Notes on Statistical and Machine Learning

Linear Methods for Classification

Chapter: *7* **Prepared by:** *Chenxi Zhou*

This note is prepared based on

- *Chapter 4, Linear Methods for Classification* in Hastie, Tibshirani, and Friedman (2009), and
- *Chapter 8, Linear Discriminant Analysis* in Izenman (2009),
- *Chapter 3, Projective Methods* in Burges (2010), and
- *Single-Layer Networks: Classification* in Bishop and Bishop (2023).

I. An Introduction to Linear Methods for Classification

1. **Assumptions:** There are W classes in total labeled as $1, \cdots, W$. The predictor $G(\mathbf{x})$ takes values in $\mathcal{W} := \{1, 2, \cdots, W\}$ so that we can partition the input space into a collection of W disjoint regions.

2. **Overview:** This chapter focuses on *linear* methods for classification. These methods are called *linear* methods since the decision boundaries these methods produce are *linear*.

   In particular, we focus on producing a *discriminant function* $\delta_w(\mathbf{x})$ for each class $w = 1, \cdots, W$, and classify $\mathbf{x}$ to the class with the largest value for its discriminant function. This discriminant approach includes

   - regression based approaches (e.g., linear regression, and logistic regression), and
   - approaches to model the boundaries between classes directly:
     - perceptron, and
     - finding an optimally separating hyperplane.

   Remark. We do *not* need the discriminant function δ_w to be linear. All we require is that some monotone transformation of it is linear.

3. **An Introduction to Linear Regression Approach:** One fits a linear regression models to the class indicator and classify to the largest fit.

   Suppose the linear model for the w-th class is of the form

$$\hat{f}_w(\mathbf{x}) = \hat{\beta}_{w,0} + \widehat{\boldsymbol{\beta}}_w^\top \mathbf{x}.$$

The decision boundary for Classes w and u is

$$\begin{aligned}\Big\{\mathbf{x} \mid \hat{f}_w(\mathbf{x}) = \hat{f}_u(\mathbf{x})\Big\} &= \Big\{\mathbf{x} \mid \hat{\beta}_{w,0} + \widehat{\boldsymbol{\beta}}_w^\top \mathbf{x} = \hat{\beta}_{u,0} + \widehat{\boldsymbol{\beta}}_u^\top \mathbf{x}\Big\} \\ &= \Big\{\mathbf{x} \mid (\hat{\beta}_{w,0} - \hat{\beta}_{u,0}) + (\widehat{\boldsymbol{\beta}}_w - \widehat{\boldsymbol{\beta}}_u)^\top \mathbf{x} = 0\Big\}. \qquad (1)\end{aligned}$$

Note that the decision boundary is an affine set or hyperplane.

4. **Example – Logistic Regression:** We model the posterior probability of Class w given the input vector $\mathbf{x}$, i.e., $\mathbb{P}(G = w \,|\, X = \mathbf{x})$.

   In the binary classification case where $W = 2$ and $\mathcal{W} = \{1, 2\}$, we let

$$\begin{aligned}\mathbb{P}(G = 1 \,|\, X = \mathbf{x}) &= \frac{\exp(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x})}{1 + \exp(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x})}, \\ \mathbb{P}(G = 2 \,|\, X = \mathbf{x}) &= \frac{1}{1 + \exp(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x})}.\end{aligned}$$

   Even though each of $\mathbb{P}(G = 1 \,|\, X = \mathbf{x})$ and $\mathbb{P}(G = 2 \,|\, X = \mathbf{x})$ is *not* linear in $\mathbf{x}$, but its logit transformation

$$\mathrm{logit}(p) := \log\left(\frac{p}{1-p}\right), \qquad \text{where } p := \mathbb{P}(G = 1 \,|\, X = \mathbf{x}),$$

   is, by noting

$$\log \frac{\mathbb{P}(G = 1 \,|\, X = \mathbf{x})}{\mathbb{P}(G = 2 \,|\, X = \mathbf{x})} = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}.$$

   In this case, the decision boundary is

$$\Big\{\mathbf{x} \mid \text{the log odds is } 0\Big\} = \Big\{\mathbf{x} \mid \beta_0 + \boldsymbol{\beta}^\top \mathbf{x} = 0\Big\},$$

   which is an affine set or hyperplane.

