Lecture 8: Multiclass Classification (I)

Hao Helen Zhang

Lecture 8: Multiclass Classification (I)

Hao Helen Zhang

Multiclass Classification

- General setup
- Bayes rule for multiclass problems
  - under equal costs
  - under unequal costs
- Linear multiclass methods
  - Linear regression model
  - Liner discriminant analysis (LDA)
  - Multiple logistic regression Models

Multiclass Problems

- Class label $Y \in \{1, ..., K\}, K \geq 3$.
- The classifier $f : R^d \longrightarrow \{1, ..., K\}$.
- Examples: zip code classification, multiple-type cancer classification, vowel recognition

The loss function

$$C(k, l) = \text{cost of classifying a sample in class } \ k \ \text{ to class } \ l.$$

In general, $C(k, k) = 0$ for any $k = 1, \cdots, K$. The loss can be described as an $K \times K$ matrix.
For example, $K = 3$,

$$C = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

Terminologies

- Prior class probabilities (marginal distribution of Y):

$$\pi_k = P(Y = k), \quad k = 1, \cdots, K$$

- Conditional distribution of $\mathbf{X}$ given Y: assume that g_k is the conditional density of X given $Y = k$, $k = 1, \cdots, K$.
- Marginal density of X (a mixture):

$$g(\mathbf{x}) = \sum_{k=1}^{K} \pi_k g_k(\mathbf{x}), \quad k = 1, \cdots, K.$$

- Posterior class probabilities (conditional dist. of Y given $\mathbf{X}$):

$$P(Y = k | \mathbf{X} = \mathbf{x}) = \frac{\pi_k g_k(\mathbf{x})}{g(\mathbf{x})}, \quad k = 1, \cdots, K.$$

Bayes Rule

The optimal (Bayes) rule aims to minimize the average loss function over the population

$$\begin{aligned} \phi_B(\mathbf{x}) &= \arg\min_f E_{\mathbf{X},Y} C(Y, f(\mathbf{X})) \\ &= \arg\min_f E_{\mathbf{X}} E_{Y|\mathbf{X}} C(Y, f(\mathbf{X})), \end{aligned}$$

where

$$E_{Y|\mathbf{X}} L(Y, f(\mathbf{X})) = \sum_{k=1}^{K} I(f(\mathbf{X}) = k) \sum_{l=1}^{K} C(l, k) P(Y = l|\mathbf{X}).$$

Bayes Rule Under Equal Costs

If $C(k,l) = I(k \neq l)$, then $R[f] = E_{\mathbf{X}} E_{Y|\mathbf{X}} I(Y \neq f(X))$ and

$$\begin{aligned} E_{Y|X} C(Y, f(\mathbf{X})) &= \sum_{k=1}^{K} I(f(\mathbf{X}) = k) \sum_{l \neq k}^{K} P(Y = l|X) \\ &= \sum_{l=k}^{K} I(f(\mathbf{X}) = k)[1 - P(Y = k|\mathbf{X})] \end{aligned}$$

Given $\mathbf{x}$, the minimizer of $E_{Y|\mathbf{X}=\mathbf{x}} C(Y, f(\mathbf{X}))$ is $f(\mathbf{x}) = k^*$, where

$$k^* = \arg \min_{k=1,..,K} 1 - P(Y = k|\mathbf{x}).$$

The Bayes rule is

$$\phi_B(\mathbf{x}) = k^* \quad \text{if} \quad P(Y = k^*|\mathbf{x}) = \max_{k=1,...,K} P(Y = k|\mathbf{x}),$$

which assigns $\mathbf{x}$ to the most probable class using $\Pr(Y = k|X = \mathbf{x})$.

Bayes Rule Under Unequal Costs

For the general loss function, the Bayes rule can be derived as

$$\phi_B(\mathbf{x}) = k^* \quad \text{if} \quad k^* = \arg \min_{k=1,\ldots,K} \sum_{l=1}^{K} C(l,k) P(Y = l|\mathbf{x}).$$

There is not a simple analytic form for the solution.

