

Comparación de la eficiencia del método de optimización BFGS en C, 0x y R para un modelo de regresión no lineal

Efficiency Comparisson for the BFGS Method in C, 0X and R for a nonlinear regression model

Luz Marina Rondón Poveda^a
franciscorincon@usantotomas.edu.co

Resumen

En este artículo se presenta una comparación de la eficiencia del método de optimización no lineal BFGS cuando es aplicado usando los programas C, 0x y R. Esta comparación es realizada evaluando el desempeño del método BFGS cuando es usado para obtener las estimativas de máxima verosimilitud de los parámetros de un modelo de regresión no lineal con errores de distribución normal. La eficiencia del método es estudiada usando algunas medidas como, por ejemplo, el número promedio de evaluaciones de la función y de la derivada de la función requeridas para la convergencia del método. Estas medidas se obtienen usando simulación de Monte Carlo.

Palabras clave: Optimización no lineal, BFGS, Modelos no lineales..

Abstract

En este artículo se presenta una comparación de la eficiencia del método de optimización no lineal BFGS cuando es aplicado usando los programas C, 0x y R. Esta comparación es realizada evaluando el desempeño del método BFGS cuando es usado para obtener las estimativas de máxima verosimilitud de los parámetros de un modelo de regresión no lineal con errores de distribución normal. La eficiencia del método es estudiada usando algunas medidas como, por ejemplo, el número promedio de evaluaciones de la función y de la derivada de la función requeridas para la convergencia del método. Estas medidas se obtienen usando simulación de Monte Carlo.

Key words: Optimización no lineal, BFGS, Modelos no lineales..

^aDocente, Facultad de estadística. Universidad Santo Tomás

1. Introducción

La necesidad de optimizar una función objetivo es una tarea que se encuentra en muchas áreas del conocimiento, en especial en estadística, donde muchos de los métodos de estimación de parámetros consisten en la maximización (maxima verosimilitud, por ejemplo) o minimización (mínimos cuadrados, por ejemplo) de una función, la cual es no lineal en muchos de los casos, dificultando la obtención de soluciones cerradas y planteando la necesidad de usar métodos iterativos. Los programas C (Kernighan y Ritchie (1980)), Ox (Doornik (2001)) y R (Venables y Ripley (2002)) ofrecen alternativas interesantes para la optimización de funciones objetivo a través de distintos métodos, entre los que se cuentan aquellos que requieren la primera y segunda derivadas de la función objetivo como aquellos que solo requieren la primera derivada e incluso aquellos que no requieren ninguna de las dos. Entre los métodos más conocidos está el método BFGS (Broyden (1970), Fletcher (1970), Goldfarb (1970) y Shanno (1970)).

El lenguaje de programación C no posee un área específica de aplicación, sin embargo, debido a su falta de restricciones y a su alta flexibilidad, es una herramienta muy conveniente y eficaz para muchas aplicaciones de lo que en realidad lo son lenguajes supuestamente más poderosos. Para optimización no lineal, está disponible un conjunto de rutinas que pueden ser usadas para la aplicación de algunos de los métodos más conocidos, entre los cuales está el método BFGS. El código fuente de estas rutinas puede ser encontrado en Press *et al.* (1992). En segundo lugar, se considera el lenguaje de programación matricial orientado a objetos Ox, el cual ofrece muchos recursos matemáticos y estadísticos. Apesar de ser un lenguaje relativamente nuevo es muy usado debido a que es una de las plataformas científicas más confiables. Finalmente, se considera el paquete estadístico R. Este es un lenguaje para computación estadística y análisis de datos que tienen código de fuente abierto, debido a esto, recibe innumerables contribuciones de varias comunidades científicas.

