Stat462/862 Unit 4
Linear Methods for Classification

C. Devon Lin

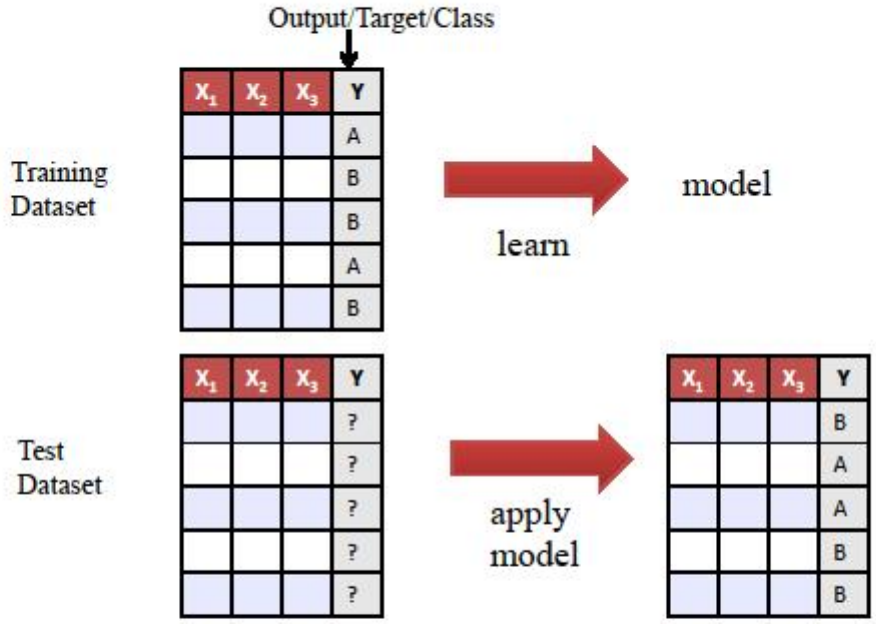
Department of Mathematics and Statistics, Queen's University

Outline

1. Introduction
2. Logistic Regression
3. Linear Regression of an Indicator Matrix
4. Linear Discriminant Analysis

Reference: Sections 4.1 to 4.4 in HTF

Classification



Simple example of a classification model ("classifier"): Use the label of the past training point which is most **similar** to the new test point, and return that as the prediction ("**nearest-neighbor**").

Linear Methods

- The predictor $G(x)$ takes values in a discrete set $\mathcal{G}$.
- $\delta_k(x)$–discriminant function, a function used to classify x to a class with the largest value of the function.

$$G(x) = k_0, \quad \text{if } k_0 = \text{argmax}_k \delta_k(x)$$

For example, $\delta_k(x) = \Pr(G = k | X = x)$.

- Decision boundary between class k and class l is the set of points for which $\delta_k(x) = \delta_l(x)$.

- Linear Methods: $\delta_k(x)$ or a monotone transformation of $\delta_k(x)$ is a linear function, i.e., $g(\delta_k(x)) = \mathbf{h}(\mathbf{x})'\beta$.
  For example:
  - Linear regression using indicator matrix.
  - Linear Discriminant Analysis.
  - Logistic regression: $\log[p/(1-p)]$.

$$p = \Pr(G = 1|X = x) = \frac{\exp(\beta_0 + \beta' x)}{1 + \exp(\beta_0 + \beta' x)}$$

$$\log \frac{\Pr(G = 1|X = x)}{\Pr(G = 0|X = x)} = \beta_0 + \beta' x$$

Linear Logistic Regression

Two-class case. $Y = 0/1$ codes the classes. Model $p(x) = Pr(Y = 1|x)$.

$$\text{logit}p(x) \equiv \log \frac{p(x)}{1 - p(x)} = \beta^T x, \quad p(x) = \frac{e^{\beta^T x}}{1 + e^{\beta^T x}}$$

$$\text{Log-Likelihood} = \sum_{i=1}^{n} \{y_i \log p_i + (1 - y_i) \log(1 - p_i)\}$$

IRLS algorithm

1. Initialize β.
2. Form linearized responses $z_i = \beta^T x_i + (y_i - p_i)/\{p_i(1 - p_i)\}$
3. Form weights $w_i = p_i(1 - p_i)$
4. Update β by weighted LS of z_i on x_i with weights w_i.