Decision (Discriminating) Functions

For multiclass problems, we generally need to estimate multiple discriminant functions $f_k(\mathbf{x}), k = 1, \cdots, K$

- Each $f_k(\mathbf{x})$ is associated with class k.
- $f_k(\mathbf{x})$ represents the evidence strength of a sample $(\mathbf{x}, y)$ belonging to class k.

The *decision rule* constructed using f_k's is

$$\hat{f}(\mathbf{x}) = k^*, \quad \text{where} \quad k^* = \arg \max_{k=1,\ldots,K} f_k(\mathbf{x}).$$

The *decision boundary* of the classification rule $\hat{f}$ between class k and class l is defined as

$$\{\mathbf{x} : \hat{f}_k(\mathbf{x}) = \hat{f}_l(\mathbf{x})\}.$$

Traditional Methods: Divide and Conquer

Main ideas:

(i) Decompose the multiclass classification problem into multiple binary classification problems.

(ii) Use the majority voting principle (a combined decision from the committee) to predict the label

Common approaches: simple but effective

- One-vs-rest (one-vs-all) approaches
- Pairwise (one-vs-one, all-vs-all) approaches

One-vs-rest Approach

One of the simplest multiclass classifier; commonly used in SVMs; also known as the one-vs-all (OVA) approach

(i) Solve K different binary problems: classify "class k" versus "the rest classes" for $k = 1, \cdots, K$.

(ii) Assign a test sample to the class giving the largest $f_k(x)$ (most positive) value, where $f_k(x)$ is the solution from the kth problem

Properties:

- Very simple to implement, perform well in practice
- Not optimal (asymptotically): the decision rule is not Fisher consistent if there is no dominating class (i.e. $\arg\max p_k(x) < \frac{1}{2}$).

Read: Rifkin and Klautau (2004) "In Defense of One-vs-all Classification"

Pairwise Approach

Also known as all-vs-all (AVA) approach

(i) Solve $\binom{K}{2}$ different binary problems: classify "class k" versus "class j" for all $j \neq k$. Each classifier is called g_{ij}.

(ii) For prediction at a point, each classifier is queried once and issues a vote. The class with the maximum number of (weighted) votes is the winner.

Properties:

- Training process is efficient, by dealing with small binary problems.
- If K is big, there are too many problems to solve. If $K = 10$, we need to train 45 binary classifiers.
- Simple to implement; perform competitively in practice.

Read: Park and Furnkranz (2007) "Efficient Pairwise Classification"

Linear Classifier for Multiclass Problems

Assume the decision boundaries are linear. Common linear classification methods:

- Multivariate linear regression methods
- Linear log-odds (logit) models
  - Linear discriminant analysis (LDA)
  - Multiple logistic regression
- Separating hyperplane – explicitly model the boundaries as linear
  - Perceptron model, SVMs

Coding for Linear Regression

For each class k, code it using an indicator variable Z_k:

$$Z_k = 1 \quad \text{if} \quad Y = k;$$

$$Z_k = 0 \quad \text{if} \quad Y \neq k.$$

- The response y is coded using a vector $\mathbf{z} = (z_1, ..., z_K)$. If $Y = k$, only the kth component of $\mathbf{z} = 1$.
- The entire training samples can be coded using a $n \times K$ matrix, denoted by Z.
- Call the matrix Z the *indicator response matrix*.

Multivariate Lineaer Regression

Consider the multivariate model

$$Z = \mathbf{XB} + \mathbf{E}, \quad \text{or} \quad \mathbf{Z}_k = f_k \equiv \mathbf{X}\mathbf{b}_k + \epsilon_k, \ k = 1, \cdots, K.$$

- $\mathbf{Z}_k$ is the kth column of the indicator response matrix Z.
- $\mathbf{X}$ is $n \times (d+1)$ input matrix
- $\mathbf{E}$ is $n \times K$ matrix of errors.
- $\mathbf{B}$ is $(d+1) \times K$ matrix of parameters, $\mathbf{b}_k$ is the kth column of $\mathbf{B}$.

