Linear Classifiers

김지수 (Jisu KIM)

통계적 기계학습(Statistical Machine Learning), 2025 1학기

The lecture note is a minor modification of the lecture notes from Prof. Larry Wasserman's "Statistical Machine Learning" and Prof. Yongdai Kim's "Statistical Machine Learning". Also, see Section 4 from [1].

1 Review

1.1 Basic Model for Supervised Learning

- Input(입력) / Covariate(설명 변수) : $x \in \mathbb{R}^d$, so $x = (x_1, \ldots, x_d)$.
- Output(출력) / Response(반응 변수): $y \in \mathcal{Y}$. If y is categorical, then supervised learning is "classification", and if y is continuous, then supervised learning is "regression".
- Model(모형) :

$$y \approx f(x), \qquad f \in \mathcal{M}.$$

  If we include the error ϵ to the model, then it can be also written as

$$y = \phi(f(x), \epsilon).$$

  For many cases, we assume additive noise, so

$$y = f(x) + \epsilon.$$

- Assumption(가정): f belongs to a family of functions $\mathcal{M}$. This is the assumption of a model: a model can be still used when the corresponding assumption is not satisfied in your data.
- Loss function(손실 함수): $\ell(y, a)$. A loss function measures the difference between estimated and true values for an instance of data. The most common ones are:
  - square: $\ell(y, a) = (y - a)^2$.
  - $0 - 1$: $\ell(y, a) = I(y \neq a)$.
- Training data(학습 자료): $\mathcal{T} = \{(y_i, x_i), i = 1, \ldots, n\}$, where (y_i, x_i) is a sample from a probability distribution P_i. For many cases we assume i.i.d., or x_i's are fixed and y_i's are i.i.d..
- Goal(목적): we want to find f that minimizes the expected prediction error,

$$f^0 = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(Y,X)\sim P} \left[\ell(Y, f(X))\right].$$

  Here, $\mathcal{F}$ can be different from $\mathcal{M}$; $\mathcal{F}$ can be smaller than $\mathcal{M}$.

- Prediction model(예측 모형): f^0 is unknown, so we estimate f^0 by $\hat{f}$ using data. For many cases we minimizes on the empirical prediction error, that is taking the expectation on the empirical distribution $P_n = \frac{1}{n}\sum_{i=1}^{n}\delta_{(Y_i,X_i)}$.

$$\hat{f} = \arg\min_{f\in\mathcal{F}} \mathbb{E}_{P_n}\left[\ell(Y, f(X))\right] = \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\ell(Y_i, f(X_i)).$$

- Prediction(예측): if $\hat{f}$ is a predicted function, and x is a new input, then we predict unknown y by $\hat{f}(x)$.

2 Introduction

The problem of predicting a discrete random variable Y from another random variable X is called *classification*, also sometimes called *discrimination*, *pattern classification* or *pattern recognition*. We observe iid data $(X_1, Y_1), \ldots, (X_n, Y_n) \sim P$ where $X_i \in \mathbb{R}^d$ and $Y_i \in \mathcal{Y} = \{0, 1, \ldots, K-1\}$. Often, the covariates X are also called *features*. The goal is to predict Y given a new X; here are some examples:

1. The Iris Flower study. The data are 50 samples from each of three species of Iris flowers, *Iris setosa*, *Iris virginica* and *Iris versicolor*; see Figure 1. The length and width of the sepal and petal are measured for each specimen, and the task is to predict the species of a new Iris flower based on these features.

2. The Coronary Risk-Factor Study (CORIS). The data consist of attributes of 462 males between the ages of 15 and 64 from three rural areas in South Africa. The outcome Y is the presence ($Y = 1$) or absence ($Y = 0$) of coronary heart disease and there are 9 covariates: systolic blood pressure, cumulative tobacco (kg), ldl (low density lipoprotein cholesterol), adiposity, famhist (family history of heart disease), typea (type-A behavior), obesity, alcohol (current alcohol consumption), and age. The goal is to predict Y from all these covariates.

3. Handwriting Digit Recognition. Here each Y is one of the ten digits from 0 to 9. There are 256 covariates $X_1, \ldots, X_{256}$ corresponding to the intensity values of the pixels in a 16×16 image; see Figure 2.

4. Political Blog Classification. A collection of 403 political blogs were collected during two months before the 2004 presidential election. The goal is to predict whether a blog is *liberal* ($Y = 0$) or *conservative* ($Y = 1$) given the content of the blog.

Figure 1: Three different species of the Iris data. *Iris setosa* (Left), *Iris versicolor* (Middle), and *Iris virginica* (Right).

We consider the K-class classification problem as

$$\mathcal{Y} = \{0, 1, \ldots, K-1\}.$$

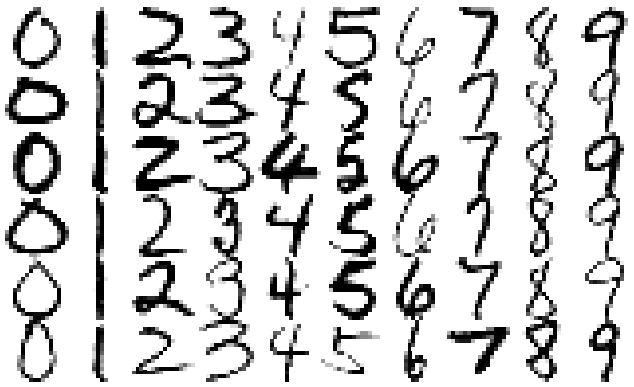

Figure 2: Examples from the zipcode data.