Steps 2-4 are repeated until convergence.

Estimation

- Let $y_i = 1$ if $g_i = 1$ and $y_i = 0$ if $g_i = 2$, Then the likelihood

$$\prod_{i=1}^{N} p_{g_i}(\mathbf{x}_i; \boldsymbol{\theta}) = \prod_{i=1}^{N} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}.$$

- The log-likelihood is $l(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p_{g_i}(\mathbf{x}_i; \boldsymbol{\theta})$, i.e.,

$$\begin{aligned} l(\boldsymbol{\beta}) &= \sum_{g_i=1} \log p_{g_i}(\mathbf{x}_i; \boldsymbol{\beta}) + \sum_{g_i=2} \log p_{g_i}(\mathbf{x}_i; \boldsymbol{\beta}) \\ &= \sum_{i=1}^{N} \left\{ y_i \boldsymbol{\beta}^T \mathbf{x}_i - \log(1 + \exp(\boldsymbol{\beta}^T \mathbf{x}_i)) \right\} \end{aligned}$$

- Minimize $l(\boldsymbol{\beta})$ to estimate $\boldsymbol{\beta}$.

Estimation

- First order derivative of $l(\boldsymbol{\beta})$ is equal to $\mathbf{0}$, called score equation.

$$\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{N} \mathbf{x}_i (y_i - p(\mathbf{x}_i, \boldsymbol{\beta})) = 0. \quad \text{Nonlinear in } \boldsymbol{\beta}$$

- Use Newton algorithm,

$$\boldsymbol{\beta}^{new} = \boldsymbol{\beta}^{old} - \left(\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} \right)^{-1} \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}.$$

- Matrix form:

$$\boldsymbol{\beta}^{new} = (\mathbf{X}^{\mathbf{T}} \mathbf{W} \mathbf{X})^{-1} (\mathbf{X}^{\mathbf{T}} \mathbf{W} \mathbf{z})$$
$$\mathbf{z} = X\boldsymbol{\beta}^{old} + W^{-1}(y - p).$$

IRLS = Newton Algorithm

$$\begin{aligned}\frac{\partial \ell(\beta)}{\partial \beta} &= \mathbf{X}^T(\mathbf{y}-\mathbf{p}) \\ \frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} &= -\mathbf{X}^T\mathbf{W}\mathbf{X}.\end{aligned}$$

A Newton step is thus

$$\begin{aligned}\beta^{new} &= \beta^{old} + (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y}-\mathbf{p}) \\ &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\left(\mathbf{X}\beta^{old} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\right) \\ &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{z}.\end{aligned}$$

In the second and third line we have re-expressed the Newton-Raphson step as a weighted least squares step, with the response

$$\mathbf{z} = \mathbf{X}\beta^{old} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p}).$$

Multiple Logistic Regression

- Multi-logit model form:

$$\log \frac{\Pr(G=1|X=x)}{\Pr(G=K|X=x)} = \beta_{10} + \boldsymbol{\beta}_1^T \mathbf{x}$$

$$\vdots$$

$$\log \frac{\Pr(G=K-1|X=x)}{\Pr(G=K|X=x)} = \beta_{(K-1)0} + \boldsymbol{\beta}_{K-1}^T \mathbf{x}.$$

equivalent to

$$\Pr(G=k|X=x) = \frac{\exp(\beta_{k0} + \boldsymbol{\beta}_k^T \mathbf{x})}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \boldsymbol{\beta}_l^T \mathbf{x})}, k = 1, \ldots, K-1$$

$$\Pr(G=K|X=x) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \boldsymbol{\beta}_l^T \mathbf{x})}.$$

- Estimation by weighted least squares or multinomial maximum likelihood.

