Note week3

Xingcheng Yao

Feb. 2019

1 Introduction

Linear regression uses MSE to estimate the error, and the optimization goal is to minimize the MSE.

Logistic regression, even though named 'regression', is a classifier by nature.

2 Mathematical Base

Bayes Formula:

$$P(A|B) = \frac{P(B)P(B|A)}{P(A)}$$

The Exponential Family:

$$p(y:\theta) = b(y)exp(\theta^T T(y) - a(\theta))$$

Assume the distribution of $(y|x:\theta)$ belongs to the exponential family, and y fits the Bernoulli distribution and x conditioned by y fits the Normal distribution, we can know that the expectation of y conditioned by x is a sigmoid function:

$$p_w(y=1|x) = \frac{1}{1+e^{-w^T x}}$$

3 Logistic Regression

Our target is to find a suitable weight matrix W to maximize the likelihood:

$$W \leftarrow \underset{W}{argmax} \prod_{i=1}^{n} p(y_i|x_i:W)$$

Equivalent to:

$$W \leftarrow \underset{W}{argmax} \sum_{i=1}^{n} ln(p(y_i|x_i : W))$$

Then we can define the Log-likelihood:

$$l(W) = \sum_{l} y^l lnP(y^l = 1|x^l; W) + (y^l - 1)lnP(y^l = 0|x^l; W)$$

Since it is a concave function, we can use the gradient descent method to get the the optmized answer. We can also add $L-1$ or $L-2$ penalty to make the model more robust. As for multi-class case, we can use softmax function as the decision funciton.

4 Boosting

Sometimes the learning model like logistic regression, decision trees performs badly in situations like hard learning problems which cause low variance but high bias.Therefore, we often combine several learners to ensure that the ensembled model performs relatively perfect at different parts of the input space. And the output class is the weighted vote of each learner.With the the Hoeffding inequality and the assumption that all the classifiers are independent, We can easily show that the the rate of mistake will decrease exponentially with the increase of the number of base classifiers.

Boosting is a family of algorithms which can convert weak learners into strong learners.

The basic principle of the algorithms is to train a base learner based on the training set. According to the performance of the base learner, we adjust the distribution of the training set such that the mistaken samples will gain more concentration in the later procedure. Then based on the new distribution, we train a new base learner. Eventually when the number of base learners reached an expected number, we take a weighted average of all the learners.

In the process of update, the loss is defined as:

$$\epsilon_k = \frac{\sum_{i=1}^{n} w_i^{k-1} I(y_i \neq h_k(x : w_i^{k-1}))}{\sum_{i=1}^{n} w_i^{k-1}}$$

The weight of each learner is :

$$\alpha_k = \frac{1}{2} ln \frac{1 - \epsilon_k}{\epsilon_k}$$

And the weight matrix is adjusted to:

$$w_i^k = w_i^{k-1} exp(-y_i \alpha_i h(x_i : w_i^{k-1}))$$

To minimize the loss. This method is called Adaboosting. We can also use gradient boosting to make the adjustment.