Regresión Lineal. Árboles y Reglas de Regresión

Aprendizaje Automático

Ingeniería Informática

Fernando Fernández Rebollo y Daniel Borrajo Millán

Grupo de Planificación y Aprendizaje (PLG)
Departamento de Informática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

27 de febrero de 2009

En Esta Sección:

Regresión

- Un proceso de regresión lineal es aquel en el que se intenta aproximar una función $f(x)$ (supuestamente lineal) con una función lineal $\hat{f}(x)$.

$$\hat{f}(\vec{x}) = w_0 + w_1 a_1(\vec{x}) + w_2 a_2(\vec{x}) + \cdots + w_n a_n(\vec{x}) \qquad (1)$$

  donde $a_i(\vec{x})$ denota el atributo i-ésimo del ejemplo $\vec{x}$
- El objetivo de la regresion es minimizar el error entre la función aproximada y el valor de la aproximación
- Una posible medida de error es la suma del error cuadrático sobre el conjunto de entrenamiento total, D:

$$E = \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2 \qquad (2)$$

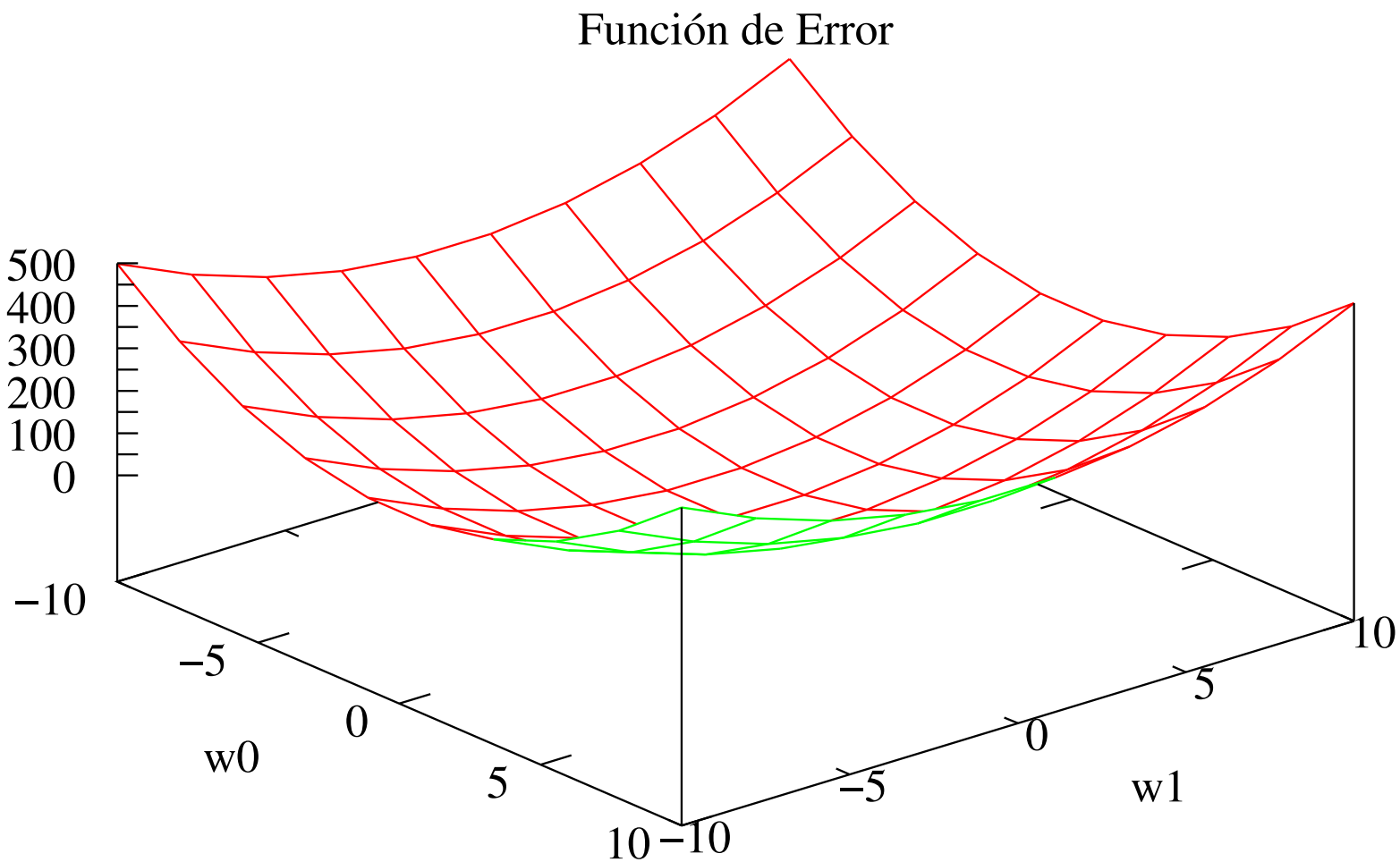
Minimizando el Error

- El problema de definir la función

$$\hat{f}(\vec{x}) = w_0 + w_1 a_1(\vec{x}) + w_2 a_2(\vec{x}) + \cdots + w_n a_n(\vec{x})$$

  se traslada a un problema de definir el vector de pesos $\vec{w}$
- Distintos vectores de pesos dan distintos valores en la función de error
- Se debe encontrar el vector $\vec{w}$ que minimice la función de error: problema de búsqueda en el espacio de pesos
- Aproximación: descenso de gradiente

La función de Error



Algoritmo de Descenso de Gradiente

- Gradiente del error respecto a $\vec{w}$:

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right]$$

- Regla de entrenamiento:

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

i.e.,

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Derivada del Error

$$
\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i}\frac{1}{2}\sum_d (t_d - o_d)^2 \\
&= \frac{1}{2}\sum_d \frac{\partial}{\partial w_i}(t_d - o_d)^2 \\
&= \frac{1}{2}\sum_d 2(t_d - o_d)\frac{\partial}{\partial w_i}(t_d - o_d) \\
&= \sum_d (t_d - o_d)\frac{\partial}{\partial w_i}(t_d - \vec{w}\cdot\vec{x_d}) \\
\frac{\partial E}{\partial w_i} &= \sum_d (t_d - o_d)(-x_{i,d})
\end{aligned}
$$

Algoritmo de Descenso de Gradiente

Descenso de Gradiente($ejemplos\_entrenamiento, \eta$)