II. Linear Regression of an Indicator Matrix

1. **Basic Setup:**

   (a) Denote the training data by $\{(\mathbf{x}_i, g_i)\}_{i=1}^n$, where $g_i \in \mathcal{W}$.

   (b) Let $\mathbf{y} := (y_1, \cdots, y_W)^\top \in \{0, 1\}^W$ be a binary W-dimensional vector with

$$y_w = \begin{cases} 1, & \text{if } G = w, \\ 0, & \text{otherwise.} \end{cases}$$

   Note that there is exactly one component in $\mathbf{y}$ is equal to 1.

Collectively, we write these binary response variables in the matrix form as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^\top \\ \mathbf{y}_2^\top \\ \vdots \\ \mathbf{y}_n^\top \end{bmatrix} \in \{0, 1\}^{n \times W},$$

where n is the number of training cases, W is the number of classes, and each $\mathbf{y}_i$ corresponds to the class label of g_i for all $i = 1, 2, \cdots, n$. Then, each row has exactly one 1 and $(W-1)$ 0's.

(c) Let $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$ be the collection of covariates of the training set, where $p+1$ columns correspond to the p inputs and a leading column of 1's for the intercept.

2. **Model Assumption:** We assume a linear model between $\mathbf{Y}$ and $\mathbf{X}$, i.e.,

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \boldsymbol{\varepsilon},$$

where $\mathbf{B} \in \mathbb{R}^{(p+1) \times W}$ is the coefficient matrix, and $\boldsymbol{\varepsilon} \in \mathbb{R}^{n \times W}$ is the random error term.

3. **Estimation of the Coefficient Matrix B:** Using the least squares method, the estimator of the coefficient matrix $\mathbf{B}$, denoted by $\widehat{\mathbf{B}}$, is

$$\begin{aligned} \widehat{\mathbf{B}} &:= \operatorname*{arg\,min}_{\mathbf{B}} \Big\{ \operatorname{trace}\big((\mathbf{Y} - \mathbf{X}\mathbf{B})^\top (\mathbf{Y} - \mathbf{X}\mathbf{B}) \big) \Big\} \\ &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} \in \mathbb{R}^{(p+1) \times W}, \end{aligned}$$

and the fitted value vector is

$$\widehat{\mathbf{Y}} = \mathbf{X}\widehat{\mathbf{B}} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}.$$

4. **Prediction at a New Case $\mathbf{x}_0$:** To classify a new case $\mathbf{x}_0 \in \mathbb{R}^{p+1}$, we do the following:

(a) Compute the fitted output $\hat{\mathbf{f}}(\mathbf{x}) = (\mathbf{x}_0^\top \widehat{\mathbf{B}})^\top \in \mathbb{R}^W$;

(b) Classify it to the class with the largest component, i.e.,

$$\widehat{G}(\mathbf{x}_0) = \operatorname*{arg\,max}_{w \in \mathcal{W}} \hat{f}_w(\mathbf{x}_0), \tag{2}$$

where $\hat{f}_w(\mathbf{x}_0)$ is the w-th component of $\hat{\mathbf{f}}(\mathbf{x}_0)$.

Remark. For each $w = 1, \cdots, W$, $\hat{f}_w(\mathbf{x})$ can be either negative or greater than 1.

5. **Another View of Linear Regression Approach to Classification:** Construct targets $\mathbf{t}_w$ for each class, where $\mathbf{t}_w$ is the w-th column of the $W \times W$ identity matrix, for all $w = 1, \cdots, W$. Then, the response vector $\mathbf{y}_i^\top \in \mathbb{R}^W$, the i-th row of $\mathbf{Y}$, for the

i-th observation, has the value $\mathbf{y}_i = \mathbf{t}_w$ if $g_i = w$. We then fit the linear model by least squares

$$\underset{\mathbf{B}}{\text{minimize}} \sum_{i=1}^{n} \left\| \mathbf{y}_i - (\mathbf{x}_i^\top \mathbf{B})^\top \right\|_2^2.$$

The criterion is a sum-of-squared Euclidean distances of the *fitted vectors* from their *targets*.