R code

```
model_logit = glm(chd ~ . ,
     family = binomial(link='logit'), data=SAheart)
require(nnet)
model_logit2 <- multinom(chd~.,data=SAheart)
pred_logit=predict(model_logit,Test_saheart,type="response'
as.numeric(pred_logit>=0.5)
pred_logit2= predict(model_logit2,Test_saheart)
```

Linear Regression using Indicator Matrix

- Let $Y(x) = (Y_1, Y_2, \ldots, Y_K)$ where $Y_k = 1$ if $G(x) = k$ and $Y_k = 0$ if $G_k(x) \neq k$. Indicator response matrix $\mathbf{Y}$ is an $N \times K$ matrix formed by the N training data point.
- Example: a response vector $g = (g_1, \ldots, g_N)'$,

Indicator response matrix

$$\mathbf{g} = \begin{pmatrix} 3 \\ 1 \\ 4 \\ \vdots \\ 2 \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \vdots & & & \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Linear Regression Formula

- Let X be the model matrix with $p+1$ columns (including the $\mathbf{1}$ column). The linear regression is

$$\mathbf{Y} = \mathbf{XB} + \epsilon.$$

- The least squares is

$$\min_{\mathbf{B}} \sum_{i=1}^{N} \|y_i - \mathbf{B}' x_i\|^2,$$

  where y_i, x_i are ith rows of $\mathbf{Y}$ and $\mathbf{X}$.
- Coefficient estimate: $\hat{\mathbf{B}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$.
  Estimation of $\mathbf{Y}$: $\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$.

Linear Regression Formula: prediction

- A new observation with input x is classified as follows:
  - compute the prediction $\hat{f}(x) = \hat{\mathbf{B}}'x$.
  - identify the largest component and classify accordingly:

$$\hat{G}(x) = \text{argmax}_{k \in \mathcal{G}} \hat{f}_k(x).$$

Masking Problem

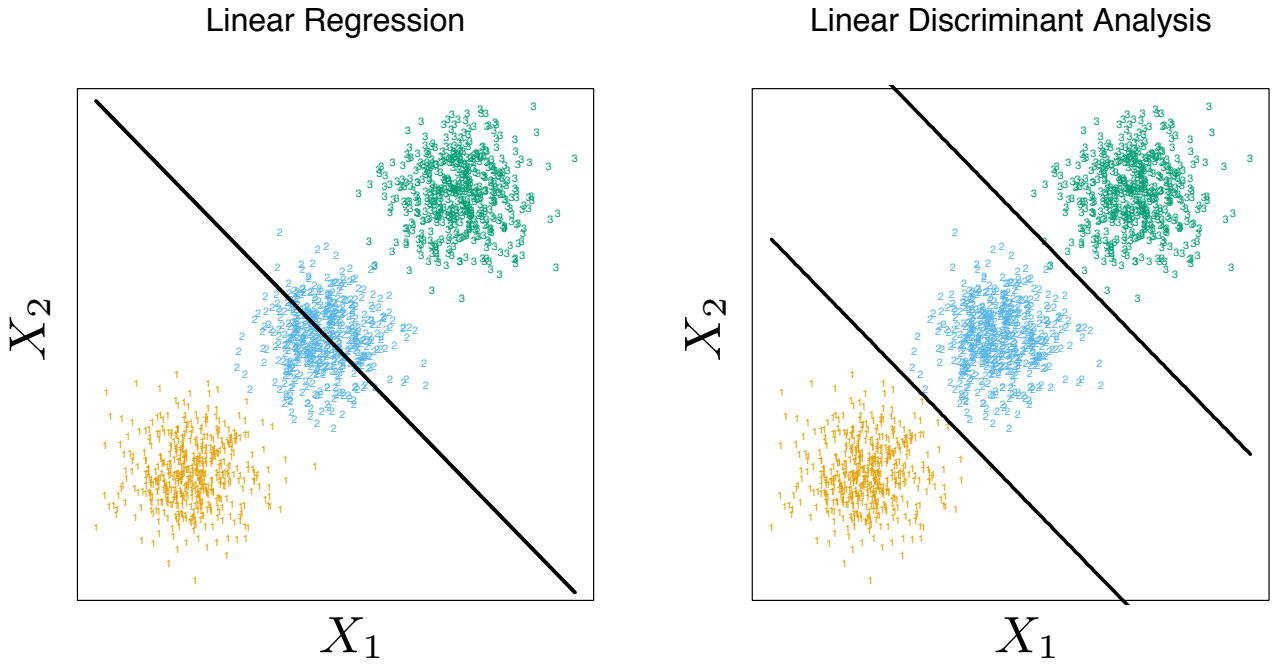
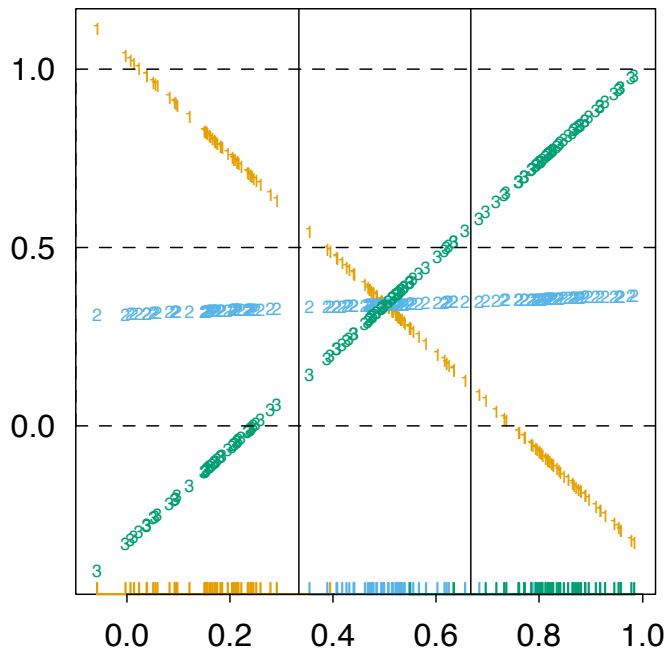