Cada ejemplo de entrenamiento es un par de la forma $\langle \vec{x}, t \rangle$, donde $\vec{x}$ es el vector de valores de entrada, y t es el valor de salida objetivo. η es el ratio de aprendizaje.

- Inicializar cada w_i a algún valor aleatorio pequeño
- Hasta que la condición de fin sea alcanzada, hacer
  - Inicializar cada Δw_i a cero.
  - Para cada $\langle \vec{x}, t \rangle$ en $ejemplos\_entrenamiento$, hacer
    - Calcular el valor $o = \hat{f}(\vec{x})$ para la instancia de entrada $\vec{x}$
    - Para cada peso w_i, hacer

$$\Delta w_i \leftarrow \Delta w_i + \eta(t - o)x_i$$

  - Para cada peso, w_i, hacer

$$w_i \leftarrow w_i + \Delta w_i$$

Bibliografía

- Machine Learning, Tom Mitchell. Capítulos 4 y 8

M5 (Quinlan, 93)

- Muchas veces las clases son numéricas y continuas
- Tradicionalmente, se ha utilizado la regresión cuando esto ocurría, pero los modelos obtenidos eran numéricos
- M5 genera árboles de decisión similares a los producidos por ID3
- M5 es una variación de CART (Breiman et al., 84)
  - Las hojas en CART son valores numéricos en lugar de modelos lineales
  - CART elige aquél atributo que maximice la reducción esperada en varianza o en desviación absoluta

Algoritmo

1. Construir el modelo (árbol de decisión con modelos lineales de clases)
2. Estimar el error
3. Construir modelos lineales en cada nodo intermedio del árbol
4. Simplificar los modelos lineales
5. Podar nodos
6. Suavizar

Características de M5

- Heurística: minimizar la variación interna de los valores de la clase dentro de cada subconjunto
- Medida concreta: elegir aquél atributo que maximice la reducción del error, de acuerdo a la siguiente fórmula:

$$\Delta\text{error} = sd(E) - \sum_i \frac{\mid E_i \mid}{\mid E \mid} \times sd(E_i)$$

  - E es el conjunto de ejemplos en el nodo a dividir,
  - E_i son los ejemplos con valor i del atributo a considerar, y
  - $sd(C)$ es la desviación típica de los valores de la clase para los ejemplos en C

Características de M5

- Heurística: minimizar la variación interna de los valores de la clase dentro de cada subconjunto
- Medida concreta: elegir aquél atributo que maximice la reducción del error, de acuerdo a la siguiente fórmula:

$$\Delta\text{error} = sd(E) - \sum_i \frac{\mid E_i \mid}{\mid E \mid} \times sd(E_i)$$

