MACHINE LEARNING

Ludovico Bessi

July 2018

Contents

1 Week 1

1.1 Linear regression

Suppose we have this data: feet and price of some houses. We want to be able to predict the price of the next house knowking only the feet. The best way to do that is using linear regression:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

We want to find θ_i such that the line behaves well with the data. This is the same as minimizing the following function, called cost function:

$$J(\theta_0, \theta_1) = \frac{1}{2n}\sum_{i=1}^{n}(h_\theta(x_i) - y_i)^2$$

We can minimize that function using "Gradient descent". We need to repeat the following operation until convergence:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

This is the same as saying:

$$\theta_j := \theta_j - \alpha * \frac{1}{n} \sum_{i=1}^{n} (h_\theta(x^{(i)} - y^{(i)}) x_j^{(i)}$$

The intuition behind this is that at the minimum the second term goes to 0 as the derivative goes to 0. α is called the learnig rate, and it is decided by the user.

2 Week 2

2.1 Multivariate linear regression

Suppose now that we have multiple information about houses. Then, we want to find a function like this:

$$h_\theta(x) = \sum_{i=1}^{n} \theta_i x_i$$

We can vectorize that function, obtaining just:

$$h_\theta(x) = \theta^T x$$

The gradient descent method does not change. However, there is a way to speed things up. We can normalize the θ_i to have them all in the same value range. It is helpful because we minimize the risk of oscillations between values. We just need to do the following:

$$x_i := \frac{x_i - \mu_i}{s_i}$$

Where μ_i is the average and s_i is the standard deviation.

2.2 Alternative to Gradient descent: Normal equation

Gradiend descent is an iterative method, it is good even when we have many features. However it is still iterative, meaning that in principle we may spend a lot of time computing if we choose the wrong α.

We can then use the normal equation if n is not too big:

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

We need to be careful for two reasons: it may be the case that $X^T X$ is not invertible, because there are linearly dependent features, or too many of them. Moreover, calculating θ in this way costs $O(n^3)$.

3 Week 3

3.1 Linear classification

Suppose we get an e-mail. We want to understand whether or not it is spam. We need a function that takes as input all the features of said email, and outputs either 0 or 1. We use the sigmoid function to achieve this. Let $g : \mathbf{R} \to (0, 1)$ be the function $g(z) = \frac{1}{1+e^{-z}}$. Then we can write $h_\theta(x) = g(\theta^T x)$. For example, if we have $h_\theta(x) = 0.7$, this means that we 70% of times the output will be 1. We define y = 1 if $h_\theta(x) \geq 0.5$ and y = 0 otherwise.

3.2 Logistic regression model

It is evident that we cannot use the same cost function as the regression model. Instead, we need to use the following:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

, with $Cost(h_\theta(x^{(i)}), y^{(i)}) = -ylog(h_\theta(x)) - (1-y)log(1-h_\theta(x))$ We then have the vectorized equation, with $h = g(X\theta)$:

$$J(\theta) = \frac{1}{m}(-y^T log(h) - (1-y)^T log(1-h))$$

3.3 Multiclass classification

We now treat the case with $y \in \{0, 1, 2, ..., n\}$. We divide the problem in n+1 sub problems of binary classification, the prediction will be the maximum value of $h_\theta(x)$.

3.4 Overfitting

If we try to fit too many features, we end up with a function that cannot predict well new results. To avoid it, we need to choose which features to analyze or to reduce the magnitude of the parameters. Let's see this last point in more detail: we need to change the cost function to avoid overfitting. Suppose we have:

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

We do not want to ditch the last terms, but we want to reduce their influence. The new cost function will then be like this:

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

When the cost function goes to 0, inevitably the terms θ_3 and θ_4 must go to 0 as well. We can rewrite it with λ as regularization parameter:

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

We must be careful with the choiche of λ, if it is too large we could risk underfitting, moreover the Gradient descent method won't converge. If it is too little, we do not remove overfitting.

3.5 Regularized linear regression

$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \frac{\lambda}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

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

3.6 Regularized logistic regression

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