Ordinary Least Square Estimates

Minimize the residual sum of squares

$$\begin{aligned} RSS(B) &= \sum_{k=1}^{K}\sum_{i=1}^{n}[z_{ik} - \beta_{0k} - \sum_{j=1}^{d}\beta_{jk}x_{ij}]^2 \\ &= \text{tr}[(Z - XB)^T(Z - XB)]. \end{aligned}$$

The LS estimates has the same form as univariate case

$$\begin{aligned} \hat{B} &= (X^TX)^{-1}X^TZ, \\ \hat{Z} &= X(X^TX)^{-1}X^TZ \equiv X\hat{B}. \end{aligned}$$

- Coefficient $\widehat{\mathbf{b}}_k$ for y_k is same as univariate OLS estimates. (Multiple outputs do not affect the OLS estimates).
- For $\mathbf{x}$, compute $\hat{f}(\mathbf{x}) = \left[(1 \;\; \mathbf{x})\hat{B}\right]^T$ and classify it as

$$\hat{f}(\mathbf{x}) = \text{argmax}_{k=1,\ldots,K}\hat{f}_k(\mathbf{x}).$$

About Multiple Linear Regression

Justification: Each $\hat{f}_k$ is an estimate of conditional expectation of Y_k, which is the conditional probability of (X, Y) belonging to the class k

$$\mathrm{E}(Y_k|\mathbf{X} = \mathbf{x}) = \Pr(Y = k|\mathbf{X} = \mathbf{x}).$$

- If the intercept is in the model (column of $\mathbf{1}$ in $\mathbf{X}$), then $\sum_{k=1}^{K} \hat{f}_k(\mathbf{x}) = 1$ for any $\mathbf{x}$.
- If the linear assumption is appropriate, then $\hat{f}_k$ give a consistent estimate of the probability for $k = 1, \cdots, K$, as the sample size n goes to infinity.

Limitations of Linear Regression

- There is no guarantee that all $\hat{f}_k \in [0, 1]$; some can be negative or greater than 1.
- Linear structure of X's can be rigid.
- Serious **masking problem** for $K \geq 3$ due to rigid linear structure.

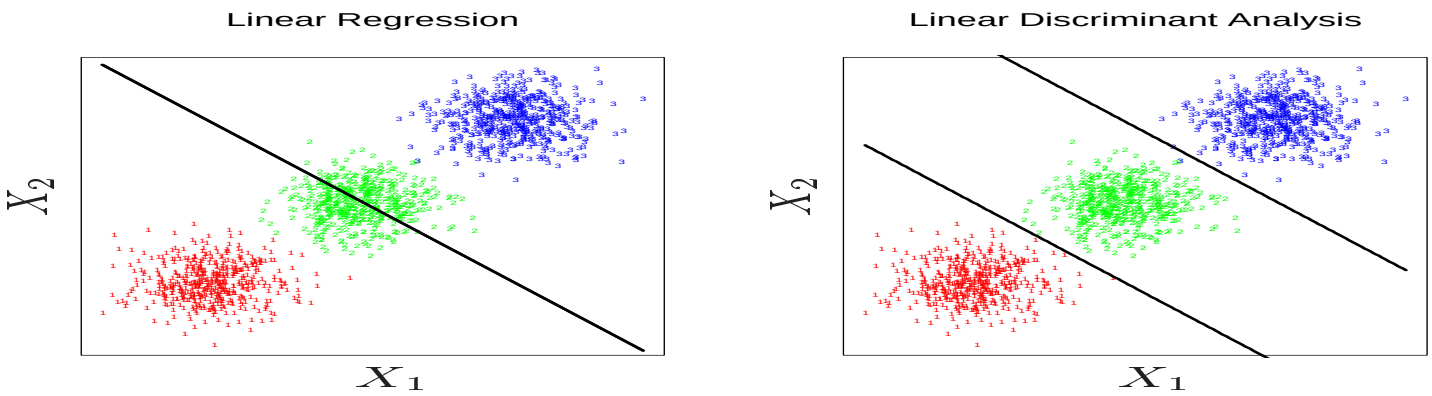

Figure 4.2: *The data come from three classes in* $\mathbb{R}^2$ *and are easily separated by linear decision boundaries. The right plot shows the boundaries found by linear discriminant analysis. The left plot shows the boundaries found by linear regression of the indicator response variables. The middle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

