Minería de datos

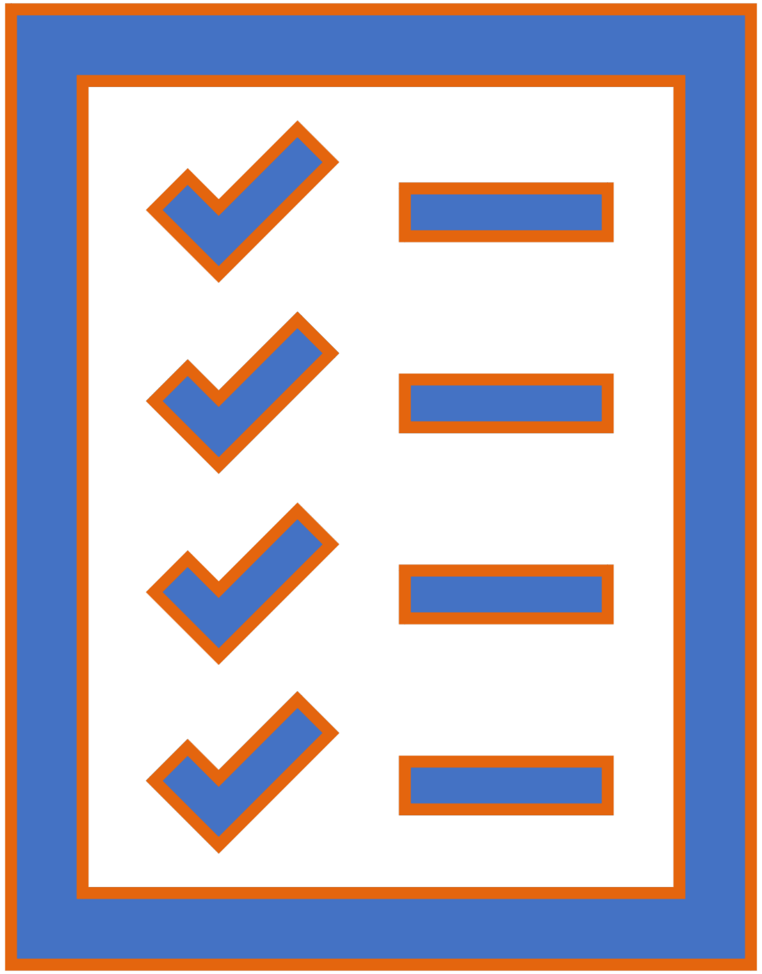
Dr. Jesús Guillermo Falcón Cardona

jfalcon@cua.uam.mx

Evaluación

Técnicas de validación

- Validación simple.
- n-Fold Cross Validation.
- Bootstrapping.



Matriz de confusión

Matriz de costos

Estimado	Real			
		Salida	Observación	UCI
	Salida	0 €	5.000 €	500.000 €
	Observación	300 €	0 €	50.000 €
	UCI	800 €	500 €	0 €

Matriz de confusión

Estimado	Real			
		Salida	Observación	UCI
	Salida	71	3	1
	Observación	8	7	1
	UCI	4	2	3

Análisis Receiver Operating Characteristics (ROC)

Estimado	Real	
	True Positives (TP)	False Positives (FP)
	False Negatives (FN)	True Negatives (TN)

Matriz de confusión normalizada

Estimado	Real	
	TPR	FPR
	FNR	TNR

Regresión lineal y lineal ponderada

Minería de datos

Clase 4: Regresión lineal

Dr. Jesús Guillermo Falcón Cardona
jfalcon@cua.uam.mx

Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA
Unidad Cuajimalpa

Outline

Definición del problema

La tarea del aprendizaje supervisado es la siguiente:

Definition 1

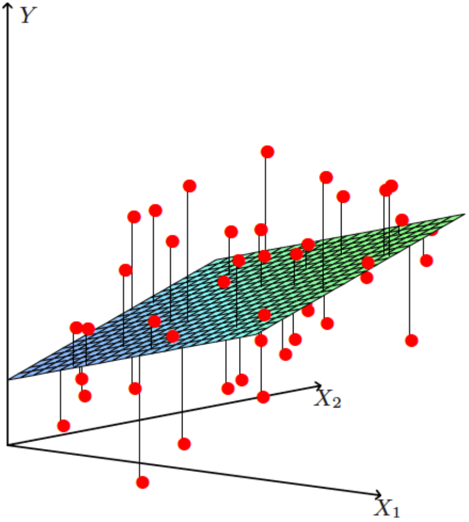
Dado un conjunto de entrenamiento de N instancias (compuestas de entradas y salidas)

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N),$$

donde cada y_j fue generada por una función desconocida $y = f(x)$, descubrir una función h que aproxime la verdadera función f.