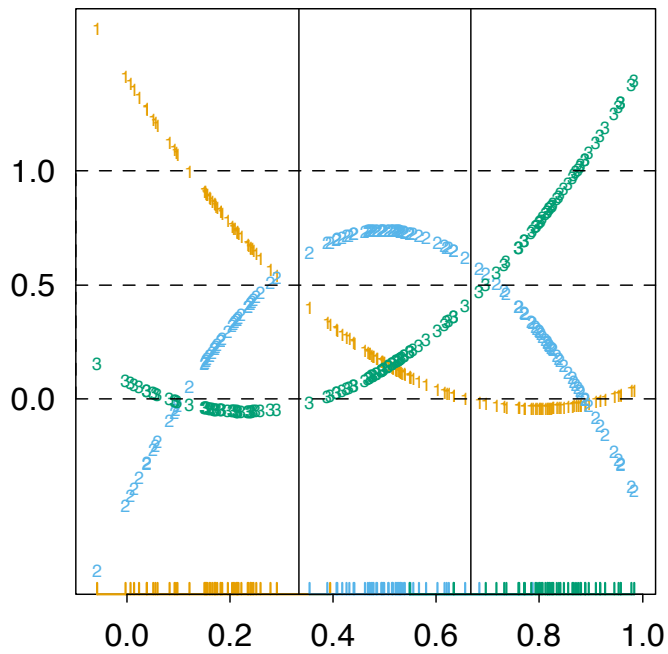
FIGURE 4.2. *The data come from three classes in* $\mathbb{R}^2$ *and are easily separated by linear decision boundaries. The right plot shows the boundaries found by linear discriminant analysis. The left plot shows the boundaries found by linear regression of the indicator response variables. The middle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

Masking Problem

Degree = 1; Error = 0.33



Degree = 2; Error = 0.04



To overcome this masking problem, include general polynomial terms and cross-product of total degree $K-1$, $O(p^{K-1})$ terms in all.

Assumption of Linear Discriminant Analysis

- Need to know $\delta_k(x) = \Pr(G = k|X = x), k = 1, \ldots, K$.
- $f_k(x)$ – the class-conditional density of X in class $G = k$, π_k – the prior probability of class k, i.e., $\Pr(G = k) = \pi_k$, thus $\sum_k^K \pi_k = 1$.
- According to Bayes theorem:

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

- Estimate the density of X for each class, $f_k(x)$.

Linear Discriminant Analysis

- Assume that in each class $G = k$, X follows a multivariate normal distribution.