El objetivo de este artículo es estudiar y comparar el desempeño del método BFGS cuando es aplicado usando las rutinas disponibles en los lenguajes de programación C, Ox, y R. Para esto se considera un modelo de regresión no lineal con errores de distribución normal, puesto que, para el análisis estadístico de datos usando este tipo de modelos, la estimación de los parámetros de interés es realizada usando el método de máxima verosimilitud y el método de mínimos cuadrados no lineales, los cuales envuelven la optimización de una función. Además, estos tipos de modelos son muy usados en la práctica, razón por la cual se hace importante conocer las características de las herramientas computacionales disponibles para su aplicación. Este artículo está organizado en la siguiente forma: En la sección 2 se presentan los modelos no lineales, y se hace referencia a la estimación de los parámetros para este tipo de modelos; en la sección 3 se describen de forma general los métodos de optimización, resaltando la diferencia entre el empleo de uno u otro método; en la sección 4 se describe una aplicación que tiene como objetivo evaluar el desempeño del método BFGS cuando es usado para optimizar una función para hallar las estimativas del vector de parámetros en un modelo de regresión no lineal usando

los lenguajes de programación C, O \mathbf{x} , y R. En la sección 5 se presentan los resultados encontrados en las simulaciones de Monte Carlo al usar el método para optimizar una función, y finalmente en la sección 6 son descritas las principales conclusiones.

2. Modelos no Lineales

Sean y_1, \dots, y_n los valores de una variable de interés medida en n individuos o unidades experimentales, los cuales son asumidos como variables aleatorias independientes de media $E(y_i|\mathbf{x}_i) = \mu_i$ y varianza $V(y_i|\mathbf{x}_i) = \sigma^2$, generadas a través del siguiente mecanismo

$$y_i = \mu(\boldsymbol{\beta}; \mathbf{x}_i) + \epsilon_i, \quad i = 1, \dots, n, \quad (1)$$

en que $\mathbf{x}_i = (x_{i1}, \dots, x_{il})$ es un vector con los valores de l variables explicativas para el individuo i y $\mu_i(\boldsymbol{\beta}) = \mu(\boldsymbol{\beta}; \mathbf{x}_i)$ es una función continua y doblemente diferenciable en relación al vector de parámetros $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^T$. Adicionalmente, se asume que la matriz de derivadas $\mathbf{D}_\beta = \partial\boldsymbol{\mu}/\partial\boldsymbol{\beta}$ tiene rango p ($p < n$) para todo $\boldsymbol{\beta} \in \Omega_\beta \subset \mathbb{R}^p$, con Ω_β un conjunto compacto con puntos interiores, siendo $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)^T$. El modelo (1) es considerado no lineal cuando la matriz de derivadas \mathbf{D}_β depende del vector de parámetros $\boldsymbol{\beta}$. Además, se asume que $\epsilon_1, \dots, \epsilon_n$ son variables aleatorias independientes con distribución normal de media 0 y varianza σ^2 , es decir, comportándose de acuerdo a la siguiente función de densidad de probabilidad

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

Se denota $\epsilon_i \sim N(0, \sigma^2)$. A continuación, se presentan algunos ejemplos de modelos no lineales, los cuales son usados frecuentemente en áreas como química y biología:

- Modelo de Gompertz

$$\mu_i(\boldsymbol{\beta}) = \beta_1 \exp\{-\beta_2 \exp(-\beta_3 x_i)\}, \quad \beta_2, \beta_3 > 0.$$

- Modelo de Weibull

$$\mu_i(\boldsymbol{\beta}) = \beta_1 - \beta_2 \exp(-\beta_3 x_i^{\beta_4}), \quad x_i > 0.$$

- Modelo de Michaelis-Menten

$$\mu_i(\boldsymbol{\beta}) = \frac{\beta_1 x_i}{x_i + \beta_2}.$$

- Modelo de crecimiento

$$\mu_i(\boldsymbol{\beta}) = \beta_1 - \frac{\beta_2}{x_i + \beta_3}. \quad (2)$$

Por ejemplo, la curva de Gompertz permite describir el crecimiento de un objeto o un individuo en relación al tiempo, donde β_1 es el tamaño máximo que puede tener el individuo y β_3 la tasa de crecimiento. Por otro lado, la curva de Michaelis-Menten es usada en el área química para describir el comportamiento cinético de las enzimas en relación a su concentración, donde β_1 es la velocidad máxima de reacción y β_2 una constante cinética. Sin embargo, debido a su flexibilidad, estas curvas también son utilizadas para describir gran variedad de fenómenos en áreas como la ingeniería y economía. En las figuras 1 y 2 se ilustra el comportamiento de las curvas de Gompertz y Michaelis-Menten para algunas combinaciones de los parámetros.

