Statistical Learning for Classification

Tianhang Zheng

https://tianzheng4.github.io

Quantitative vs Qualitative Outputs

Regression mainly study quantitative outputs

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

Classification mainly study qualitative outputs

Qualitative variables take values in an unordered set, i.e.,
$eye\ color \in \{brown, black, green\}$

Why not Linear Regression

Binary Classification: $Y = \{0, 1\}$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

As long as $\hat{\beta}_1$ is not zero, the prediction could be larger than 1 or smaller than 0

Logistic regression is more appropriate!

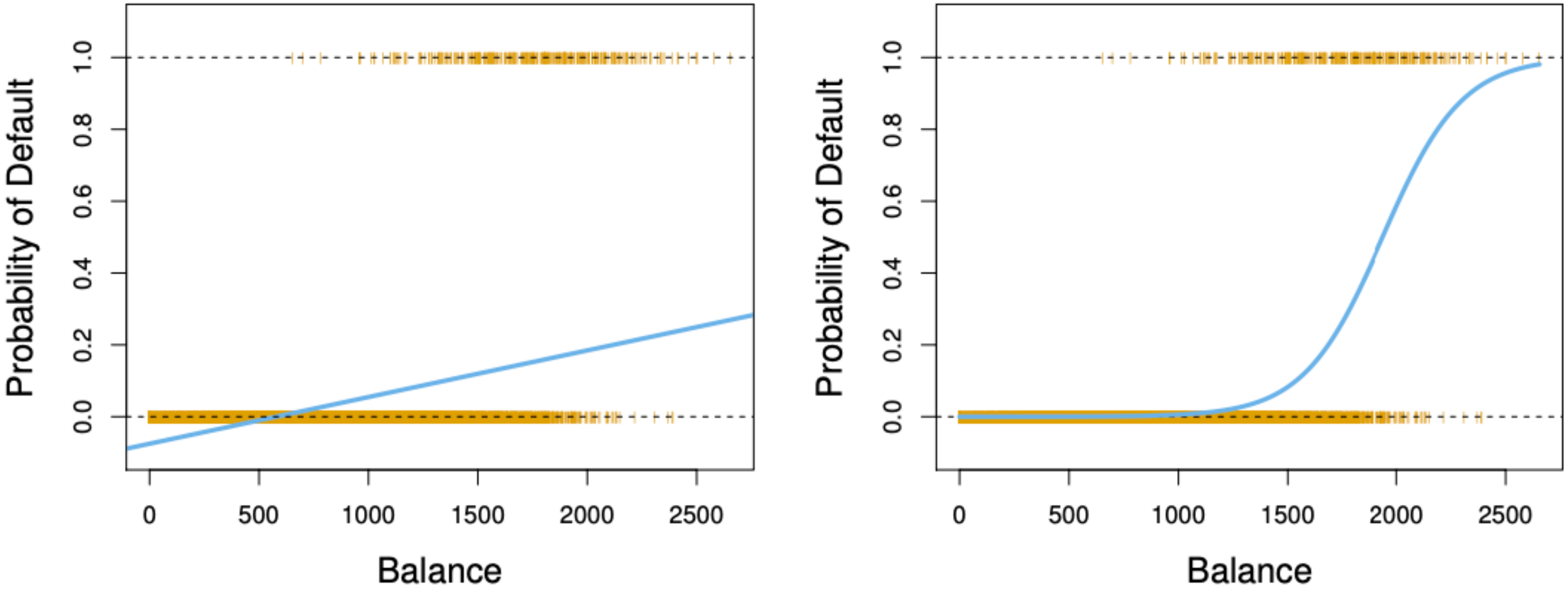
Logistic Regression

$$p(Y = 1|X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

$p(Y = 1|X)$ always has values between 0 and 1

$$\log\left(\frac{p}{1 - p}\right) = \beta_0 + \beta_1 \mathrm{X}$$

Linear Regression vs Logistic Regression



Maximum Likelihood Method

Commonly used for parameter estimation of logistic regression

Assume that the predictors are independent

$$p(y|x) = \prod_{i:y_i=1} p_i \prod_{i:y_i=0} (1 - p_i) \qquad p_i = p(y_i = 1|x_i)$$

This likelihood characterizes the conditional probability of the observed data

Maximum Likelihood Method

$$\max_{\hat{\beta}_0,\hat{\beta}_1} p(D) = \prod_{i:y_i=1} p_i \prod_{i:y_i=0} (1 - p_i)$$

May need to use an optimizer to solve the problem

In practice, we could use sklearn.linear_model.LogisticRegression

Multi-Class Logistic Regression

A linear function for each class

$$p(Y = k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X}}{\sum_{l=1}^{K} e^{\beta_{0l}+\beta_{1l}X}}$$

Multi-class logistic regression is also called multinomial regression

Discriminant Analysis

Model the data distribution for each class as Gaussian distribution

Use the **Bayes theorem** to obtain $p(Y|X)$

$$p(Y = y|X = x) = \frac{p(X = x|Y = y)p(Y = y)}{p(X = x)}$$

Discriminant Analysis

$$p(Y = k|X = x) = \frac{p(X = x|Y = k)p(Y = k)}{p(X = x)}$$

Prior distribution: $\pi_k = p(Y = k)$

Data density for class k: $f_k(x) = p(X = x|Y = k)$

$$p(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

Discriminant Analysis

Model the data distribution for each class as Gaussian distribution

$$f_k(x) = \frac{1}{\sqrt{2\pi\sigma_k}} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2}$$

Use the data from class k to estimate the mean and standard deviation

$$\hat{\pi}_k = \frac{n_k}{n}, \qquad \hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i\,, \qquad \hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \mu_k)^2$$