To classify a new observation $\mathbf{x}_0$, we compute its fitted value $\hat{\mathbf{f}}(\mathbf{x}_0) = (\mathbf{x}_0^\top \widehat{\mathbf{B}})^\top$ and classify it to the closet target, i.e.,

$$\widehat{G}(\mathbf{x}_0) = \underset{w \in \mathcal{W}}{\arg\min} \left\| \hat{\mathbf{f}}(\mathbf{x}_0) - \mathbf{t}_w \right\|_2^2. \tag{3}$$

Remarks.

(a) The sum-of-squared-norm criterion is exactly the criterion for multiple response linear regression.

(b) The classification rule (3) is exactly the same as the rule (2).

6. **Problem of Linear Regression Approach to Classification:** When $W \geq 3$, especially when W is large, this linear regression approach is problematic as classes can be *masked* by others.

   A loose but general rule is that if $W \geq 3$ classes are lined up, polynomial terms up to degree $W - 1$ are needed.

III. Linear Discriminant Analysis

1. **Bayes' Rule Classifier:** We view classification problem from the decision-theoretical perspective and use the 0-1 loss function.

   Let $\widetilde{G}$ be a classification rule. The expected prediction error (EPE) is

$$\begin{aligned} \text{EPE}(\widetilde{G}) &:= \mathbb{E}_{(X,G)}[L(G, \widetilde{G}(X))] \\ &= \mathbb{E}_X \left[\mathbb{E}_{G|X=\mathbf{x}} \left[L(G, \widetilde{G}(\mathbf{x})) \,|\, X = \mathbf{x} \right] \right] \\ &= \mathbb{E}_X \left[\sum_{w=1}^{W} L(w, \widetilde{G}(\mathbf{x})) \mathbb{P}(G = w \,|\, X = \mathbf{x}) \right]. \end{aligned}$$

   In order to find a classification rule $\widehat{G}$ that minimizes EPE, it is sufficient to consider

$$\begin{aligned} \widehat{G} &:= \underset{\widetilde{G}}{\arg\min} \left\{ \sum_{w=1}^{W} L(w, \widetilde{G}(\mathbf{x})) \mathbb{P}(G = w \,|\, X = \mathbf{x}) \right\} \\ &= \underset{\widetilde{G}}{\arg\min} \left\{ 1 - \mathbb{P}(G = \widetilde{G}(\mathbf{x}) \,|\, X = \mathbf{x}) \right\} \\ &= \underset{\widetilde{G}}{\arg\max} \left\{ \mathbb{P}(G = \widetilde{G}(\mathbf{x}) \,|\, X = \mathbf{x}) \right\}. \end{aligned} \tag{4}$$

In this view, we need to know the class posteriors $\mathbb{P}(G \,|\, X = \mathbf{x})$ for the optimal classification. Suppose that f_w is the class-conditional density of X in class $G = w$, and let π_w be the prior probability of Class w so that $\sum_{w=1}^{W} \pi_w = 1$. Then, Bayes theorem yields that

$$\mathbb{P}(G = w \,|\, X = \mathbf{x}) = \frac{\pi_w f_w(\mathbf{x})}{\sum_{j=1}^{W} \pi_j f_j(\mathbf{x})}. \tag{5}$$

Therefore, knowing f_w, the class conditional densities, is almost equivalent to knowing $\mathbb{P}(G = w \,|\, X = \mathbf{x})$.

2. **Assumption:** We assume that each class-conditional density has the multivariate Gaussian distribution as

$$f_w(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_w|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_w)^\top \boldsymbol{\Sigma}_w^{-1}(\mathbf{x} - \boldsymbol{\mu}_w)\right),$$

where the covariance matrix of Class w, $\boldsymbol{\Sigma}_w$, may differ from that of Class u, $\boldsymbol{\Sigma}_u$, for $w \neq u$.

