Elección discreta: Teoría, estimación y métodos numéricos

Tarea 2

El Colegio de México
Prof. Edwin Muñoz Rodríguez
Alumno: Max Brando Serna Leyva

Contents

1 Pregunta 1

Considere el modelo el modelo logit condicional usando los datos en `yogurt.csv`. La utilidad representativa es

$$\nu_{nj} = \alpha_j + \beta' x_{nj}$$

donde $x_{nj} = (\text{price}_j, \text{feat}_j)$, y α_j es la constante específica para la alternativa j.

1.1 Primer inciso

Normalice $\alpha_4 = 0$. Argumente por qué con esta normalización el modelo está identificado.
Respuesta: Bajo este escenario, las utilidades para cada alternativa del tipo n se representan en el siguiente modelo como:

$$\begin{aligned} u_{n1} &= \alpha_1 + \beta' x_{n1} + \varepsilon_1 \\ &\vdots \\ u_{n4} &= \alpha_4 + \beta' x_{n4} + \varepsilon_4 \\ &\vdots \\ u_{nj} &= \alpha_j + \beta' x_{nj} + \varepsilon_j \end{aligned}$$

Donde $\alpha_k - \alpha_j = d_{kj}$, $\forall j \neq k$. Esta representación es equivalente a:

$$\begin{aligned} u_{n1} &= \hat{\alpha}_1 + \beta' x_{n1} + \varepsilon_1 \\ &\vdots \\ u_{n4} &= \hat{\alpha}_4 + \beta' x_{n4} + \varepsilon_4 \\ &\vdots \\ u_{nj} &= \hat{\alpha}_j + \beta' x_{nj} + \varepsilon_j \end{aligned}$$

siempre que $\hat{\alpha}_k - \hat{\alpha}_j = d_{kj}$, $\forall j \neq k$. Este hecho implica una imposibilidad para estimar los parámetros α_j $\forall j \in |B|$, dado que existen infinitos valores que resultan en la misma diferencia d_{kj}. Es por ello que **al normalizar, obtenemos**:

$$\begin{aligned} \alpha_4 = 0 &\iff \alpha_k - \alpha_4 = \alpha_k - 0 = d_{k4} \\ &\iff \alpha_k = d_{k4} \end{aligned}$$

Por lo que nuestro modelo se convierte en:

$$\begin{aligned} u_{n1} &= d_{14} + \beta' x_{n1} + \varepsilon_1 \\ &\vdots \\ u_{n4} &= \beta' x_{n4} + \varepsilon_4 \\ &\vdots \\ u_{nj} &= d_{j4} + \beta' x_{nj} + \varepsilon_j \end{aligned}$$

Con este modelo no tenemos el problema anterior de imposibilidad en la estimación, puesto que ahora α_{jk} es igual a d_{j4} y éste es único; por tanto, el modelo está identificado, y d_{j4} se puede interpretar como el impacto promedio que factores no observados tienen sobre la utilidad de la alternativa j con respecto a la alternativa 4.

1.2 Segundo inciso

Estime el modelo usando Máxima Verosimilitud.

a. No use ninguna rutina integrada para estimar modelos logit multinomiales, como `mlogit` en R o equivalentes. Puede (y debe) usar rutinas de optimización numérica no lineal de propósito general como `optim` en R o `scipy.optimize` en Python. Recomiendo experimentar con diferentes métodos de optimización (`method='...'` en `scipy.optimize`).

b. Preste atención a la calidad del código y comentarios. Trate de hacer el código eficiente, elegante, legible y reusable, pues este código sera la base para las siguientes tareas.

c. Provea pseudo-codigo y codigo.

Queremos estimar

$$\hat{\alpha}_{MLE}, \hat{\beta}_{MLE} = \arg\max_{\alpha,\beta} LL\Big(\alpha, \beta \mid \{(m, j^*(m), s_m)\}_{m=1}^{M}\Big)$$

donde

$$LL\Big(\alpha, \beta \mid \{(m, j^*(m), s_m)\}_{m=1}^{M}\Big) = \sum_{m=1}^{M} log\left(\frac{e^{\alpha_{j^*(m)} + \beta' x_{mj^*(m)}}}{\sum\limits_{i \in B} e^{\alpha_i + \beta' x_{mi}}}\right)$$

recordando que normalizamos haciendo $\alpha_4 = 0$.

A continuación, proveo el pseudo-código para este ejercicio.

