(Plag(s[i], p, alpha, n, j, 1), i = 1 .. b)])
od;
Pdat := [seq(Pn[j], j = 1 .. m)]
elif op(0, c) = binomial then
N := op(1, c);
p := op(1, c)*op(2, c);
for i to b do s[i] := stats[describe, mean](mbinb[i])
od;
for j to m do Pn[j] := evalf(
  [seq(Pkrawt(s[i], p, N, n, j), i = 1 .. b)])
od;
Pdat := [seq(Pn[j], j = 1 .. m)]
elif op(0, c) = poisson then
p := op(1, c);
for i to b do
  s[i] := stats[describe, mean](mpoisb[i])
od;
for j to m do Pn[j] :=
  evalf([seq(Pcharl(s[i], p, n, j), i = 1 .. b)])
od;
Pdat := [seq(Pn[j], j = 1 .. m)]
elif op(0, c) = uniform then
p := op(1, c);
for i to b do s[i] := Maximo(munifb[i]) od;
for j to m do Pn[j] :=
  evalf([seq(Pjac(s[i], p, n, j), i = 1 .. b)])
od;
Pdat := [seq(Pn[j], j = 1 .. m)]
fi
end)

Ydata = (proc(h, c, mn, b)
local Tn, i;
  if op(0, c) = normal then
    if type(h, list) then Tn := unapply(op(2, h), x, n)
    else Tn := unapply(op(1, Umvue(h, c)), x, n)

```

```

    fi;
    RETURN([seq(Tn(stats[describe, mean](mnorb[i]), nn),
        i = 1 .. b)])
  elif op(0, c) = binomial then
    if type(h, list) then Tn := unapply(op(2, h), x, n)
    else Tn := unapply(op(1, Umvue(h, c)), x, n)
    fi;
    RETURN([seq(Tn(stats[describe, mean](mbinb[i]), nn),
        i = 1 .. b)])
  elif op(0, c) = gamma then
    if type(h, list) then Tn := unapply(op(2, h), x, n)
    else Tn := unapply(op(1, Umvue(h, c)), x, n)
    fi;
    RETURN([seq(Tn(stats[describe, mean](mgamb[i]), nn),
        i = 1 .. b)])
  elif op(0, c) = poisson then
    if type(h, list) then Tn := unapply(op(2, h), x, n)
    else Tn := unapply(op(1, Umvue(h, c)), x, n)
    fi;
    RETURN([seq(Tn(stats[describe, mean](mpoisb[i]), nn),
        i = 1 .. b)])
  elif op(0, c) = uniform then
    if type(h, list) then Tn := unapply(op(2, h), x, n)
    else Tn := unapply(op(1, Umvue(h, c)), x, n)
    fi;
    RETURN([seq(Tn(Maximo(munifb[i]), nn), i = 1 .. b)])
  fi
end)

```

```

Aproximar = (proc(h, c, l::list, b::posint, m::posint)
local nn, mm, ecc, hh, v, vv, v1, e, i, j, yy, er, sesg,
ydata, pdata, ec, bb, X, Xt, A, IA, Dg;
global mboot, mgamb, mnorb, mbinb, mpoisb, munifb, resultado,
sigma, alpha, N_;
  v1 := 0;
  e := 0;
  nn := nops(l);

```

```

if op(0, c) = normal then
  sigma := op(2, c);
  if type(h, list) then hh := unapply(op(1, h), mu)
  else hh := unapply(h, mu)
  fi;
  mm := stats[describe, mean](1);
  if nargs = 6 then
    for i to b do mnorb[i] := op(i, args[6]) od
  else for i to b do mnorb[i] :=
    [stats[random, normald[mm, sigma]](nn)]
    od
  fi;
  mboot := [seq(mnorb[i], i = 1 .. b)];
  ydata := Ydata(h, c, nn, b);
  pdata := Pdata(normal[mm, sigma], nn, b, m);
  Xt := array([[seq(1, i = 1 .. b)], op(pdata)]);
  X := linalg[transpose](Xt);
  A := linalg[multiply](Xt, X);
  IA := linalg[inverse](A);
  Dg := sum(
    IA[j, j]*Estima[Norma](Pherm[mu, sigma], nn, j)^2
    , j = 1 .. m);
  ec := stats[
    fit, leastsquare[[seq(x[i], i = 1 .. m), y]]](
    [op(pdata), ydata]);
  bb := [op(1, rhs(ec)),
    seq(op(i, rhs(ec))/x[i - 1], i = 2 .. m + 1)];
  ecc := unapply(rhs(ec), seq(x[j], j = 1 .. m));
  yy := stats[transform, multiapply[ecc]]([op(pdata)]);
  er :=
    stats[transform, multiapply[(a1, b1) -> a1 - b1]](
    [ydata, yy]);
  sesg := op(1, bb) - hh(mm);
  for j to b do e := e + op(j, er)^2 od;
  e := e/(b - m - 1);
  for j from 2 to nops(bb) do v1 := v1 + op(j, bb)^2*
    Estima[Norma](Pherm[mu, sigma], nn, j - 1)^2
  od;
od;

```

```

v := v1 + e;
vv := v - e*Dg;
resultado := table([Ecuacion = rhs(ec), var('e') = e,
    var(Tn) = v, ivar(Tn) = vv, sesgo = sesg]);
RETURN(eval(resultado))
elif op(0, c) = gamma then
alpha := op(1, c);
if type(h, list) then hh := unapply(op(1, h), mu)
else hh := unapply(h, mu)
fi;
mm := stats[describe, mean](1);
if nargs = 6 then
    for i to b do mgamb[i] := op(i, args[6]) od
else for i to b do mgamb[i] :=
    [stats[random, gamma[alpha, mm/alpha]](nn)]
    od
fi;
mboot := [seq(mgamb[i], i = 1 .. b)];
ydata := Ydata(h, c, nn, b);
pdata := Pdata(gamma[alpha, mm/alpha], nn, b, m);
Xt := array([[seq(1, i = 1 .. b)], op(pdata)]);
X := linalg[transpose](Xt);
A := linalg[multiply](Xt, X);
IA := linalg[inverse](A);
Dg := sum(IA[j, j]*
    Estima[Norma](Plag[alpha, mm/alpha], nn, j)^2,
    j = 1 .. m);
ec := stats[
    fit, leastsquare[[seq(x[i], i = 1 .. m), y]]](
    [op(pdata), ydata]);
bb := [op(1, rhs(ec)),
    seq(op(i, rhs(ec))/x[i - 1], i = 2 .. m + 1)];
ecc := unapply(rhs(ec), seq(x[j], j = 1 .. m));
yy := stats[transform, multiapply[ecc]]([op(pdata)]);
er :=
    stats[transform, multiapply[(a1, b1) -> a1 - b1]](
    [ydata, yy]);
sesg := op(1, bb) - hh(mm);

```

```

for j to b do e := e + op(j, er)^2 od;
e := e/(b - m - 1);
for j from 2 to nops(bb) do v1 := v1 + op(j, bb)^2*
  Estima[Norma](Plag[alpha, mm/alpha], nn, j - 1)^2
od;
v := v1 + e;
vv := v - e*Dg;
resultado := table([Ecuacion = rhs(ec), var('e') = e,
  var(Tn) = evalf(v), ivar(Tn) = evalf(vv),
  sesgo = sesg]);
RETURN(eval(resultado))
elif op(0, c) = binomial then
N_ := op(1, c);
if type(h, list) then hh := unapply(op(1, h), mu)
else hh := unapply(h, mu)
fi;
mm := stats[describe, mean](1);
if nargs = 6 then
  for i to b do mbinb[i] := op(i, args[6]) od
else for i to b do
  mbinb[i] := Estima[Bin](N_, mm/N_, nn)
  od
fi;
mboot := [seq(mbinb[i], i = 1 .. b)];
ydata := Ydata(h, c, nn, b);
pdata := Pdata(binomial[N_, mm/N_], nn, b, m);
Xt := array([[seq(1, i = 1 .. b)], op(pdata)]);
X := linalg[transpose](Xt);
A := linalg[multiply](Xt, X);
IA := linalg[inverse](A);
Dg := sum(IA[j, j]*
  Estima[Norma](Pkrawt[N_, mm/N_], nn, j)^2,
  j = 1 .. m);
ec := stats[
  fit, leastsquare[[seq(x[i], i = 1 .. m), y]]](
  [op(pdata), ydata]);
bb := [op(1, rhs(ec)),
  seq(op(i, rhs(ec))/x[i - 1], i = 2 .. m + 1)];

```