Hence, a *classification rule*, or *classifier*, is a function $f : \mathcal{X} \to \mathcal{Y} = \{0, \ldots, K-1\}$ where $\mathcal{X}$ is the domain of X, e.g., $\mathcal{X} = \mathbb{R}^d$. When we observe a new X, we predict Y to be $f(X)$. For $K = 2$, we have a *binary classification* problem. For $K > 2$, we have a *multiclass classification* problem. To simplify the discussion, we mainly discuss binary classification, and briefly explain how methods can extend to the multiclass case.

Hence the *classification risk*, or *error rate*, of f is defined as

$$R(f) = \mathbb{E}_{(Y,X)\sim P}\left[\ell(Y, f(X))\right],$$

and the *empirical classification error* or *training error* is

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(Y_i, f(X_i)).$$

The usual loss function for classification is the 0-1 loss given as

$$l(y, a) = I(y \neq a).$$

Hence for this case, the classification risk and the empirical classification error become

$$R(f) = \mathbb{P}\big(Y \neq f(X)\big), \qquad \hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} I\big(Y_i \neq f(X_i)\big).$$

2.1 Linear Classifiers

Intuitively, the classification rule f partitions the input space $\mathcal{X}$ into K disjoint *decision regions* whose boundaries are called decision boundaries. In these notes, we consider *linear classifiers* whose decision boundaries are linear functions of the covariate X.

Let the classifier(분류기) $f : \mathbb{R}^d \to \mathcal{Y}$ be returning an output label given input. A *decision boundary* (결정 경계) is the region of a problem space in which the output label of a classifier is ambiguous. In other words, A decision boundary between class i and j is the set of points whose neighbor always intersect both regions classified as label i and label j, i.e.,

$$\{x \in \mathcal{X} : \text{for all } r > 0,\, B(x, r) \cap f^{-1}\{i\} \neq \emptyset,\, B(x, r) \cap f^{-1}\{j\} \neq \emptyset\},$$

where $B(x, r) = \{y \in \mathcal{X} : \|y - x\|_2 < r\}$ is the ball centered at x and radius r. And the decision boundary is the

union over all pairs i and j, i.e.,

$$\bigcup_{1\le i<j\le K} \{x \in \mathcal{X} : \text{for all } r > 0,\ B(x,r) \cap f^{-1}\{i\} \neq \emptyset,\ B(x,r) \cap f^{-1}\{j\} \neq \emptyset\}.$$

The *linear classifier* (선형 분류기) is the case where the decision boundary between class i and j is given as a subset of an affine hyperspace

$$\{x : H(x) = 0\}, \qquad \text{where } H(x) = \beta_0 + x^\top \beta.$$

In this case, $H(x)$ is called a *linear discriminant function*. Specifically, for the linear binary classifier, the classifier f can be expressed as

$$f(x) = I(H(x) > 0).$$

There are three approaches for linear classifier(선형 분류기):

- model joint probability distribution (결합확률분포) : Linear Discriminant Analysis (선형판별분석), Naive Bayes Classifier (나이브 베이즈 분류기)
- model conditional density function $P(\text{class}|X)$: many classifier (분류기) based on regressions (회귀분석), in particular, Logistic Classification (로지스틱 분류) using Logistic Regression (로지스틱 회귀)
- Maximize difference between differeng groups: perceptron (퍼셉트론), support vector machine (서포트 벡터 머신)

This class covers Linear Discriminant Analysis (선형판별분석) and Logistic Classification (로지스틱 분류).

2.2 Bayes Classifier

For the classification so that $y \in \mathcal{Y} = \{0, \dots, K-1\}$, the Risk is

$$\begin{aligned} R(f) &= \mathbb{E}_{(Y,X)\sim P}\left[\ell(Y, f(X))\right] = \mathbb{E}\left[I(Y \neq f(X))\right] \\ &= \mathbb{E}_X \left[\sum_{j=0}^{K-1} \ell(j, f(X)) P(Y = j|X)\right]. \end{aligned}$$

Hence $R(f)$ is minimized when

$$f^0(x) = \arg\min_{k=0,\dots,K-1} \sum_{j=0}^{K-1} \ell(j,k) P(Y = j|X = x).$$

For the classification, the most common loss is the 0-1 loss

$$\ell(y, a) = I(y \neq a).$$

And the optimal prediction function is

$$f^0(x) = \arg\max_{j=1,\dots,J} P(Y = j|X = x).$$

This optimal classifier is called Bayes rule / Bayes classifier (베이즈분류 / 베이즈모형), and the error rate is called the Bayes risk (베이즈위험).

3 Bayes Classifier

Theorem 1. *The rule f that minimizes $R(f)$ is*

$$f^*(x) = \arg\min_{k=0,\dots,K-1} \sum_{j=0}^{K-1} \ell(j,k)\mathbb{P}\left(Y=j|X=x\right), \tag{1}$$

and in particular when $\ell(y,a) = I(y \neq a)$,

$$f^*(x) = \arg\max_{k=0,\dots,K-1} \mathbb{P}\left(Y=k|X=x\right).$$

This optimal rule f^* is called the *Bayes rule / Bayes classifier* (베이즈분류 / 베이즈모형). The risk $R^* = R(f^*)$ is called the *Bayes risk* (베이즈위험). The decision boundary $\{x \in \mathcal{X} : \ m(x) = 1/2\}$ is called the *Bayes decision boundary.*

Proof. Note that the classification risk is

$$R(f) = \mathbb{E}\left[\ell(Y, f(X))\right] = \mathbb{E}_X\left[\mathbb{E}_{Y|X}\left[\ell(Y,f(X))|\, X\right]\right]. \tag{2}$$

Now for each $x \in \mathcal{X}$,

$$\mathbb{E}_{Y|X}\left[\ell(Y,f(X))|\, X=x\right] = \sum_{j=0}^{K-1} \ell(j,f(x))\mathbb{P}\left(Y=j|X=x\right).$$

Hence for each $x \in \mathcal{X}$, this is minimized when

$$f(x) = \arg\min_{k=0,\dots,K-1} \sum_{j=0}^{K-1} \ell(j,k)\mathbb{P}\left(Y=j|X=x\right),$$

and this f is the minizer for (2) as well.