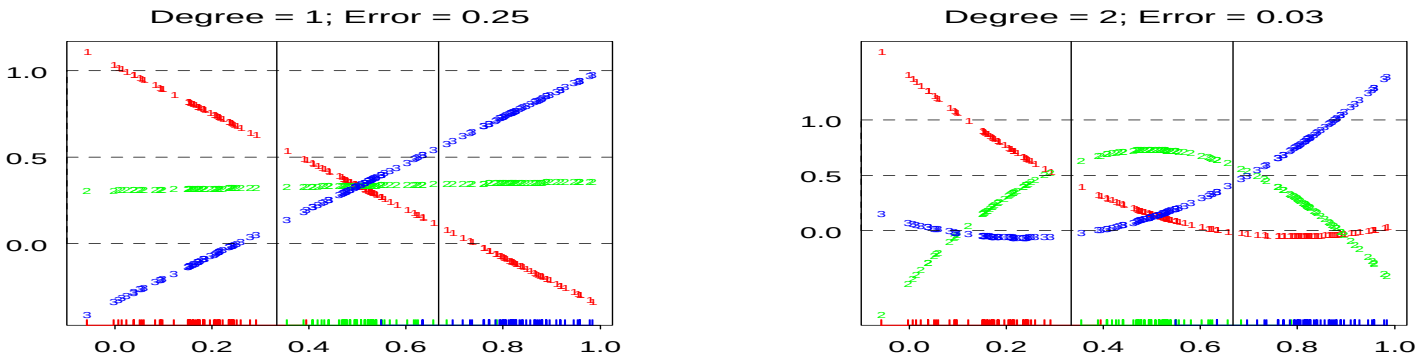

Figure 4.3: *The effects of masking on linear regression in* $\mathbb{R}$ *for a three-class problem. The* rug plot *at the base indicates the positions and class membership of each observation. The three curves in each panel are the fitted regressions to the three-class indicator variables; for example, for the red class,* y_{red} *is* 1 *for the red observations, and* 0 *for the green and blue. 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.*

About Masking Problems

One-dimensional example: (see the previous figure)

- Projected the data onto the line joining the three centroid
- There is no information in the northwest-southeast direction
- Three curves are regression lines
  - The left panel is for the linear regression fit: the middle class is never dominant!
  - The right panel is for the quadratic fit (masking problem is solved!)

How to Solve Masking Problems

In the previous example with $K = 2$

- Using Quadratic regression rather than linear regression
- Linear regression error 0.25; Quadratic regression error 0.03; Bayes error 0.025.

In general,

- If $K \geq 3$ classes are lined up, we need polynomial terms up to degree $K - 1$ to resolve the masking problem.
- In the worst case, need $O(p^{K-1})$ terms.
- Though LDA is also based on linear functions, it does not suffer from masking problems.

Large K, Small d Problem

Masking often occurs for large K and small d.
Example: Vowel data $K = 11$ and $d = 10$. (Two-dimensional for training data using LDA projection)

- A difficult classification problem, as the class overlap is considerate
- the best methods achieve 40% test error

Method	Traing Error	Test Error
Linear regression	0.48	0.67
LDA	0.32	0.56
QDA	0.01	0.53
Logistic regression	0.22	0.51