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp(-\frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k))$$

- One more assumption for LDA: $\Sigma_k = \Sigma$ for all k, i.e., common covariance matrix.
- Comparing any two classes k and l, it is sufficient to look at their log-ratio

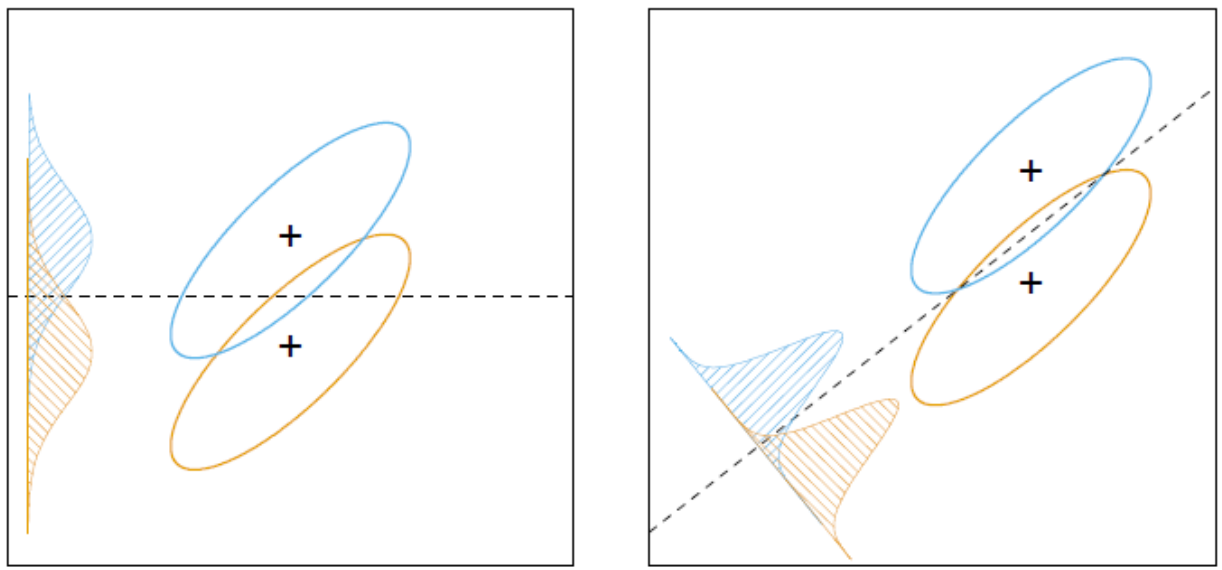
$$\begin{aligned}\log \frac{\Pr(G=k|X=x)}{\Pr(G=l|X=x)} &= \log \frac{f_k(x)}{f_l(x)} + \log \frac{\pi_k}{\pi_l} \\ &= \log \frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k+\mu_l)'\Sigma^{-1}(\mu_k-\mu_l) \\ &\quad + x'\Sigma^{-1}(\mu_k-\mu_l)\end{aligned}$$

Illustration for Two-Class LDA

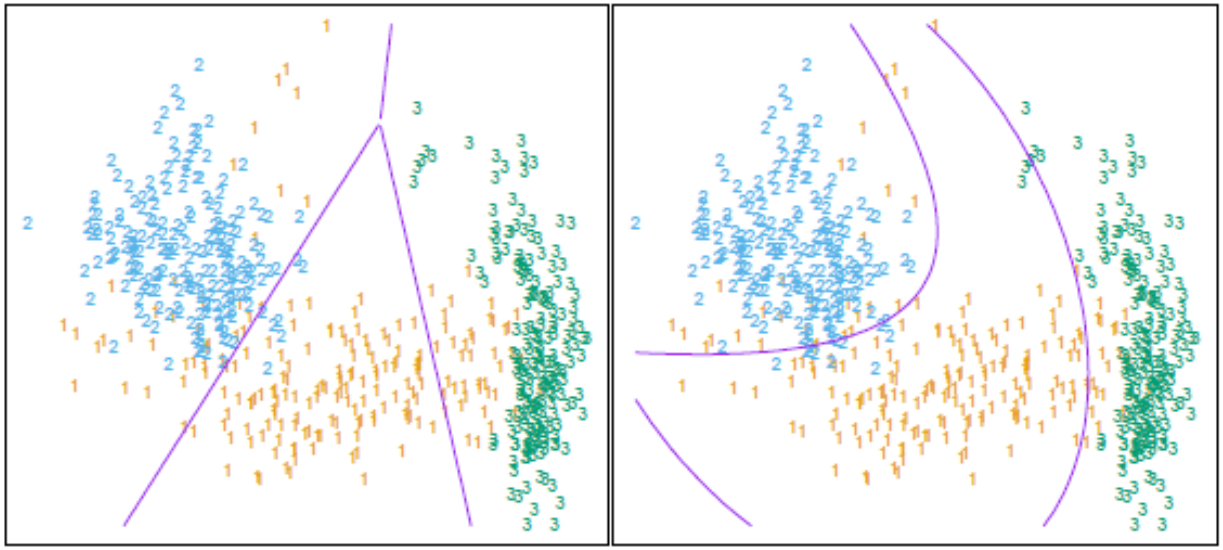
- For two class, we classify to class 1 if