Regresión lineal

¿Cuál es la idea detrás?



Modelo de regresión lineal

Dados $\vec{\beta}, \vec{x} \in \mathbb{R}^{m+1}$, el modelo de regresión lineal se establece como:

$$\hat{y} = h_{\vec{\beta}}(\vec{x}) = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m = \vec{\beta}^T \cdot \vec{x}.$$

- m: número de features.
- x_i: i-ésima característica.
- β_i: i-ésimo parámetro del modelo.
- $\hat{y}$: valor predecido.
- $h_{\vec{\beta}}$: función de hipótesis, usando parámetros de modelo $\vec{\beta}$.

Entradas del modelo

Vector de parámetros de ajuste del modelo:

$$\vec{\beta} = (\beta_0, \beta_1, \cdots, \beta_m).$$

Vector de características (features) de entrada:

$$\vec{x} = (x_0 = 1, x_1, \ldots, x_m).$$

β_0 es un valor de sesgo para el modelo.

Conjunto de entrenamiento

De nuestro dataset de N instancias, vamos a tomar $n < N$ elementos para constituir nuestro conjunto de entrenamiento (training set). Las $N - n$ instancias restantes van a formar el conjunto de prueba (test set).

$$\boldsymbol{X}_{\text{training}} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_1 \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix}$$

$$\vec{y}_{\text{labels}} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

¿Cómo lo entrenamos?

Necesitamos una medida que nos indique qué tan bueno o malo es el ajuste del modelo al conjunto de entrenamiento.

$$\mathrm{MSE}(\boldsymbol{X}, h_{\vec{\beta}}) = \frac{1}{n}\sum_{i=1}^{n}\Big(h_{\vec{\beta}}(\vec{x}_i) - y_i\Big)^2,$$

donde $\hat{y}_i = h_{\vec{\beta}}(\vec{x}_i) = \vec{\beta}^T \cdot \vec{x}_i$.

Esta función es el error cuadrático medio. Para este caso la usamos como medida de calidad del modelo y también le llamaremos función de costo.

Dato adicional sobre MSE

MSE también es empleado de forma general para medir la efectividad de modelos de aprendizaje automático. La ocupamos para medir la efectividad sobre el conjunto de prueba, es decir, con las $N - n$ instancias que aún nos quedan del dataset.

Retomando el entrenamiento

$$\vec{\beta}_* = \underset{\vec{\beta} \in \mathbb{R}^{m+1}}{\arg\min}\, \mathsf{MSE}(\boldsymbol{X}, h_{\vec{\beta}})$$

Solución cerrada

De sus clases de álgebra lineal, tenemos:

$$\vec{\beta}_* = (\boldsymbol{X}^T \cdot \boldsymbol{X})^{-1} \cdot \boldsymbol{X}^T \cdot \vec{y}$$

Solución alternativa

El cálculo de la inversa en la ecuación normal nos podría traer muchos problemas (mucho tiempo de cómputo en matrices muy grandes, no existencia de la inversa, problemas de estabilidad numérica) . Por lo tanto, otra opción que tenemos es aplicar el método del Gradiente Descendente. Para ello, necesitamos calcular $\nabla \text{MSE}(\vec{\beta})$.

$$\nabla \text{MSE}(\vec{\beta}) = \frac{2}{n}\boldsymbol{X}^T \cdot (\boldsymbol{X} \cdot \vec{\beta} - \vec{y})$$