Figura 1: *Comportamiento de la familia de curvas de Gompertz para diferentes valores de β_3 con $\beta_1 = 10$ e $\beta_2 = 7$.*

PSfrag replacements

$$\begin{array}{c} \mu(\beta) \\ x \\ \beta_3 \end{array}$$

Una de las ventajas de los modelos no lineales es que permiten la obtención de parametrizaciones que son fácilmente interpretables, además, generalmente los modelos no lineales requieren menor número de parámetros para lograr un ajuste satisfactorio. Algunas veces es posible “linealizar” el modelo de interés usando transformaciones de la respuesta y de los parámetros del modelo, lo que facilita el proceso de inferencia (estimación y pruebas de hipótesis de los parámetros de interés), pues permite la aplicación de las herramientas inferenciales clásicas en modelos de regresión. Sin embargo, en estos casos los parámetros del modelo pueden perder su interpretación intrínseca, adicionalmente, se puede alterar la estructura de la distribución del error, invalidando los supuestos de normalidad e independencia, por ejemplo. Las expresiones usadas para evaluar la adecuación de la aproximación lineal son conocidas en la literatura estadística como medidas de no linealidad. En la práctica, las medidas de curvatura descritas en Bates y Watts (1988) son las herramientas utilizadas con mayor frecuencia en la evaluación de la

Figura 2: Comportamiento de la familia de curvas de Michaelis-Menten para diferentes valores de β_2 con $\beta_1 = 10$.

PSfrag replacements

$$\frac{\mu(\boldsymbol{\beta})}{x}$$

β_2

no linealidad intrínseca de los modelos no lineales.

2.1. Estimación de los parámetros

Para la estimación de los parámetros en los modelos de regresión no lineal es posible aplicar el método de máxima verosimilitud. Este método de estimación consiste en maximizar una función que expresa el chance de observar los datos que componen la muestra en función de los parámetros del modelo. La utilización de este método presenta varias ventajas, pues, los estimadores obtenidos de esta manera poseen propiedades estadísticas deseables, como eficiencia, consistencia y normalidad asintóticas. Para la aplicación del método de máxima verosimilitud es posible considerar la función de log-verosimilitud de $\boldsymbol{\theta} = (\boldsymbol{\beta}^T, \sigma^2)^T$, que es dada por la siguiente expresión

$$l(\boldsymbol{\theta}) = -\frac{n}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} Q(\boldsymbol{\beta}), \quad (3)$$

siendo $Q(\boldsymbol{\beta}) = (\mathbf{y} - \boldsymbol{\mu})^T (\mathbf{y} - \boldsymbol{\mu})$. Los estimadores de máxima verosimilitud de los parámetros de interés son obtenidos maximizando la función dada en (3) en relación a $\boldsymbol{\theta}$. De esta manera son consideradas las derivadas de $l(\boldsymbol{\theta})$, que están dadas por

$$\mathbf{U}_{\boldsymbol{\beta}}(\boldsymbol{\theta}) = \frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\beta}} = \frac{1}{\sigma^2} \mathbf{D}_{\boldsymbol{\beta}}^T (\mathbf{y} - \boldsymbol{\mu})$$

y

$$\mathbf{U}_{\sigma^2}(\boldsymbol{\theta}) = \frac{\partial l(\boldsymbol{\theta})}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} Q(\boldsymbol{\beta}).$$