```

ecc := unapply(rhs(ec), seq(x[j], j = 1 .. m));
yy := stats[transform, multiapply[ecc]]([op(pdata)]);
er :=
  stats[transform, multiapply[(a1, b1) -> a1 - b1]](
    [ydata, yy]);
sesg := op(1, bb) - hh(mm);
for j to b do e := e + op(j, er)^2 od;
e := e/(b - m - 1);
for j from 2 to nops(bb) do v1 := v1 + op(j, bb)^2*
  Estima[Norma](Pkrawt[N_, mm/N_], nn, j - 1)^2
od;
v := v1 + e;
vv := v - e*Dg;
resultado := table([Ecuacion = rhs(ec),
  var('e') = evalf(e), var(Tn) = evalf(v),
  ivar(Tn) = evalf(vv), sesgo = sesg]);
RETURN(eval(resultado))
elif op(0, c) = 'poisson' then
  if type(h, list) then hh := unapply(op(1, h), mu)
  else hh := unapply(h, mu)
  fi;
mm := stats[describe, mean](1);
if nargs = 6 then
  for i to b do mpoisb[i] := op(i, args[6]) od
else for i to b do
  mpoisb[i] := [stats[random, poisson[mm]](nn)]
  od
fi;
mboot := [seq(mpoisb[i], i = 1 .. b)];
ydata := Ydata(h, c, nn, b);
pdata := Pdata(poisson[mm], nn, b, m);
Xt := array([[seq(1, i = 1 .. b)], op(pdata)]);
X := linalg[transpose](Xt);
A := linalg[multiply](Xt, X);
IA := linalg[inverse](A);
Dg := sum(
  IA[j, j]*Estima[Norma](Pcharl[mm], nn, j)^2,
  j = 1 .. m);

```

```

ec := stats[
  fit, leastsquare[[seq(x[i], i = 1 .. m), y]]](
  [op(pdata), ydata]);
bb := [op(1, rhs(ec)),
  seq(op(i, rhs(ec))/x[i - 1], i = 2 .. m + 1)];
ecc := unapply(rhs(ec), seq(x[j], j = 1 .. m));
yy := stats[transform, multiapply[ecc]]([op(pdata)]);
er :=
  stats[transform, multiapply[(a1, b1) -> a1 - b1]](
  [ydata, yy]);
sesg := op(1, bb) - hh(mm);
for j to b do e := e + op(j, er)^2 od;
e := e/(b - m - 1);
for j from 2 to nops(bb) do v1 := v1 + op(j, bb)^2*
  Estima[Normal](Pcharl[mm], nn, j - 1)^2
od;
v := v1 + e;
vv := v - e*Dg;
resultado := table([Ecuacion = rhs(ec), var('e') = e,
  var(Tn) = v, ivar(Tn) = vv, sesgo = sesg]);
RETURN(eval(resultado))
elif op(0, c) = uniform then
  if type(h, list) then hh := unapply(op(1, h), theta)
  else hh := unapply(h, theta)
  fi;
mm := Estima[Maximo](l);
if nargs = 6 then
  for i to b do munifb[i] := op(i, args[6]) od
else for i to b do munifb[i] :=
  [stats[random, uniform[0, mm]](nn)]
  od
fi;
mboot := [seq(munifb[i], i = 1 .. b)];
ydata := Ydata(h, c, nn, b);
pdata := Pdata(uniform[mm], nn, b, m);
Xt := array([[seq(1, i = 1 .. b)], op(pdata)]);
X := linalg[transpose](Xt);
A := linalg[multiply](Xt, X);

```

```

IA := linalg[inverse](A);
Dg := sum(IA[j, j]*Estima[Norma](Pjac[mm], nn, j)^2,
  j = 1 .. m);
ec := stats[
  fit, leastsquare[[seq(x[i], i = 1 .. m), y]]](
  [op(pdata), ydata]);
bb := [op(1, rhs(ec)),
  seq(op(i, rhs(ec))/x[i - 1], i = 2 .. m + 1)];
ecc := unapply(rhs(ec), seq(x[j], j = 1 .. m));
yy := stats[transform, multiapply[ecc]]([op(pdata)]);
er :=
  stats[transform, multiapply[(a1, b1) -> a1 - b1]](
  [ydata, yy]);
sesg := op(1, bb) - hh(mm);
for j to b do e := e + op(j, er)^2 od;
e := e/(b - m - 1);
for j from 2 to nops(bb) do v1 := v1 +
  op(j, bb)^2*Estima[Norma](Pjac[mm], nn, j - 1)^2
od;
v := v1 + e;
vv := v - e*Dg;
resultado := table([Ecuacion = rhs(ec), var('e') = e,
  var(Tn) = v, ivar(Tn) = vv, sesgo = sesg]);
RETURN(eval(resultado))
fi
end)

```

```

MC = (proc(c, mo, l)
local i, j, m, _B, _e, vb, em, sn, _p, sb, h_;
  _B := nops(l);
  _p := unapply(op(2, c), x, n);
  if op(0, c) = uniform then
    h_ := unapply(op(1, c), theta);
    sn := Estima[Maximo](mo);
    for i to _B do
      m[i] := Estima[Maximo](op(i, l));
      _e[i] := _p(m[i], nops(mo))
    od

```

```

else
  h_ := unapply(op(1, c), mu);
  sn := stats[describe, mean](mo);
  for i to _B do
    m[i] := stats[describe, mean](op(i, 1));
    _e[i] := _p(m[i], nops(mo))
  od
fi;
em := sum(_e[j], j = 1 .. _B)/_B;
if op(3, c) = var then
  vb := evalf(sum((_e[j] - em)^2, j = 1 .. _B)/_B)
elif op(3, c) = bias then sb := evalf(em - h_(sn))
fi
end)
])

```

9.5.2 NPestima.

El código del paquete Npestim:

```

Npestim := table([

Boot = (proc(1, B)
local i, j, n, mues, ran;
global mboot;
  n := nops(1);
  mboot := [];
  for i to B do
    mues[i] := [];
    for j to n do
      ran := rand(1 .. n);
      mues[i] := [op(mues[i]), op(ran(), 1)]
    od;
    mboot := [op(mboot), mues[i]]
  od;
od;

```

```

    RETURN(op(mboot))
end)

Esp = (proc(h, l::list)
local t, i;
    t := unapply(h, x);
    evalf(sum(t(op(i, l)), i = 1 .. nops(l))/nops(l))
end)

Spo = (proc(s, l, k)
local i, a, c, r, p;
    if k = 0 then RETURN(1)
    elif k = 1 then RETURN(s - NPestima[Esp](x, l))
    else
        for i to 2 do a[i] :=
            NPestima[Esp](NPestima[Spo](x, l, k - i)^2, l)
        od;
        a[3] :=
            NPestima[Esp](x*NPestima[Spo](x, l, k - 1)^2, l);
        c := a[3]/a[1];
        r := a[1]/a[2];
        p := (s - c)*NPestima[Spo](s, l, k - 1)
            - r*NPestima[Spo](s, l, k - 2)
    fi;
    if type(s, numeric) then RETURN(evalf(p))
    else RETURN(sort(expand(p), s))
    fi
end)

Nspo = (proc(l, k)
    sqrt(NPestima[Esp](NPestima[Spo](x, l, k)^2, l))
end)

ECMedio = (proc(sb, si)
local j, e1, e2, n, md;
    n := nops(s);
    md := sum(si[j], j = 1 .. n)/n;
    e1 := sum((sb[j] - md)^2, j = 1 .. n)/n;

```

```

    e2 := sum((sb[j] - si[j])^2, j = 1 .. n)/n;
    RETURN([e1, e2])
end)

```

```

NPdata = (proc(l, n, B, m)
local i, j, b;
global Pn, Pdat, Pd;
  for i to m do
    for j to n do Pn[i, j] := [seq(
      NPestima[Spo](op(j, op(b, mboot))), l, i),
      b = 1 .. B)]
    od;
    Pd[i] := seq(Pn[i, j], j = 1 .. n)
  od;
  Pdat := [seq(Pd[i], i = 1 .. m)]
end)

```

```

NYdata = (proc(h, nn, B)
local i, b, Tn;
  Tn := unapply(h, seq(x[i], i = 1 .. nn));
  RETURN([seq(Tn(op(op(b, mboot))), b = 1 .. B)])
end)

```

```

NPaproximar = (proc(h, l::list, B::posint, m::posint)
local nn, N, t, s, ss, yy, T, v, v1, e, b, bb, i, j, mb, er,
sesg, Yd, X, Xt, A, IA, M, Y, Y1, m1, P1, P2, P3, q, q1, Pd,
Pn, k;
global mboot, resultado;
  v1 := 0;
  nn := nops(l);
  T := unapply(h, seq(x[i], i = 1 .. nn));
  i := 'i';
  if nargs = 5 then
    for b to B do mb[b] := op(b, args[5]) od
  else for b to B do mb[b] := NPestima[Boot](l, 1) od
  fi;
  b := 'b';
  mboot := [seq(mb[b], b = 1 .. B)];

```

