Week Three

Zhang Yichi, 2018011390
zhangyic18@mails.tsinghua.edu.cn

May 18, 2019

Abstract

In this week's group discussion, Lv Tian introduced the topic of *Logisitic Regression and Boosting*.

1 Logistic Regression

At the beginning, we recalled the linear regression. By using least square method, we can work out a line which can predict the value. Regression involves the problem where input and output are both contiunous variables. From the simple linear regression, we developed the generalized linear model, which is $y = g^{-1}(w^T x + b)$. While the linear regression is used for prediction, what if we want the model to do classification? Since the output of the linear regression model is concrete, we need to transform the output z to $0/1$ by applying unit-step function, which is

$$f(x) = \begin{cases} 0, z < 0 \\ 0.5, z = 0 \\ 1, z > 0 \end{cases}$$

However, the unit-step function is not continuous and cannot be used int the generalized linear model. Therefore, we would like to find a surrogate function which is monotonic and differentiable. We choose **Sigmoid Function**

$$y = \frac{1}{1 + e^{-z}}$$

1.1 Training the Model

If we apply sigmoid function is $g^{-1}()$, we will have $ln\frac{y}{1-y} = w^T x + b$. Here, if we view y as the possibility

that x is positive, then $ln\frac{y}{1-y}$ will represent the log odds. Therefore, we can see that we use the linear regression model to approach the log odds of correct mark.
Remarks:
• although the name is *regression*, this model is actually a classification;
• we design a model only based on the possibility of the classification instead of the data distribution;
• the model doesn't only classify, but also gives the probability prediction.
If we view y as $P(Y=1|X)$, then the expression above can be tured into:

$$P(Y=1|X) = \frac{1}{1+e^{(w_0+\sum_{i=1}^{n} w_i x_i)}}$$

$$P(Y=0|X) = \frac{e^{(w_0+\sum_{i=1}^{n} w_i x_i)}}{1+e^{(w_0+\sum_{i=1}^{n} w_i x_i)}}$$

Then, we can apply maximum likelihood method to estimate w_i and b which can be composed as one n+1-dimension-vector w. Our target becomes to find a suitable W to maximize $\prod_i P(y_i|x_i;W)$ and if we turn it into the form of log, it will become $\sum_i ln(P(y_i|x_i;W))$.
Based on the estimation, we define the log-likelihood as below

$$l(W) = \sum_i y_i ln(P(y_i=1|x_i;W)) + (1-y_i) ln(P(y_i=0|x_i;W))$$

It is obvious that only one of the terms can be non-zero. With some transformation, we can have,

$$l(W) = \sum_l y^l (w_0 + \sum_{i=1}^{n} w_i x_i^l) - ln(1+e^{(w_0+\sum_{i=1}^{n} w_i x_i^l)}$$

By applying Gradient Descent Method, we can estimate W with the training rule:$\Delta w = \eta \bigtriangledown E(w)$
According to the log-likelihood expression, we have $\frac{\partial l(W)}{\partial w_i} = \sum_l = x_i^l(y^l - P(y^l=1|x^l;W))$

1.2 Regularization in Logistic Regression

In order to reduce overfitting effectively, regularization will be introduced to solve the problem.

$$W \leftarrow arg\,max_W \sum_l lnP(y_l|x_l;W) - \frac{\lambda}{2}||W||^2$$

The term $-\frac{\lambda}{2}||W||^2$ works as a penalization of large value of W, therefore there will be a tendency not to overfit.

1.3 Exponential Family of Distributions

Firstly, we define the exponential family. We call a distribution belongs to the exponential family, if it can be shown in the form below.

$$p(y;\eta) = b(y)e^{\eta^T T(y) - a(\eta)}$$

$b(y)$ is a given function;
η is a natural parameter;
$T(y)$ is a sufficient statistic which is often simply $T(y) = y$;
$a(\eta)$ is log normalizer to make sure $\int_{-\infty}^{\infty} p(y;\eta)dy = 1$
With different values of η, we have different distributions. For example, Bernoulli distribution and Normal distribution can both be expressed as exponential distributions.