Dado que $l(\boldsymbol{\theta})$ es una función cóncava, los estimadores de máxima verosimilitud de $\boldsymbol{\beta}$ y σ^2 son obtenidos resolviendo conjuntamente las ecuaciones $\mathbf{U}_{\boldsymbol{\beta}}(\hat{\boldsymbol{\theta}}) = 0$ y $\mathbf{U}_{\sigma^2}(\hat{\boldsymbol{\theta}}) = 0$. Sin embargo, se puede observar que la solución de la ecuación $\mathbf{U}_{\boldsymbol{\beta}}(\hat{\boldsymbol{\theta}}) = 0$ no depende de σ^2 y equivale a minimizar $Q(\boldsymbol{\beta})$ en relación a $\boldsymbol{\beta}$ con el método de mínimos cuadrados no lineales. Para los modelos no lineales, a diferencia de los modelos lineales, la solución de las ecuaciones normales de los mínimos cuadrados requiere un método numérico iterativo, pues la solución analítica generalmente no puede ser encontrada. El gradiente de $Q(\boldsymbol{\beta})$ está dado por $\mathbf{g}(\boldsymbol{\beta}) = (g_1(\boldsymbol{\beta}), \dots, g_p(\boldsymbol{\beta}))^T$, con

$$\mathbf{g}(\boldsymbol{\beta}) = -2\mathbf{D}_{\boldsymbol{\beta}}^T(\mathbf{y} - \boldsymbol{\mu}).$$

En resumen, la estimación de los parámetros de interés $\boldsymbol{\beta}$ y σ^2 puede ser realizada en dos pasos. En el primero, la estimativa de máxima verosimilitud de $\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}}$, es obtenida via mínimos cuadrados minimizando la función $Q(\boldsymbol{\beta})$. En el segundo paso, la estimativa de máxima verosimilitud de σ^2 , $\hat{\sigma}^2$, se calcula como $\hat{\sigma}^2 = Q(\hat{\boldsymbol{\beta}})/n$, que se obtiene al resolver $\mathbf{U}_{\sigma^2}(\hat{\boldsymbol{\theta}}) = 0$.

3. Métodos de optimización

La minimización de Q para este tipo de modelos debe ser realizada usando procedimientos de optimización no lineal. Diferentes enfoques para la aplicación de estos métodos deben ser adoptados dependiendo de la dimensión y de la complejidad del conjunto Ω , en el cual está definida la solución óptima para la función Q . En los modelos no lineales, $Q(\boldsymbol{\beta})$ es una función continua y diferenciable. De esta manera, es posible aplicar los *métodos gradiente*. El proceso de búsqueda de la solución óptima usado por estos métodos puede ser descrito como

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \lambda_t \boldsymbol{\Delta}_t,$$

onde $\boldsymbol{\theta}_t$ es la solución viable, λ_t es el tamaño del paso y $\boldsymbol{\Delta}_t = M_t \mathbf{g}_t$ es el vector de dirección, todos son obtenidos en la iteración t . Para el vector de dirección se tiene que M_t es una matriz definida positiva y $\mathbf{g}_t = \mathbf{g}(\boldsymbol{\theta}_t)$ es el vector gradiente, en que $\mathbf{g}(\boldsymbol{\beta}) = \frac{\partial Q(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$.

Este proceso requiere la especificación por parte del usuario de un punto inicial $\boldsymbol{\theta}_0$ y de una regla de parada para el algoritmo. Esta regla es el criterio usado por el algoritmo para finalizar el proceso y puede ser enunciada en diferentes vías: *i*) en términos del número de iteraciones especificando un número máximo; *ii*) a través de la diferencia entre las soluciones viables estudiadas en iteraciones consecutivas, especificando una variación mínima; *iii*) usando el criterio de la derivada

especificando un intervalo alrededor de cero, en el cual la derivada será considerada como cero; y *iv*) utilizando una combinación de las anteriores reglas.

Entre los métodos gradiente, uno de los mas conocidos es el método de Newton-Rapshon. Este método usa la matriz $-H^{-1}$ (H es la matriz hessiana) como matriz M , sin embargo, cuando el valor del punto inicial θ_0 no esta próximo del punto óptimo, la matriz $-H^{-1}$ puede no ser definida positiva, dificultando el uso del método. Además, en algunas aplicaciones es difícil calcular la matriz hessiana, por lo tanto, la literatura científica proporciona otros métodos llamados *Quasi-Newton*; estos métodos usan una aproximación iterativa de la matriz hessiana $M_{t+1} = M_t + N_t$, donde N_t es una matriz definida positiva, garantizando que esta aproximación es definida positiva en cada paso del proceso iterativo, pues M_{t+1} es la suma de dos matrices definidas positivas. Para la implementación de estos algoritmos es necesario seleccionar un punto inicial θ_0 y una matriz M_0 definida positiva.