Discriminant Analysis

Classify x as k with the largest probability $p_k(x)$

$$p_k(x) = p(Y = k|X = x) = \frac{\pi_k \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x-\mu_l}{\sigma}\right)^2}}$$

Discriminant score

$$\delta_k(x) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

Multi-Variable Discriminant Analysis

$$f(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)}$$

Use the data from class k to estimate the mean and covariance

$$\delta_k(x) = x^T \mathbf{\Sigma}^{-1} \mu_k - \frac{1}{2} \mu_k^T \mathbf{\Sigma}^{-1} \mu_k + \log \pi_k$$

Linear or Quadratic Discriminant Analysis

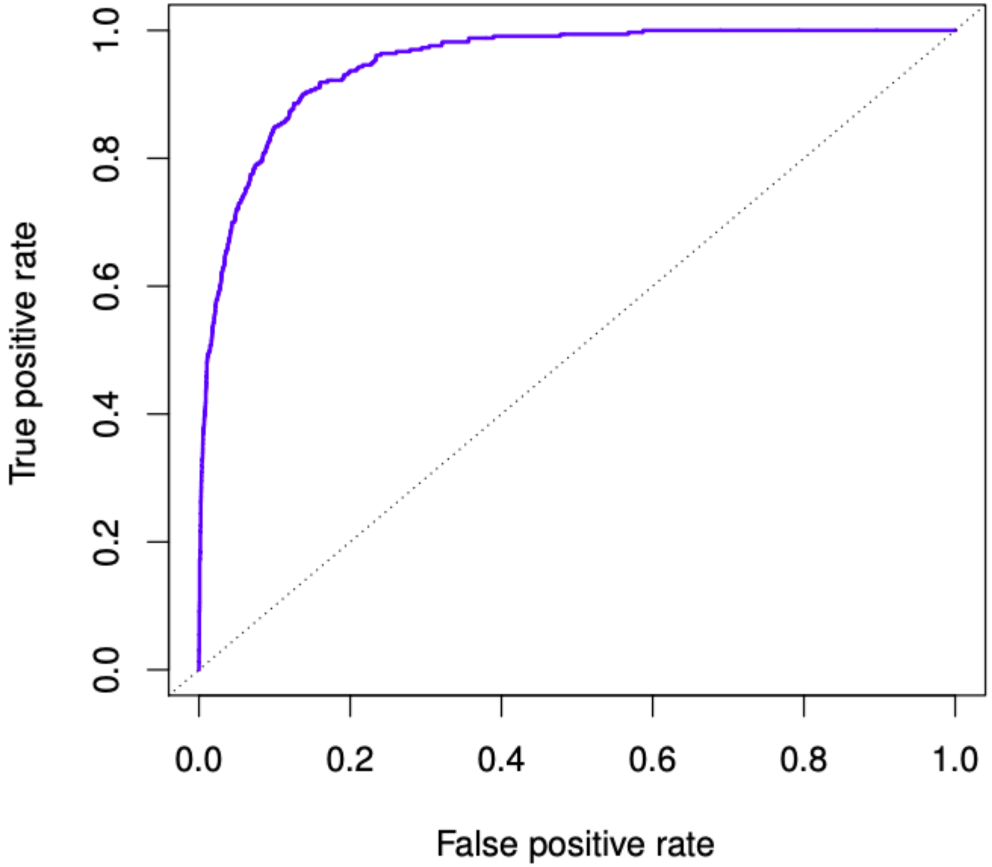
Linear Discriminant Analysis: same estimation of standard deviation or covariance for all the classes

Quadratic Discriminant Analysis: different estimations of standard deviation or covariance for all the classes

ROC Curve

True Positive Rate and False Positive Rate

$$p(Y = 1|X) \geq threshold$$



Logistic Regression vs Linear DA

Linear Discriminant Analysis

$$\log\left(\frac{p_1(x)}{p_2(x)}\right) = \delta_1(x) - \delta_2(x) = \left(\frac{\mu_1}{\sigma^2} - \frac{\mu_2}{\sigma^2}\right) x - \frac{\mu_1^2}{2\sigma^2} + \frac{\mu_2^2}{2\sigma^2} + \log(\frac{\pi_1}{\pi_2})$$

Logistic Regression maximizes conditional likelihood for estimation (discriminative learning)

Linear Discriminant Analysis use full likelihood (generative learning)

Naïve Bayes

Assume the features are independent, which means the covariance matrix is diagonal.

$$f_k(x) = \prod_{j=1}^{p} f_{jk}(x_j)$$

Discriminant score

$$\begin{aligned}\delta_k(x) &\propto \log\left[\pi_k \prod_{j=1}^{p} f_{kj}(x_j)\right] \\ &= -\frac{1}{2}\sum_{j=1}^{p}\left[\frac{(x_j-\mu_{kj})^2}{\sigma_{kj}^2} + \log\sigma_{kj}^2\right] + \log\pi_k\end{aligned}$$

K Nearest Neighbors

A non-parametric supervised learning algorithm for classification

Assign the label of x based on a majority vote mechanism

Select the k training points that are nearest to the target point x

Assign the majority label for the k training points as the label of x

Grid Search for k

To search the best k, we could create a validation dataset

Try different k, and see the performance on the validation dataset

Select the best-performed k (can be done by sklearn)

Summary

Logistic Regression is very popular when K=2

Linear Discriminant Analysis is useful when n is small, K is large and the Gaussian distribution assumption makes sense

Naïve Bayes is useful when p (number of features) is large, and features are not correlated

Q & A