3. **Linear Discriminant Analysis (LDA):** In the linear discriminant analysis (LDA), we further assume that $\boldsymbol{\Sigma}_w = \boldsymbol{\Sigma}$ for all $w = 1, \cdots, W$. Then, when we compare two classes w and u, it is sufficient to look at the log-ratio of the posterior probabilities, i.e.,

$$\begin{aligned}
&\log \frac{\mathbb{P}(G = w \,|\, X = \mathbf{x})}{\mathbb{P}(G = u \,|\, X = \mathbf{x})} \\
&= \log \frac{\pi_w}{\pi_u} + \log \frac{f_w(\mathbf{x})}{f_u(\mathbf{x})} \\
&= \log \frac{\pi_w}{\pi_u} + \left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_w)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_w) + \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_u)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_u)\right] \\
&= \log \frac{\pi_w}{\pi_u} - \frac{1}{2}(\boldsymbol{\mu}_w + \boldsymbol{\mu}_u)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_w - \boldsymbol{\mu}_u) + \mathbf{x}^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_w - \boldsymbol{\mu}_u),
\end{aligned}$$

which is a linear function in $\mathbf{x}$.

The *linear discriminant function* in LDA is defined by

$$\delta_w(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_w - \frac{1}{2}\boldsymbol{\mu}_w^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_w + \log \pi_w \qquad \text{for all } w = 1, \cdots, W, \tag{6}$$

which is an equivalent description of the decision rule with

$$G(\mathbf{x}) = \operatorname*{arg\,max}_{w=1,\cdots,W} \delta_w(\mathbf{x}). \tag{7}$$

Remark. This linear log-odds function implies that the *decision boundary* between Classes w and u, i.e., the set where $\mathbb{P}(G = w \,|\, X = \mathbf{x}) = \mathbb{P}(G = u \,|\, X = \mathbf{x})$, is linear in $\mathbf{x}$ and is a hyperplane in p-dimensional space. In other words, we can partition $\mathbb{R}^p$ into W subregions and the boundaries between two subregions is a hyperplane.

4. **Total Misclassification Probability of LDA When $W = 2$:** When there are only 2 classes, the LDA rule classifies to Class 2 if and only if

$$\begin{aligned} & \log \frac{\mathbb{P}(G = 2 \,|\, X)}{\mathbb{P}(G = 1 \,|\, X)} > 0 \\ \iff \ & \log\left(\frac{\pi_2}{\pi_1}\right) - \frac{1}{2}(\boldsymbol{\mu}_2 + \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + X^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) > 0. \end{aligned}$$

For simplicity, we let

$$\begin{aligned} u &:= \log\left(\frac{\pi_2}{\pi_1}\right) - \frac{1}{2}(\boldsymbol{\mu}_2 + \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1), \\ V &:= X^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1), \end{aligned}$$

and notice that u is a constant and V is a random variable. Also, notice that, for $i = 1, 2$, the expectation of V is

$$\mathbb{E}[V \,|\, X \text{ actually belongs to Class } i] = \boldsymbol{\mu}_i^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1),$$

and the variance is

$$\begin{aligned} \operatorname{Var}[V \,|\, X \text{ actually belongs to Class } i] &= (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1} \operatorname{Var}[X] \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \\ &= (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) =: \Delta^2, \end{aligned}$$

where Δ^2 is the squared Mahalanobis distance between the means of the two classes. Misclassification occurs when either

- $\mathbf{x}$ actually belongs to Class 1 but is assigned to Class 2, or
- $\mathbf{x}$ actually belongs to Class 2 but is assigned to Class 1.

The *total misclassification probability* is given by

$$\begin{aligned} \mathbb{P}(\text{Misclassification}) = \pi_1 \times \mathbb{P}(&\text{Classify } X \text{ to Class } 2 \,|\, X \text{ belongs to Class } 1) + \\ & \pi_2 \times \mathbb{P}(\text{Classify } X \text{ to Class } 1 \,|\, X \text{ belongs to Class } 2). \end{aligned}$$