□

Note that for $0-1$ loss and binary classifier,

$$f^*(x) = I\left(\mathbb{P}\left(Y=1|X=x\right) > \frac{1}{2}\right).$$

For $0-1$ loss case, we can rewrite f^* in a different way. Assume X is discrete for simplicity, then from Bayes theorem,

$$\begin{aligned} P(Y=j|X=x)P(X=x) &= P(Y=j, X=x) \\ &= P(X=x|Y=j)P(Y=j) = \pi_j p_j(x), \end{aligned}$$

where $\pi_j = \mathbb{P}(Y=j)$ and $p_j(x) = P(X=x|Y=j)$. And hence

$$\begin{aligned} \mathbb{P}(Y=j|X=x) &= \frac{P(Y=j,X=x)}{P(X=x)} \\ &= \frac{P(Y=j,X=x)}{\sum_{l=0}^{K-1} P(Y=l,X=x)} \\ &= \frac{\pi_j p_j(x)}{\sum_{l=0}^{K-1} \pi_l p_l(x)}. \end{aligned} \tag{3}$$

From the above equality, we have that

$$\mathbb{P}(Y = k|X = x) > \mathbb{P}(Y = j|X = x) \quad \text{is equivalent to} \quad \frac{p_k(x)}{p_j(x)} > \frac{\pi_j}{\pi_k}.$$

Thus the Bayes rule can be rewritten as

$$f^*(x) = k \text{ when } \frac{p_k(x)}{p_j(x)} > \frac{\pi_j}{\pi_k} \text{ for all } j \neq k,$$

or

$$f^*(x) = \arg \max_{j=0,\ldots,K-1} \pi_j f_j(x).$$

If $\mathcal{H}$ is a set of classifiers then the classifier $f^o \in \mathcal{H}$ that minimizes $R(f)$ is the *oracle classifier*. Formally,

$$R(f^o) = \inf_{f \in \mathcal{H}} R(f)$$

and $R_o = R(f^o)$ is called the *oracle risk* of $\mathcal{H}$. In general, if f is any classifier and R^* is the Bayes risk then,

$$R(f) - R^* = \underbrace{R(f) - R(f^o)}_{\text{distance from oracle}} + \underbrace{R(f^o) - R^*}_{\text{distance of oracle from Bayes error}}.$$

The first term is analogous to the variance, and the second is analogous to the squared bias in linear regression.

4 Logistic Classification (로지스틱 분류)

One approach to the classification is to estimate the regression function $m_j(x) = \mathbb{E}(I(Y = j)|X = x) = \mathbb{P}(Y = j|X = x)$ and, once we have an estimate $\hat{m}_j(x)$, use the classification rule

$$\hat{f}(x) = k \qquad \text{if } \hat{m}_k(x) > \hat{m}_j(x) \text{ for all } j \neq k. \tag{4}$$

For binary classification problems, one possible choice is the linear regression model

$$I(Y = k) = m(X) + \epsilon = \beta_0 + \sum_{j=1}^{d} \beta_j X_j + \epsilon.$$

The linear regression model does not explicitly constrain $I(Y = k)$ to take on binary values. A more natural alternative is to use *logistic regression*, which is the most common binary classification method.

Before we describe the logistic regression model, let's recall some basic facts about binary random variables. If Y takes values 0 and 1, we say that Y has a Bernoulli distribution with parameter $\pi_1 = \mathbb{P}(Y = 1)$. The probability mass function for Y is $p(y; \pi_1) = \pi_1^y (1 - \pi_1)^{1-y}$ for $y = 0, 1$. The likelihood function for π_1 based on iid data $Y_1, \ldots, Y_n$ is

$$\mathcal{L}(\pi_1) = \prod_{i=1}^{n} p(Y_i; \pi_1) = \prod_{i=1}^{n} \pi_1^{Y_i} (1 - \pi_1)^{1-Y_i}.$$

The logistic model assumes that for $1 \leq k \leq K - 1$,

$$\mathbb{P}(Y = k|x) = \frac{\exp(\beta_{k0} + x^\top \beta_k)}{1 + \sum_{j=1}^{K-1} \exp(\beta_{j0} + x^\top \beta_j)} \equiv \pi_k(x, \{\beta_{j0}\}, \{\beta_j\}). \tag{5}$$

and

$$\mathbb{P}(Y=0|x) = \frac{1}{1+\sum_{l=1}^{K-1}\exp(\beta_{l0}+x^\top\beta_l)} \equiv \pi_0(x,\{\beta_{j0}\},\{\beta_j\}).$$

In other words, we set $\beta_{00}=0$ and $\beta_0=0$, which is for identifiablity problem. In other words, given $X=x$, $I(Y=k)$ is Bernoulli with mean $\pi_k(x,\{\beta_{j0}\},\{\beta_j\})$. The Bayes classifier is

$$f^*(x) = \arg\max_{j=0,\dots,K-1} \mathbb{P}(y=j|x).$$

Lemma 2. *Both linear regression and logistic regression models have linear decision boundaries.*

Proof. Recall that the Bayes classifier is

$$f_*(x) = \arg\max_{j=0,\dots,K-1} \mathbb{P}(Y=j|X=x).$$

Now, the decision boundary for i and j is a subset of where $\mathbb{P}(Y=i|X=x)=\mathbb{P}(Y=j|X=x)$, and for both linear regression and the logistic regression, this is a hyperplane defined as

$$\left\{x\in\mathbb{R}^d : (\beta_{i0}-\beta_{j0})+x^\top(\beta_i-\beta_j)=0\right\}.$$

In particular for the logistic regression, this follows from the monotonicity of the logistic function.

