02_Linear_Regression_and_Linear_Classification

1. Linear Regression

Regression

In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome' or 'response' variable) and one or more independent variables (often called 'predictors', 'covariates', 'explanatory variables' or 'features').

In regression, we always use squared loss at the criterion

$$
\begin{aligned}
L &= \int (f(x)-y)^2 \cdot p(x,y) dx dy \\
f^* &= \operatorname{argmin} \iint (f-y)^2 p(x,y) dx dy = E(Y(x)) \quad \rightarrow \frac{dL}{df} = 0
\end{aligned}
$$

However, in practical ML scenarios, we do not know p and real f, we need to assume model for f and p

Least-Square Linear Regression

In linear regression, we consider f in a linear function

Without Probability

We can first use a model without probability estimation (the following equation also without interception).

$$
\begin{aligned}
L &= \sum_{i=1}^{N} \left(w^\top x_i - y_i\right)^2 = \|x_w - Y\|^2 \\
w^* &= \frac{\sum x_i y_i}{\sum x_i^2} \quad \left(\frac{dL}{dw} = 0\right)
\end{aligned}
$$

Notes:

the **interception/bias** can be added:

$$f(x)=w^{\top}x+w_0=\left[w^{\top}w_0\right]\cdot\left[\begin{array}{c}x\\1\end{array}\right]$$

Extension to Probabilistic Model

Previous model assume all points has the same probability or the same weight, but in reality it is not always the case.

For example, we can assume a gaussian condition: $p(y|x)=N(y|w^Tx,\sigma^2)$. Which means for a given x, y is a gaussian distribution. (And we also assume, the same w for all x, which means, we can find a linear model).

Two criterion then can be used: m**aximum likelihood and maximum posterior**

Maximum Likelihood

$$\begin{gathered}\prod_{2=1}^{N}N\left(y_i\mid\omega^{\top}x_i,\sigma^2\right)\\ \text{Assume }\sigma\text{ is known}\quad\hat{w}=(X^TX)^{-1}X^TY\\ \text{Assume }w\text{ known}\quad\hat{\sigma}^2=\frac{1}{N}\cdot\Sigma\left(x_i^{\top}w-y_i\right)^2\end{gathered}$$

Maximum Posterior

$$\begin{aligned}&\left(\prod_{i=1}^{N}N\left(y_2\mid w^{\top}x_2,\sigma^2\right)\right)\cdot N(w\mid 0,\alpha I)\\ \hat{w}_{\text{map}}&=\left(x^{\top}x+\frac{\sigma^2}{\alpha}I\right)^{-1}x^{\top}Y\end{aligned}\tag{1}$$

Nonlinear Cases

we can introduce nonlinear relations by introducing **feature transformation** $\phi(x)$. such as $x^2\ sin(x)$ etc. Then we will have criterion like:

$$\sum_{i=1}^{N}\left(w^{\top}\phi\left(x_i\right)-y_i\right)^2$$

2. Linear Classification

Logistic Regression

For bayesian classification, it is hard for us to use optimization perspective to deal with classification, because the gradient may not exist. The **Logistic Regression** aims to solve classification problem as an optimization problem.

Logistic Regression Model

$$p\left(y_1\mid x\right)=\frac{1}{\exp\left(-w^{\top}x_c+w_0\right)+1}=1-y_2(x)$$

Optimization Target: Maximize Likelihood

$$\sum_{x \in y_1} \log_2 \left(\frac{1}{\exp(-f(x)) + 1} \right) + \sum_{x \in y_2} \log_2 \left(\frac{1}{\exp(f(x)) + 1} \right)$$

Optimization Model

$$f^*(x) = \operatorname{argmin}_x \sum_{i=1}^{n} \log_2 \left(\exp\left(-y_i f(x) \right) + 1 \right)$$

Linear Classifier

use $\sum_{i=1}^{M} \left(w^\top x + w_0 - y_i \right)^2$ as criterion

Perceptron

> https://zhuanlan.zhihu.com/p/25696112

Perceptron is a bi-classified linear classification model. It takes feature vector as input and use class as output.

Perceptron

$$f(x) = \operatorname{sign}(w * x + b)$$

Optimization Strategy

- Minimize the number of misclassification points is not tractable
- So we use the sum of the distance between misclassification point and the decision boundary: $\frac{1}{\|w\|} \left| w \cdot x_0 + b \right|$

$$-\frac{1}{\|w\|} y_i \left(w \cdot x_i + b \right)$$

我们知道每一个误分类点都满足 $-y_i \left(w * x_0 + b \right) > 0$

- 因为当我们数据点正确值为 $+1$ 的时候，你误分类了，那么你判断为 -1 ，则算出来 $\left(w * x_0 + b \right) < 0$ ，所以满足 $-y_i \left(w * x_0 + b \right) > 0$
- 当数据点是正确值为 -1 的时候，你误分类了，那么你判断为 $+1$ ，则算出来 $\left(w * x_0 + b \right) > 0$ ， 所以满足 $-y_i \left(w * x_0 + b \right) > 0$

Optimization Model

$$L(w, b) = -\sum_{x_i \in M} y_i \left(w \cdot x_i + b \right)$$

Perceptron vs SVM

/qu

When using SVM, we also consider other target: we want a large margin.

Actually, SVM $\approx$ L2 regularized Perceptron

3. General Setup for Fitting a Classification/Regression Learner

- choose **class of models**: linear functions, Gaussian classes, sigmoid posteriors
- choose a **loss function**: log-likelihood, squared loss, MAP
- Sum over individual training elements

Notes:

- Different Configuration may generate the same formula. For example, logistic regression can be regarded as: sigmoidal posterior+LL or linear model+logistic loss
- Based on the setting, most classifier don't directly minimize error rate

Summary

This chapter we first introduce linear regression and least-square method. Then we extend the regression to probabilistic model, we can use maximum likelihood and maximum posterior to solve it. For nonlinear cases, we can use feature transformation to overcome.

Then we introduce the linear classification problem, mainly the logistic regression method. The logistic regression method use MLE as criterion and use logistic function so that we can use optimization method to solve it.

Finally, we set up the general setup steps for build a classification or regression learner.