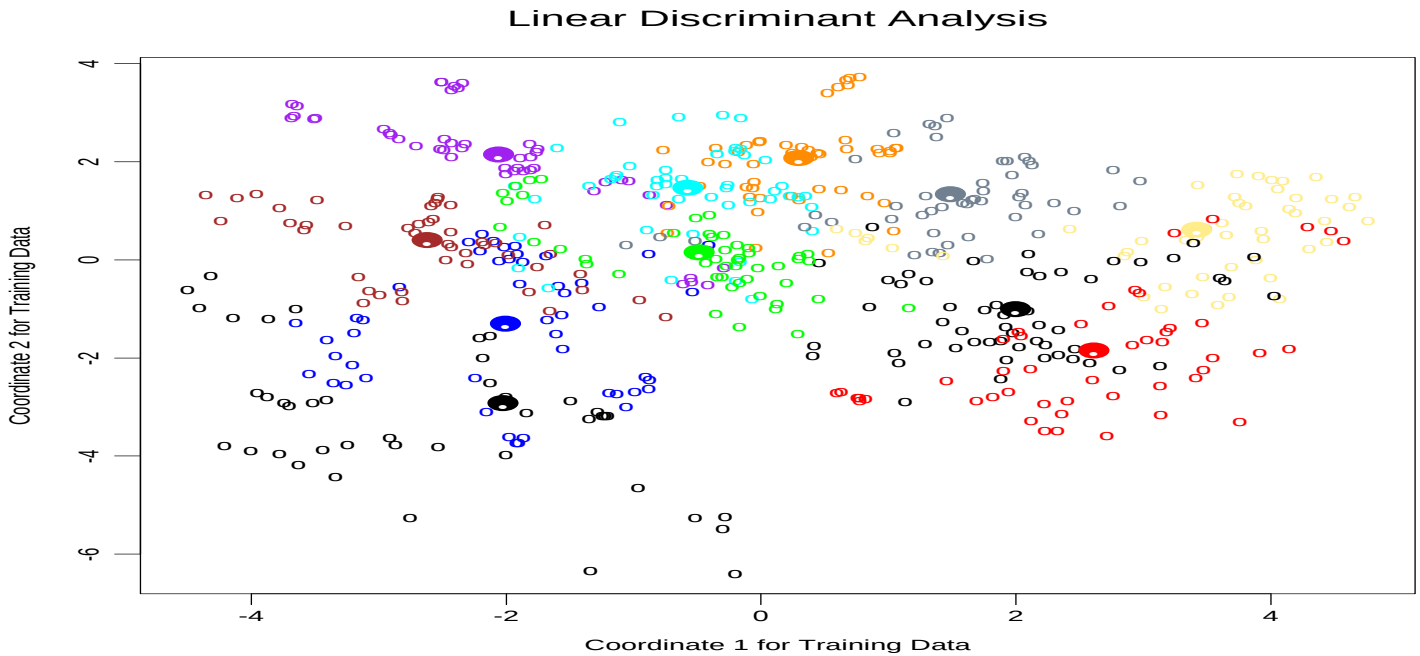

Figure 4.4: *A two-dimensional plot of the vowel training data. There are eleven classes with* $X \in \mathbb{R}^{10}$, *and this is the best view in terms of a LDA model (Section 4.3.3). The heavy circles are the projected mean vectors for each class. The class overlap is considerable.*

Model Assumptions for Linear Discriminant Analysis

Model Setup

- Let π_k be prior probabilities of class k
- Let $g_k(\mathbf{x})$ be the class-conditional densities of $\mathbf{X}$ in class k.
- The posterior probability

$$\Pr(Y = k|\mathbf{X} = \mathbf{x}) = \frac{g_k(\mathbf{x})\pi_k}{\sum_{l=1}^{K} g_l(\mathbf{x})\pi_l}.$$

Model Assumptions:

- Assume each class density is multivariate Gaussian $N(\mu_k, \Sigma_k)$.
- Further, we assume $\Sigma_k = \Sigma$ for all k (equal covariance required!)

Linear Discriminant Function

The log-ratio between class k and l is

$$\begin{aligned}\log \frac{\Pr(Y=k|\mathbf{X}=\mathbf{x})}{\Pr(Y=l|\mathbf{X}=\mathbf{x})} &= \log \frac{\pi_k}{\pi_l} + \log \frac{g_k(\mathbf{x})}{g_l(\mathbf{x})} \\ &= [\mathbf{x}^T\Sigma^{-1}\mu_k - \frac{1}{2}\boldsymbol{\mu}_k^T\Sigma^{-1}\boldsymbol{\mu}_k + \log \pi_k] - \\ &\quad [\mathbf{x}^T\Sigma^{-1}\mu_l - \frac{1}{2}\boldsymbol{\mu}_l^T\Sigma^{-1}\boldsymbol{\mu}_l + \log \pi_l]\end{aligned}$$

For each class k, its discriminant function is

$$f_k(\mathbf{x}) = \mathbf{x}^T\Sigma^{-1}\mu_k - \frac{1}{2}\boldsymbol{\mu}_k^T\Sigma^{-1}\boldsymbol{\mu}_k + \log \pi_k,$$

The quadratic term canceled due to the equal covariance matrix.