```

b;
'b';
T := unapply(h, seq(x[i], i = 1 .. nn));
i := 'i';
Yd := [seq(T(op(op(b, mboot))), b = 1 .. B)];
b := 'b';
m1 := sum(l[i], i = 1 .. nn)/nn;
P1 := unapply(x - m1, x);
P3 := unapply(P1(x)*P1(y), x, y);
P2 := unapply(NPestima[Spo](x, 1, 2), x);
if m = 1 then
  for j to nn do
    q[j] := [seq(P1(mboot[b][j]), b = 1 .. B)];
    b := 'b'
  od;
  j := 'j';
  Pd := [seq(q[j], j = 1 .. nn)];
  j := 'j'
elif m = 2 then
  for j to nn do
    q[j] := [seq(P1(mboot[b][j]), b = 1 .. B)];
    b := 'b';
    q1[j] := [seq(P2(mboot[b][j]), b = 1 .. B)];
    b := 'b'
  od;
  j := 'j';
  Pd := [seq(q[j], j = 1 .. nn)];
  j := 'j';
  Pd := [op(Pd), seq(q1[j], j = 1 .. nn)];
  j := 'j'
else
  for j to nn do
    q[j] := [seq(P1(mboot[b][j]), b = 1 .. B)];
    b := 'b';
    q1[j] := [seq(P2(mboot[b][j]), b = 1 .. B)];
    b := 'b'
  od;
  j := 'j';

```

```

Pd := [seq(q[j], j = 1 .. nn)];
j := 'j';
Pd := [op(Pd), seq(q1[j], j = 1 .. nn)];
j := 'j';
for i to nn - 1 do for j from i + 1 to nn do
    Pn[i, j] := [seq(
        P3(mboot[b][i], mboot[b][j]), b = 1 .. B)
    ];
    b := 'b'
od
od;
Pd := [op(Pd), seq(seq(Pn[i, j], j = i + 1 .. nn),
    i = 1 .. nn - 1)]
fi;
i := 'i';
j := 'j';
Xt := array([[seq(1, i = 1 .. B)], op(Pd)]);
b := 'b';
X := linalg[transpose](Xt);
j := 'j';
i := 'i';
A := linalg[multiply](Xt, X);
IA := linalg[inverse](A);
M := linalg[multiply](IA, Xt);
Y := array(Yd);
bb := linalg[multiply](M, Y);
Y1 := linalg[multiply](X, bb);
er := linalg[add](Y, Y1, 1, -1);
if m = 1 or m = 2 then
    e := sum(er[i]^2, i = 1 .. B)/(B - nn*m - 1)
else e :=
    sum(er[i]^2, i = 1 .. B)/(B - 1/2*nn*(nn + 3) - 1)
fi;
j := 'j';
i := 'i';
if m = 1 then
    t[1] := sum(bb[i]^2, i = 2 .. nn + 1);
    N[1] := NPestima[Nspo](1, 1)^2;

```

```

v1 := v1 + t[1]*N[1];
v := v1 + e;
sesg := bb[1] - T(op(1))
elif m = 2 then
  t[1] := sum(bb[i]^2, i = 2 .. nn + 1);
  t[2] := sum(bb[i]^2, i = nn + 2 .. 2*nn + 1);
  N[1] := NPestima[Nspo](1, 1)^2;
  N[2] := NPestima[Nspo](1, 2)^2;
  v1 := v1 + t[1]*N[1] + t[2]*N[2];
  v := v1 + e;
  sesg := bb[1] - T(op(1))
else
  t[1] := sum(bb[i]^2, i = 2 .. nn + 1);
  t[2] := sum(bb[i]^2, i = nn + 2 .. 2*nn + 1);
  t[3] :=
    sum(bb[i]^2, i = 2*nn + 2 .. 1/2*nn*(nn + 3) + 1)
    ;
  N[1] := NPestima[Nspo](1, 1)^2;
  N[2] := NPestima[Nspo](1, 2)^2;
  N[3] := N[1]^2;
  for i to 3 do v1 := v1 + t[i]*N[i] od;
  i := 'i';
  v := v1 + e;
  sesg := bb[1] - T(op(1))
fi;
resultado := table(
  [ET = bb[1], var_e = e, var_Tn = v, sesgo = sesg]);
RETURN(eval(resultado))
end)

Vboot2d = (proc(h, l::list, B::posint, m::posint)
local nn, s, ss, yy, T, v, v1, e, b, bb, i, j, mb, er, sesg,
ydata, pdata, X, Xt, A, IA, M, Y, Y1, mx, my, Px, Py, qx, qy,
tx, ty;
global mboot, resultado;

v1 := 0;
nn := nops(l);

```

```

T := unapply(h, seq(x[i], i = 1 .. nn));
i := 'i';
if nargs = 5 then
  for b to B do mb[b] := op(b, args[5]) od
else for b to B do mb[b] := NPestima[Boot](1, 1) od
fi;
b := 'b';
mboot := [seq(mb[b], b = 1 .. B)];
mb := 'mb';
for b to B do
  mb[b, 1] := [seq(mboot[b][i][1], i = 1 .. nn)];
  i := 'i';
  mb[b, 2] := [seq(mboot[b][i][2], i = 1 .. nn)];
  i := 'i'
od;
mx := [seq(l[i][1], i = 1 .. nn)];
i := 'i';
my := [seq(l[i][2], i = 1 .. nn)];
i := 'i';
T := unapply(h, seq(x[i], i = 1 .. nn),
  seq(y[j], j = 1 .. nn));
i := 'i';
j := 'j';
ydata := [seq(T(op(mb[b, 1]), op(mb[b, 2])), b = 1 .. B)];
b := 'b';
pdata := [];
for i to m do
  Px[i] := unapply(NPestima[Spo](x, mx, i), x);
  Py[i] := unapply(NPestima[Spo](x, my, i), x);
  for j to nn do
    qx[i, j] := [seq(Px[i](mb[b, 1][j]), b = 1 .. B)]
    ;
    b := 'b';
    qy[i, j] := [seq(Py[i](mb[b, 2][j]), b = 1 .. B)]
    ;
    b := 'b'
  od;
  j := 'j';

```

```

        pdata := [op(pdata), seq(qx[i, j], j = 1 .. nn),
                  seq(qy[i, j], j = 1 .. nn)];
        j := 'j'
    od;
    i := 'i';
    Xt := array([[seq(1, i = 1 .. B)], op(pdata)]);
    X := linalg[transpose](Xt);
    j := 'j';
    i := 'i';
    A := linalg[multiply](Xt, X);
    IA := linalg[inverse](A);
    M := linalg[multiply](IA, Xt);
    Y := array(ydata);
    bb := linalg[multiply](M, Y);
    Y1 := linalg[multiply](X, bb);
    er := linalg[add](Y, Y1, 1, -1);
    e := sum(er[i]^2, i = 1 .. B)/(B - 2*nn*m - 1);
    j := 'j';
    i := 'i';
    for j to m do
        tx[j] := sum(bb[2*(j - 1)*nn + i]^2, i = 2 .. nn + 1)
        ;
        ty[j] := sum(bb[(2*j - 1)*nn + i]^2, i = 2 .. nn + 1)
    od;
    j := 'j';
    for i to m do v1 := v1 + tx[i]*NPestima[Nspo](mx, i)^2
        + ty[i]*NPestima[Nspo](my, i)^2
    od;
    v := v1 + e;
    sesg := bb[1] - T(op(mx), op(my));
    resultado := table(
        [ET = bb[1], var_e = e, var_Tn = v, sesgo = sesg]);
    RETURN(eval(resultado))
end)

MonteCarlo = (proc(T, mo, B)
local i, j, b, n, MT, T0, t, Tn, mb, mx, my, mbx, mby;
global Rs, Rv;

```