Dentro de los métodos Quasi-Newton se encuentra el algoritmo DFP (Davidon, Fletcher e Powell), el cual usa

$$M_{t+1} = M_t \frac{\delta_t \delta_t'}{\delta_t' \nu_t} + \frac{M_t \nu_t \nu_t' M_t}{\nu_t' M_t \delta_t},$$

donde $\delta_t = \theta_{t+1} - \theta_t$ y $\nu_t = g(\theta_{t+1}) - g(\theta_t)$. Sin embargo, el mas usado de estos métodos es el algoritmo BFGS (Broyden, Fletcher, Goldfarb y Shanno), que usa

$$M_{t+1} = M_t \frac{\delta_t \delta_t'}{\delta_t' \nu_t} + \frac{M_t \nu_t \nu_t' M_t}{\nu_t' M_t \delta_t} - \nu_t' M_t \nu_t \left(\frac{\delta_t}{\delta_t' \nu_t} - \frac{M_t \nu_t}{\nu_t' M_t \delta_t} \right) \left(\frac{\delta_t}{\delta_t' \nu_t} - \frac{M_t \nu_t}{\nu_t' M_t \delta_t} \right).$$

El método Steepest Ascent es el algoritmo mas sencillo de esta familia; en este caso $M_t = I$ en todos los pasos del esquema iterativo, donde I es la matriz identidad de orden k , lo que hace $\Delta_t = g_t$. El algoritmo de este método consiste en moverse del punto θ_t al punto θ_{t+1} maximizando a lo largo de la linea entre θ_t y la dirección dada por $h_t = g_t$. El método Fletcher-Reeves es una modificación del método Steepest Ascent con

$$h_{t+1} = g_t + \gamma_t h_t,$$

donde

$$\gamma_t = \frac{g_{t+1}' g_{t+1}}{g_t' g_t}.$$

El método Polak-Ribiere modifica el calculo del vector direccional h_t haciendo

$$\gamma_t = \frac{[g_{t+1} - g_t]' g_{t+1}}{g_t' g_t}.$$

La literatura recomienda el cálculo de γ_t segun Polak-Ribiere.

4. Aplicación

De los métodos de optimización no lineal descritos en la sección anterior se considera el método BFGS. Con el objetivo de estudiar y comparar el desempeño de este método cuando es aplicado usando las rutinas disponibles en los lenguajes de programación C (Kernighan y Ritchie (1980)), Ox (Doornik (2001)), y R (Venables y Ripley (2002)) se considera un modelo de regresión no lineal con respuesta normal, puesto que, para el análisis estadístico de datos usando este tipo de modelos, la estimación de los parámetros de interés es realizada usando el método de máxima verosimilitud y el método de mínimos cuadrados no lineales, los cuales envuelven la optimización de una función. Además, estos tipos de modelos son muy usados en la práctica, razón por la cual se hace importante conocer las características de las herramientas computacionales disponibles para su aplicación. Se considera el *Modelo de Crecimiento* presentado en la ecuación (2).