The decision rule is

$$\hat{Y}(\mathbf{x}) = \text{argmax}_{k=1,\ldots,K} f_k(\mathbf{x}).$$

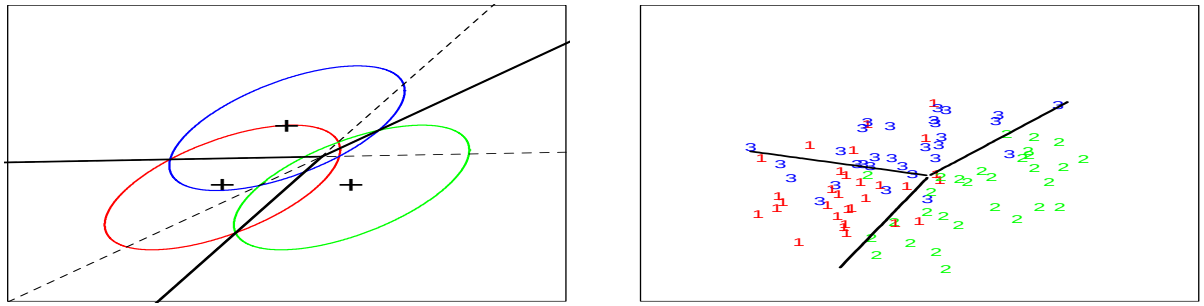

Figure 4.5: *The left panel shows three Gaussian distributions, with the same covariance and different means. Included are the contours of constant density enclosing 95% of the probability in each case. The Bayes decision boundaries between each pair of classes are shown (broken straight lines), and the Bayes decision boundaries separating all three classes are the thicker solid lines (a subset of the former). On the right we see a sample of 30 drawn from each Gaussian distribution, and the fitted LDA decision boundaries.*

Interpretation of LDA

$$\log \Pr(Y = k|\mathbf{X} = \mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) + \log \pi_k + \text{const.}$$

- The constant term does not involve $\mathbf{x}$.
- If prior probabilities are same, the LDA classifies $\mathbf{x}$ to the class with centroid closest to $\mathbf{x}$, using the *squared Mahalanobis distance*, $d_\Sigma(x, \mu)$, based on the common covariance matrix.
- If $d_\Sigma(\mathbf{x} - \boldsymbol{\mu}_k) = d_\Sigma(\mathbf{x} - \boldsymbol{\mu}_l)$, then the prior determines the classifier.

Special Case: $\Sigma_k = I$ for all k

$$f_k(\mathbf{x}) = -\frac{1}{2}\|\mathbf{x} - \boldsymbol{\mu}_k\|^2 + \log \pi_k.$$

- only Euclidean distance is needed!

Mahalanobis Distance

Mahalanobis distance is a distance measure introduced by P. C. Mahalanobis in 1936.

Def: The Mahalanobis distance of $\mathbf{x} = (x_1, \ldots, x_d)^T$ from a set of points with mean $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)^T$ and covariance matrix Σ is defined as:

$$d_\Sigma(\mathbf{x}, \boldsymbol{\mu}) = \left[(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]^{1/2}.$$

It differs from Euclidean distance in that

- it takes into account the correlations of the data set
- it is scale-invariant.

It can also be defined as a dissimilarity measure between $\mathbf{x}$ and $\mathbf{x}'$ of the same distribution with the covariance matrix Σ:

$$d_\Sigma(\mathbf{x}, \mathbf{x}') = \left[(\mathbf{x} - \mathbf{x}')^T \Sigma^{-1} (\mathbf{x} - \mathbf{x}') \right]^{1/2}.$$

Special Cases

- If Σ is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance.
- If Σ is diagonal $\text{diag}(\sigma_1^2, \cdots, \sigma_d^2)$, then the resulting distance measure is called the normalized Euclidean distance:

$$d_\Sigma(\mathbf{x}, \mathbf{x}') = \left[\sum_{i=1}^{d} \frac{(x_i - x_i')^2}{\sigma_i^2} \right]^{1/2},$$

where σ_i is the standard deviation of the x_i and x_i'.

