Estimación de un modelo lineal generalizado del salario mediante métodos numéricos

Valentina Caldiroli, Maximiliano Saldaña

Noviembre 2020

Introducción

- El objetivo planteado es estimar un modelo logístico mediante un método de máxima verosimilitud que aplique los métodos de optimización del curso y comparar el desempeño de nuestra estimación con el de las funciones base de R
- Algoritmos de optimización empleados: el método del ascenso más rápido, el de Newton y el de Broyden-Fletcher-Goldfarg-Shanno.
- Se hace una comparación términos de velocidad (cantidad de iteraciones que toman los algoritmos en converger) y precisión respecto a replicar el resultado del método de mínimos cuadrados iterativamente re-ponderados empleado por la función glm() para la estimación de modelos lineales generalizados.

Presentación del problema

El modelo en cuestión es de la forma siguiente:

$$P(Y_i = 1) = \frac{exp(x_i'\beta)}{1 + exp(x_i'\beta)} = \pi_i = E(y_i)$$

Donde se tiene que:

$$y_i = sal\_med_i$$

$$x_i'\beta = \beta_0 + \beta_1 female_i + \beta_2 educ_i + \beta_3 exper_i + \beta_4 tenure_i + \beta_5 northcen$$

$$+\beta_6 south_i + \beta_7 west_i + \beta_8 reg.metro + \beta_9 construc_i + \beta_{10} services_i$$

$$+\beta_{11} trade_i + \beta_{12} profocc_i + \beta_{13} clerocc_i + \beta_{14} servocc_i$$

Presentación del problema

La función de unión (la que vincula $E(y_i)$ con x_i') es:

$$log_e\left[\frac{\pi_i}{1-\pi_i}\right] = exp(x_i'\beta)$$

- El problema de estimar el vector de parámetros β se traduce en un problema de optimización, ya que buscamos que dicho vector sea el que maximice la función de verosimilitud de Y. Como Y_i son variables aleatorias Bernoulli independientes e idénticamente distribuidas $\forall\ i = 1, \ldots, n$ la función de verosimilitud es:

Presentación del problema

$$L(\pi_1, \dots, \pi_n | Y_1, \dots, Y_n) = \prod_{i=1}^{n} f_{Y_i}(y_i) = \prod_{i=1}^{n} \pi_i^{y_i}(1 - \pi_i^{y_i})^{1-y_i}$$

Resulta equivalente trabajar con el logaritmo de la verosimilitud, que es:

$$log\ L(\pi_1, \dots, \pi_n | Y_1, \dots, Y_n) = log\ \prod_{i=1}^{n} \pi_i^{y_i}(1 - \pi_i^{y_i})^{1-y_i}$$

$$= \sum_{i=1}^{n} \left[Y_i log \left(\frac{\pi}{1 + \pi} \right) \right] + \sum_{i=1}^{n} log(1 - \pi_i)$$

Presentación del problema

Teniendo en cuenta que:

$$1 - \pi_i = [1 + exp(x_i'\beta)]^{-1}$$

$$\frac{\pi}{1 + \pi} = exp(x_i'\beta)$$

$$log\ L(\beta|Y) = \sum_{i=1}^{n} Y_i(x_i'\beta) - \sum_{i=1}^{n} log(1 + exp(x_i'\beta))$$

Estimaremos el vector β que maximice la función anterior.

Presentación de los datos

- Se hará uso de una base de salarios de Estados Unidos del año 1976, extraída del libro *Introducción a la econometría* (2009) de Jeffrey M. Wooldridge.
- La base cuenta con observaciones de 526 personas y con 22 variables.
- La variable a ser explicada es una indicadora que vale 1 cuando el salario de la persona es mayor o igual del promedio (5,896103) y 0 cuando es menor -*sal_med*-.

	sal_med	Proporción
1	Bajo el promedio	0.62
2	Sobre promedio	0.38

Table 1: Proporción de personas con salarios por debajo y sobre el promedio

Presentación de los datos

- Las variables explicativas (que se eligieron en base a lo ya trabajado en el curso Modelos Lineales) son:
  - Cuantitativas: educación -*educ*-, experiencia -*exper*- y antigüedad -*tenure*-
  - Cualitativas: sexo de la persona -*female*-, región (dividida en Norte -*northcen*-, Sur -*south*-, Oeste -*west*- y Este -categoría de referencia-), si vive en una región metropolitana -*reg.metro*-, rama de actividad (dividida en Construcción -*cosntruc*-, Comercio -*trade*-, Servicios -*services*- y Otros -categoría de referencia-) y Ocupación (dividida en Profesional -*profocc*-, Administrativos -*clerocc*-, Servicios -*servocc*- y Otros -categoría de referencia-).

