Note: Logistic Regression

Sun Zhao

December 26, 2012

1 Classification

In classification problem, we will predict the label for the given data instead of continuous values like regression problem. For example, a classifier may automatically label an email as spam or not spam, an online transaction fraudulent or not fraudulent, or a tumor malignant or benign. In the simplest case, there are only two classes. Usually, we use 0 and 1 to denote the negative class and the positive class separately. Figure1 shows a training data set of tumor classification. When using linear regression algorithm and exclude blue cross example, we get a hypothesis of purple line. Naturally, we define 0.5 as the boundary of malignant or benign, ie, if the hypothesis h(x) is equal to or greater than 0.5, then x is malignant, otherwise benign. Let $h(x_0) = 0.5$, then if x ¿= x_0, x is malignant, otherwise benign. In this case, we get a perfect classifier. However, if the blue cross is included, we may get a hypothesis of green line. And let $h(x_1) = 0.5$, obviously, x_1 is worth than x_0 when classifying tumors. This is because in the binary classification problem, the value of y is always 0 or 1, but the output of linear regression can be larger than 1 or smaller than 0. Logistic regression will use a hypothesis $h_\theta(x)$ whose output is always between 0 and 1.

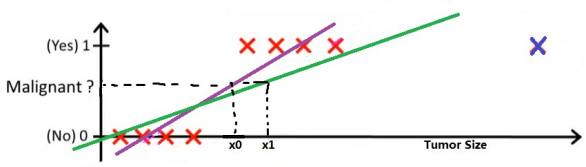

Figure 1:

2 Hypothesis Representation

How to control the output of our hypothesis between 0 and 1? The answer is using a sigmoid function shown in (1). The sigmoid function plotted in Figure2 maps an interval of (-∞, ∞) to (0, 1). Combining the sigmoid function and hypothesis function $h_\theta(x) = \theta^T x$, we will get the hypothesis function of logistic regression shown in (2). The output of the new hypothesis $h_\theta(x)$ means the probability that y = 1 on input of x. For example, in the tumor classification problem, if $h_\theta(x_0) = 0.7$, then we can tell the patient whose tumor size is equal to x_0 that there are 70% chance that her/his tumor to be malignant. Formally, the hypothesis predicts the probability of x being positive class given x and θ. This definition is described in (3), and obviously, we can induce a property shown in (4).

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{2}$$

$$h_\theta(x) = P(y = 1|x; \theta) \tag{3}$$

$$P(y = 0|x; \theta) + P(y = 1|x; \theta) = 1 \tag{4}$$

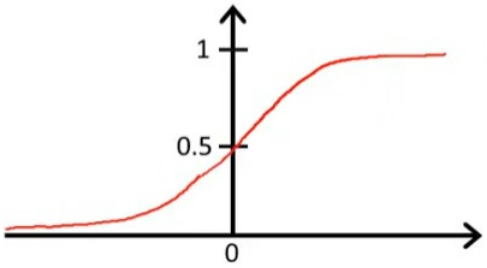

Figure 2:

3 Decision Boundary

Because we judge a sample x whether is positive class by checking whether its hypothesis value $h_\theta(x)$ equal to or greater than 0.5, and note Sigmoid(0) = 0.5, we can use the condition that whether $\theta^T x$ is equal to or greater than 0 to decide its class. If we plot the examples and hypothesis function when given θ, we will find that $\theta^T x = 0$ defines the decision boundary between two classes. Figure3 shows two decision boundaries. In Figure3A, all positive examples are from the up-right direction of the decision line, and in Figure3B, they are all from the outside direction of the decision circle.

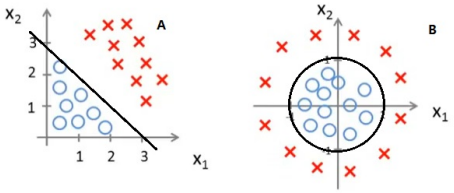

Figure 3:

4 Cost Function

Now we have the logistic regression hypothesis shown in (2), if we use square error cost function as linear regression, the cost function is much complicated and non-convex, hence the gradient descent algorithm may not converge with a global optima. To ensure a convex cost function, we define the cost between $h_\theta(x)$ and y as $Cost(h_\theta(x), y)$ shown in (5).

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if y = 1} \\ -log(1 - h_\theta(x)) & \text{if y = 0} \end{cases} \tag{5}$$

The intuition behind the cost function is that if $h_\theta(x) = 0$, but y = 1 or $h_\theta(x) = 1$, but y = 0, we'll penalize learning algorithm by a very large cost, and if $h_\theta(x) =$ y, the cost is zero. Figure4 shows $-log(h_\theta(x))$ against $h_\theta(x)$ (when y = 1, the cost function will be $-log(h_\theta(x))$, and it is clear that the cost decreases as $h_\theta(x)$ goes 1.
Since y only equal 1 or 0, we can refine $Cost(h_\theta(x), y)$ in a simple form shown in (6).

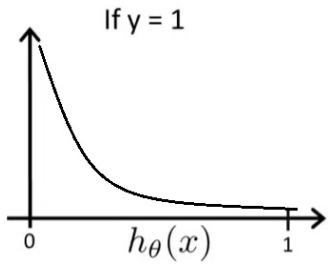

Figure 4:

$$Cost(h_\theta(x), y) = -ylog(h_\theta(x)) - (1 - y)log(1 - h_\theta(x)) \tag{6}$$

Thus, the cost function of logistic regression becomes (7).

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) \tag{7}$$

$$= -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}log(h_\theta(x^{(i)})) + (1 - y^{(i)})log(1 - h_\theta(x^{(i)})) \tag{8}$$

5 Gradient Descent

According to cost function shown in (7) and gradient descent idea, we get the gradient descent algorithm for logistic regression. Algorithm1 shows the update

iteration, and it is cool that the update function looks identical with linear regression! The only difference is the hypothesis function.

Algorithm1: Gradient Descent for Logistic Regression

Repeat{

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

}

6 Multiple Class Classification

Contrasting with binary classification, multiple class classification predicts more than two classes. Here are a list of examples:

- Email tagging: Work, Friends, Family.
- Medical diagrams: Not ill, Cold, Flu.
- Weather: Sunny, Cloudy, Rain, Snow.

The idea of One-vs-All is to train a binary logistic regression classifier $h_\theta^{(c)}(x)$ for each class c(treat examples belonging to c as positive class and examples not belonging to class c as negative class) to predict the probability that x is a member of class c. On a new input of x, its class is predicted as c that maximize $h_\theta^{(c)}(x)$.

7 Summary

Logistic regression solves the classification problem. When using a sigmoid like hypothesis and logarithms cost function, the update iteration is identical to linear regression except the hypothesis. Multiple binary classifiers are trained for each class when solving multiple class classification problem.