Normalized Distance

Assume we like to decide whether $\mathbf{x}$ belongs to a class. Intuitively, the decision relies on the distance of $\mathbf{x}$ to the class center $\boldsymbol{\mu}$: the closer to $\boldsymbol{\mu}$, the more likely $\mathbf{x}$ belongs to the class.

- To decide whether a given distance is large or small, we need to know if the points in the class are spread out over a large range or a small range. Statistically, we measure the spread by the standard deviation of distances of the sample points to $\boldsymbol{\mu}$.
- If $d_{\Sigma}(\mathbf{x}, \boldsymbol{\mu})$ is less than one standard deviation, then it is highly probable that $\mathbf{x}$ belongs to the class. The further away it is, the more likely that $\mathbf{x}$ not belonging to the class. This is the concept of the *normalized* distance.

Motivation of Mahalanobis Distance

The drawback of the normalized distance is that the sample points are assumed to be distributed about $\boldsymbol{\mu}$ in a spherical manner.

- For non-spherical situations, for instance ellipsoidal, the probability of $\mathbf{x}$ belonging to the class depends not only on the distance from $\boldsymbol{\mu}$, but also on the direction.
- In those directions where the ellipsoid has a short axis, $\mathbf{x}$ must be closer. In those where the axis is long, $\mathbf{x}$ can be further away from the center.

The ellipsoid that best represents the class's probability distribution can be estimated by the *covariance matrix* of the samples.

Linear Decision Boundary of LDA

Linear discriminant functions

$$f_k(\mathbf{x}) = \boldsymbol{\beta}_{1k}^T \mathbf{x} + \beta_{0k}, \quad f_j(\mathbf{x}) = \boldsymbol{\beta}_{1j}^T \mathbf{x} + \beta_{0j}.$$

Decision boundary function between class k and j is

$$(\boldsymbol{\beta}_{1k} - \boldsymbol{\beta}_{1j})^T \mathbf{x} + (\beta_{0k} - \beta_{0j}) = \tilde{\boldsymbol{\beta}}_1^T \mathbf{x} + \tilde{\beta}_0 = 0.$$

Since $\boldsymbol{\beta}_{1k} = \Sigma^{-1}\boldsymbol{\mu}_k$ and $\boldsymbol{\beta}_{1j} = \Sigma^{-1}\boldsymbol{\mu}_j$, the decision boundary has the directional vector

$$\tilde{\boldsymbol{\beta}}_1 = \Sigma^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_j)$$

- $\tilde{\boldsymbol{\beta}}_1$ in generally not in the direction of $\boldsymbol{\mu}_k - \boldsymbol{\mu}_j$
- The discriminant direction $\boldsymbol{\beta}_1$ minimizes the overlap for Gaussian data

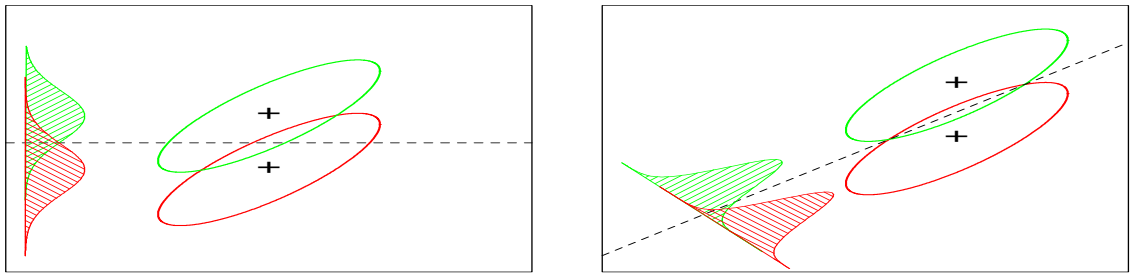

Figure 4.9: *Although the line joining the centroids defines the direction of greatest centroid spread, the projected data overlap because of the covariance (left panel). The discriminant direction minimizes this overlap for Gaussian data (right panel).*