```

n := nops(mo);
if type(mo[1], list) then
  mx := [seq(mo[i][1], i = 1 .. n)];
  i := 'i';
  my := [seq(mo[i][2], i = 1 .. n)];
  i := 'i';
  Tn := unapply(T, seq(x[i], i = 1 .. n),
    seq(y[j], j = 1 .. n));
  i := 'i';
  j := 'j'
else Tn := unapply(T, seq(x[i], i = 1 .. n))
fi;
i := 'i';
if nargs = 4 then
  for i to B do mb[i] := op(i, args[4]) od
else for i to B do mb[i] := Boot(mo, 1) od
fi;
if type(mo[1], list) then
  for b to B do
    mbx[b] := [seq(mb[b][i][1], i = 1 .. n)];
    i := 'i';
    mby[b] := [seq(mb[b][i][2], i = 1 .. n)];
    i := 'i';
    t[b] := Tn(op(mbx[b]), op(mby[b]))
  od;
  b := 'b';
  T0 := Tn(op(mx), op(my))
else
  for i to B do t[i] := Tn(op(mb[i])) od;
  T0 := Tn(op(mo))
fi;
i := 'i';
MT := sum(t[i], i = 1 .. B)/B;
Rs := MT - T0;
Rv := sum((t[i] - MT)^2, i = 1 .. B)/(B - 1);
RETURN(Rs, Rv)
end)

```

```

Frecboot = (proc(mb, mo)
local i, j, b, n, fr, cont, B;
  n := nops(mo);
  if nargs = 3 then
    if type(mb, listlist) then
      B := nops(mb);
      for b to B do for j to n - 1 do
        cont := 0;
        for i to n do
          if op(i, mb[b]) = mo[j] then
            cont := cont + 1
          fi
        od;
        fr[b, j] := cont
      od;
      j := 'j';
      b := 'b';
      RETURN([seq([seq(fr[b, j], j = 1 .. n - 1)],
        b = 1 .. B)])
    else
      for j to n - 1 do
        cont := 0;
        for i to n do
          if mb[i] = mo[j] then cont := cont + 1 fi
        od;
        fr[j] := cont
      od;
      RETURN([seq(fr[j], j = 1 .. n - 1)])
    fi
  else
    if type(mb, listlist) then
      B := nops(mb);
      for b to B do for j to n do
        cont := 0;
        for i to n do
          if op(i, mb[b]) = mo[j] then
            cont := cont + 1
          fi
        od;
        fr[b, j] := cont
      od;
      RETURN([seq(fr[b, j], b = 1 .. B, j = 1 .. n)])
    else
      for j to n do
        cont := 0;
        for i to n do
          if mb[i] = mo[j] then cont := cont + 1 fi
        od;
        fr[j] := cont
      od;
      RETURN([seq(fr[j], j = 1 .. n)])
    fi
  fi
end

```

```

                fi
            od;
            fr[b, j] := cont
        od
    od;
    j := 'j';
    b := 'b';
    RETURN([
        seq([seq(fr[b, j], j = 1 .. n)], b = 1 .. B)])
else
    for j to n do
        cont := 0;
        for i to n do
            if mb[i] = mo[j] then cont := cont + 1 fi
        od;
        fr[j] := cont
    od;
    RETURN([seq(fr[j], j = 1 .. n)])
fi
fi
end)

```

```

PolMultinom = (proc(n, s)
local i, j, k, u, y, u0, m0, V, IA, A, uu0, f0, f, x, xx, m,
mm, g0, yy, uu, g, p1, p, q1, q, t1, t;
global Pm, Qm;
    V := array(1 .. n - 1, 1 .. n - 1, []);
    for i to n - 1 do
        for j to n - 1 do
            if i = j then V[i, j] := 1 - 1/n
            else V[i, j] := - 1/n
            fi
        od;
        j := 'j'
    od;
    i := 'i';
    m0 := array(1 .. n - 1, [seq(1, i = 1 .. n - 1)]);
    i := 'i';

```

```

with(linalg);
IA := cholesky(V);
A := inverse(IA);
u0 := multiply(A, m0);
uu0 := seq(u0[i], i = 1 .. n - 1);
i := 'i';
f0 := product(m[i]^x[i], i = 1 .. n - 1)*
      (n - sum(m[j], j = 1 .. n - 1))^
      (n - sum(x[k], k = 1 .. n - 1));
f := unapply(f0, seq(x[i], i = 1 .. n - 1),
             seq(m[j], j = 1 .. n - 1));
i := 'i';
j := 'j';
y := array(1 .. n - 1, []);
u := array(1 .. n - 1, []);
x := multiply(IA, y);
xx := seq(eval(x[i]), i = 1 .. n - 1);
m := multiply(IA, u);
mm := seq(eval(m[j]), j = 1 .. n - 1);
i := 'i';
j := 'j';
g0 := f(xx, mm);
yy := seq(y[i], i = 1 .. n - 1);
uu := seq(u[j], j = 1 .. n - 1);
i := 'i';
j := 'j';
g := unapply(g0, yy, uu);
if s = 1 then
  for i to n - 1 do Pm[1, i] :=
    unapply(z[i] - u0[i], seq(z[j], j = 1 .. n - 1))
  od;
  i := 'i';
  j := 'j'
elif s = 2 then
  for i to n - 1 do
    p1 := simplify(diff(g(yy, uu), u[i])/g(yy, uu));
    p[1, i] := unapply(p1, yy, uu);
    Pm[1, i] :=

```

```

        unapply(simplify(p[1, i](yy, uu0)), yy);
q1 := diff(p[1, i](yy, uu), u[i]);
q := unapply(q1, yy, uu);
Pm[2, i] := unapply(expand(
    simplify(q(yy, uu0)) + Pm[1, i](yy, uu0)^2),
    yy)
od;
i := 'i'
else
for i to n - 1 do
    p1 := simplify(diff(g(yy, uu), u[i])/g(yy, uu));
    p[1, i] := unapply(p1, yy, uu);
    Pm[1, i] :=
        unapply(simplify(p[1, i](yy, uu0)), yy);
    q1 := diff(p[1, i](yy, uu), u[i]);
    q := unapply(q1, yy, uu);
    Pm[2, i] := unapply(expand(
        simplify(q(yy, uu0)) + Pm[1, i](yy, uu0)^2),
        yy)
od;
i := 'i';
for i to n - 2 do
    for j from i + 1 to n - 1 do
        t1 := diff(p[1, i](yy, uu), u[j]);
        t := unapply(t1, yy, uu);
        Qm[i, j] := unapply(expand(
            simplify(t(yy, uu0))
            + Pm[1, i](yy, uu0)*Pm[1, j](yy, uu0)),
            yy)
    od;
    j := 'j'
od;
i := 'i'
fi
end)

Multinomaprox = (proc(h, l::list, B::posint, m::posint, d::posint)
local nn, t, s, ss, yy, T, v, v1, e, b, bb, i, j, mb, er,

```

```

sesg, ydata, pdata, X, Xt, A, IA, M, Y, Y1, mx, my, mb1, mb2,
AA, mbt, p, P, L0, L, L2, L3, v2, v3;
global mboot, Mboot, resultado;
  v1 := 0;
  nn := nops(1);
  if type(l[1], list) then
    mx := [seq(l[i][1], i = 1 .. nn)];
    i := 'i';
    my := [seq(l[i][2], i = 1 .. nn)];
    i := 'i';
    T := unapply(h, seq(x[i], i = 1 .. nn),
      seq(y[j], j = 1 .. nn));
    i := 'i';
    j := 'j'
  else T := unapply(h, seq(x[i], i = 1 .. nn)); i := 'i'
  fi;
  if nargs = 6 then
    for b to B do mb[b] := op(b, args[6]) od
  else for b to B do mb[b] := NPestima[Boot](1, 1) od
  fi;
  mboot := [seq(mb[b], b = 1 .. B)];
  b := 'b';
  if type(l[1], list) then
    for b to B do
      mb1[b] := [seq(mb[b][i][1], i = 1 .. nn)];
      i := 'i';
      mb2[b] := [seq(mb[b][i][2], i = 1 .. nn)];
      i := 'i'

    od;
    b := 'b'
  fi;
  Mboot := NPestima[Frecboot](mboot, 1, -1);
  if d = 10 then AA := array([
    [1/3*sqrt(10), 0, 0, 0, 0, 0, 0, 0, 0],
    [1/12*sqrt(2), 3/4*sqrt(2), 0, 0, 0, 0, 0, 0, 0], [
    1/28*sqrt(14), 1/28*sqrt(14), 2/7*sqrt(14), 0, 0, 0,
    0, 0, 0], [1/42*sqrt(42), 1/42*sqrt(42),

```

