Tema 02. Regresión y clasificación

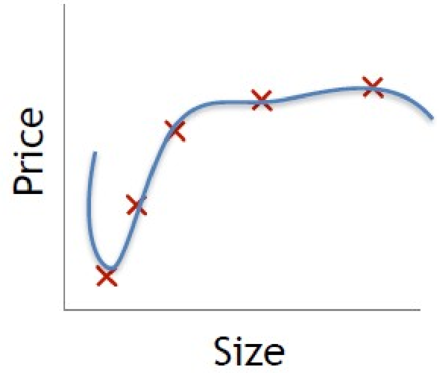
Autor: Ismael Sagredo Olivenza

2.4. Regularización

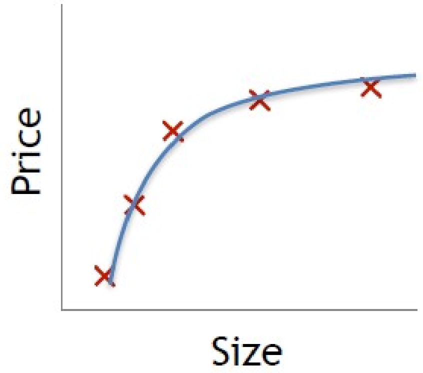
- Is a technique used in machine learning to avoid **overtitting** of models.
- A model fits too closely to the training data and loses the ability to generalise.
- Regularising models helps us to reduce model complexity and avoid overfitting.

2.4.1 Overfitting and how addressing

- Collect more training examples
- Select manually features => useful features could be lost
- Regularization => reduce de size of certain parameters of W



$$f(x) = 28x - 385x^2 + 39x^3 - 174x^4 + 100$$



$$f(x) = 13x - 0.23x^2 + 0.000014x^3 - 0.000x^4 + 10$$

2.4.2 Regularization benefits

- Keep all the features, but reduce magnitude of parámeters
- Works well when we have a lot of features, each of which contributes a bit to predicting
- It is necessary to reduce variance and increase bias.

2.4.3 Cost Function

- The intuition is to reduce weights in order to reduce the complexity of the model
- Let's imagine that a lnear regression has generated the following model

$$w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$$

- but the best model is

$$w_1 x + w_2 x^2 + b$$

We need minimising w_3 and w_4 to close to zero

To reduce the cost function we add a panalty function $p(w)$ to $J(\vec{w}, b)$

We used $\lambda > 0$ as a learning rate. The parameter is a hiperparameter of the model (similar to α in gradient descent)

$$J(w, b) = \frac{1}{2m} \cdot \sum_{i=1}^{m} (f_{w,b}(x_i) - y_i^2) + \lambda p(w)$$

2.4.4 Regularization L1 (Lasso)

The L1 regularisation adds a penalty function proportional to the sum of the absolute values of the model coefficients.

$$p(w) = \sum_{j=1}^{m} |w_j|$$

- This has the effect of **forcing some coefficients to zero**, which implies simpler models.

2.4.5 Regularization L2 (Ridge)

The L2 regularisation adds a penalty term proportional to the sum of the squares of the model coefficients.

$$p(w) = \sum_{j=1}^{m} w_j^2$$

- This has the effect of reducing the values of the coefficients, which can help to avoid over-adjustment, but they will never become zero.

2.4.6 Regularization in linar regression (L2)

$$\min_{\vec{w},b} J(\vec{w},b) = \min_{\vec{w},b} \left[\frac{1}{2m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2 \right]$$

Gradient descent

repeat {

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w},b) \qquad = \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} w_j$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w},b) \qquad = \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)})$$

don't have to regularize *b*

2.4.6.1 Simplified notation

$$J(w,b) = \frac{1}{2m} \cdot \sum_{i=1}^{m} (y_i' - y_i)^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2$$

$$w_j = w_j - \alpha [\frac{1}{m} \sum_{i=1}^{m} (y_i\prime - y_i) * x_i + \frac{\lambda}{m} w_j]$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} (y_i\prime - y_i)$$

where n = number of features, m = number of examples and
$y' = f_{\vec{w},b}(\vec{x}_i) = \vec{w}\vec{x} + b$

2.4.7 Regularization in logistic regression

$$J(\overrightarrow{w}, b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}log(f_{\overrightarrow{w},b}(\overrightarrow{x}^{(i)})) + (1 - y^{(i)})log(1 - f_{\overrightarrow{w},b}(\overrightarrow{x}^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

Gradient descent

repeat {

$$w_j = w_j - \alpha\frac{\partial}{\partial w_j}J(\overrightarrow{w}, b) \quad \Bigg| \quad = \frac{1}{m}\sum_{i=1}^{m}(f_{\overrightarrow{w},b}(\overrightarrow{x}^{(i)}) - y^{(i)})x_j^{(i)} + \frac{\lambda}{m}w_j$$

$$b = b - \alpha\frac{\partial}{\partial b}J(\overrightarrow{w}, b) \quad \Bigg| \quad = \frac{1}{m}\sum_{i=1}^{m}(f_{\overrightarrow{w},b}(\overrightarrow{x}^{(i)}) - y^{(i)})$$

} simultaneous updates

$$f_{\overrightarrow{w},b}(\overrightarrow{x}) = \frac{1}{1 + e^{-(\overrightarrow{w}\cdot\overrightarrow{x}+b)}}$$

2.4.7.1 Simplified notation

$$J(\vec{w}, b) = -\frac{1}{m}\sum_{i=1}^{m}[y_i log(y_i') + (1 - y_i) log(y_i')] + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

$$w_j = w_j - \alpha[\frac{1}{m}\sum_{i=1}^{m} y_i' - y_i x_{j,i} + \frac{\lambda}{m}\sum_{j=1}^{n} w_j]$$

$$b = b - \alpha\frac{1}{m}\sum_{i=1}^{m} y_i' - y_i$$

where n = number of features, m = number of examples and
$y' = f_{\vec{w},b}(\vec{x}_i) = \frac{1}{1+e^{-(\vec{w}\vec{x}+b)}}$