□

There are several ways to motivate logistic regression: for simplicity consider binary classifiction, so $\mathcal{Y}=\{0.1\}$.
1. Consider a linear regression

$$\mathbb{P}(y=1|x) = \beta_0 + x^\top\beta.$$

It violates that the constraint

$$\mathbb{P}(y=1|x)\in[0,1].$$

A simple remedy for this problem is to set

$$\mathbb{P}(y=1|x) = F(\beta_0+x^\top\beta),$$

where F is a continuous and strictly increasing function with $F(-\infty)=0$ and $F(\infty)=1$. Some examples for F are:

- Gaussian: Probit model
- Gompertz: $F(x)=\exp(-\exp(x))$, popularly used in Insurance
- Logistic: $F(x)=\exp(x)/(1+\exp(x))$.

2. Consider the decision boundary

$$\{x : \Pr(Y=1|X=x)=0.5\}.$$

This is equivalent to

$$\{x : \log(\Pr(Y=1|X=x)/\Pr(Y=2|X=x))=0\}.$$

Suppose that the log-odds is linear. That is,

$$\log(\Pr(Y=1|X=x)/\Pr(Y=2|X=x)) = \beta_0+x^\top\beta,$$

and this implies that

$$\Pr(Y = 1|X = x) = \frac{\exp(\beta_0 + x^\top \beta)}{1 + \exp(\beta_0 + x^\top \beta)}.$$

The parameters β_{j0} and $\beta_j = (\beta_{j1}, \ldots, \beta_{jd})^\top$ can be estimated by maximum conditional likelihood. The likelihood is simply the probability of the observations given as

$$\mathcal{L}(\{\beta_{j0}\}, \{\beta_j\}) = \prod_{i=1}^{n} \Pr(Y = Y_i|X = X_i),$$

and we estimate β by maximizing the log-likelihood. For simplicity, we consider the binary case where $\mathcal{Y} = \{0, 1\}$, and write β_0 for β_{10}, β for β_1. The conditional likelihood function for β_0, β is

$$\mathcal{L}(\{\beta_{j0}\}, \{\beta_j\}) = \prod_{i=1}^{n} \pi_1(x_i, \beta_0, \beta)^{Y_i}(1 - \pi_1(x_i, \beta_0, \beta))^{1-Y_i}.$$

Thus the conditional log-likelihood is

$$\begin{aligned} \ell(\beta_0, \beta) &= \sum_{i=1}^{n} \Big\{ Y_i \log \pi_1(X_i, \beta_0, \beta) - (1 - Y_i) \log(1 - \pi_1(X_i, \beta_0, \beta) \Big\} && (6) \\ &= \sum_{i=1}^{n} \Big\{ Y_i(\beta_0 + X_i^\top \beta) - \log\big(1 + \exp(\beta_0 + X_i^\top \beta)\big) \Big\}. && (7) \end{aligned}$$

The maximum conditional likelihood estimators $\hat{\beta}_0$ and $\hat{\beta}$ cannot be found in closed form. However, the loglikelihood function is concave and can be efficiently solve by the Newton's method in an iterative manner as follows.

Note that the logistic regression classifier is essentially replacing the 0-1 loss with a smooth loss function. In other words, it uses a *surrogate loss function.*

For notational simplicity, we redefine (local to this section) the d-dimensional covariate x_i and parameter vector β as the following $(d + 1)$-dimensional vectors:

$$x_i \leftarrow (1, x_i^\top)^T \text{ and } \beta \leftarrow (\beta_0, \beta^T)^T.$$

Thus, we write $\pi_1(x, \beta_0, \beta)$ as $\pi(x, \beta)$ and $\ell(\beta_0, \beta)$ as $\ell(\beta)$.

To maximize $\ell(\beta)$, the $(k + 1)$th Newton step in the algorithm replaces the kth iterate $\hat{\beta}^{(k)}$ by

$$\hat{\beta}^{(k+1)} \leftarrow \hat{\beta}^{(k)} - \left(\frac{\partial^2 \ell(\hat{\beta}^{(k)})}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial \ell(\hat{\beta}^{(k)})}{\partial \beta}. \quad (8)$$

The gradient $\partial s \frac{\partial \ell(\hat{\beta}^{(k)})}{\partial \beta}$ and Hessian $\partial s \frac{\partial^2 \ell(\hat{\beta}^{(k)})}{\partial \beta \partial \beta^T}$ are both evaluated at $\hat{\beta}^{(k)}$ and can be written as

$$\frac{\partial \ell(\hat{\beta}^{(k)})}{\partial \beta} = \sum_{i=1}^{n} (\pi(x_i, \hat{\beta}^{(k)}) - Y_i) X_i \text{ and } \frac{\partial^2 \ell(\hat{\beta}^{(k)})}{\partial \beta \partial \beta^T} = -\mathbb{X}^T \mathbb{W} \mathbb{X}$$

where $\mathbb{W} = \text{diag}(w_{11}^{(k)}, w_{22}^{(k)}, \ldots, w_{dd}^{(k)})$ is a diagonal matrix with

$$w_{ii}^{(k)} = \pi(x_i, \hat{\beta}^{(k)})\big(1 - \pi(x_i, \hat{\beta}^{(k)})\big). \quad (9)$$