```

1/42*sqrt(42), 1/6*sqrt(42), 0, 0, 0, 0, 0], [
1/30*sqrt(30), 1/30*sqrt(30), 1/30*sqrt(30),
1/30*sqrt(30), 1/5*sqrt(30), 0, 0, 0, 0], [
1/10*sqrt(5), 1/10*sqrt(5), 1/10*sqrt(5),
1/10*sqrt(5), 1/10*sqrt(5), 1/2*sqrt(5), 0, 0, 0], [
1/6*sqrt(3), 1/6*sqrt(3), 1/6*sqrt(3), 1/6*sqrt(3),
1/6*sqrt(3), 1/6*sqrt(3), 2/3*sqrt(3), 0, 0], [
1/6*sqrt(6), 1/6*sqrt(6), 1/6*sqrt(6), 1/6*sqrt(6),
1/6*sqrt(6), 1/6*sqrt(6), 1/6*sqrt(6), 1/2*sqrt(6), 0
], [1/2*sqrt(2), 1/2*sqrt(2), 1/2*sqrt(2),
1/2*sqrt(2), 1/2*sqrt(2), 1/2*sqrt(2), 1/2*sqrt(2),
1/2*sqrt(2), sqrt(2)]]
fi;
if d = 8 then AA := array([
2/7*sqrt(14), 0, 0, 0, 0, 0, 0, 0], [
1/42*sqrt(42), 1/6*sqrt(42), 0, 0, 0, 0, 0, 0], [
1/30*sqrt(30), 1/30*sqrt(30), 1/5*sqrt(30), 0, 0, 0, 0, 0], [
1/10*sqrt(5), 1/10*sqrt(5), 1/10*sqrt(5),
1/2*sqrt(5), 0, 0, 0], [1/6*sqrt(3), 1/6*sqrt(3),
1/6*sqrt(3), 1/6*sqrt(3), 2/3*sqrt(3), 0, 0], [
1/6*sqrt(6), 1/6*sqrt(6), 1/6*sqrt(6), 1/6*sqrt(6),
1/6*sqrt(6), 1/2*sqrt(6), 0], [1/2*sqrt(2),
1/2*sqrt(2), 1/2*sqrt(2), 1/2*sqrt(2),
1/2*sqrt(2), sqrt(2)]]
fi;
mbt := [];
for b to B do
p := array(Mboot[b]);
P := linalg[multiply](AA, p);
mbt := [op(mbt), eval([seq(P[i], i = 1 .. nn - 1)])];
i := 'i';
p := 'p';
P := 'P'
od;
b := 'b';
if type(l[1], list) then
ydata := [seq(T(op(mb1[b]), op(mb2[b])), b = 1 .. B)]
;

```

```

    b := 'b'
else ydata := [seq(T(op(mb[b])), b = 1 .. B)]; b := 'b'
fi;
LO := [seq(1, b = 1 .. B)];
b := 'b';
if m = 1 then
    if d = 10 then read 'spomult1.m' fi;
    if d = 8 then read 'spomn8_1.m' fi;
    for i to nn - 1 do
        L[i] := [seq(Pm[1, i](op(mbt[b])), b = 1 .. B)];
        b := 'b'
    od;
    i := 'i';
    pdata := [seq(L[i], i = 1 .. nn - 1)];
    i := 'i'
elif m = 2 then
    if d = 10 then read 'spomult2.m' fi;
    if d = 8 then read 'spomn8_2.m' fi;
    for i to nn - 1 do
        L[i] := [
            seq(evalf(Pm[1, i](op(mbt[b]))), b = 1 .. B)
        ];
        b := 'b';
        L2[i] := [
            seq(evalf(Pm[2, i](op(mbt[b]))), b = 1 .. B)
        ];
        b := 'b'
    od;
    i := 'i';
    pdata := [seq(L[i], i = 1 .. nn - 1)];
    i := 'i';
    pdata := [op(pdata), seq(L2[i], i = 1 .. nn - 1)];
    i := 'i'
elif m = 3 then
    if d = 10 then read 'spomult3.m' fi;
    if d = 8 then read 'spomn8_3.m' fi;
    for i to nn - 1 do
        L[i] := [

```

```

        seq(evalf(Pm[1, i](op(mbt[b]))), b = 1 .. B]
    ;
    b := 'b';
    L2[i] := [
        seq(evalf(Pm[2, i](op(mbt[b]))), b = 1 .. B]
    ;
    b := 'b'
od;
i := 'i';
pdata := [seq(L[i], i = 1 .. nn - 1)];
i := 'i';
pdata := [op(pdata), seq(L2[i], i = 1 .. nn - 1)];
i := 'i';
for i to nn - 2 do
    for j from i + 1 to nn - 1 do
        L3[i, j] := [seq(evalf(Qm[i, j](op(mbt[b]))),
            b = 1 .. B)];
        b := 'b'
    od;
    j := 'j'
od;
i := 'i';
pdata := [op(pdata), seq(
    seq(L3[i, j], j = i + 1 .. nn - 1),
    i = 1 .. nn - 2)];
i := 'i';
j := 'j'
fi;
Xt := array([L0, op(pdata)]);
X := linalg[transpose](Xt);
j := 'j';
i := 'i';
A := linalg[multiply](Xt, X);
IA := linalg[inverse](A);
M := linalg[multiply](IA, Xt);
Y := array(ydata);
bb := linalg[multiply](M, Y);
Y1 := linalg[multiply](X, bb);

```

```

er := linalg[add](Y, Y1, 1, -1);
if m = 1 or m = 2 then
    e := sum(er[i]^2, i = 1 .. B)/(B - (nn - 1)*m - 1)
elif m = 3 then
    e := sum(er[i]^2, i = 1 .. B)/(B - 1/2*nn*(nn + 1))
fi;
j := 'j';
i := 'i';
if m = 1 then
    v1 := sum(bb[i]^2, i = 2 .. nn); v := v1 + e
elif m = 2 then
    v1 := sum(bb[i]^2, i = 2 .. nn);
    v2 :=
        2*(1 - 1/nn)*sum(bb[i]^2, i = nn + 1 .. 2*nn - 1)
    ;
    v := v1 + v2 + e
elif m = 3 then
    v1 := sum(bb[i]^2, i = 2 .. nn);
    v2 :=
        2*(1 - 1/nn)*sum(bb[i]^2, i = nn + 1 .. 2*nn - 1)
    ;
    v3 := (1 - 1/nn)*
        sum(bb[i]^2, i = 2*nn .. 1/2*nn*(nn + 1));
    v := v1 + v2 + v3 + e
fi;
if type(l[1], list) then
    sesg := bb[1] - T(op(mx), op(my))
else sesg := bb[1] - T(op(l))
fi;
resultado := table(
    [ET = bb[1], var_e = e, var_Tn = v, sesgo = sesg]);
RETURN(eval(resultado))
end)
]:

```

9.5.3 Repita.

El código del paquete Repita:

```

Repita := table([

Recortar = (proc(h, c, l::list, b::integer, m::integer, q::integer)
  local h1, T, va, iva, ecua, ses, j, tabla1;
  global tabla;
  h1 := h;
  if not type(h, list) then
    T := op(1, Estima[Umvue](h, c)); h1 := [h, T]
  fi;
  if nargs = 7 then
    tabla1 := Estima[Aproximar](h1, c, l, b, m, args[7])
  else tabla1 := Estima[Aproximar](h1, c, l, b, m)
  fi;
  va[1] := tabla1[var(Tn)];
  iva[1] := tabla1[ivar(Tn)];
  ses[1] := tabla1[sesgo];
  for j to q do
    tabla1 :=
      Estima[Aproximar](h1, c, l, b, m + j, mboot);
    va[j + 1] := tabla1[var(Tn)];
    iva[j + 1] := tabla1[ivar(Tn)];
    ses[j + 1] := tabla1[sesgo]
  od;
  tabla := array(1 .. q + 2, 1 .. 4, [(1, 1) = 'M',
    (1, 2) = 'var[B]', (1, 3) = 'ivar[B]',
    (1, 4) = 'sesg[B]']);
  for j from 2 to q + 2 do
    tabla[j, 1] := m + j - 2;
    tabla[j, 2] := va[j - 1];
    tabla[j, 3] := iva[j - 1];
    tabla[j, 4] := ses[j - 1]
  od;
  RETURN(eval(tabla))
end)

```

```

Resimular = (proc(h, c, l::list, b_::integer, m::integer, q::integer, s::int
local t, T, h1, mb, va, iva, ecua, ses, j_;
global tabla, Mboot;
  h1 := h;
  if not type(h, list) then
    T := op(1, Estima[Umvue](h, c)); h1 := [h, T]
  fi;
  if nargs = 8 then
    t := Estima[Aproximar](h1, c, l, b_, m, args[8])
  else
    t := Estima[Aproximar](h1, c, l, b_, m);
    Mboot := mboot
  fi;
  va[q + 1] := t[var(Tn)];
  iva[q + 1] := t[ivar(Tn)];
  ses[q + 1] := t[sesgo];
  for j_ to q do
    mb := Estima[Suprimir](s, mboot);
    t := Estima[Aproximar](h1, c, l, b_ - j_*s, m, mb);
    va[q + 1 - j_] := t[var(Tn)];
    iva[q + 1 - j_] := t[ivar(Tn)];
    ses[q + 1 - j_] := t[sesgo];
    mb := 'mb'
  od;
  tabla := array(1 .. q + 2, 1 .. 4, [(1, 1) = 'B',
    (1, 2) = 'var[B]', (1, 3) = 'ivar[B]',
    (1, 4) = 'sesg[B]']);
  for j_ from 2 to q + 2 do
    tabla[j_, 1] := b_ - (q - j_ + 2)*s;
    tabla[j_, 2] := va[j_ - 1];
    tabla[j_, 3] := iva[j_ - 1];
    tabla[j_, 4] := ses[j_ - 1]
  od;
  RETURN(eval(tabla))
end)