El lenguaje de programación C no posee un área específica de aplicación, sin embargo, debido a su falta de restricciones y a su alta flexibilidad, es una herramienta muy conveniente y eficaz para muchas aplicaciones de lo que en realidad lo son lenguajes supuestamente más poderosos. Para optimización no lineal, está disponible un conjunto de rutinas que pueden ser usadas para la aplicación de algunos de los métodos más conocidos, entre los cuales está el método BFGS. El código fuente de estas rutinas puede ser encontrado en Press *et al.* (1992). En C, se usa la rutina `dfpmin` para la aplicación del método BFGS en la estimación de los parámetros del modelo. En segundo lugar, se considera el lenguaje de programación matricial orientado a objetos Ox, el cual ofrece muchos recursos matemáticos y estadísticos. Apesar de ser un lenguaje relativamente nuevo es muy usado debido a que es una de las plataformas científicas más confiables. Para la aplicación del método de optimización no lineal BFGS se usa la función `MaxBFGS`, siendo flexible pues permite la aplicación de este método usando derivadas numéricas calculadas por la plataforma. En este trabajo se considera la rutina `MaxBFGS` con derivadas analíticas y numéricas para la estimación de los parámetros del modelo (2). Finalmente, fue usado el lenguaje R. Este es un lenguaje para computación estadística y análisis de datos que tienen código de fuente abierto. Debido a esto, recibe innumerables contribuciones de varias comunidades científicas. En este caso usamos la función `optim`, la cual permite de la misma manera que en Ox, la optimización de funciones usando el método BFGS con derivadas analíticas (proporcionadas por el usuario) y numéricas (calculadas por la plataforma).

Para ilustrar la aplicación del método de optimización BFGS se generaron r muestras independientes de tamaño n del vector (y_i, x_i) , siguiendo la relación descrita por el modelo de crecimiento (2). Los valores de la variable explicativa x_i fueron generados siguiendo distribución uniforme en el intervalo $(0, 1000)$. Estos valores fueron mantenidos fijos a lo largo del proceso y fueron los mismos en los tres programas usados. En cada una de las r réplicas fue calculado el estimador de máxima verosimilitud del vector de parámetros β usando el métodos BFGS. En cada réplica fueron obtenidas las siguientes medidas de eficiencia: número de evaluaciones de

la función objetivo, número de evaluaciones de la derivada de la función objetivo, el valor de $\hat{\beta}$, el valor de la función evaluada en $\hat{\beta}$ (es decir, $Q(\hat{\beta})$) y el tiempo de ejecución. Los valores de los parámetros del modelo fueron $\beta_1 = 6$, $\beta_2 = 130$ y $\beta_3 = 37$, el número de réplicas fue $r=10000$ y los tamaños muestrales considerados fueron $n = 50, 100$ y 150 . El criterio de parada del algoritmo (`gtol`) en el caso de `C`, `Ox` y `R` fue el mismo, y es el siguiente

$$\text{Max}\{|g_1(\beta_t)|, \dots, |g_p(\beta_t)|\} < \text{gtol},$$

con g_r el r -ésimo elemento del vector gradiente, en este trabajo `gtol`= $10 \times e^{-4}$.

5. Resultados

En esta sección se describen los resultados obtenidos de los experimentos de simulación para estudiar y comparar el desempeño del método de optimización no lineal BFGS, cuando es aplicado usando las rutinas disponibles en los lenguajes de programación `C`, `Ox`, y `R`. En las tablas 1 a 3 se presentan las medias y las varianzas de las medidas de eficiencia obtenidas en los lenguajes `C`, `Ox` y `R` con derivadas analíticas, es decir, proporcionadas por el usuario; además, para `Ox` y `R` son presentados los resultados obtenidos con derivadas numéricas; todo lo descrito anteriormente fue realizado para cada uno de los tamaños muestrales considerados.

En la tabla 1 se presenta el caso en que el tamaño de la muestra fue 50, se consideran las medidas de eficiencia para comparar el desempeño del método BFGS. A partir de los valores de las medias se puede observar que el tiempo de ejecución promedio es menor usando `C`, seguido por `Ox` y por último `R`, la misma relación se observa cuando se usa la derivada numérica. El lenguaje que necesita en promedio de un número menor de evaluaciones de la derivada de la función, cuando son usadas las derivadas analíticas, es `C`. Aunque la variación de esta es la mayor entre los tres lenguajes y el que presenta menor variación es `Ox`. La misma relación se observa para los valores promedios de la evaluación de las derivadas de la función. El cambio entre los resultados es presentado en `Ox` porque presenta la mayor media de los tres lenguajes. Cuando son usadas las derivadas numéricas el lenguaje que presenta menor media y variación de evaluaciones de la función es `R`, sin embargo este presenta tiempo promedio de ejecución mayor que `Ox`. Note que el valor promedio del mínimo de la función objetivo para el lenguaje `Ox` es menor, indicando la eficacia de este lenguaje para encontrar el mínimo de la función, sin encontrar diferencia si es usada la derivada numérica o analítica.

