Lecture 3 Report

EnHsien Chou
zex18@mails.tsinghua.edu.cn

Abstract

Reading Notes for Lecture 3, "Logistic regression and boosting"

1 Introduction

First, we consider an easy case — Linear Regression of discrete cases. Recall "Mean Squared Error" in lecture one :

$$E(f;D) = \frac{1}{m}\sum_{i=1}^{m}(\hat{y_i} - y_i)^2$$

And the linear function:

$$y_i = \omega x_i + b$$

We want to minimize MSE. By partial derivative, it is easy to obtain:

$$\omega = \frac{\sum_{i=1}^{m} y_i(x_i - \bar{x})}{\sum_{i=1}^{m} x_i^2 - \frac{1}{m}(\sum_{i=1}^{m} x_i)^2}$$

$$b = \frac{1}{m}\sum_{i=1}^{m}(y_i - \omega x_i)$$

However, what if the input and output are both continuous variables? In this case, we have to do "regression". Be aware that Logistic "regression", is actually a "classifier" by nature :

$$P(Y=1|X)^{1} = \frac{1}{1 + exp(\omega_0 + \sum_{i=1}^{n}\omega_i x_i)}$$

$$P(Y=0|X) = 1 - P(Y=1|X) = \frac{exp(\omega_0 + \sum_{i=1}^{n}\omega_i x_i)}{1 + exp(\omega_0 + \sum_{i=1}^{n}\omega_i x_i)}$$

2 Mathmetical Foundation

- Sigmoid Function: $f(z) = \frac{1}{1+e^{-z}}$
- The Exponentional Family: $p(y;\eta) = b(y)exp(\eta^T T(y) - a(\eta))$

In the above formula, $b(y)$ is a given function; η is the natural parameter; $T(y)$ is the sufficient statistic, often simplified that $T(y) = y$; $a(\eta)$ is the log normalizer to make sure $p(y;\eta)$ is a probability density function.

[1] Using Bayes Formula to calculate conditional probability : $\mathrm{P(B}_i|A) = \frac{P(B_i)P(A|B_i)}{\sum_{j=1}^{n} P(B_j)P(A|B_j)}$

3 Logistic Regression

By the formula mentioned in Section one, we get:

$$ln\frac{P(Y=0|X)}{P(Y=1|X)} = \omega_0 + \sum_{i=1}^{n} \omega_i x_i$$

Recall linear classification in Lecture 2. The above equation has the same mode, which means logistic regression can be taken through the linear classification.

3.1 Estimation Target

Our target is now to find $W = (\omega_1, ..., \omega_n)$ such that:

$$W \leftarrow arg \max_W \Pi_i P(y_i|x_i; W)$$

And is equivalent to:

$$W \leftarrow arg \max_W \Pi_i lnP(y_i|x_i; W)$$

3.2 Log-likelihood and Gradient Descend Method

We define log-likelihood function:

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

Since it is a concave function, we can apply gradient Descend method:

$$\Delta\vec{\omega} = -\eta\nabla F(\vec{\omega})$$

3.3 Regularization and Penalties

Recall Lecture 1, overfitting is a common problem when training classification problem. Therefore, we could add a penalty term: $-\frac{\lambda}{2}||W||^2$ (The constant $\frac{1}{2}$ is multiplied so it is more convenient to calculate gradient.)

Therefore, the gradient descend method becomes:

$$\frac{\partial l(W)}{\partial \omega_i} = \sum_l x_i^l(y^l - \hat{P}(y^l = 1|x^l; W)) - \lambda\omega_i$$

$$\omega_i \leftarrow \omega_i + \eta \sum_l x_i^l(y^l - \hat{P}(y^l = 1|x^i; W)) - \eta\lambda\omega_i$$

4 Boosting

Boosting is the algorithms that transform weak learners into strong learners.

4.1 Adaboost Algorithm

See "Appendix" for algorithm. This is the way to minimize the loss function.

Acknowledgments

Thanks for the speecher, Lv Tian, and every one in our reading group. Additionally, thanks for all the learning materials provided from our leading teacher, Pro. Su Hang.

References

[1] Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

Appendix

Input: Training Set D; A weak learner $h(x, \theta)$; Numbers of learning round T;
Output: A strong learner $H(x, \theta)$

Algorithm 1 Adaboost Algorithm

procedure BOOST(D, h, T)
  Initialize example weights: $\omega_t = \frac{1}{N}$
  for t in range T **do**
    Obtain hypothesis: $h_t = h(x, \theta_t)$
    Compute hypothesis error: $\epsilon_t = \frac{\sum_{i=1}^{n} \omega_i^{t-1} I(y_i \neq h(x_i; \theta^t))}{\sum_{i=1}^{n} \omega_i^{t-1}}$
    Compute hypothesis weight: $\alpha_t = \frac{1}{2} ln \frac{1-\epsilon_t}{\epsilon_t}$
    Update example weights: $\omega_i^t \leftarrow \omega_i^{t-1} exp(-y_i \alpha_t h(x_i; \theta_t))$
  return $H(x, \theta) = sign(\sum_{i=1}^{T} \alpha_i h_i(x))$