IdealRep = (proc(T, Mo, B)

```

```

local i, n, N, sgi, vri;
global SGI, VRI;
  N := nops(Mo);
  n := nops(Mo[1]);
  for i to N do
    Repita[MC](T, Mo, B); sgi[i] := Rs; vri[i] := Rv
  od;
SGI := [seq(sgi[i], i = 1 .. N)];
i := 'i';
VRI := [seq(vri[i], i = 1 .. N)];
i := 'i';
RETURN([SGI, VRI])
end)

Repetir = (proc(T, Mo, M, B, m, c)
local i, j, n, N, Tn, rmc_s, rmc_v, rbo_s, rbo_v;
global RMC_S, RMC_V, RBO_S, RBO_V;
  n := nops(Mo[1]);
  N := nops(Mo);
  if nops(Mo[1][1]) = 2 then
    Tn := unapply(T, seq(x[i], i = 1 .. n),
      seq(y[j], j = 1 .. n));
    i := 'i';
    j := 'j'
  else Tn := unapply(T, seq(x[i], i = 1 .. n)); i := 'i'
  fi;
  for i to N do for j to M do
    if nargs = 7 then NPestima[c](T, Mo[i], B, m,
      op(j, op(i, args[7])))
    else NPestima[c](T, Mo[i], B, m)
    fi;
    rbo_s[i, j] := resultado[sesgo];
    rbo_v[i, j] := resultado[var_Tn];
    NPestima[MonteCarlo](T, Mo[i], B, mboot);
    rmc_s[i, j] := Rs;
    rmc_v[i, j] := Rv
  od
od;

```

```

RMC_S :=
  [seq([seq(rmc_s[i, j], j = 1 .. M)], i = 1 .. N)];
RMC_V :=
  [seq([seq(rmc_v[i, j], j = 1 .. M)], i = 1 .. N)];
RBO_S :=
  [seq([seq(rbo_s[i, j], j = 1 .. M)], i = 1 .. N)];
RBO_V :=
  [seq([seq(rbo_v[i, j], j = 1 .. M)], i = 1 .. N)];
RETURN([RMC_S, RMC_V], [RBO_S, RBO_V])
end)

```

```

PMCrepeter = (proc(c, mo, l, q, sal)
local i, B, tb;
  tb := array(1 .. q + 2, 1 .. 2, []);
  tb[1, 1] := 'B';
  if op(3, c) = var then tb[1, 2] := 'var[MC]'
  elif op(3, c) = bias then tb[1, 2] := 'bias[MC]'
  fi;
  B := nops(1);
  tb[q + 2, 1] := B;
  tb[q + 2, 2] := Estima[MC](c, mo, 1);
  for i to q do
    tb[q + 2 - i, 1] := B - sal*i;
    tb[q + 2 - i, 2] :=
      Estima[MC](c, mo, Estima[Suprimir](sal*i, 1))
  od;
  RETURN(print(tb))
end)

```

```

]):

```

9.5.4 NEF

El Código del paquete NEF:

```

NEF := table([

```

```

Cumulante = proc(k,c)
  local N,p,m,f,g,s,i;
  if op(0,c) = normal then if k = 1 then op(1,c) elif k = 2
    then op(2,c)^2 else 0 fi
  elif op(0,c) = gamma then (k-1)!*op(1,c)*op(2,c)^k
  elif op(0,c) = poisson then op(1,c)
  elif op(0,c) = binomial then
    N := op(1,c);
    p := op(2,c);
    m := N*p;
    f := unapply(x-x^2/N,x);
    if k = 1 then eval(m)
    elif k = 2 then eval(m-m^2/N)
    elif k = 3 then
      g := unapply(f(x)*diff(f(x),x),x);
      g := collect(simplify(expand(g(x))),x);
      subs(x = N*p,g(x))
    else
      g := unapply(f(x)*diff(f(x),x),x);
      for i from 4 to k do g:=unapply(f(x)*diff(g(x),x),x)
      od:
      s := collect(simplify(expand(g(x))),x);
      subs(x = N*p,s)
    fi
  elif op(0,c) = binomialnegativa then
    N := op(1,c);
    p := op(2,c);
    m := N*(1-p)/p;
    f := unapply(x+x^2/N,x);
    if k = 1 then eval(m)
    elif k = 2 then eval(m+m^2/N)
    elif k = 3 then
      g := unapply(f(x)*diff(f(x),x),x);
      g := collect(simplify(expand(g(x))),x);
      subs(x = m,g(x))
    else
      g := unapply(f(x)*diff(f(x),x),x);
      for i from 4 to k do g:=unapply(f(x)*diff(g(x),x),x)

```

```

        od;
        s := collect(simplify(expand(g(x))),x);
        subs(x = m,s)
    fi
elif op(0,c) = shg then
    N := op(1,c);
    p := op(2,c);
    m := N*tan(p);
    f := unapply(N+x^2/N,x);
    if k = 1 then eval(m)
    elif k = 2 then eval(N+m^2/N)
    elif k = 3 then
        g := unapply(f(x)*diff(f(x),x),x);
        g := collect(simplify(expand(g(x))),x);
        subs(x = m,g(x))
    else
        g := unapply(f(x)*diff(f(x),x),x);
        for i from 4 to k do g := unapply(f(x)*diff(g(x),x),x)
        od;
        s := collect(simplify(expand(g(x))),x);
        subs(x = m,s)
    fi
fi
end

MomentoCentrado = proc(k,c)
    local i,j,S,s,f,g,a,b,m;
    s := 0;
    if op(0,c) = normal then
        if k = 1 or k = 3 then 0
        elif k = 2 then op(2,c)^2
        else (k-1)*op(2,c)^2*NEF[MomentoCentrado](k-2,c)
        fi
    elif op(0,c) = gamma then
        a := op(1,c);
        b := op(2,c);
        if k = 1 then 0
        elif k = 2 then a*b^2

```

```

elif k = 3 then 2*a*b^3
else
  for i from 2 to k-2 do
    S[i]:=NEF[MomentoCentrado](i,c)
  od;
  (k-1)!*(a*b^k+sum(b^(k-j)*S[j]/j!,j = 2 .. k-2))
fi:
elif op(0,c) = poisson then
  a := op(1,c);
  if k = 1 then 0
  elif k = 2 or k = 3 then a
  else
    for i from 2 to k-1 do
      S[i]:=NEF[MomentoCentrado](i,c)
    od;
    collect(simplify(a*(1+sum((k-1)!*S[j]/j!/(k-j-1)!,
      j=2.. k-2))),a)
  fi:
elif op(0,c) = binomial then
  a := op(1,c);
  b := op(2,c);
  m := a*b;
  for j from 0 to k do
    S[j] := eval(NEF[Momento](j,c));
    s[j] := eval(k!*(-m)^(k-j)/j!/(k-j)!)
  od;
  simplify(sum(simplify(s[i]*S[i]),i = 0 .. k))
elif op(0,c) = binomialnegativa then
  a := op(1,c);
  b := op(2,c);
  m := a*(b-1)/b;
  for j from 0 to k do
    S[j] := eval(NEF[Momento](j,c));
    s[j] := eval(k!*(-m)^(k-j)/j!/(k-j)!)
  od;
  simplify(sum(simplify(s[i]*S[i]),i = 0 .. k))
elif op(0,c) = shg then
  a := op(1,c);

```

```

    b := op(2,c);
    m := a*tan(b);
    for j from 0 to k do
        S[j] := eval(NEF[Momento](j,c));
        s[j] := eval(k!*(-m)^(k-j)/j!/(k-j)!)
    od;
    simplify(sum(simplify(s[i]*S[i]),i = 0 .. k))
fi
end