Note that

$$\begin{aligned} & \mathbb{P}(\text{Classify } X \text{ to Class } 2 \,|\, X \text{ belongs to Class } 1) \\ = & \mathbb{P}\left(\log \frac{\mathbb{P}(Y = 2 \,|\, X)}{\mathbb{P}(Y = 1| \, X)} > 0 \,\middle|\, X \text{ belongs to Class } 1\right) \\ = & \mathbb{P}\Big(u + V > 0 \,\Big|\, X \text{ belongs to Class } 1\Big) \\ = & \mathbb{P}\left(\frac{V - \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)}{\Delta} > -\frac{u + \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)}{\Delta}\right) \\ = & \mathbb{P}\left(Z > \frac{\Delta}{2} - \frac{1}{\Delta}\log\left(\frac{\pi_2}{\pi_1}\right)\right) \\ = & \Phi\left(-\frac{\Delta}{2} + \frac{1}{\Delta}\log\left(\frac{\pi_2}{\pi_1}\right)\right), \end{aligned}$$

where Z denotes the standard normal random variable and Φ is the corresponding cumulative distribution function.

Similarly, we have

$$\mathbb{P}(\text{Classify } X \text{ to Class } 1 \,|\, X \text{ belongs to Class } 2) = \Phi\bigg(-\frac{\Delta}{2} - \frac{1}{\Delta}\log\Big(\frac{\pi_2}{\pi_1}\Big)\bigg).$$

Therefore, the total misclassification probability is

$$\mathbb{P}(\text{Misclassification}) = \pi_1 \times \Phi\bigg(-\frac{\Delta}{2} + \frac{1}{\Delta}\log\Big(\frac{\pi_2}{\pi_1}\Big)\bigg) + \pi_2 \times \Phi\bigg(-\frac{\Delta}{2} - \frac{1}{\Delta}\log\Big(\frac{\pi_2}{\pi_1}\Big)\bigg).$$

If, in particular, $\pi_1 = \pi_2 = \frac{1}{2}$, we have

$$\mathbb{P}(\text{Misclassification}) = \frac{1}{2}\Phi\Big(-\frac{\Delta}{2}\Big) + \frac{1}{2}\Phi\Big(-\frac{\Delta}{2}\Big) = \Phi\Big(-\frac{\Delta}{2}\Big).$$

5. **Estimation in LDA:** In practice, we do *not* know the parameters necessary to perform LDA, and we estimate them by the following approach:

- estimate π_w by
$$\hat{\pi}_w := \frac{n_w}{n}, \qquad \text{for all } w = 1, 2, \cdots, W,$$
where n_w is the number of observations in Class w;
- estimate $\boldsymbol{\mu}_w$ by
$$\hat{\boldsymbol{\mu}}_w := \frac{\sum_{\{i \,|\, g_i = w\}} \mathbf{x}_i}{n_w}, \qquad \text{for all } w = 1, 2, \cdots, W,$$
- estimate the common covariance matrix by
$$\widehat{\boldsymbol{\Sigma}} := \frac{1}{n - W} \sum_{w=1}^{W} \sum_{\{i \,|\, g_i = w\}} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_w)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_w)^\top.$$

Case for $W = 2$. When there are only 2 classes, the LDA rule classifies to Class 2 if and only if

$$\begin{aligned}
& \log \frac{\mathbb{P}(G = 2 \,|\, X = \mathbf{x})}{\mathbb{P}(G = 1 \,|\, X = \mathbf{x})} > 0 \\
\iff\ & \log\Big(\frac{n_2}{n_1}\Big) - \frac{1}{2}(\hat{\boldsymbol{\mu}}_2 + \hat{\boldsymbol{\mu}}_1)^\top \widehat{\boldsymbol{\Sigma}}^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) + \mathbf{x}^\top \widehat{\boldsymbol{\Sigma}}^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) > 0 \\
\iff\ & \mathbf{x}^\top \widehat{\boldsymbol{\Sigma}}^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) > \frac{1}{2}(\hat{\boldsymbol{\mu}}_2 + \hat{\boldsymbol{\mu}}_1)^\top \widehat{\boldsymbol{\Sigma}}^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) - \log\Big(\frac{n_2}{n_1}\Big),
\end{aligned} \tag{8}$$

and Class 1 otherwise.

