Logistic Regression Problems

Huy Tr.

October 3rd, 2016

In the previous post, we learned **Linear Regression** to predict a continuous value as a linear function of input values. Now we take another problem: **classification**

1 Classification Problems

It's just like the regression problem, except the value y now only takes a small number of discrete values.

For example, with **binary classification**, the y values should only takes 0 (**negative class**) and 1 (**positive class**).

Classification can be found in *spam detection problems*, etc...

2 Logistic Regression

Logistic Regression is the approach to solve classification by ignoring the fact that y is discrete-valued and use the old good **linear regression** algorithm to predict y by x. But we need to change our **hypothesis function** $h_\theta(x)$:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{1}$$

Which:

$$g(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

This called **logistic function** or **sigmoid function**

Notice that $g(z)$ tends towards 1 as $z \to \infty$ and towards 0 as $z \to -\infty$, and $g(z)$ (or $h(x)$ as well) is always bounded between 0 and 1.

We also have the **derivative of the sigmoid function** as follow:

$$g'(z) = \frac{d}{dz}\frac{1}{1 + e^{-z}} = g(z)(1 - g(z)) \tag{3}$$

3 How to fit θ for this logistic regression model?

We endow our classification model with a set of probabilistic assumptions then fit the parameters (θ) via **maximum likelihood**

Let's assume that:

$$\begin{aligned} P(y=1|\ x;\theta) &= h_\theta(x) \\ P(y=0|\ x;\theta) &= 1-h_\theta(x) \end{aligned} \tag{4}$$

Let's rewrite it more compactly:

$$p(y|\ x;\theta) = (h_\theta(x))^y(1-h_\theta(x))^{1-y} \tag{5}$$

Assume we have m training examples, the **likelihood** of the parameters is:

$$\begin{aligned} L(\theta) &= p(\vec{y} \mid X;\theta) \\ &= \prod_{i=1}^{m}(h_\theta(x^{(i)}))^{y^{(i)}}(1-h_\theta(x^{(i)}))^{1-y^{(i)}} \end{aligned} \tag{6}$$

And we have the **log likelihood** $\ell(\theta)$:

$$\begin{aligned} &\ell(\theta) = logL(\theta) \\ &= \sum_{i=1}^{m} y^{(i)}logh(x^{(i)}) + (1-y^{(i)})log(1-h(x^{(i)})) \end{aligned} \tag{7}$$

To maximize the **likelihood**, we use gradient ascent. We have the **derivative** for stochastic gradient ascent:

$$\begin{aligned} \frac{\partial}{\partial\theta_j}\ell(\theta) &= (y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x})\frac{\partial}{\theta_j}\ g(\theta^T x) \\ &= (y-h_\theta(x))x_j \end{aligned} \tag{8}$$

So we have the stochastic gradient ascent rule as follow:

$$\theta_j = \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \tag{9}$$

Note: This may looks similar to LMS update rule, but this is not the same algorithm, because $h_\theta(x^{(i)})$ now defined as a **non-linear function** of $\theta^T x$