Let $\pi_1^{(k)} = \big(\pi_1(x_1, \hat{\beta}^{(k)}), \ldots, \pi_1(x_n, \hat{\beta}^{(k)})\big)^T$, (8) can be written as

$$\begin{aligned} \hat{\beta}^{(k+1)} &= \hat{\beta}^{(k)} + (\mathbb{X}^T \mathbb{W} \mathbb{X})^{-1} \mathbb{X}^T (y - \pi_1^{(k)}) && (10) \\ &= (\mathbb{X}^T \mathbb{W} \mathbb{X})^{-1} \mathbb{X}^T \mathbb{W} \big(\mathbb{X} \hat{\beta}^{(k)} + \mathbb{W}^{-1} (y - \pi_1^{(k)})\big) && (11) \\ &= (\mathbb{X}^T \mathbb{W} \mathbb{X})^{-1} \mathbb{X}^T \mathbb{W} z^{(k)} && (12) \end{aligned}$$

where $z^{(k)} \equiv (z_1^{(k)}, \ldots, z_n^{(k)})^T = \mathbb{X}^T \hat{\beta}^{(k)} + \mathbb{W}^{-1}(y - \pi_1^{(k)})$ with

$$z_i^{(k)} = \log \left(\frac{\pi_1(x_i, \hat{\beta}^{(k)})}{1 - \pi_1(x_i, \hat{\beta}^{(k)})} + \frac{y_i - \pi_1(x_i, \hat{\beta}^{(k)})}{\pi_1(x_i, \hat{\beta}^{(k)})(1 - \pi_1(x_i, \hat{\beta}^{(k)}))} \right). \tag{13}$$

Given the current estimate $\hat{\beta}^{(k)}$, the above Newton iteration forms a quadratic approximation to the negative log-likelihood using Taylor expansion at $\hat{\beta}^{(k)}$:

$$-\ell(\beta) = \underbrace{\frac{1}{2}(z - \mathbb{X}\beta)^T \mathbb{W} (z - \mathbb{X}\beta)}_{\ell_Q(\beta)} + \text{constant}. \tag{14}$$

The update equation (12) corresponds to solving a quadratic optimization

$$\hat{\beta}^{(k+1)} = \text{argmin}_\beta \ell_Q(\beta). \tag{15}$$

We then get an iterative algorithm called *iteratively reweighted least squares*. See Algotirhm 1.

Algorithm 1 Finding the Logistic Regression MLE.

Iteratively Reweighted Least Squares Algorithm

Choose starting values $\hat{\beta}^{(0)} = (\hat{\beta}_0^{(0)}, \hat{\beta}_1^{(0)}, \ldots, \hat{\beta}_d^{(0)})^T$ and compute $\pi_1(x_i, \hat{\beta}^{(0)})$ using Equation (5), for $i = 1, \ldots, n$ with β_j replaced by its initial value $\hat{\beta}_j^{(0)}$.

For $k = 1, 2, \ldots$, iterate the following steps until convergence.

1. Calculate $z_i^{(k)}$ according to (13) for $i = 1, \ldots, n$.
2. Calculate $\hat{\beta}^{(k+1)}$ according to (12). This corresponds to doing a weighted linear regression of z on $\mathbb{X}$.
3. Update the $\pi(x_i, \hat{\beta})$'s using (5) with the current estimate of $\hat{\beta}^{(k+1)}$.

We can get the estimated standard errors of the final solution $\hat{\beta}$. For the kth iteration, recall that the Fisher information matrix $I(\hat{\beta}^{(k)})$ takes the form

$$I(\hat{\beta}^{(k)}) = -\mathbb{E}\left(\frac{\partial^2 \ell(\hat{\beta}^{(k)})}{\partial\beta\partial\beta^T}\right) \approx \mathbb{X}^T\mathbb{W}\mathbb{X}, \tag{16}$$

we estimate the standard error of $\hat{\beta}_j$ as the jth diagonal element of $I(\hat{\beta})^{-1}$.

Example 3. We apply the logistic regression on the Coronary Risk-Factor Study (CORIS) data and yields the following estimates and Wald statistics W_j for the coefficients:

Covariate	$\hat{\beta}_j$	se	W_j	p-value
Intercept	-6.145	1.300	-4.738	0.000
sbp	0.007	0.006	1.138	0.255
tobacco	0.079	0.027	2.991	0.003
ldl	0.174	0.059	2.925	0.003
adiposity	0.019	0.029	0.637	0.524
famhist	0.925	0.227	4.078	0.000
typea	0.040	0.012	3.233	0.001
obesity	-0.063	0.044	-1.427	0.153
alcohol	0.000	0.004	0.027	0.979
age	0.045	0.012	3.754	0.000

5 Linear Discriminant Analysis (선형판별분석)

Let $p_j(x)$ is the class conditional density of x in class $y = j$ where $y \in \mathcal{Y} = \{0, \ldots, K-1\}$, i.e.,

$$p_j(x) = p(x|y = j), \qquad j = 0, \ldots, K-1.$$

Let

$$\pi_j = \Pr(y = j), \qquad j = 0, \ldots, K-1,$$

be the prior probabilities, with $\sum_{j=0}^{K-1} \pi_j = 1$. Recall from the above that the Bayes classifier is

$$f^*(x) = \arg \max_{j=0,\ldots,K-1} p_j(x)\pi_j.$$

Suppose that we model each class density $p_j(x) = p(x|Y = k)$ as multivariate Gaussian:

$$p_j(x) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)\right), \quad j = 0, 1\ldots, K-1,$$

where μ_j is the mean vector, Σ_j is the $d \times d$ covariance matrix, so that $X|Y = j \sim \mathcal{N}(\mu_j, \Sigma_j)$.

Given a square matrix A, we define $|A|$ to be the determinant of A. For a binary classification problem with Gaussian distributions, we have the following theorem.