En la tabla 2 se pueden observar los resultados cuando el tamaño de la muestra fue 100, en los valores promedios de las evaluaciones de la función se observa que el menor valor es presentado por el lenguaje `C`, aunque también presenta la variación más alta. El lenguaje `Ox` es el que mas evaluaciones de la función necesita, tanto con derivadas numéricas como analíticas pero además presenta la menor variación. Para `R`, el valor promedio de evaluaciones de la función están muy próximos tanto cuando son usadas las derivadas analíticas y las numéricas,

en cuanto a las variaciones de esta medida para este lenguaje son mas altas para el caso de las derivadas numéricas, manteniendose constante cuando son usadas las derivadas numéricas. Los valores promedios de las evaluaciones de la derivada da función son menores para el lenguaje C, seguido por R y por último 0x.

En la tabla 3 se presentan los valores promedios y las varianzas de las medidas consideradas para evaluar el desempeño del método BFGS, cuando el tamaño de la muestra es 150. En este caso se puede observar resultados parecidos a los anteriores tamaños considerados, es decir, el lenguaje C es el que tiene menor valor promedio en las evaluaciones de la función objetivo, asi como también tiene la mayor variación, comparado con los otros dos lenguajes. El tiempo de ejecución para C es el menor cuando son usadas las derivadas numéricas; 0x presenta el mayor valor promedio de las evaluaciones de la función, aunque presenta un tiempo menor de ejecución cuando es comparado con R.

Se puede observar en las tablas que, en la medida que aumenta el tamaño de la muestra, el número promedio de evaluaciones de la función en los dos casos (es decir, con derivadas analíticas y numéricas) disminuye en los tres lenguajes, lo mismo se puede observar en el caso de las varianzas para esta medida. El valor promedio de las evaluaciones de la derivada de la función también va disminuyendo a medida que el tamaño de la muestra aumenta. Además de esto, los valores promedios de $\hat{\beta}$ estan mas cercanos a los valores de los parámetros; la variación de estas medidas disminuye a medida que el tamaño de la muestra aumenta, esto va de acuerdo con las propiedades asintóticas de los estimadores de máxima verosimilitud. Observe que los valores promedios del mínimo de la función objetivo en el lenguaje 0x, en todos los escenarios considerados, siempre es menor indicando la eficacia de este lenguaje para encontrar el mínimo de la función, no encontrando diferencia cuando son consideradas las derivadas analíticas o numéricas.

6. Conclusiones

En general, en el desempeño del método BFGS en los lenguajes C, 0x y R, pueden ser observadas algunas diferencia, tales como, el lenguaje que necesita menos evaluaciones de la función y evaluaciones de la derivada de la función objetivo es C, aunque presenta la mayor variación cuando son usadas las derivadas analíticas, además, este lenguaje presenta tiempos promedios de ejecución menores en todos los escenarios considerados; el lenguaje R presenta el menor valor promedio de las evaluaciones de la función cuando son usadas las derivadas numéricas, aunque este lenguaje presenta tiempos promedios de ejecución mayores. De esta manera, la elección de un lenguaje entre C, 0x y R cuando es de interés optimizar una función usando el método BFGS depende del criterio o de la medida de evaluación considerada por el usuario. Por ejemplo, si él opta por usar un lenguaje que necesite del menor número de evaluaciones de la función cuando es usada la derivada analítica, él debe escoger C. Pero si el criterio fuera el uso de las derivadas numéricas, entonces se debe utilizar R, o si, por el contrario, él desea usar el lenguaje que tiene

mayor eficiencia para encontrar el valor mínimo de la función puede usar Ox.