Momento = proc(kk,c)
local k,f,g,m,s,p,N;
k := kk;
if op(0,c) = normal then
    if k = 0 then 1
    elif k = 1 then op(1,c)
    else
        m := op(1,c);
        s := op(2,c);
        collect(m*NEF[Momento](k-1,c)+
            (k-1)*s^2*NEF[Momento](k-2,c),m)
    fi:
elif op(0,c) = gamma then m := op(1,c); s := op(2,c);
    simplify(s^k*GAMMA(m+k)/GAMMA(m))
elif op(0,c) = poisson then
    if k = 0 then 1
    else
        f := (t,mu) -> exp(mu*(exp(t)-1));
        g := unapply(diff(f(t,mu),t $ k),t);
        subs(mu = op(1,c),g(0))
    fi
elif op(0,c) = binomial then
    if k = 0 then 1
    else
        f := unapply((op(2,c)*exp(t)-op(2,c)+1)^op(1,c),t);
        g := unapply(diff(f(t),t $ k),t);
        simplify(g(0))
    fi
fi

```

```

    elif op(0,c) = binomialnegativa then
      if k = 0 then 1
      else
        f := unapply((op(2,c)/(1-(1-
          op(2,c))*exp(t)))^op(1,c),t);
        g := unapply(diff(f(t),t $ k),t);
        expand(simplify(g(0)))
      fi
    elif op(0,c) = shg then
      if k = 0 then 1
      else
        f := unapply(1/(cos(t)-tan(op(2,c))*sin(t))^op(1,c),t);
        g := unapply(diff(f(t),t $ k),t);
        RETURN(simplify(g(0)))
      fi
    fi
  end

```

```

Resumen = (proc(c)
  if c = normal then table([Normal = N(mu, sigma^2),
    'Densidad '*f(x, mu) =
    exp(- 1/2*(x - mu)^2/sigma^2)/(sqrt(2*Pi)*sigma),
    'Funcion generatriz de cumulantes '*psi[mu](t) =
    t*mu + 1/2*t^2*sigma^2, Media = mu,
    'Funcion de Varianza'*V(mu) = sigma^2, Cumulantes =
    (C[1] = theta, C[2] = sigma^2, C[k] = 0, 2 <= k),
    Momentos = (mu[0] = 1, mu[1] = mu,
    mu[k] = mu[k - 1]*mu + (k - 1)*sigma^2*mu[k - 2],
    2 <= k), 'Momentos centrados' =
    (m[2*k + 1] = 0, m[2*k] = '(2k-1)!*U{a1}*sigma^(2*k)),
    'Estadistico suficiente y completo' =
    'media muestral'*x[n],
    'Densidad del estadistico s. y c.'*f[n](s, mu) =
    N(mu, sigma^2/n),
    'umvue de la densidad' = N(x[n], (1 - 1/n)*sigma^2)])
  elif c = gamma then table(['Gamma' = G(alpha, theta),
    'Densidad '*f(x, theta) = (
    x^(alpha - 1)*exp(- x/theta)/(

```

```

theta^alpha*Gamma(alpha)), 'con x>0'),
'Funcion generatriz de cumulantes '*psi[theta](t) =
- alpha*log(1 - theta*t),
Cumulantes = (C[k] = (k - 1)!*alpha*theta^k, ' k>0'),
Momentos = (
mu[k] = theta^k*Gamma(alpha + k)/Gamma(alpha), 0 <= k
), 'Momentos centrados' = (m[k] = theta^k*Sum(
Gamma(alpha + k)*(-alpha)^(k - j)/Gamma(alpha),
j = 0 .. k), 1 <= k),
'Estadistico suficiente y completo' =
'media muestral'*x[n],
'Densidad del estadistico s. y c. '*f[n](s, theta) =
G(n*alpha, theta/n)])
elif c = poisson then table([Poisson = P(mu),
'Densidad '*f(x, mu) = exp(-mu)*mu^x/x!,
'Funcion generatriz de cumulantes '*psi[mu](t) =
mu*(exp(t) - 1), Cumulantes = (C[k] = mu, 1 <= k),
Media = mu, 'Funcion de Varianza '*V(mu) = mu,
Momentos = (mu[0] = 1, mu[k] = mu*
Sum((k - 1)!*mu[i]/((k - i - 1)!*i!), i = 0 .. k - 1)
, 1 <= k), 'Momentos centrados' = (m[1] = 0,
m[2] = mu, m[3] = mu, mu[k] = mu*(1 +
Sum((k - 1)!*m[i]/((k - i - 1)!*i!), i = 2 .. k - 2))
, 4 <= k), 'Estadistico suficiente y completo' =
'media muestral'*x[n],
'Densidad del estadistico s. y c. '*f[n](s, mu) = (
exp(- n*mu)*(n*mu)^(n*s)/(n*s)!, 's={0,1/n,2/n,...}')
, 'umvue de la densidad' = B(S[n], 1/n)])
elif c = binomial then tabe([Binomial = B(N, p),
'Densidad '*f(x, p) = N!*p^x*q^(N - x)/((N - x)!*x!),
'Funcion generatriz de cumulantes '*psi[p](t) =
N*log(p*exp(t) + q), 'Media '*mu = Np,
'Funcion de varianza '*V(mu) = Npq*='*mu - mu^2/N,
Cumulantes = (C[1] = mu, C[2] = V(mu),
C[k + 1](mu) = V(mu)*Diff(C[k](mu), mu), 2 <= k),
'Estadistico suficiente y completo' =
'media muestral'*x[n],
'Densidad del estadistico s. y c. '*f[n](s, p) = (

```

```

nN!*p^ns*q^(n*(N - s))/(ns!*(nN - ns!)),
' s={0,1/n,2/n,...}' ]])
elif c = bnegativa then table([
'Binomial Negativa' = BN(r, p), 'Densidad '*f(x, p)
= (
(x + r - 1)!/((r - 1)!*x!) + p^r*q^x, ' x={0,1,...}' )
, 'Funcion generatriz de cumulantes '*psi[p](t) =
(r*log(p) - r*log(1 - q*exp(t))), t < -log(q)),
'Media '*mu = rq/p, 'Funcion de varianza '*V(mu) =
rq*'='*mu/p^2 + 1/2*mu^2, Cumulantes = (C[1] = mu,
C[2] = V(mu), C[k + 1](mu) = V(mu)*Diff(C[k](mu), mu),
2 <= k), 'Estadistico suficiente y completo' =
'media muestral'*x[n],
'Densidad del estadistico s. y c. '*f[n](s, p) = (
(nr + ns - 1)!*p^nr*q^ns/((nr - 1)!*ns!),
' s={0,1/n,2/n,...}' ]])
elif c = shg then table([
'Secante Hiperbolica Generalizada' = (GHS(r, theta),
abs(theta) < 1/2*Pi, rho*'='*tan(theta), 0 < r),
'Densidad '*f(x, r, rho) = (
(1 + rho^2)^(- 1/2*r)*exp(x*arctan(p))*f(x, r, 0),
' con ', f(x, r, 0) = 2^(r - 2)*
Product(1/(1 + x^2/(r + 2*j)^2), j = 0 .. infinity)/
Gamma(r)), 'Funcion generatriz de cumulantes' =
- r*log(cos(t) - rho*sin(t)), 'media '*mu = r*rho,
'Varianza '*V(mu) = r^2*(1 + rho^2)*'=' + mu^2/r,
'estadistico suficiente y completo' =
'media muestral'*x[n], 'Densidad de '*x[n] = n*
(1 + rho^2)^(- 1/2*nr)*exp(ns*arctan(rho))*
f(ns, nr, 0)]])
fi
end)
]:

```

9.5.5 FFTestima.

El Codigo del paquete FFTestima:

```
FFTestima := TABLE([
```

```

DensidadPuntos = proc (mo, m, T)
  local md, P, s, n, j, k, C, F, f, Dx, Df, w, w0;
  global Result; n := nops(mo);
  md := sum(mo[j], j = 1 .. n)/n;
  s := sqrt(sum((mo[j]-md)^2, j = 1 .. n)/n);
  w := 2*Pi*T/m/s/sqrt(n);
  w0 := exp(Pi/m*I);
  if nargs = 4 then
    P := unapply(args[4], t);
    for k from -m to m do C[k] := P(2*Pi*k*T/m) end do;
    k := 'k';
  else
    for k from -m to m do
      C[k] := (sum(exp(k*w*(mo[j]-md)*I), j = 1 .. n)/n)^n;
    end do;
    k := 'k';
  end if;
  C[-m] := 1/2*C[-m]+1/2*C[m];
  for j from -m to m-1 do
    F[j] := sum(C[k]*w0^(-j*k), k = -m .. m-1);
    f[j] := T*F[j]/m;
  end do; j := 'j';
  Df := [seq(Re(f[j]), j = -m .. m-1)]; j := 'j';
  Dx := [seq(.5000000000*j/T, j = -m .. m-1)]; j := 'j';
  Result := evalf([Dx, Df]);
RETURN(Result)
end proc

```