  - E es el conjunto de ejemplos en el nodo a dividir,
  - E_i son los ejemplos con valor i del atributo a considerar, y
  - $sd(C)$ es la desviación típica de los valores de la clase para los ejemplos en C

Características de M5

- Hojas: se calcula un modelo lineal utilizando regresión estándar en función de los valores de los atributos, que proporciona un valor numérico (clase predecida)
- Criterio de parada en cada nodo: pocos ejemplos, o poca variación de los valores de la clase

Características de M5

- Hojas: se calcula un modelo lineal utilizando regresión estándar en función de los valores de los atributos, que proporciona un valor numérico (clase predecida)
- Criterio de parada en cada nodo: pocos ejemplos, o poca variación de los valores de la clase

Estimación del error

- Para estimar el error en posteriores instancias, calcula la media del error residual producido al clasificar con el modelo creado, m, cada instancia del conjunto de test I:

$$e(I, m) = \frac{1}{n} \sum_{i \in I} \|c(i) - c(m, i)\|$$

  - $n = \mid I \mid$,
  - $c(i)$ es la clase de la instancia i, y
  - $c(m, i)$ es la clasificación con el modelo m de la instancia i

Estimación del error

- Como esto subestima el error en posteriores instancias, se multiplica por (ν es el número de atributos en el modelo m):

$$\frac{n+\nu}{n-\nu}$$

- Esto consigue incrementar el error en modelos construidos con muchos parámetros y pocas instancias.

Siguientes pasos

- Construcción de modelos lineales: se calculan para cada nodo del árbol, considerando sólo los atributos que aparecen en su subárbol como test o en modelos lineales
- Simplificación de los modelos lineales: en cada modelo lineal se eliminan atributos, utilizando escalada, para reducir el error estimado. Esto, normalmente, hace que aumente el error residual, pero también reduce el factor por el que luego se multiplica. Puede llegar a dejar sólo una constante
- Poda: cada nodo interno del árbol tiene ahora un modelo simplificado lineal y un modelo subárbol. Se elige aquél que minimice el error. Si es el modelo lineal, el subárbol se queda reducido a ese nodo
- Suavizar el árbol: se tienen en cuenta los demás modelos desde el nodo hoja al nodo raíz

Siguientes pasos

- Construcción de modelos lineales: se calculan para cada nodo del árbol, considerando sólo los atributos que aparecen en su subárbol como test o en modelos lineales
- Simplificación de los modelos lineales: en cada modelo lineal se eliminan atributos, utilizando escalada, para reducir el error estimado. Esto, normalmente, hace que aumente el error residual, pero también reduce el factor por el que luego se multiplica. Puede llegar a dejar sólo una constante
- Poda: cada nodo interno del árbol tiene ahora un modelo simplificado lineal y un modelo subárbol. Se elige aquél que minimice el error. Si es el modelo lineal, el subárbol se queda reducido a ese nodo
- Suavizar el árbol: se tienen en cuenta los demás modelos desde el nodo hoja al nodo raíz