Parameter Estimation in LDA

In practice, we estimate the parameters from the training data

- $\hat{\pi}_k = n_k/n$, where n_k is sample size in
- $\hat{\mu}_k = \sum_{Y_i=k} \mathbf{x}_i/n_k$
- The sample covariance matrix for the kth class:

$$S_k = \frac{1}{n_k - 1} \sum_{Y_i=k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T.$$

- (Unbiased) pooled sample covariance is a weighted average

$$\begin{aligned} \hat{\Sigma} &= \sum_{k=1}^{K} \frac{n_k - 1}{\sum_{l=1}^{K}(n_l - 1)} S_k \\ &= \sum_{k=1}^{K} \sum_{Y_i=k} (\mathbf{x}_i - \hat{\mu}_k)(\mathbf{x}_i - \hat{\mu}_k)^T/(n - K). \end{aligned}$$

Implemenation of LDA

- Compute the eigen-decomposition:

$$\hat{\Sigma} = UDU^T$$

- Sphere the data using $\hat{\Sigma}$, using the following transformation

$$X^* = D^{-1/2}U^T X.$$

  The common covariance estimate of X^* is the identity matrix.
- In the transformed space, classify the point to the closest *transformed* centroid, with the adjustment using π_k's.

Connection between LDA and Linear Regression for $K = 2$

Suppose we code the targets in the two classes as $+1$ and -1. Fit linear model

$$y_i = \beta_0 + \boldsymbol{\beta}_1^T \mathbf{x}_i, \quad i = 1, \cdots, n$$

Then the OLS $(\hat{\beta}_0, \widehat{\boldsymbol{\beta}}_1)$ satisfy

- $\widehat{\boldsymbol{\beta}}_1 \propto \hat{\boldsymbol{\Sigma}}^{-1}(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_2)$
- $\hat{\beta}_0$ is different from the LDA unless $n_1 = n_2$.

The linear regression gives a different decision rule from LDA, unless $n_1 = n_2$.

Multiple Logistic Models

Model the K posterior probabilities by linear functions of $\mathbf{x}$.
We use logit transformations

$$\begin{aligned}
\log \frac{\Pr(Y=1|\mathbf{X}=\mathbf{x})}{\Pr(Y=K|\mathbf{X}=\mathbf{x})} &= \beta_{10} + \boldsymbol{\beta}_1^T \mathbf{x} \\
\log \frac{\Pr(Y=2|\mathbf{X}=\mathbf{x})}{\Pr(Y=K|\mathbf{X}=\mathbf{x})} &= \beta_{20} + \boldsymbol{\beta}_2^T \mathbf{x} \\
&\cdots \qquad \cdots \\
\log \frac{\Pr(Y=K-1|\mathbf{X}=\mathbf{x})}{\Pr(Y=K|\mathbf{X}=\mathbf{x})} &= \beta_{(K-1)0} + \boldsymbol{\beta}_{K-1}^T \mathbf{x}
\end{aligned}$$

The choice of denominator in the odd-ratios is arbitrary.
The parameter vector

$$\boldsymbol{\theta} = \{\beta_{10}, \boldsymbol{\beta}_1, ..., \beta_{(K-1)0}, \boldsymbol{\beta}_{K-1}\}.$$

Probability Estimates

We use the logit transformation to assure that

- the total probabilities sum to one
- each estimated probability lies in $[0, 1]$

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

$$\text{for} \quad k = 1, ..., K-1.$$

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

Check $\sum_{k=1}^{K} p_k(\mathbf{x}) = 1$ for any $\mathbf{x}$.

Maximum Likelihood Estimate for Logistic Models

The joint conditional likelihood of y_i given $\mathbf{x}_i$ is

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p_{y_i}(\boldsymbol{\theta}; \mathbf{x}_i)$$

Using the Newton-Raphson method:

- solve iteratively by minimizing re-weighted least squares.
- provide consistent estimates if the models are specified correctly.