```
DendidadEstima = proc (x, mPuntos)
```

```

  local g, j, xj, fj;
  xj := mPuntos[1];
  fj := mPuntos[2];
  if x < xj[1] or xj[2*m] < x then g := 0 else
    for j to 2*m do
      if xj[j] <= x and x < xj[j+1] then break end if;

```

```

        end do;
        g := (2*T*x-j+m+1)*(fj[j+1]-fj[j])+fj[j]
    end if;
    RETURN(evalf(g))
end proc

```

```

DistribucionEstima = proc (x, mPuntos)
    local g, g0, g1, g2, j, k, xj, fj;
    xj := mPuntos[1];
    fj := mPuntos[2];
    if x < xj[1] then g := 0:
    elif xj[2*m] < x then g := 1:
    else
        for j to 2*m do
            if xj[j] <= x and x < xj[j+1] then break end if:
        end do;
        g1 := sum(fj[k]+fj[k+1],k = 1 .. j-1);
        g2 := (x-1/2*(j-m-1)/T)*((2*T*x-j+m+1)*(fj[j+1]-fj[j])+2*fj[j]);
        g := 1/4*g1/T+1/2*g2:
    end if;
    RETURN(evalf(g))
end proc

```

```

InversionDirectDistrib = proc (mo, x, a, Fa, m, T)
    local md, S, n, i, j, k, C, Ca, Rx, r, r0, fx, w0;
    global Fx; n := nops(mo);
    md := sum(mo[i],i = 1 .. n)/n;
    S := sum((mo[i]-md)^2,i = 1 .. n)/n;
    w0 := exp(2*I*Pi*T/m);
    r0 := 2*Pi*T/m;
    r := r0/sqrt(n*S)*I;
    for k from -m to m do
        if k <> 0 then
            C[k] := -I*(w0^(-a*k)-w0^(-x*k))/r0/k;
            Ca[k] := (sum(exp(r*k*(mo[j]-md)),j = 1 .. n)/n)^n
        else
            C[0] := x-a; Ca[0] := 1:
        end if;
    end for;
end proc

```

```
        end if:
        end do;
        k := 'k';
        Rx := 1/2*T*(C[-m]*Ca[-m]+C[m]*Ca[m])/m+Fa;
        fx := sum(C[k]*Ca[k],k = -m+1 .. m-1);
        Fx := Re(evalf(T/m*fx+Rx));
        RETURN(Fx)
    end proc
]:
```

Referencias.

1. Abbey, J. L. and David, H. T. , (1970). The construction of uniform minimum variance unbiased estimators for exponential distributions. *Annals of Mathematical Statistics* **41** (4), 1217-1226.
2. Akhiezer, N. I. (1965). The classical moments problem (1st Ed.), Oliver and Boyd, London.
3. Bhattacharaya, R. N. and Qumsiyeh, M. (1989). Second order and L^p -comparisons between the bootstrap and empirical Edgeworth expansion methodologies, *Ann. Statist.* **17**, 160-169.
4. Billingsley, P. (1979). *Probability and measure*, Wiley, New York.
5. Brown, L. D. (1986). *Fondamentals of statistical exponential families with applications in statistical decision theory*. Lecture Notes-Monograph series Vol. 9, Institute of Mathematical Statistics.
6. Casalis, M. (1992). Les familles exponentielles sur \mathbb{R}^2 de fonction-variance $am \otimes m + b(m) + c$. C. R. Acad. Sciennces, Serie I, **314**, 635-638.
7. Casalis, M. (1994). Les $2d + 4$ types de familles exponentielles naturelles quadratiques simples sur \mathbb{R}^d . C. R. Acad. Sciennces, Serie I, **318**, 261-264.
8. Chihara, T. S, (1978). *An introduction to orthogonal polynomials*. Gordon & Breach, New York.
9. Davison, A. C., Henkley, D. V. (1997). *Bootstrap methods and their application*. Cambridge University Press.
10. Davison, A. C., Henkley, D. V. and Schechtman, E. (1986). Efficient bootstrap simulations. *Biometrika* **73**, 555-566.
11. Do, K. A. (1992). A simulation study of balanced and antithetic bootstrap resampling methods, *J. Statist. Compu. Simul.*, **40**, 153-156.
12. Efron, B. (1979). Bootstrap methods: Another look at the jackknife, *Ann. Statist.* **7**, 1-26.
13. Efron, B. (1981). Nonparametric standard errors and confidence intervals (with discussion), *Canadian J. Statist.* **9**, 139-172.

14. Efron, B. (1982). *The Jackknife, the Bootstrap, and Other Resampling Plans*, SIAM, Philadelphia.
15. Efron, B. (1987). Better bootstrap confidence intervals (with Discussion). *Journal of the American Statistical Association* **82**, 171-200.
16. Efron, B. (1990). More efficient bootstrap computations, *J. Amer. Statist. Assoc.* **85**, 79-89.
17. Efron, B. (1992). Jackknife-after-bootstrap standard errors and influence functions (with discussions), *J. R. Statist. Soc. B*, **54**, 83-127.
18. Efron, B, and Stein, C., (1981). The Jackknife estimate of variance. *Ann. Statist.* **9**, 586-596.
19. Efron, B. and Tibshirani, R. J. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy, *Statist. Science* **1**, 54-77.
20. Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*, Chapman and Hall, New York.
21. Graham, R. L., Hinkley, D. V., John, P. W. M. and Shi, S. (1990). Balanced design of bootstrap simulations, *J. R. Statist. Soc. B*, **52**, 185-202.
22. Hall, P. (1989a). On the efficient bootstrap simulations, *Biometrika* **76**, 613-617.
23. Hall, P. (1989b). Antithetic reasampling for the bootstrap, *Biometrika* **76**, 713-724.
24. Hall, P. (1990a). Asymptotic properties of the bootstrap for heavy-tailed distributions, *Ann. Prob.* **18**, 1342-1360.
25. Hall, P. (1990b). On the relative performance of bootstrap and Edgeworth approximations of a distribution function, *J. Multivariate Anal.* **35**, 108-129.
26. Hall, P. (1990c). Performance of bootstrap balanced reasampling in distributions function and quantile problems, *Prob. Theory and Related fields* **85**, 239-267.
27. Henrici, P. (1986). *Applied and computational complex analysis*, John Wiley.

28. Kingman, J. F. C. and Taylor, S. J., (1966). *Introduction to measures and probability*. Cambridge University Press, Cambridge.
29. Konishi, S. (1991). Normalizing transformations and bootstrap confidence intervals, *Ann. Statist.* **19**, 2209-2225.
30. Letac, G and Mora, M., (1990). Natural real exponential families with quadratic variance functions. *The Annals of Statistics* **18**(1), 1-37.
31. Morris, C. N., (1982). Natural exponential families with quadratic variance functions. *The Annals of Statistics* **10**, 65-82.
32. Morris (1983). Natural exponential families with quadratic variance functions: Statistical theory. *The Annals of Statistics* **11**, 515-529.
33. Pfanzagl, J. (1985). *Asymptotic Expansions for General Statistical Models*, Springer-Verlag, Berlin.
34. Pommeret, D. (1996). Orthogonal polynomials and natural exponential families. *Test* **5**, 77-111.
35. Pommeret, D. (1997). Multidimensional Bhattacharyya matrices and exponential families. *Journal of Multivariate Analysis* **63**, 105-118.
36. Quenouille, M. H., (1949). Approximate tests of correlation in time-series. *Journal of the Royal Statistical Society series B* **11**, 68-84.
37. Serfling, R. J. (1980). *Approximation theorems of Mathematical Statistics*, Wiley, New York.
38. Shao, J. (1990). Bootstrap estimation of the asymptotic variance of statistical functionals. *Ann. Inst. Statist. Math.* **42**, 737-752.
39. Shao, J. (1992a). Some results for diferenciable statistical functionals, *Non-parametric statistics and related Topics*, A. K. Md. E. Saleh ed., 179-188, North-Holland, Amsterdam.
40. Shao, J. and Tu, D. *The Jackknife and Bootstrap*, (1995). Springer-Verlag New York, Inc.
41. Singh, K. (1981). On the asymptotic accuracy of Efron's bootstrap, *Ann. Statist.* **9**, 1187-1195.

- 42. Tukey, J. W., (1958). Bias and confidence in not quite large samples (Abstract). *Annals of Mathematical Statistics* 29, 614.

Mohamed Ghezziel
 Secretaries of the Board of Directors

Abraham E. Green
 Board for Unanimous
 17
 2024


 EC PRESIDENT,