$$Bernoulli\,distribution:\ f(y;p) = p^y(1-p)^y = e^{(yln\frac{p}{1-p} + ln(1-p))}$$

$$Normal\,distribution:\ f(y;\mu) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(y-\mu)^2} = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}y^2}e^{\mu y - \frac{1}{2}\mu^2}$$

It is interesting that if we assume:
• (y|x;w) ExponentialFamily(η);
• $h_w(x) = E(y|x;w)$;
• $\eta = w^T x$,
then the expectation of a Bernoulli distribution will be $E(y|x;w) = p(y=1|x) = \frac{1}{1+e^{-\eta}} = \frac{1}{1+e^{-w^T x}}$, which is obviously a sigmoid.

2 Boosting

2.1 Ensemble learning

Ensemble learning accomplishes the learning assignment by constructing and combining several learners.
The general structure of ensemble learning is described as follow. First we train a set of individual learners, then we combine them together with some certain strategies. A good ensemble learner should possess individual learner which are accurate and diverse. If an ensemble learner is composed of several individual learners of the same kind, we call them base learner. Otherwise, we call them component learner.
Ensemble learning usually can acquire much better generalization ability than individual learner.
Considering a binary classification, we assume that every

base learner has a rate of mistake $\epsilon = P(h_i(x) \neq f(x))$.
$H(x) = sign(\sum_{i=1}^{T} h_i(x))$ represents the correction rate of ensemble learner relys on whether more than half of base classifiers' judgements are correct.
According to Hoeffding inequality,

$$P(H(x) \neq f(x)) = P(m \leq \lfloor \frac{T}{2} \rfloor) \leq P(m \leq \frac{T}{2}) \leq e^{(-\frac{1}{2}T(1-2\epsilon)^2}$$

From the inequality, we can draw the conclusion that the rate of mistake will decrease exponentially while the number of base classifiers increases. However, we assume that the base learners are mutually independent.

2.2 Boosting

Boosting is a family of algorithms which can convert weak learners into strong ones. The process is to train a base learner based on the training set and adjust the distribution of the training set according to the performance of the trained base learner, so that the mstaken samples will be focused on more. By repeating the process above, we will have a set of base learners and get the ensemble learner by taking a weighted average of them.

$$\hat{h(x)} = \alpha_1 h(x; \theta_1) + ... + \alpha_m h(x; \theta_m)$$

2.3 Adaboost Algorithm

Here we talk about the additive model which can be represents as the expression above. We use the linear combinition to minimize the exponential loss function: $\ell_{exp}(H|D = \mathbb{E}_{x\ D}[e^{-f(x)H(x)}]$. Adaboost Algorithm is shown as below.

Input: Training Set $D = (x_1, y_1), ..., (x_m, y_m)$
Base learning algorithm ζ
Training rounds T
Process:
1: $D_1(x) = \frac{1}{m}$
2: **for** t=1,2,...,T **do**
3: $h_t = \zeta(D, D_t)$;

4: $\epsilon_t = P_{x\ D_t}(h_t(x) \neq f(x))$;
5: **if** $\epsilon_t > 0.5$ **then break**
6: $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$;
7: $D_{t+1}(x) = \frac{D_t}{Z_t} \times \begin{cases} exp(-\alpha_t), \ if\, h_t(x) = f(x) \\ exp(\alpha_t), \ if\, h_t(x) \neq f(x) \end{cases}$
$= \frac{D_t(x) exp(-\alpha_t f(x) h_t(x))}{Z_t}$
8: **end for**
Output: $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

Note:
• 1^{st} step is to initialize the distribution of samples;
• 3^{rd} step is to train the learner $h_t(x)$ based on the training set D under the distribution D_t;
• 4^{th} step is to estimate the rate of mistake of this learner;
• 6^{th} step is to determine the weight of this learner;
• 7^{th} step is to update the distribution of samples according to the performance of this learner, Z_t is a regularization factor to make sure it's a distribution.
Here we leave out the mathematical derivation.
From the view of bias-variation, Boosting focuses on minimizing the bias. Therefore, Boosting can construct powerful ensemble learner based on learners with weak generalization ability.