Algorithm 1 Estimación de un Modelo de Logit Condicional

Entrada: Base de datos de yogurt con características y elecciones
Cargar los datos desde un archivo CSV
Definir número de alternativas $A = 4$
Definir conjunto de alternativas disponibles $B = \{1, 2, 3, 4\}$
Extraer las características de cada alternativa (precio y publicidad)
Construir la matriz de características X con dimensiones $(M \times A \times K)$
Determinar la alternativa elegida por cada individuo j^*
function LogVerosimilitud(θ, X, j^*, B)
  Calcular utilidades $v_{jn} = \alpha_j + \beta' X_{m,j}$
  Calcular probabilidades de elección $P_{j,m} = \frac{e^{v_{j,m}}}{\sum_{i \in B} e^{v_{i,m}}}$ y sumar sobre m
  Retornar el logaritmo de verosimilitud negativo
end function
function EstimaciónMLE(X, j^*, B)
  Definir valores iniciales de β y α (con $\alpha_4 = 0$)
  Minimizar la función de log-verosimilitud con algún método usando `scipy.optimize.minimize`
  Retornar los valores estimados de β y α
end function
Ejecutar: Estimación de parámetros con MLE
Guardar los valores de β y α

En el teams se encuentra el archivo `.py` con el código.

1.3 Tercer inciso

Reporte los coeficientes estimados.

Parámetro	**Estimación de ML**
β_1	-37.058
β_2	0.487
α_1	1.388
α_2	0.644
α_3	-3.086
α_4	0.000

1.4 Cuarto inciso

Reporte el error estándar de sus estimados usando:

a. Teoria asintotica de primer orden.

b. Bootstrap (con 3211 repeticiones). En este caso, además de los errores, grafique la distribución marginal empírica de los estimadores. Provea pseudo-codigo y codigo.

Para el primer caso, usaremos la matriz hessiana inversa estimada mediante `scipy.optimize.minimize`. El resultado es el siguiente:

Parámetro	**Estimación**	**Error Est. Asintótico**
β_1	-37.058	2.399
β_2	0.487	0.120
α_1	1.388	0.088
α_2	0.644	0.054
α_3	-3.086	0.145
α_4	0.000	0.000

Estimaciones de β y α con errores estándar asintóticos calculados mediante `numdifftools`.

Para el caso con bootstrap, los resultados son:

Parámetro	**Estimación**	**Error Est. Bootstrap**
β_1	-37.058	2.377
β_2	0.487	0.133
α_1	1.388	0.087
α_2	0.644	0.055
α_3	-3.086	0.144
α_4	0.000	0.000

Estimaciones de β y α con errores estándar de Bootstrap

Existen diferencias que considero pueden deberse a que el método de optimización utilizado, `BFGS`, arroja una inversa de la Hessiana aproximada, que según foros de internet no es una aproximación acertada. Para solucionar esto habría que calcular el Hessiano analíticamente e integrarlo a la optimización en Python o calcularlo manualmente. Por facilidad, me quedaré con el resultado de `BFGS`.

A continuación el pseudo-código para la implementación del bootstrap:

Algorithm 2 Bootstrap para estimación de errores estándar

Entrada: Número de iteraciones *n_bootstraps*, datos x, *j_star*, alternativas B
Salida: Errores estándar de Bootstrap *boot_se_betas*, *boot_se_alphas*

Inicialización:
Obtener número de observaciones $M \leftarrow$ longitud de *df*
Crear lista vacía *resample_indices*

Generación de índices de resampling:
for $b = 1$ to *n_bootstraps* **do**
    Seleccionar M índices aleatorios con reemplazo
    Almacenar los índices en *resample_indices*[b]
end for

Definir función de Bootstrap: ▷ Estimación por iteración
function BootstrapIteration(*resample_idx*)
    Extraer muestra *x_boot* $\leftarrow$ x[*resample_idx*]
    Extraer elecciones *j_star_boot* $\leftarrow$ *j_star*[*resample_idx*]
    Estimar parámetros (β*_boot*, α*_boot*) $\leftarrow$ EstimateBetaAlpha(*x_boot*, *j_star_boot*, B)
    return (β*_boot*, α*_boot*)
end function

Ejecutar el Bootstrap mediante paralelización:
Ejecutar BootstrapIteration para cada *resample_idx* en *resample_indices*
Guardar resultados en *results*

