Regresión Logística

desde el punto de vista de optimización

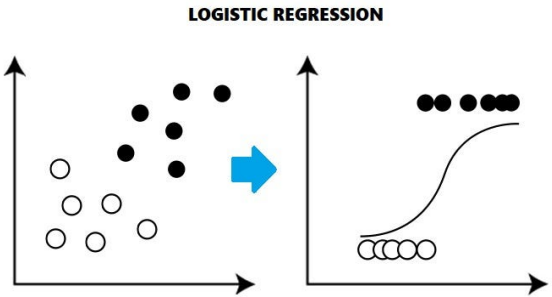
Mariel Guamuche 21150, Adrián López 21357

Universidad del Valle de Guatemala

14/Noviembre/2024

Qué es regresión logística

La regresión logística es un modelo de clasificación supervisado, donde busca predecir el resultado de una variable discreta $\boldsymbol{y}$ en función de las variables predictoras $\boldsymbol{X}$, donde $\boldsymbol{X} = \{x_i\}$ para $i = 1, ..., n$.



Su objetivo es poder trazar una recta (o hiperplano) que separe ambas clases.
Se comenzará a explicar la deducción de la regresión logística en términos de una sola variable.

Deducción de la regresión logística

Debido a que este modelo está asociado a los ensayos de Bernoulli, donde se conoce el número de éxitos más no sus probabilidades, se trabaja con la proporción de que algo pasa entre la proporción de que no sucede, i.e. las posibilidades (*odds*).
La probabilidad asociada a los *odds* se calcula como

$$P(odds) = log(odds) = log\left(\frac{p}{1-p}\right) \tag{1}$$

donde p es la probabilidad de que el evento ocurra.
Tomar en cuenta que se aplica logarítmico para asegurar que la probabilidad esté entre 0 y 1.
Recordando la forma del modelo de regresión lineal, se tiene que:

$$g\left(\boldsymbol{X}\right) = \beta_0 + \beta_1 x \tag{2}$$

donde β_0 es el intercepto en y, β_1 es la pendiente de la recta, x es el valor de la coordenada x y y es el valor de la coordenada y.

Deducción de la regresión logística

Para que el modelo asigne valores de 0 o 1, se realiza la trasformación de una nueva función $F(g(x))$. Se comienza igualando la ecuación (1) y (2).

$$log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$$

$$e^{\ln\left(\frac{p(x)}{1-p(x)}\right)} = e^{\beta_0+\beta_1 x}$$

$$\left(\frac{p(x)}{1-p(x)}\right) = e^{\beta_0+\beta_1 x}$$

Sea $y = e^{\beta_0+\beta_1 x}$

$$\left(\frac{p(x)}{1-p(x)}\right) = y$$

$$p(x) = y(1-p(x)) = y - y(p(x))$$

$$p(x) + y(p(x)) = y$$

$$p(x)(1+y) = y$$

Deducción de la regresión logística

$$p(x) = \frac{y}{1+y} = \frac{1}{1+\frac{1}{y}}$$

$$p(x) = \frac{1}{1+\frac{1}{e^{\beta_0+\beta_1 x}}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

$$\therefore F(g(x)) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

Esta es la **función sigmoide** asociada a una sola variable.Para $\boldsymbol{x_i} \in R^n$ para $i = 1, ..., n$ se busca $\boldsymbol{y} \in R^n$ y β tal que:

$$\frac{exp\left(\boldsymbol{x}_i^T \boldsymbol{y} + \beta\right)}{1 + exp\left(\boldsymbol{x}_i^T \boldsymbol{y} + \beta\right)} \tag{3}$$

Deducción de la Maximum Likelihood Estimation

Sea

$$\beta^T = [\beta_1, \beta_2, \ldots, \beta_n]$$

los parámetros del modelo y

$$\hat{x} = [x_1, x_2, \ldots, x_n]$$

las variables independientes.
Entonces, la hipótesis del modelo, $h(\hat{x})$, está dada por la función sigmoide:

$$h(\hat{x}) = \frac{1}{1 + e^{-\beta^T \hat{x}}}$$

La probabilidad de que $y = 1$ dado $\hat{x}$ y los parámetros β es:

$$P(y = 1|\hat{x}; \beta^T) = h(\hat{x})$$

y, por otro lado, la probabilidad de que $y = 0$ es:

$$P(y = 0|\hat{x}; \beta^T) = 1 - h(\hat{x})$$

Deducción de la Maximum Likelihood Estimation

Para cada observación en la muestra, estas probabilidades se combinan en función de si $y_i = 1$ o $y_i = 0$, como sigue:

$$P(y|\hat{x}; \beta^T) = h(\hat{x})^y (1 - h(\hat{x}))^{1-y}$$

Entonces Dado un conjunto de datos con m observaciones, la función de verosimilitud para los parámetros β es el producto de las probabilidades conjuntas para cada observación:

$$L(\beta) = \prod_{i=1}^{m} P(y^{(i)}|\hat{x}^{(i)}; \beta^T) = \prod_{i=1}^{m} \left(h(\hat{x}^{(i)})^{y^{(i)}} \cdot (1 - h(\hat{x}^{(i)}))^{1-y^{(i)}} \right)$$

Este producto representa la verosimilitud de los parámetros β dado el conjunto de datos y es la función que se busca maximizar para encontrar los mejores valores de β.