6. **Computation for LDA:** Referring to (6) and (7), LDA classification rule is equivalent to the following minimization problem

$$G(\mathbf{x}) = \operatorname*{arg\,min}_{w \in \mathcal{W}} \left\{ \frac{1}{2} (\mathbf{x} - \hat{\boldsymbol{\mu}}_w)^\top \widehat{\boldsymbol{\Sigma}}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_w) - \log \hat{\pi}_w \right\}. \tag{9}$$

Using the eigen-decomposition of the symmetric positive definite matrix $\widehat{\boldsymbol{\Sigma}}$, we have

$$\widehat{\boldsymbol{\Sigma}} = \widehat{\mathbf{U}} \widehat{\mathbf{D}} \widehat{\mathbf{U}}^\top,$$

where $\widehat{\mathbf{U}} \in \mathbb{R}^{p \times p}$ is an orthogonal matrix and $\widehat{\mathbf{D}} \in \mathbb{R}^{p \times p}$ is a diagonal matrix, and obtain

$$\begin{aligned}
(\mathbf{x} - \hat{\boldsymbol{\mu}}_w)^\top \widehat{\boldsymbol{\Sigma}}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_w) &= (\mathbf{x} - \hat{\boldsymbol{\mu}}_w)^\top \widehat{\mathbf{U}} \widehat{\mathbf{D}}^{-1} \widehat{\mathbf{U}}^\top (\mathbf{x} - \hat{\boldsymbol{\mu}}_w) \\
&= \|\widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top (\mathbf{x} - \hat{\boldsymbol{\mu}}_w)\|_2^2 \\
&= \|\widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top \mathbf{x} - \widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top \hat{\boldsymbol{\mu}}_w\|_2^2,
\end{aligned}$$

which is the squared distance between the transformed variable $\tilde{\mathbf{x}} := \widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top \mathbf{x}$ and the transformed mean vector $\tilde{\boldsymbol{\mu}}_w := \widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top \hat{\boldsymbol{\mu}}_w$.

Hence, the LDA classifier can be implemented by the following steps:

(1) Find the eigen-decomposition of $\widehat{\boldsymbol{\Sigma}} = \widehat{\mathbf{U}} \widehat{\mathbf{D}} \widehat{\mathbf{U}}^\top$;

(2) Transform the class centroids $\tilde{\boldsymbol{\mu}}_w = \widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top \hat{\boldsymbol{\mu}}_w$, for all $w = 1, 2, \cdots, W$;

(3) Given any point $\mathbf{x} \in \mathbb{R}^P$, transform to $\tilde{\mathbf{x}} = \widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top \mathbf{x} \in \mathbb{R}^p$, and then classify according to the closest centroid in the *transformed* space, adjusting for class proportions.

Remark. Applying the transformation $\tilde{\mathbf{x}} = \widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top \mathbf{x}$ is basically *sphering* the data points, because if we consider $\mathbf{x}$ were a random variable with covariance matrix $\widehat{\boldsymbol{\Sigma}}$, then

$$\mathrm{Var}[\widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top \mathbf{x}] = \widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top \widehat{\boldsymbol{\Sigma}} \widehat{\mathbf{U}} \widehat{\mathbf{D}}^{-\frac{1}{2}} = \widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top (\widehat{\mathbf{U}} \widehat{\mathbf{D}} \widehat{\mathbf{U}}^\top) \widehat{\mathbf{U}} \widehat{\mathbf{D}}^{-\frac{1}{2}} = \mathbf{I}_p.$$

7. **Derivation of LDA from Regression When $W = 2$:** Suppose $W = 2$ with class sizes being n_1 and n_2, respectively. Label the target as $-\frac{n}{n_1}$ and $\frac{n}{n_2}$, respectively. Consider minimization of the least squares criterion

$$\begin{aligned}
L(\beta_0, \boldsymbol{\beta}) :=& \frac{1}{2} \sum_{i=1}^n (y_i - \beta_0 - \mathbf{x}_i^\top \boldsymbol{\beta})^2 \\
=& \frac{1}{2} \|\mathbf{Y} - \beta_0 \mathbf{1}_n - \mathbf{X} \boldsymbol{\beta}\|_2^2,
\end{aligned}$$

where $\mathbf{Y} := (y_1, y_2, \cdots, y_2)^\top \in \mathbb{R}^n$ and $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the design matrix. Let $(\hat{\beta}_0, \widehat{\boldsymbol{\beta}}) := \operatorname*{arg\,min}_{\beta_0, \boldsymbol{\beta}} L(\beta_0, \boldsymbol{\beta})$.