Siguientes pasos

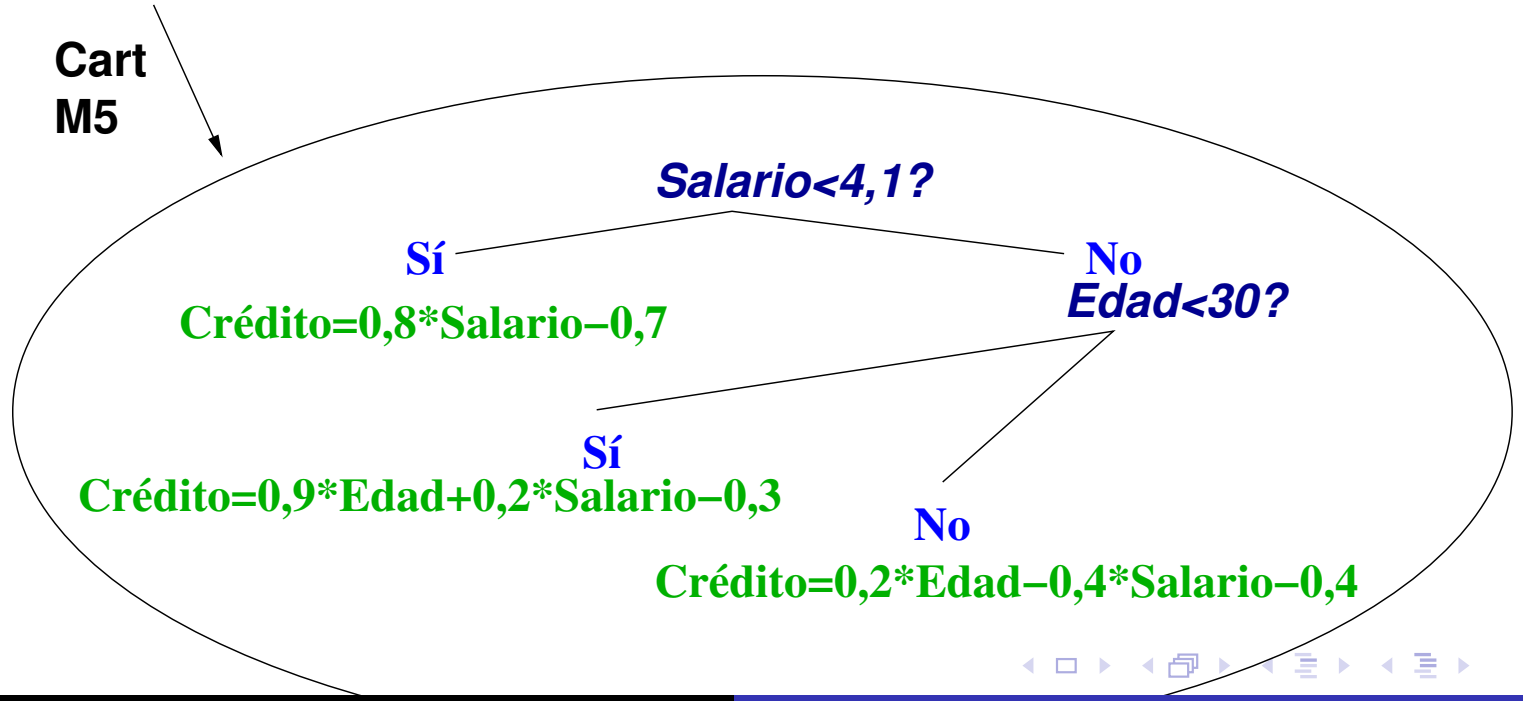
- Construcción de modelos lineales: se calculan para cada nodo del árbol, considerando sólo los atributos que aparecen en su subárbol como test o en modelos lineales
- Simplificación de los modelos lineales: en cada modelo lineal se eliminan atributos, utilizando escalada, para reducir el error estimado. Esto, normalmente, hace que aumente el error residual, pero también reduce el factor por el que luego se multiplica. Puede llegar a dejar sólo una constante
- Poda: cada nodo interno del árbol tiene ahora un modelo simplificado lineal y un modelo subárbol. Se elige aquél que minimice el error. Si es el modelo lineal, el subárbol se queda reducido a ese nodo
- Suavizar el árbol: se tienen en cuenta los demás modelos desde el nodo hoja al nodo raíz

Siguientes pasos

- Construcción de modelos lineales: se calculan para cada nodo del árbol, considerando sólo los atributos que aparecen en su subárbol como test o en modelos lineales
- Simplificación de los modelos lineales: en cada modelo lineal se eliminan atributos, utilizando escalada, para reducir el error estimado. Esto, normalmente, hace que aumente el error residual, pero también reduce el factor por el que luego se multiplica. Puede llegar a dejar sólo una constante
- Poda: cada nodo interno del árbol tiene ahora un modelo simplificado lineal y un modelo subárbol. Se elige aquél que minimice el error. Si es el modelo lineal, el subárbol se queda reducido a ese nodo
- Suavizar el árbol: se tienen en cuenta los demás modelos desde el nodo hoja al nodo raíz

Ejemplo de salida

Salario	Cliente	Edad		Hijos	Crédito
4,5	1	34	...	1	12,3
2,3	0	27	...	2	14,4
9,5	0	51	...	2	4,6
1,2	1	29	...	3	21,7
...	...	...	...	...	...



Resumen

- Árboles de Decisión
- Criterio de división de hojas basada en la entropía
- Generación de reglas de decisión a partir de los árboles
- Aspectos metodológicos
- Árboles de regresión

Bibliografía

- Machine Learning, Tom Mitchell. Capítulo 3
- Data Mining: Practical Machine Learning Tools and Techniques. Ian H. Witten, Eibe Frank. Capítulo 6