UNIVERSIDAD DE LOS LAGOS

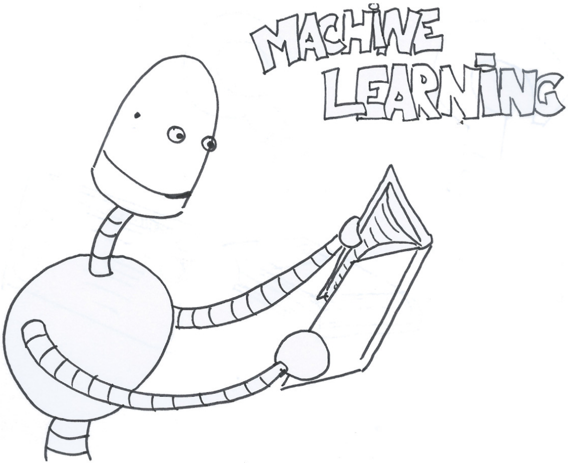
Electivo III: Machine Learning

Problema de Overfitting

Joel S. Torres

Departamento de Ciencias de la Ingeniería
Ingeniería Civil Informática

MACHINE LEARNING



Overfitting

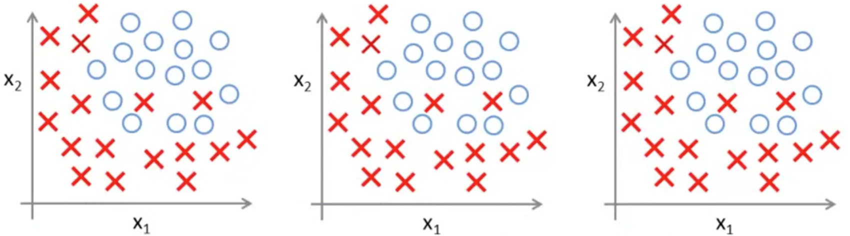
Volvamos un poco al modelo de regresión lineal

$\theta_0 + \theta_1 x$ $\qquad$ $\theta_0 + \theta_1 x + \theta_2 x^2$ $\qquad$ $\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Si existen demasiados atributos, el modelo puede ajustarse excesivamente bien al conjunto de entrenamiento ($J(\theta) \approx 0$), pero falla al generalizar las predicciones a nuevos datos de entrada.

Overfitting

Si vemos el mismo fenómeno en clasificación



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

(g = sigmoid function)

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2$
$+\theta_3 x_1^2 + \theta_4 x_2^2$
$+\theta_5 x_1 x_2)$

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2$
$+\theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2$
$+\theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$

Overfitting

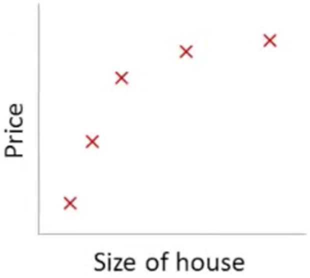
Opciones para evitar este problema

1. Reducir el número de atributos
   - Seleccionar un conjunto manualmente
   - Utilizar un modelo de selección
2. Regularización
   - Mantiene los atributos, pero reduce la magnitud de los parámetros θ_j
   - Funciona bien con un conjunto de atributos, ya que cada atributo contribuye un poco a predecir la salida

Regularización

Regularizando la función de costo: Intuición

$$\theta_0 + \theta_1 x + \theta_2 x^2$$



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Se penalizan los valores θ

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\theta_3^2 + 1000\theta_4^2$$

$$\theta_3 \approx 0, \theta_4 \approx 0$$

Regularización

¿Qué es Regularización? Se penalizan los valores $\theta_0, ..., \theta_n$

- Es una hipótesis más simple
- Compensa el sobreajuste

Ejemplo: Boston Housing

- Atributos: $x_1, ..., x_m$ (Boston: 13 atributos)
- Parámetros: $\theta_0, ..., \theta_n$

$$J(\theta) = \frac{1}{2m}[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2]$$

El simbolo λ es el parámetro de regularización

- Permite que el modelo se ajuste adecuadamente
- Minimiza los parámetros
- Es un valor de penalización (si es demasiado grande, entonces el modelo no se ajustará)

Regularización

Algorithm 1: Descenso de Gradiente

Result: Encuentra $\min\limits_{\theta_0,\theta_1} J(\theta_0, \theta_1)$

repeat

$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$;

$\theta_j := \theta_j - \alpha[\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}) * x^{(i)} + \frac{\lambda}{m}\theta_j], j \in \{1, ..., n\}$;

until *Converger*;

Regularización

Algorithm 2: Descenso de Gradiente

Result: Encuentra $\min_{\theta_0,\theta_1} J(\theta_0, \theta_1)$

repeat

$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$;

$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) * x^{(i)}, j \in \{1, ..., n\}$;

until *Converger*;

$$1 - \alpha \frac{\lambda}{m} < 1$$

Ecuación Normal

$$\theta = (X^T + \lambda L)^{-1} X^T y$$

donde,

$$L = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \dots & \\ & & & & 1 \end{bmatrix}$$

Siguiente Clase

- Redes Neuronales
- Aprendizaje No Supervisado