Optimización

Maximización de la Función Log-verosimilitud de el Modelo de Regresion Logística para la Clasificación de dos Clases.

Tarea 4

3 de marzo de 2021

Erika Rivadeneira Pérez
erika.rivadeneira@cimat.mx
Matemáticas Aplicadas - CIMAT

1. Resumen

En el presente reporte se muestran los resultados de maximizar la función log-verosimilitud del modelo de regresión logística para la clasificación de dos clases utilizando el método del descenso por gradiente. Se comparan los resultados obtenidos al utilizar este método con dos técnicas de búsquedas en línea, backtracking y bisección.

2. Introducción

Dada una muestra observada $(x_1, \ldots, x_n)$ y una ley de probabilidad P_θ, la verosimilitud cuantifica la probabilidad de que las observaciones provengan efectivamente, de una muestra (teórica) de la ley P_θ.

El método más común para estimar parámetros es el método de máxima verosimilitud. Sea $X_1, \ldots, X_n$ independientes e idénticamente distribuidas con función de densidad de probabilidad $p(x;\theta)$ entonces la función de verosimilitud se define como

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} p\left(x_i;\theta\right)$$

y la log-verosimilitud se define como

$$l(\theta) = \log \mathcal{L}(\theta) = \sum_{i=1}^{n} \log p\left(x_i;\theta\right)$$

Luego, el estimador de máxima verosimilitud es el valor de θ que maximiza $\mathcal{L}(\theta)$.
En este reporte se utiliza el método numérico del descenso por gradiente (steepest descent) para encontrar el máximo de la función log-verosimilitud de un modelo de regresión logística para la clasificación de dos clases.

3. Metodología

La función $h(\beta, \beta_0)$ (1) corresponde al log-verosimilitud del modelo de regresión logística para las clases $y_i \in \{0, 1\}$

$$\begin{aligned} h\left(\boldsymbol{\beta}, \beta_0\right) &= \sum_{i=1}^{n} y_i \log \pi_i + (1 - y_i) \log\left(1 - \pi_i\right), \\ \pi_i := \pi_i\left(\boldsymbol{\beta}, \beta_0\right) &= \frac{1}{1 + \exp\left(-\boldsymbol{x}_i^T \boldsymbol{\beta} - \beta_0\right)}, \end{aligned} \tag{1}$$

donde $x_i \in \mathbb{R}^{784}$ y $y_i \in \{0, 1\}$, los cuales vienen dados en un conjunto de datos, en el cual $x_i \in \mathbb{R}^{784}$ representa una observación de una matriz de dimensión 50000×784 y y_i es una entrada de un vector de tamaño 50000.

Como lo menciona en la introducción se quiere encontrar un máximo de la función (1) por el método de descenso por gradiente. Para esto se requiere encontrar el gradiente de $h(\beta, \beta_0)$, el cual viene dado por la siguiente expresión

$$\nabla h(\beta, \beta_0) = \sum_{i=1}^{n} \frac{\partial h}{\partial \pi_i} \cdot \nabla \pi_i. \tag{2}$$

Notemos que,

$$\frac{\partial h}{\partial \pi_i} = \frac{y_i - \pi_i}{\pi_i(1 - \pi_i)},$$

y sabiendo que $\pi_i = \frac{1}{1+exp(-x_i^T\beta-\beta_0)}$ y $(1 - \pi_i) = \frac{exp(-x_i^T\beta-\beta_0)}{1+exp(-x_i^T\beta-\beta_0)}$ se tiene que

$$\begin{aligned} \nabla \pi_i &= \frac{exp(-x_i^T\beta - \beta_0)}{(1 + exp(-x_i^T\beta - \beta_0))^2} \cdot (\vec{x_i}, 1) \\ &= \frac{exp(-x_i^T\beta - \beta_0)}{(1 + exp(-x_i^T\beta - \beta_0))(1 + exp(-x_i^T\beta - \beta_0))} \cdot (\vec{x_i}, 1) \\ &= (1 - \pi_i) \cdot \pi_i \cdot (\vec{x_i}, 1) \end{aligned}$$

Ahora, sustituyendo estos resultados en (2)

$$\nabla h(\beta, \beta_0) = \sum_{i=1}^{n} (y_i - \pi_i) \cdot \pi_i \cdot (\vec{x_i}, 1). \tag{3}$$

Notemos que, si bien se utiliza el método del descenso por gradiente para encontrar mínimos de funciones también podemos usarlo para encontrar máximos ya que resulta equivalente encontrar el maximo de una función $f(\cdot)$ y el mínimo de la función $-f(\cdot)$.
En la siguiente sección se hace uso de (1) y (3) para el implemento del algoritmo del descenso por gradiente.

4. Resultados

Se consideró el vector inicial $x_0 = (\vec{\beta^*}, \beta_0^*) = (0, 0, ..., 0)$ de tamaño 785, el cual está compuesto por valores iniciales para $\vec{\beta}$ y β_0 respectivamente. Utilizando el método del descenso por gradiente en este vector inicial se obtuvieron los siguiente resultados:

- **Con backtracking:**
  - $||g(x*)|| = 4{,}017e - 12$
  - Iteraciones: 849

- Tiempo de ejecución: 95.67 seg

- **Con bisección:**
  - $||g(x*)||$: 0,00096
  - Iteraciones: 557
  - Tiempo de ejecución: 90.88 seg.

Los vectores solución $x*$ se encuentran adjuntos. La solución con bisección fue guardada en el archivo *solucion_bis.txt* y el vector solución por backtracking en el archivo *solucion_back.txt*.
Por otro lado, el error de $x* = (\hat{\beta}, \hat{\beta_0})$ fue calculado usando la siguiente expresión

$$\text{error } = \frac{1}{n}\sum_{i=1}^{n}\left|\mathbf{1}_{\pi_i(\hat{\boldsymbol{\beta}},\hat{\beta}_0)>0,5}\left(\boldsymbol{x}_i\right) - y_i\right|.$$

Los errores conseguidos fueron cero en ambos casos de los métodos de búsqueda en línea.

5. Conclusiones y discusión

No fue posible evaluar la función $h(\beta, \beta_0)$ (1) en el vector solución, esto pudo deberse a que hay operaciones que no están definidas o no son posibles debido a la evaluación en la función logaritmo. Además, se observó que al utilizar backtracking el tamaño de paso α no cambió, esto debido a que la primera condición de Wolfe siempre se cumplía, para un $\alpha = 1$ inicial.

Es importate mencionar que se pudo obtener los vectores solución deseados haciendo uso de la librería numba, ya que si no se lo utiliza el tiempo de ejecución sería de aproximadamente 40 minutos, dependiendo de los criterios de paro que se utilicen. Finalmente, se puede concluir que con ambos métodos de búsqueda en línea funcionan bien para encontrar un punto crítico por el método de descenso por gradiente, esto debido a que los errores fueron iguales en ambos casos y el resto de resultados como iteraciones, $||\nabla f(x*)||$ y tiempo de ejecución fueron muy similares. Aunque el método de bisección realiza un número menor de iteraciones y de tiempo de ejecución.

Referencias

[1] J. Nocedal and S. J. Wright. Numerical Optimization. Springer Series in Operation Research, 2000.