Theorem 4. *If $X|Y = j \sim \mathcal{N}(\mu_j, \Sigma_j)$, then the Bayes rule is*

$$f^*(x) = j \qquad \text{if } \delta_j(x) > \delta_l(x) \text{ for all } l \neq j,$$

where

$$\delta_j(x) \coloneqq -\frac{1}{2}\log|\Sigma_j| - \frac{1}{2}(x-\mu_j)^\top\Sigma_j^{-1}(x-\mu_j) + \log\pi_j.$$

Note that $(x-\mu_j)^\top\Sigma_j^{-1}(x-\mu_j)$ is the *Mahalanobis distance* between x and μ_j.

Proof. By definition, the Bayes rule is

$$f^*(x) = j \qquad \text{if } \pi_j p_j(x) > \pi_l p_l(x) \text{ for all } l \neq j.$$

By plugging in the specific forms of p_j's and taking the logarithms,

$$\log(\pi_j p_j(x)) = -\frac{1}{2}\log|\Sigma_j| - \frac{1}{2}(x-\mu_j)^\top\Sigma_j^{-1}(x-\mu_j) + \log\pi_j - \frac{d}{2}\log(2\pi).$$

Hence we get $f^*(x) = k$ if and only if

$$\delta_j(x) > \delta_l(x) \text{ for all } l \neq j.$$

□

An equivalent way of expressing the Bayes rule is

$$f^*(x) = \text{argmax}_{j=0,\ldots,K-1}\delta_j(x).$$

The functions δ_j is called the *(Gaussian) discriminant function*. The decision boundary of the above classifier can be characterized by the set $\{x \in \mathcal{X} : \delta_i(x) = \delta_j(x)\}$, which is quadratic so this procedure is called *quadratic discriminant analysis* (QDA).

In practice, we use sample quantities of π_j, μ_j, Σ_j in place of their population values, namely

$$\hat{\pi}_j = \frac{n_j}{n}, \qquad \text{where } n_j = \sum_{i=1}^n I(y_i = j), \tag{17}$$

$$\hat{\mu}_j = \frac{1}{n_j}\sum_{i=1}^n X_i I(Y_i = j), \tag{18}$$

$$\hat{\Sigma}_j = \frac{1}{n_j - 1}\sum_{i=1}^n (X_i - \hat{\mu}_0)(X_i - \hat{\mu}_0)^\top I(Y_i = j). \tag{19}$$

(Note: we could also estimate Σ_j using its maximum likelihood estimate, which replace $n_j - 1$ with n_j.)

A simplification occurs if we assume that

$$\Sigma_j = \Sigma \text{ for all } j.$$

In this case, we can see that

$$\begin{aligned}\log \frac{P(Y=j|X=x)}{P(Y=l|X=x)} &= \log \frac{p_j(x)}{p_l(x)} + \log \frac{\pi_j}{\pi_l} \\ &= \log \frac{\pi_j}{\pi_l} - \frac{1}{2}(\mu_j + \mu_l)^\top \Sigma(\mu_j - \mu_l) + x^\top \Sigma(\mu_j - \mu_l).\end{aligned}$$

Hence the Bayes classifier is given as

$$f^*(x) = \text{argmax}_k \delta_k(x),$$

where now

$$\delta_j(x) \coloneqq x^\top \Sigma^{-1} \mu_j - \frac{1}{2} \mu_j{}^\top \Sigma^{-1} \mu_j + \log \pi_j. \tag{20}$$

The decision boundary between class i and j, i.e., $\{x \in \mathcal{X} : \ \delta_i(x) = \delta_j(x)\}$ is linear, so the Bayes classifier is a linear classifier. The functions δ_j are called the linear discriminant functions, and this method is called *linear discrimination analysis* (LDA). The parameters are estimated as before, except that we use a pooled estimate of the Σ:

$$\hat{\Sigma} = \frac{1}{n-K} \sum_{j=1}^{K} (n_j - 1) \hat{\Sigma}_j. \tag{21}$$

When the dimension d is large, fully specifying the QDA decision boundary requires $d + d(d-1)$ parameters, and fully specifying the LDA decision boundary requires $d + d(d-1)/2$ parameters. Such a large number of free parameters might induce a large variance. To further regularize the model, two popular methods are *diagonal quadratic discriminant analysis* (DQDA) and *diagonal linear discriminant analysis* (DLDA). The only difference between DQDA and DLDA with QDA and LDA is that after calculating $\hat{\Sigma}_1$ and $\hat{\Sigma}_0$ as in (19), we set all the off-diagonal elements to be zero. This is also called the independence rule.

Example 5. Let us return to the Iris data example. Recall that there are 150 observations made on three classes of the iris flower: *Iris setosa*, *Iris versicolor*, and *Iris virginica*. There are four features: sepal length, sepal width, petal length, and petal width. In Figure 3 we visualize the datasets. Within each class, we plot the densities for each feature. It's easy to see that the distributions of petal length and petal width are quite different across different classes, which suggests that they are very informative features.

Figures 4 and 5 provide multiple figure arrays illustrating the classification of observations based on LDA and QDA for every combination of two features. The classification boundaries and error are obtained by simply restricting the data to these a given pair of features before fitting the model. We see that the decision boundaries for LDA are linear, while the decision boundaries for QDA are highly nonlinear. The training errors for LDA and QDA on this data are both 0.02. From these figures, we see that it is very easy to discriminate the observations of class *Iris setosa* from those of the other two classes.