7. Referencias

- Bates, D.M. y Watts, D.G. (1988). *Nonlinear Regression Analysis and its Applications*. John Wiley, New York.
- Broyden, C.G. (1970). The Convergence of a Class of Double-rank Minimization Algorithms. *Journal of the Institute of Mathematics and Its Applications*, **6**, 76-90.
- Doornik J.A., (2001). *Ox: an Object-oriented Matrix Programming Language*. 4 ed. Londres: Timberlake Consultants.
- Fletcher, R., (1970). A New Approach to Variable Metric Algorithms. *Computer Journal*, **13**, 317-322.
- Goldfarb, D., (1970). A Family of Variable Metric Updates Derived by Variational Means. *Mathematics of Computation*, **24**, 23-26.
- Kernighan, B.W. y Ritchie, D.M. (1980). *The C Programming Language*. 2 ed. Englewood Cliffs: Prentice Hall.
- Nocedal, J. y Wright, S.J. (1999). *Numerical Optimization*. Springer-Verlag, New York.
- Press, W.H.; Flanery, B.P.; Teolofsky, S.A. y Vetterling, W.T. (1992). *Numerical Recipes in C: The Art of Scientific Computing*. 2 ed. Cambridge University.
- Seber, G.A.F. y Wild, C.J. (1989). *Nonlinear Regression*. John Wiley, New York.
- Shanno, D.F., (1970). Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computation*, **24**, 647-656.
- Venables, W.N. y Ripley, B.D., (2004). *Modern Applied Statistics with S*. 4 ed. Springer.

		C	Ox
μ	Evaluaciones de la función	19.522	32.123
	Evaluaciones de la derivada de la función	11.545	22.999
	Valor de $\hat{\alpha}$	6.000	6.001
	Valor de $\hat{\beta}$	130.315	130.250
	Valor de $\hat{\gamma}$	37.113	37.099
	Valor de $Q(\hat{\beta})$	23.485	23.422
	Tiempo de ejecución en segundos	0.000156	0.000939
σ^2	Evaluaciones de la función	393.597	4.871
	Evaluaciones de la derivada de la función	131.457	2.875
	Valor de $\hat{\alpha}$	0.004	0.001
	Valor de $\hat{\beta}$	54.192	54.520
	Valor de $\hat{\gamma}$	5.863	5.879
	Valor de $Q(\hat{\beta})$	23.653	23.222
	Tiempo de ejecución en segundos	0.000002	0.000013

Tabla 2: Medias (μ) y varianzas (σ^2) de las medidas de eficiencia del método BFGS en un modelo de regresión no lineal, con $n = 100$

	Medidas	Derivada analítica	
		C	Ox
μ	Evaluaciones de la función	19.131	32.043
	Evaluaciones de la derivada de la función	10.338	22.023
	Valor de $\hat{\alpha}$	6.000	6.000
	Valor de $\hat{\beta}$	130.129	130.150
	Valor de $\hat{\gamma}$	37.062	37.061
	Valor de $Q(\hat{\beta})$	48.508	48.380
	Tiempo de ejecución en segundos	0.000307	0.001085
σ^2	Evaluaciones de la función	393.785	2.796
	Evaluaciones de la derivada de la función	88.662	2.320
	Valor de $\hat{\alpha}$	0.001	0.002
	Valor de $\hat{\beta}$	29.839	29.176
	Valor de $\hat{\gamma}$	3.815	3.729
	Valor de $Q(\hat{\beta})$	48.596	49.341
	Tiempo de ejecución en segundos	0.000004	0.000015

		C	Ox
μ	Evaluaciones de la función	18.560	31.365
	Evaluaciones de la derivada de la función	8.899	21.328
	Valor de $\hat{\alpha}$	6.000	6.000
	Valor de $\hat{\beta}$	130.135	130.110
	Valor de $\hat{\gamma}$	37.055	37.045
	Valor de $Q(\hat{\beta})$	73.585	73.360
	Tiempo de ejecución en segundos	0.000379	0.001249
σ^2	Evaluaciones de la función	351.984	2.871
	Evaluaciones de la derivada de la función	88.674	2.468
	Valor de $\hat{\alpha}$	0.001	0.001
	Valor de $\hat{\beta}$	21.275	20.380
	Valor de $\hat{\gamma}$	2.920	2.793
	Valor de $Q(\hat{\beta})$	74.292	72.715
	Tiempo de ejecución en segundos	0.000005	0.000018