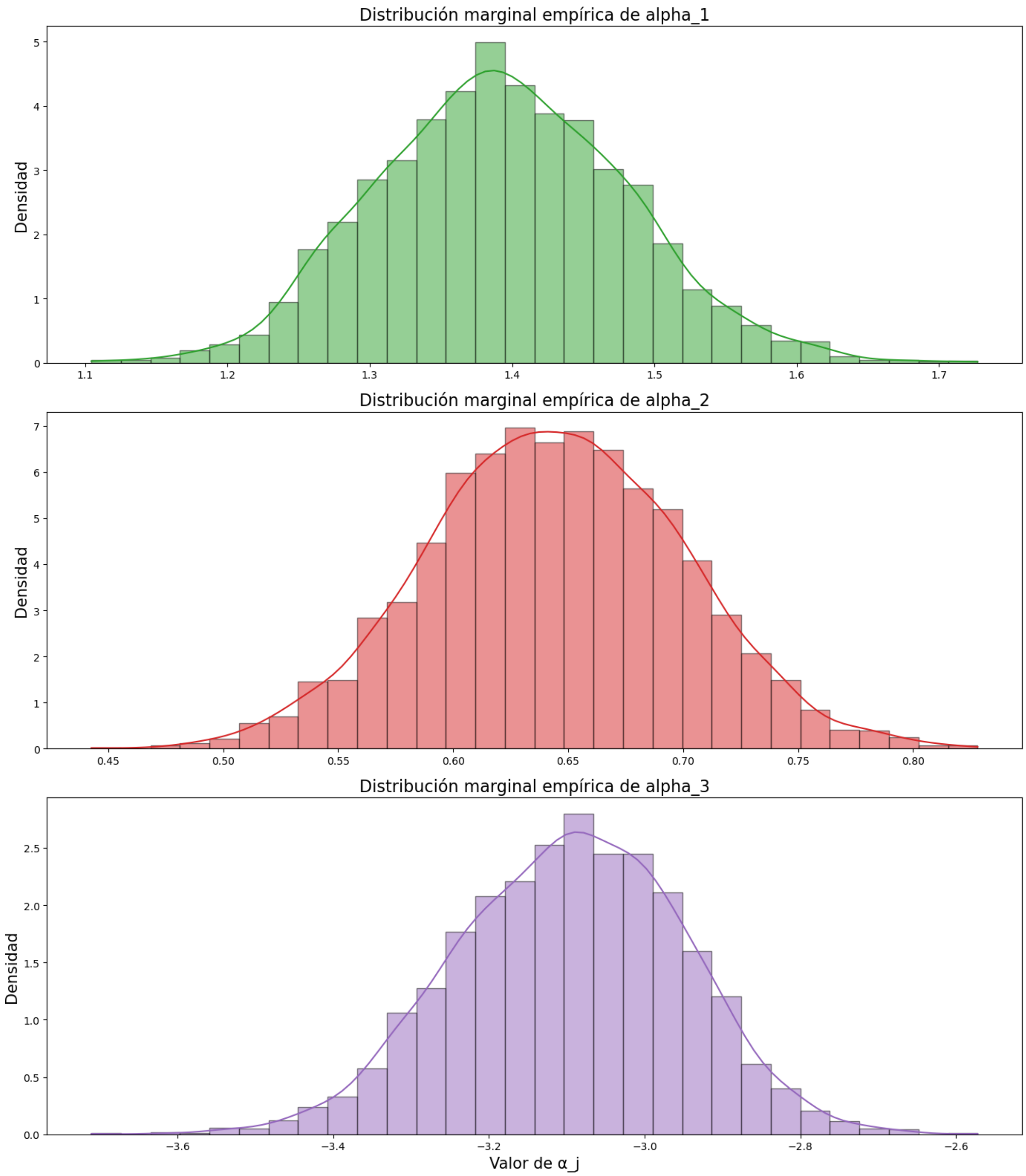
Extraer estimaciones de Bootstrap:
Crear matriz *bootstrap_betas* con los valores β*_boot* de *results*
Crear matriz *bootstrap_alphas* con los valores α*_boot* de *results*

Calcular errores estándar de Bootstrap:
boot_se_betas $\leftarrow$ desviación estándar de *bootstrap_betas*
boot_se_alphas $\leftarrow$ desviación estándar de *bootstrap_alphas*