Estimando mediante las funciones base

- Inicialmente para tener como referencia y para comparar, estimaremos el modelo mediante la función de R *glm()* del paquete base *stats*, la cual hace uso del método de estimación por mínimos cuadrados iterativamente re-ponderados (IWLS).
- Evaluando los coeficientes resultantes en la función de verosimilud el valor del máximo obtenido es -232,3441.
- Para emplear los métodos numéricos de optimización antes mencionados haremos uso de las funciones para aplicarlos vistas en el curso, las que fue necesario modificar para que fuera posible emplear aproximaciones numéricas del gradiente y la hessiana de la log-verosimilitud en vez de ser estas calculadas manualmente.

Estimación mediante el método del ascenso más rápido

- Para comenzar con este método y verificar que las funciones se desempeñan correctamente, probamos suministrándole como valores iniciales los estimados mediante la función *glm()*. A continuación, se presenta un cuadro comparando los valores de ambas estimaciones:

Estimación mediante el método del ascenso más rápido

	glm()	Ascenso
(Intercept)	-4.442367437	-4.442367437
female	-1.194755469	-1.194755469
educ	0.277912495	0.277912308
exper	0.007568890	0.007568376
tenure	0.083587709	0.083587627
northcen	-0.012727340	-0.012727340
south	0.035935951	0.035935951
west	0.688820844	0.688820844
reg.metro	0.745975971	0.745975971
construc	-0.363769611	-0.363769611
services	-0.493415175	-0.493415175
trade	-1.429567604	-1.429567604
profocc	0.854540015	0.854540015
clerocc	-0.585495085	-0.585495085
servocc	-1.209387935	-1.209387935

Table 2: Comparación de glm() y ascenso más rápido

Estimación mediante el método del ascenso más rápido

- Se puede apreciar, como era de esperarse, la similitud entre ambas estimaciones, difiriendo a lo sumo recién a partir del sexto valor después de la coma (se fijó una tolerancia de 10^{-5}). Toma solo una iteración tener este resultado.
- Reduciendo la tolerancia la diferencia entre las estimaciones por los dos métodos se hace menor, pero toma más iteraciones y es necesario permitir más iteraciones de la búsqueda lineal, además de que se necesita hacer *forward-tracking* en varios pasos.
- El valor del máximo obtenido es -232,3441, lo que lo hace equivalente al de la función de control.
- La elección del vector inicial es un problema, tenemos que partir de un punto inicial "cercano" al que maximiza la función para que converja rápidamente.

Estimación mediante el método del ascenso más rápido

- Se puede utilizar una grilla de valores, pero el problema con este camino es de caracter práctico.
- Incluso considerando un número reducido de valores para la grilla (por ejemplo: -2, -1, 0, 1 y 2), el espacio que ocuparía en la memoria es demasiado grande debido al número de coeficientes a estimar.
- Sumándose a esto el elevado tiempo que implicaría evaluar suficientes valores iniciales distintos para encontrar los que hagan que el método del ascenso más rápido (o cualquier otro) converja.
- Reduciendo aún más la grilla (los valores posibles son -1, 0 y 1) el primero problema se soluciona, pero sigue presente el segundo.

Estimación mediante el método del ascenso más rápido

- Probemos ahora con un vector de valores iniciales (x_0) compuesto solo de ceros. Nos lleva a que el algoritmo no converja a una solución, dadas las 1000 iteraciones asignadas y presente varios fallos en la búsqueda lineal (el número de iteraciones -10- resultaba muy reducido para encontrar una longitud del paso del algoritmo apropiada).
- Incluso asignando 5000 iteraciones el resultado continúa siendo el mismo. Esto nos lleva a pensar que se necesita un mejor punto de inicio para que el algoritmo converja más velozmente, es decir, uno más cercano al verdadero valor que máximiza la log-verosimilitud.