$$x'\hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_2) > \frac{1}{2}(\hat{\mu}_1 + \hat{\mu}_2)'\hat{\Sigma}(\hat{\mu}_1 - \hat{\mu}_2) - \log\frac{N_1}{N_2},$$

where N_1, N_2 are numbers of observations in each class.



Linear Boundaries and Their Projections



The left plot shows some data from three classes, with linear decision boundaries found by linear discriminant analysis. The right plot shows quadratic decision boundaries. These were obtained by finding linear boundaries in the five-dimensional space $X_1, X_2, X_1X_2, X_1^2, X_2^2$. Linear inequalities in this space are quadratic inequalities in the original space.

Linear Discriminant Analysis

- Discriminant functions:

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

- Estimate:
  - $\hat{\pi}_k = N_k/N$, where N_k is the number of observations of class-k.
  - $\hat{\mu}_k = \sum_{g_i=k} x_i/N_k$.
  - $\hat{\Sigma} = \sum_{k=1}^{K}\sum_{g_i=k}(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)'/(N-K)$.
- Number of parameters (apart from $\hat{\Sigma}$): only need the difference $\delta_k(x) - \delta_K(x)$,

$$(p+1) \times (K-1).$$

Quadratic Discriminant Analysis

- If Σ_k are not assumed to be equal,

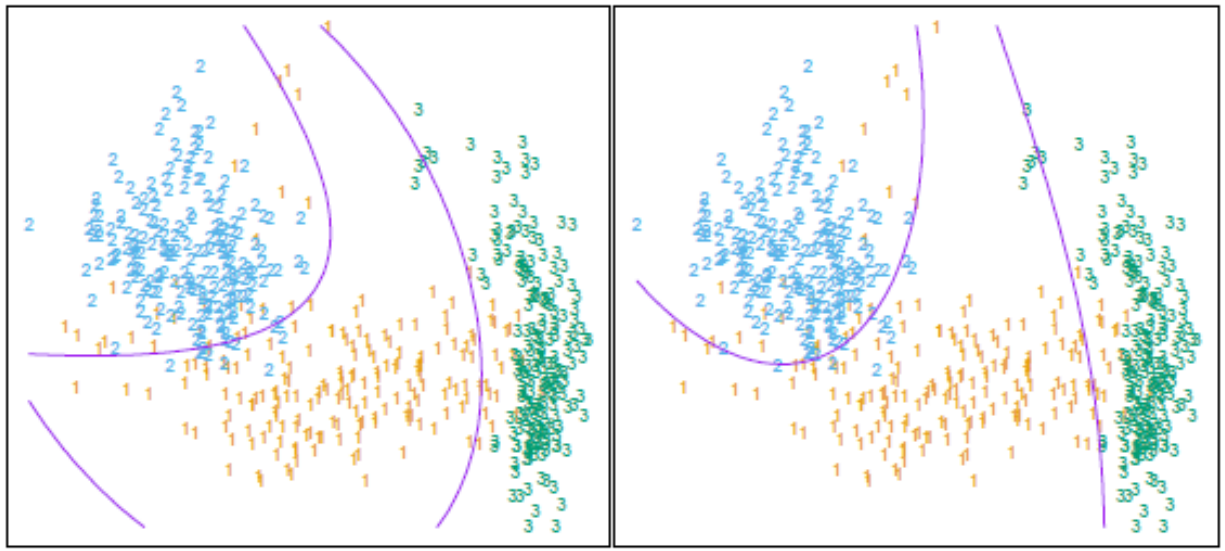
$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k) + \log\pi_k.$$

- Number of parameters: only need the difference $\delta_k(x) - \delta_K(x)$

$$\{1(\pi_k) + p(\mu_k) + \frac{p(p+1)}{2}(\Sigma)\} \times (K-1) = \{\frac{p(p+3)}{2} + 1\} \times (K-1)$$

- Both LDA and QDA have good records of performances.