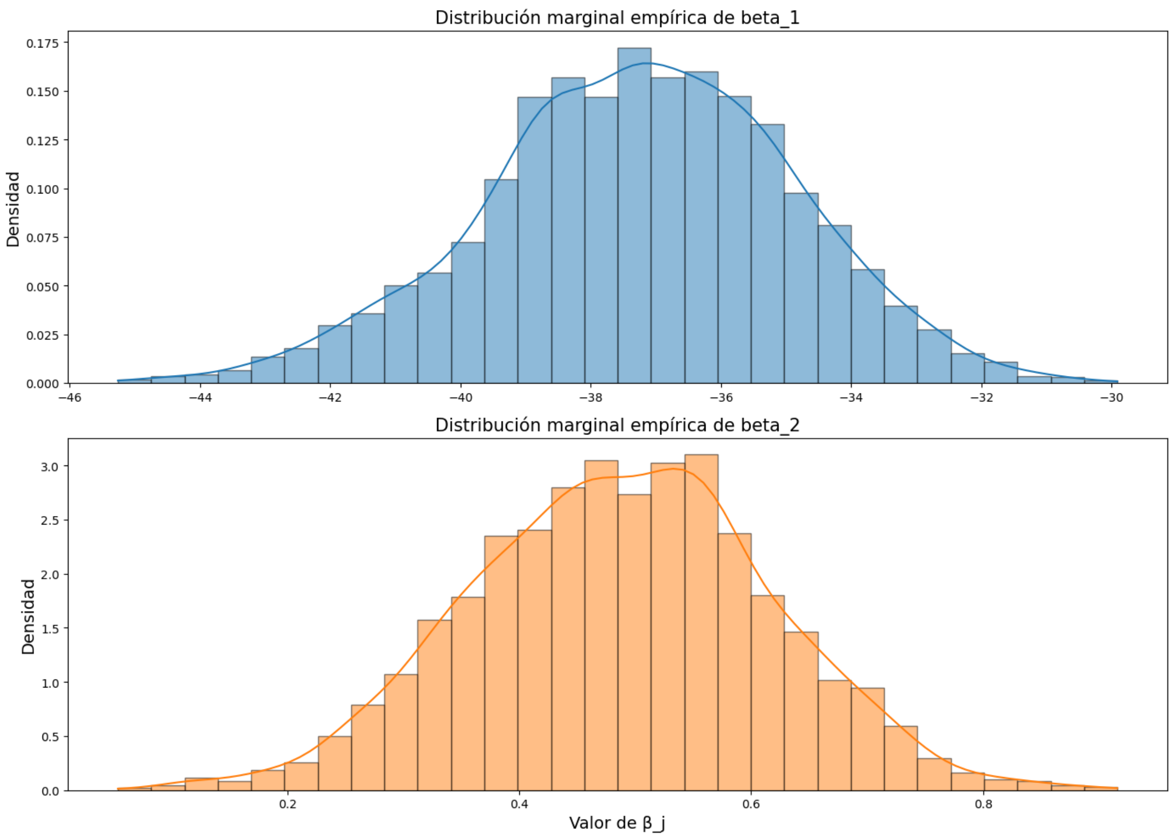
Fin del algoritmo.

En lo sucesivo presento las distribuciones marginales empíricas de los estimadores de ML obtenida mendiante bootstrap.

Distribución marginal empírica de los estimadores de β obtenida mediante bootstrap.



Distribución marginal empírica de los estimadores de α obtenida mediante bootstrap.



1.5 Quinto inciso

Calcule las elasticidades precio propias y cruzadas, e interprete los resultados.

Esto se obtiene usando las fórmulas derivadas en la tarea anterior. Los resultados son los siguientes:

	$j = 1$	$j = 2$	$j = 3$	$j = 4$
$j = 1$	-2.701	1.177	0.054	0.659
$j = 2$	1.236	-1.844	0.054	0.659
$j = 3$	1.236	1.177	-1.933	0.659
$j = 4$	1.236	1.177	0.054	-2.288

Elasticidades precio propias (diagonal) y cruzadas (fuera de la diagonal), promediadas entre individuos.

Esta tabla presenta las elasticidades precio cruzadas y propias para cada una de las cuatro alternativas, promediadas entre todas las M observaciones. Dado que la elasticidad propia es negativa para todas las alternativas, nos encontramos ante bienes normales. Además, las elasticidades cruzadas son positivas, lo que implica que son sustitutos entre ellas.