Estimación mediante el método del ascenso más rápido

- Alternativamente se puede intentar llegar a una solución utilizando estimaciones iniciales dadas por el método de mínimos cuadrados ordinarios (MCO) y considerando el salario en vez de la indicadora utilizada.
- Las estimaciones pueden dar una idea aproximada de la relación entre nuestra variable dependiente y sus regresores.
- Pero utilizar las estimaciones por MCO no nos lleva a que se alcance la convergencia del algoritmo, incluso permitiéndole al algoritmo utilizar 5000 iteraciones.
- Aunque puede que se encuentre la solución con un mayor número de iteraciones, conviene encontrar un punto de arranque que permita llegar a la convergencia en un número menor de iteraciones y por lo tanto en menor tiempo.

Estimación mediante el método de Newton

- Dado los problemas de la velocidad de convergencia y con el vector inicial del método anterior, parece una buena opción intentar utilizar el método de Newton.
- Este algoritmo acelera la convergencia del método del ascenso más rápido al incorporar información de la hessiana de la función en el algoritmo.
- Para que en efecto el método de Newton converja a un máximo y no a un mínimo, tenemos que tener una hessiana definida negativa.
- A modo de comprobar esto, utilizamos una función vista en el curso que calcula los vectores propios de la hessiana inicial, si todos ellos resultan negativos la matriz es definida negativa.
- Al igual que con el método anterior, comencemos con el punto inicial de "control", el vector de estimaciones dadas por la función *glm()*.

Estimación mediante el método de Newton

- Análogamente a lo que ocurrió con el método del ascenso más rápido, el de Newton converge en este caso en una iteración a un vector de coeficientes muy similares y un máximo equivalente a los obtenidos con la función *glm()*.
- Cabe destacar que la búsqueda lineal necesito hacer *forward-tracking* y se da un fallo en la búsqueda lineal.
- Ahora intentemos utilizar como vectores iniciales el compuesto por ceros y el de las estimaciones MCO. La hessiana de la log-verosimilitud resulta definida negativa al evaluarla en ambos vectores, por lo que en un principio pueden ser candidatos.

Estimación mediante el método de Newton

Variables	glm()	Vector de 0's
(Intercept)	-4.442367437	-4.442363082
female	-1.194755469	-1.194755231
educ	0.277912495	0.277912242
exper	0.007568890	0.007568853
tenure	0.083587709	0.083587710
northcen	-0.012727340	-0.012727760
south	0.035935951	0.035935444
west	0.688820844	0.688820173
reg.metro	0.745975971	0.745975951
construc	-0.363769611	-0.363771043
services	-0.493415175	-0.493415066
trade	-1.429567604	-1.429567877
profocc	0.854540015	0.854540092
clerocc	-0.585495085	-0.585495421
servocc	-1.209387935	-1.209388297

Table 3: Comparación glm() y el método de newton

Estimación mediante el método de Newton

Variables	glm()	$\hat{\beta}_{MCO}$
(Intercept)	-4.442367437	-4.442361732
female	-1.194755469	-1.194755497
educ	0.277912495	0.277912116
exper	0.007568890	0.007568843
tenure	0.083587709	0.083587707
northcen	-0.012727340	-0.012727487
south	0.035935951	0.035935708
west	0.688820844	0.688820410
reg.metro	0.745975971	0.745975895
construc	-0.363769611	-0.363770666
services	-0.493415175	-0.493415144
trade	-1.429567604	-1.429567922
profocc	0.854540015	0.854540730
clerocc	-0.585495085	-0.585494736
servocc	-1.209387935	-1.209387806

Table 4: Comparación glm() y el método de newton

Estimación mediante el método de Newton

- Podemos ver que este método convergió rápidamente, en tan solo cuatro iteraciones en el caso del vector de ceros y en siete usando las estimaciones MCO.
- Se necesito hacer *forward-tracking* solo una vez en ambos casos.
- En este caso el vector nulo parece ser una buena opción de punto de arranque si no se cuenta con información sobre los coeficientes a estimar, incluso mejor que utilizar las estimaciones MCO ya que nos ahorramos iteraciones.