Maximización de la Verosimilitud (Máxima Verosimilitud)

El objetivo es maximizar la función de verosimilitud para encontrar los valores de los coeficientes $(\beta_0, \beta_1, \ldots, \beta_k)$ que hacen que el modelo se ajuste mejor a los datos.
Maximizar el producto de muchas probabilidades puede ser complejo, por lo que en la práctica se maximiza el logaritmo de la verosimilitud (log-verosimilitud), que convierte el producto en una suma:

$$\ln(L(\beta_0, b\beta_1, \ldots, \beta_k)) = \sum_{i=1}^{m} (y_i \ln(P(Y = 1|X_i)) + (1 - y_i) \ln(1 - P(Y = 1|X_i))) \quad (4)$$

La log-verosimilitud permite simplificar los cálculos y hacer más eficiente el proceso de maximización.
Con la función de la log-verosimilitud ya establecida , se busca maximizar la función pero, ¿por qué maximizarla?

Maximización de la Verosimilitud (Máxima Verosimilitud)

La maximización de la función de verosimilitud (y de su logaritmo, la log-verosimilitud) se realiza porque, en estadística, **la verosimilitud mide cuán probable es que los datos observados hayan sido generados por el modelo** con un conjunto específico de parámetros. Al **maximizar la verosimilitud**, buscamos el conjunto de parámetros que hace que los datos observados sean lo **más probable** bajo el modelo.

Métodos para maximizar la log-verosimilitud

Métodos para maximizar la log-verosimilitud:

- **Gradiente Descendente:** Simple, pero puede ser lento en algunos casos.
- **Newton-Raphson:** Usa la segunda derivada para una convergencia más rápida, pero requiere el cálculo de la Hessiana.
- **Fisher Scoring:** Variante de Newton-Raphson usando la matriz de información de Fisher, más estable.
- **Cuasi-Newton (BFGS, L-BFGS):** Aproximan la Hessiana, son rápidos y eficientes para grandes datasets.

En la práctica, la elección del método depende del tamaño de los datos, la complejidad del modelo y la disponibilidad de recursos computacionales.

Aplicación del método gradiente descendente

Recordando la ecuación de log-verosimilitud:

$$\ln(L(\beta_0, \beta_1, \ldots, \beta_k)) = \sum_{i=1}^{m} (y_i \ln(P(Y = 1|X_i)) + (1 - y_i) \ln(1 - P(Y = 1|X_i)))$$

Para la implementación del gradiente descendente, usaremos el promedio de la **log-verosimilud negativa** para asegurar la **escalabilidad**, el valor de pérdida será independiente del tamaño del conjunto de datos, y la **estabilidad en la optimización**, evita oscilaciones bruscas entre cada iteración. Por lo tanto la función a minimizar es la siguiente:

$$-\frac{1}{m} \sum_{i=1}^{m} (y_i \ln(P(Y = 1|X_i)) + (1 - y_i) \ln(1 - P(Y = 1|X_i)))$$

Adicionalmente, el gradiente de la log-verosimilitud negativa es el siguiente:

$$\frac{\partial - Ln(L(\beta))}{\partial \beta} = -\frac{\sum_{i=1}^{m} (y_i - P(Y = 1|X_i)) X_i}{m} \tag{5}$$

Aplicación del método gradiente descendente: Pseudocódigo

function GradientDescent(X,y,alpha,epochs)
    $m, n \leftarrow$ size X
    $b \leftarrow$ vector zero tamaño n
    for i:=0 epochs **do**
        $z \leftarrow prodPunto(X, beta)$
        $y_{pred} \leftarrow sigmoid(z)$
        $gradient \leftarrow prodPunto(X.T, (y_{pred} - y))/m$
        $beta[i] = beta[i-1] - alpha * gradient$
    end for
end function

Métodos en sklearn y recomendaciones

Métodos en librería Sklearn:

- **Liblinear (por defecto)**
  - **Método**: Basado en descenso de gradiente.
  - **Uso**: Clasificación binaria y conjuntos de datos pequeños.
  - **Regularización**: Admite L1 y L2.
  - **Recomendado**: Datasets pequeños; interpretación rápida de resultados.
- **Newton-CG**
  - **Método**: Método de Newton modificado con gradiente conjugado.
  - **Uso**: Problemas grandes y multiclase.
  - **Ventaja**: Convergencia precisa.
  - **Recomendado**: Modelos multiclase con muchas características.
- **L-BFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno)**
  - **Método**: Optimización cuasi-Newton.
  - **Uso**: Problemas multiclase y conjuntos de datos grandes.
  - **Ventaja**: Menor uso de memoria que Newton-CG.
  - **Recomendado**: Modelos multiclase con grandes datasets.

- **SAG (Stochastic Average Gradient)**
  - **Método**: Optimización estocástica.
  - **Uso**: Grandes conjuntos de datos con características densas.
  - **Ventaja**: Más eficiente para problemas masivos.
  - **Recomendado**: Datasets grandes y densos.
- **SAGA**
  - **Método**: Similar a SAG con soporte para L1.
  - **Uso**: Datos dispersos (sparse).
  - **Ventaja**: Permite regularización L1.
  - **Recomendado**: Problemas con regularización L1 o datasets dispersos grandes.

Gracias