Algoritmo

Algorithm 1 Batch Gradient Descent

Establecer learning rate $\alpha \in (0, 1]$
Inicializar $\vec{\beta}_0$
Establecer $j = 0$
while convergencia no alcanzada **do**
    $\vec{\beta}_{j+1} = \vec{\beta}_j - \alpha \nabla f(\vec{\beta})$
    $j = j + 1$
end while
return $\vec{\beta}_j$

Suposiciones de la regresión lineal

1. Linealidad.
2. Normalidad.
3. Independencia.
4. Homocedasticidd.
5. Baja multi-colinealidad.

Homocedasticidad

En general, el modelo lineal establece que $\vec{y} = X\vec{\beta} + \vec{e}$, donde $\vec{e} \in \mathbb{R}^n$ es un vector de errores del modelo que son independientes e idénticamente distribuidos (variables aleatorias normales). Por lo tanto, tenemos:

$$\boldsymbol{C} = E[\vec{e} \cdot \vec{e}^T] = \sigma^2 \boldsymbol{I} = \begin{pmatrix} \sigma^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma^2 & 0 & \dots & 0 \\ 0 & 0 & \sigma^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma^2 \end{pmatrix}$$

El error cometido por el modelo tiene siempre la misma varianza.

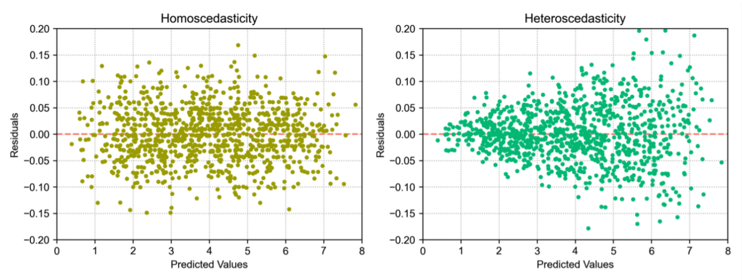
Heterocedasticidad

En algunas aplicaciones, los errores de observación no están uniformemente distribuidos. Entonces, tenemos que $\boldsymbol{C}$ está dada como sigue.

$$\boldsymbol{C} = \begin{pmatrix} \sigma_1^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_n^2 \end{pmatrix}$$

La varianza de los errores no es constante en todas las observaciones realizadas.

Detectar heterocedasticidad



Buscar patrones en los datos

Regresión lineal ponderada

$$\vec{\beta}_* = \arg\max_{\vec{\beta}} \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{C}|}} e^{\left(-\frac{1}{2}(\vec{y}-\boldsymbol{X}\vec{\beta})^T C^{-1} (\vec{y}-\boldsymbol{X}\vec{\beta})\right)}$$

Después de mucha álgebra...

$$\vec{\beta}_* = (\boldsymbol{X}^T \boldsymbol{C}^{-1} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{C}^{-1} \vec{y}$$

Si $w_i = 1/\sigma_i^2, \forall i = 1, 2, \ldots, n$, entonces $\boldsymbol{W} = \boldsymbol{C}^{-1}$ y, por lo tanto, tenemos:

$$\vec{\beta}_* = (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W} \vec{y}$$

De vuelta al MSE

$$\mathsf{MSE}(\vec{\beta}) = \frac{1}{n} \sum_{i=1}^{n} w_i (\vec{\beta}^T \cdot \vec{x}_i - y_i)^2$$

Después de algo de álgebra...

$$\nabla \mathsf{MSE}(\vec{\beta}) = \frac{2}{n} \boldsymbol{X}^T \boldsymbol{W} \cdot (\boldsymbol{X} \vec{\beta} - \vec{y})$$

Queda algo pendiente...

¿Cómo se obtiene $\boldsymbol{C}$ y por lo tanto $\boldsymbol{W}$?

Habrá que aproximar $\boldsymbol{C}$.

1. Aplicar la regresión lineal clásica.
2. Calcular los residuos $r_i = y_i - \hat{y}_i$
3. Definir $c_{ii} = r_i^2, \forall i = 1, 2, \ldots, n$.
4. Entonces, $w_{ii} = 1/r_i^2$.

¿Para qué nos sirve la regresión lineal ponderada?

Resistencia a los outliers y heterocedasticidad