Estimación mediante el método de Broyden-Fletcher-Goldfarg-Shanno.

- Al ahorrarse la evaluación de la hessiana, estos métodos pueden resultar más veloces que el de Newton, aunque esto también dependerá del vector inicial empleado.
- El método de Newton tenderá a tomar menos iteraciones si se comienza cerca de la solución.
- Resulta necesario una evaluación inicial de la hessiana, que se puede elegir como el opuesto de la matriz identidad cuya dimensión es la cantidad de variables de la función a optimizar, o emplear una estimaciones de la hessiana en x_0.

Estimación mediante el método de Broyden-Fletcher-Goldfarg-Shanno.

- Se quiso también hacer una comparación con el resultado obtenido con el algoritmo bfgs de la función *optim()* de la biblioteca *stats*, pero tuvo problemas al aproximar la hessiana.
- Continuamos presentando las estimaciones resultantes de utilizar como vector inicial el nulo y el de las estimaciones MCO.

Estimación mediante el método de Broyden-Fletcher-Goldfarg-Shanno.

Variable	glm()	Vector de 0's
(Intercept)	-4.442367437	-4.441287082
female	-1.194755469	-1.194356801
educ	0.277912495	0.277846624
exper	0.007568890	0.007565239
tenure	0.083587709	0.083585123
northcen	-0.012727340	-0.012964024
south	0.035935951	0.035914771
west	0.688820844	0.688620025
reg.metro	0.745975971	0.745850209
construc	-0.363769611	-0.363854741
services	-0.493415175	-0.493346501
trade	-1.429567604	-1.429542202
profocc	0.854540015	0.854472462
clerocc	-0.585495085	-0.585483998
servocc	-1.209387935	-1.209298455

Estimación mediante el método de Broyden-Fletcher-Goldfarg-Shanno.

Variables	glm()	$\hat{\beta}_{MCO}$
(Intercept)	-4.442367437	-4.442571372
female	-1.194755469	-1.194609027
educ	0.277912495	0.277942647
exper	0.007568890	0.007576988
tenure	0.083587709	0.083582161
northcen	-0.012727340	-0.012858416
south	0.035935951	0.035839134
west	0.688820844	0.688680786
reg.metro	0.745975971	0.745871069
construc	-0.363769611	-0.363864731
services	-0.493415175	-0.493359194
trade	-1.429567604	-1.429527372
profocc	0.854540015	0.854280808
clerocc	-0.585495085	-0.585675743
servocc	-1.209387935	-1.209619458

Estimación mediante el método de Broyden-Fletcher-Goldfarg-Shanno.

- Si bien se llega al mismo máximo que en el método anterior, el desempeño de este método resultó peor que el de Newton en cuanto a la cantidad de iteraciones que toma para converger y la precisión de las estimaciones respecto a *glm()*.
- En cuanto a lo primero, se puede deber a que se partió de un vector lo suficientemente similar al que maximiza la log-verosimilitud para que el método de Newton fuera más veloz que el BFGS.
- Lo segundo puede ser consecuencia de que al aproximarse y no evaluarse la hessiana se pierde precisión, esto se puede solucionar reduciendo la tolerancia, aunque la consecuencia de esto es un aumento en las iteraciones necesarias para la convergencia.

Resumen de los resultados

	Método	Iter.	f(x_max)	cif. = a glm()
1	Ascenso más rápido	No converge	No converge	No converge
2	Newton (vector 0's)	4	-232.344056	5
3	Newton (beta_mco)	7	-232.344056	5
4	BFGS (vector 0's)	35	-232.344059	2
5	BFGS (beta_mco)	24	-232.344057	2

Table 7: Resumen de resultados de los distintos métodos.

Pasos futuros

- A futuro se podría profundizar en métodos para la elección de vectores iniciales, como las mencionadas grillas de valores, ya que si bien en este caso se tenía una idea de donde partir para enfrentar el problema esto no va a ocurrir en la mayoría de aplicaciones.
- La comparación con otros métodos de optimización adicionales también sería otro punto para expandir sobre lo trabajado.