Illustration of QDA



Two methods for fitting quadratic boundaries. [Left] Quadratic decision boundaries, obtained using LDA in the five-dimensional "quadratic" space. [Right] Quadratic decision boundaries found by QDA. The differences are small, as is usually the case.

LDA and QDA in R

```
library(MASS)
SAheart<-read.table('SAheart.txt',header=T)
attach(SAheart)

model_lda = lda(chd~sbp+tobacco+ldl+adiposity+
typea+obesity+alcohol+age,data=SAheart)

# test data
Test_saheart = SAheart[1:50,1:8]

# prediction
pred_lda = predict(model_lda,Test_saheart )

# quadratic discriminant analysis
model_qda = qda(chd~sbp+tobacco+ldl+adiposity+
typea+obesity+alcohol+age,data=SAheart)
pred_qda = predict(model_qda,Test_saheart)
```

Example: iris data

- The goal is to study variation between and among the different species.
- There are 260 species of iris; this data set focuses of three of them (Iris setosa, Iris virginica and Iris versicolor)
- Four features were measured on 50 samples for each species: sepal width, sepal length, petal width, and petal length.

LDA and QDA in R

```
> table(pred_lda$class, iris2[101:150,5])

            setosa versicolor virginica
  setosa        18          0         0
  versicolor     0         13         1
  virginica      0          2        16
> table(pred_qda$class, iris2[101:150,5])

            setosa versicolor virginica
  setosa        18          0         0
  versicolor     0         13         0
  virginica      0          2        17
```

LDA and QDA in SAS

```
* LDA with equal prior;
PROC DISCRIM DATA=SAheart;
 CLASS chd;
 title 'LDA with equal prior';
RUN;

*Prediction;
PROC DISCRIM DATA=SAheart TESTDATA=SAheart TESTOUT=Pred;
  CLASS chd;
  title 'prediction using LDA with equal prior'
RUN;
```

LDA and QDA in SAS

```
* LDA with proportional  prior;
PROC DISCRIM DATA=SAheart;
 CLASS chd;
 PRIORS prop;
 title 'LDA with proportional prior';
RUN;

* QDA with proportional prior;
PROC DISCRIM DATA=SAheart pool=no;
 CLASS chd;
 PRIORS prop;
 title 'QDA with proportional prior';
RUN;
```

Logistic Regression vs LDA

- LDA:

$$\begin{aligned}\log\frac{\Pr(G=j|X=x)}{\Pr(G=K|X=x)} &= \log\frac{\pi_j}{\pi_K} - \frac{1}{2}(\mu_j+\mu_K)^T\Sigma^{-1}(\mu_j-\mu_K) \\ &\quad +x^T\Sigma^{-1}(\mu_j-\mu_K) \\ &= \alpha_{j0}+\alpha_j^T x.\end{aligned}$$

  This linearity is a consequence of the Gaussian assumption for the class densities, as well as the assumption of a common covariance matrix.

- Logistic model:

$$\log\frac{\Pr(G=j|X=x)}{\Pr(G=K|X=x)} = \beta_{j0}+\beta_j^T x.$$

  They use the same form for the logits

Logistic Regression vs LDA

- Discriminative vs generative (informative) learning: logistic regression uses the conditional distribution of Y given x to estimate parameters, while LDA uses the full joint distribution (assuming normality).

$$\Pr(X, G = j) = \Pr(X)\Pr(G = j|X),$$

- If normality holds, LDA is up to 30% more efficient; o/w logistic regression can be more robust. But the methods are similar in practice.
- The additional efficiency is obtained from using observations far from the decision boundary to help estimate Σ (dubious!)

Comparison between LDA and Logistic Regression

Properties	LDA	Logistic Regression
Model form (odd-ration)	$\alpha_{k0} + \alpha_k^T x$	$\beta_{k0} + \beta_k^T x$
Assumption	X is Gaussian	No
Likelihood	unconditional	conditional
Performance	similar	safer, more robust

Summary

1. Linear Methods in Classification.
2. Logistic Regression, model form and estimation.
3. Linear Regression of the Indicator Matrix.
4. LDA and QDA: assumption, formula and their variants.
5. Comparison of Logistic Regression and LDA.