6 LDA or Logistic Classification

- Logistic classification and LDA both have linear decision boundaries.
- Logistic classification only needs the specification of the conditional distribution $\Pr(Y = j|X = x)$, that is, $\Pr(X = x)$ is completely undetermined. On the other hand, the LDA needs the specification of the joint

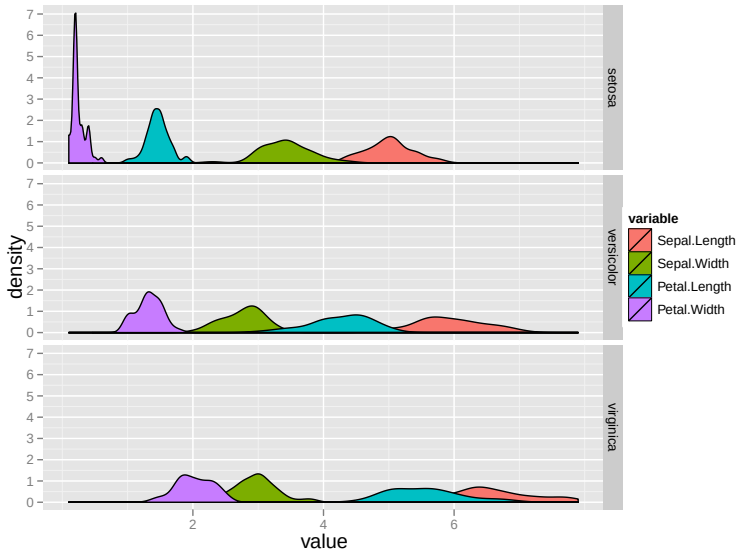

Figure 3: The Iris data: The estimated densities for different features are plotted within each class. It's easy to see that the distributions of petal length and petal width are quite different across different classes, which suggests that they are very informative features.

distribution $\Pr(Y, X)$. In fact, in LDA, the marginal distribution of x is a mixture of Gaussians

$$\Pr(x) = \sum_{j=1}^{K} \pi_j N(\mu_j, \Sigma).$$

Hence, LDA needs more assumptions and hence less applicability than the logistic regression.

- Categorical input variables are allowable for the logistic regression (using dummy variables) while LDA has troubles with such inputs.
- However, LDA is a useful tool when some of the output are missing (semi-supervised learning).
- LDA is useful when Gaussian assumptions are reasonable.
- LDA works better for multi-class problems ($K > 2$).
- In practice, for a two-class problem, logistic classification and LDA are often very similar.

6.1 Multi-class problems with regression approach

There is a serious problem with the regression approach when the number of classes $K \geq 3$, especially prevalent when K is large. Because of the rigid nature of the regression model, classes can be masked by others. Figure 6 illustrates an extreme situation when $K = 3$. The three classes are perfectly separated by linear decision boundaries, yet linear regression misses the middle class completely.

In Figure 7 we have projected the data onto the line joining the three centroids (there is no information in the orthogonal direction in this case), and we have included and coded the three response variables Y_1, Y_2, and Y_3. The three regression lines (left panel) are included, and we see that the line corresponding to the middle class is horizontal and its fitted values are never dominant! Thus, observations from class 2 are classified either as class 1 or class 3. The right panel uses quadratic regression rather than linear regression. For this simple example a quadratic

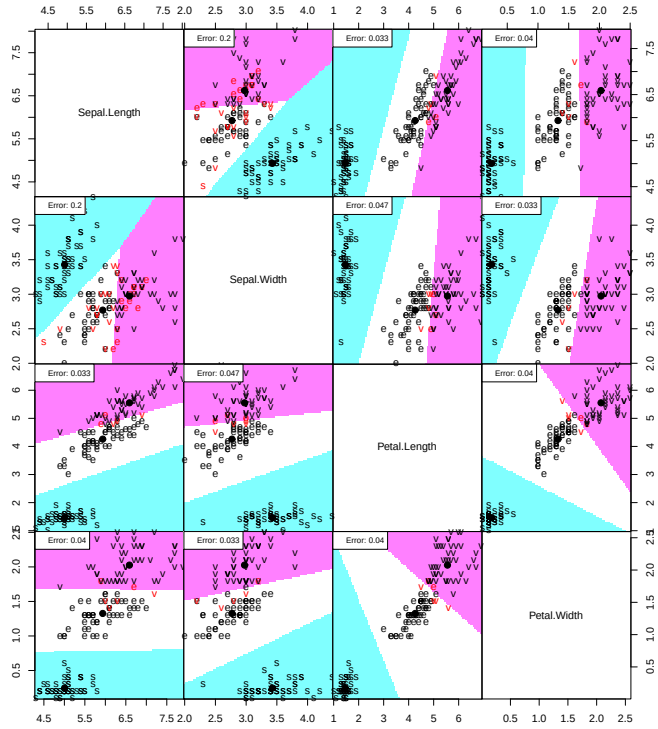

Figure 4: Classifying the Iris data using LDA. The multiple figure array illustrates the classification of observations based on LDA for every combination of two features. The classification boundaries and error are obtained by simply restricting the data to a given pair of features before fitting the model. In these plots, "s" represents the class label *Iris setosa*, "e" represents the class label *Iris versicolor*, and "v" represents the class label *Iris virginica*. The red letters illustrate the misclassified observations.

rather than linear fit (for the middle class at least) would solve the problem, but in general, if $K \geq 3$ classes are lined up, polynomial terms up to degree $K-1$ might be needed to resolve them.

Note: masking problem is severe in ordinary regression (that is, regress Y on X), but it is also present in logistic regression as well. For tackling masking problem, LDA is better than logistic classification.

References

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer, New York, second edition, 2009. Data mining, inference, and prediction.