We first show that $\widehat{\boldsymbol{\beta}}$ satisfies

$$\big[(n-2)\widehat{\boldsymbol{\Sigma}} + n\widehat{\boldsymbol{\Sigma}}_B\big]\widehat{\boldsymbol{\beta}} = n(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1), \tag{10}$$

where $\widehat{\boldsymbol{\Sigma}}_B := \frac{n_1 n_2}{n^2}(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^\top$.

Taking the derivatives of $L(\beta_0, \boldsymbol{\beta})$ with respect to β_0 and $\boldsymbol{\beta}$ and setting the derivatives to 0 yield

$$\begin{aligned}
\frac{\partial}{\partial \beta_0} L(\beta_0, \boldsymbol{\beta}) &= \mathbf{1}_n^\top(\mathbf{Y} - \beta_0 \mathbf{1}_n - \mathbf{X}\boldsymbol{\beta}) \overset{\text{set}}{=} 0, \\
\frac{\partial}{\partial \boldsymbol{\beta}} L(\beta_0, \boldsymbol{\beta}) &= \mathbf{X}^\top(\mathbf{Y} - \beta_0 \mathbf{1}_n - \mathbf{X}\boldsymbol{\beta}) \overset{\text{set}}{=} 0.
\end{aligned}$$

The minimizer $(\hat{\beta}_0, \widehat{\boldsymbol{\beta}})$ must satisfy

$$0 = \mathbf{1}_n^\top \mathbf{Y} - \mathbf{1}_n^\top \mathbf{1}_n \hat{\beta}_0 - \mathbf{1}_n^\top \mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{1}_n^\top \mathbf{Y} - n\hat{\beta}_0 - \mathbf{1}_n^\top \mathbf{X}\widehat{\boldsymbol{\beta}}, \tag{11}$$

$$\mathbf{0}_p = \mathbf{X}^\top \mathbf{Y} - \hat{\beta}_0 \mathbf{X}^\top \mathbf{1}_n - \mathbf{X}^\top \mathbf{X}\widehat{\boldsymbol{\beta}}. \tag{12}$$

Due to the way we label y_i's, we have

$$\mathbf{1}_n^\top \mathbf{Y} = n_1\left(-\frac{n}{n_1}\right) + n_2\left(\frac{n}{n_1}\right) = 0.$$

Using this result, we can simplify (11) as $0 = -n\hat{\beta}_0 - \mathbf{1}_n^\top \mathbf{X}\widehat{\boldsymbol{\beta}}$, and thus

$$\hat{\beta}_0 = -\frac{1}{n}\mathbf{1}_n^\top \mathbf{X}\widehat{\boldsymbol{\beta}}.$$

Plugging the preceding equation into (12) yields

$$\mathbf{0}_p = \mathbf{X}^\top \mathbf{Y} + \frac{1}{n}(\mathbf{X}^\top \mathbf{1}_n)(\mathbf{1}_n^\top \mathbf{X})\widehat{\boldsymbol{\beta}} - \mathbf{X}^\top \mathbf{X}\widehat{\boldsymbol{\beta}},$$

that is,

$$\left(\mathbf{X}^\top \mathbf{X} - \frac{1}{n}(\mathbf{X}^\top \mathbf{1}_n)(\mathbf{1}_n^\top \mathbf{X})\right)\widehat{\boldsymbol{\beta}} = \mathbf{X}^\top \mathbf{Y}.$$

The desired result follows if we can show

$$\mathbf{X}^\top \mathbf{X} - \frac{1}{n}(\mathbf{X}^\top \mathbf{1}_n)(\mathbf{1}_n^\top \mathbf{X}) = (n-2)\widehat{\boldsymbol{\Sigma}} + n\widehat{\boldsymbol{\Sigma}}_B, \tag{13}$$

and

$$\mathbf{X}^\top \mathbf{Y} = n(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1). \tag{14}$$