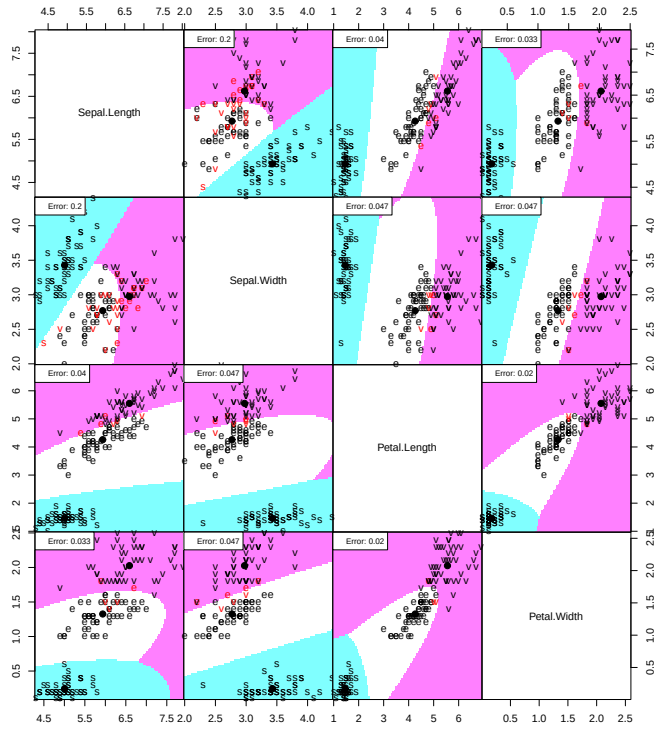

Figure 5: Classifying the Iris data using QDA. The multiple figure array illustrates the classification of observations based on QDA for every combination of two features. The classification boundaries are displayed and the classification error by simply casting the data onto these two features are calculated. In these plots, "s" represents the class label *Iris setosa*, "e" represents the class label *Iris versicolor*, and "v" represents the class label *Iris virginica*. The red letters illustrate the misclassified observations.

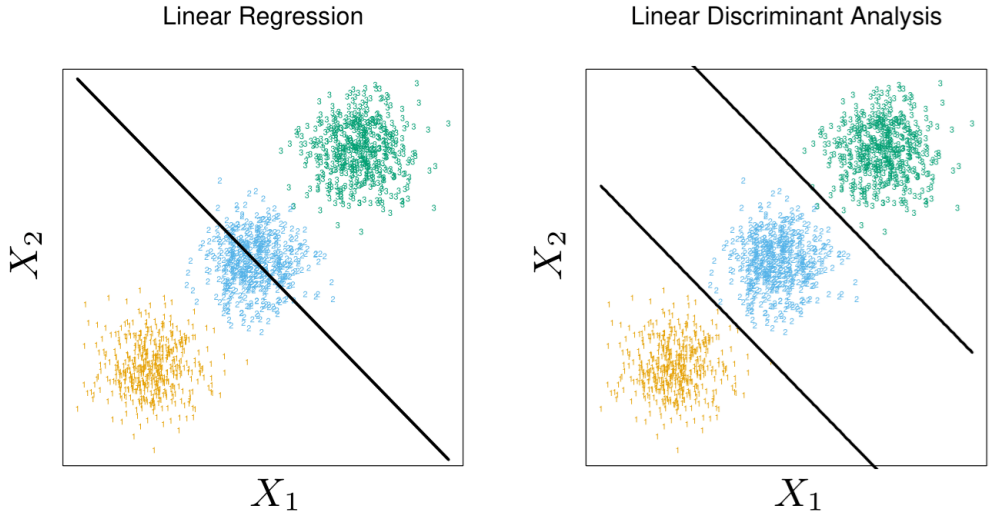

Figure 6: The data come from three classes in $\mathbb{R}^2$ and are easily separated by linear decision boundaries. The right plot shows the boundaries found by linear discriminant analysis. The left plot shows the boundaries found by linear regression of the indicator response variables. The middle class is completely masked (never dominates). Figure 4.2 from [1].

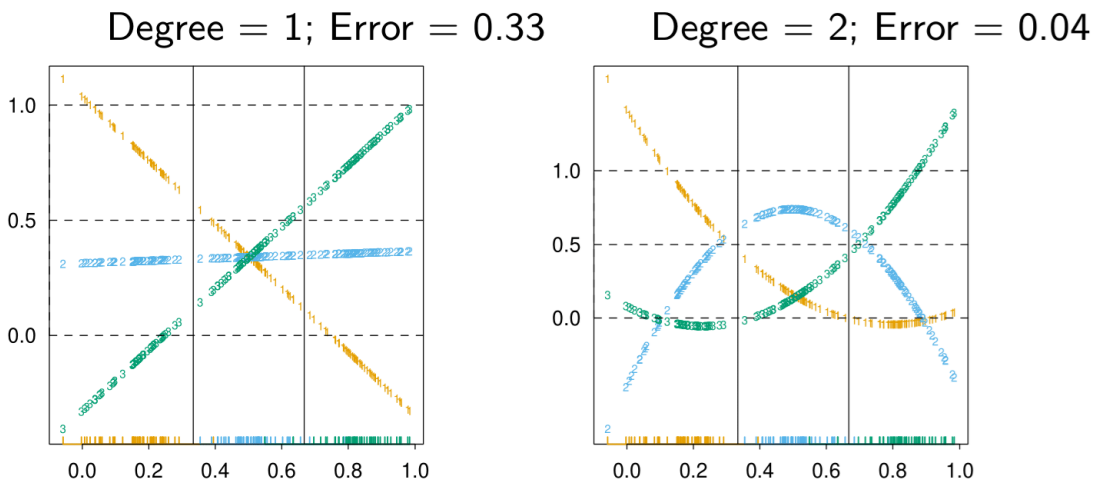

Figure 7: The effects of masking on linear regression in IR for a three-class problem. The rug plot at the base indicates the positions and class membership of each observation. The three curves in each panel are the fitted regressions to the three-class indicator variables; for example, for the blue class, yblue is 1 for the blue observations, and 0 for the green and orange. The fits are linear and quadratic polynomials. Above each plot is the training error rate. The Bayes error rate is 0.025 for this problem, as is the LDA error rate. Figure 4.3 from [1].