To show (13), note that

$$\begin{aligned}
(n-2)\widehat{\mathbf{\Sigma}} &= \sum_{\{i \,|\, g_i=1\}} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_1)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_1)^\top + \sum_{\{i \,|\, g_i=2\}} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_2)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_2)^\top \\
&= \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top - \left(\sum_{\{i \,|\, g_i=1\}} \mathbf{x}_i \right) \hat{\boldsymbol{\mu}}_1^\top - \hat{\boldsymbol{\mu}}_1 \left(\sum_{\{i \,|\, g_i=1\}} \mathbf{x}_i \right)^\top + n_1 \hat{\boldsymbol{\mu}}_1 \hat{\boldsymbol{\mu}}_1^\top \\
&\qquad - \left(\sum_{\{i \,|\, g_i=2\}} \mathbf{x}_i \right) \hat{\boldsymbol{\mu}}_2^\top - \hat{\boldsymbol{\mu}}_2 \left(\sum_{\{i \,|\, g_i=2\}} \mathbf{x}_i \right)^\top + n_2 \hat{\boldsymbol{\mu}}_2 \hat{\boldsymbol{\mu}}_2^\top \\
&= \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top - n_1 \hat{\boldsymbol{\mu}}_1 \hat{\boldsymbol{\mu}}_1^\top - n_2 \hat{\boldsymbol{\mu}}_2 \hat{\boldsymbol{\mu}}_2^\top,
\end{aligned}$$

and that

$$n\widehat{\mathbf{\Sigma}}_B = \frac{n_1 n_2}{n} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^\top.$$

Their sum is

$$\begin{aligned}
(n-2)\widehat{\mathbf{\Sigma}} + n\widehat{\mathbf{\Sigma}}_B &= \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top - \frac{1}{n} \left(n n_1 \hat{\boldsymbol{\mu}}_1 \hat{\boldsymbol{\mu}}_1^\top + n n_2 \hat{\boldsymbol{\mu}}_2 \hat{\boldsymbol{\mu}}_2^\top - n_1 n_2 (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^\top \right) \\
&= \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top - \frac{1}{n} \left(n_1^2 \hat{\boldsymbol{\mu}}_1 \hat{\boldsymbol{\mu}}_1^\top + n_1 n_2 \hat{\boldsymbol{\mu}}_1 \hat{\boldsymbol{\mu}}_2^\top + n_1 n_2 \hat{\boldsymbol{\mu}}_2 \hat{\boldsymbol{\mu}}_1^\top + n_2 \hat{\boldsymbol{\mu}}_2 \hat{\boldsymbol{\mu}}_2^\top \right) \\
&= \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top - \frac{1}{n} \left(n_1 \hat{\boldsymbol{\mu}}_1 + n_2 \hat{\boldsymbol{\mu}}_2 \right) \left(n_1 \hat{\boldsymbol{\mu}}_1 + n_2 \hat{\boldsymbol{\mu}}_2 \right)^\top \\
&= \mathbf{X}^\top \mathbf{X} - \frac{1}{n} (\mathbf{X}^\top \mathbf{1}_n)(\mathbf{1}_n^\top \mathbf{X}).
\end{aligned}$$

To show (14), note that

$$\mathbf{X}^\top \mathbf{Y} = -\frac{n}{n_1} \sum_{\{i \,|\, g_i=1\}} \mathbf{x}_i + \frac{n}{n_2} \sum_{\{i \,|\, g_i=2\}} \mathbf{x}_i = -n\hat{\boldsymbol{\mu}}_1 + n\hat{\boldsymbol{\mu}}_2 = n(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1).$$

Hence, with the definition of $\widehat{\mathbf{\Sigma}}_B$ above, we see $\widehat{\mathbf{\Sigma}}_B \boldsymbol{\beta}$ is in the direction $(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)$. By (10), we conclude that

$$\widehat{\boldsymbol{\beta}} \propto \widehat{\mathbf{\Sigma}}(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1),$$

and that the least-squares regression coefficient is identical to the LDA coefficient, up to a scalar multiple.

Remarks.

(a) Since the derivation of the LDA via least squares above does *not* use a Gaussian distribution assumption for the features, one can extend LDA to non-Gaussian data. However, the derivation of the particular intercept or *cut-point* given in (8) does require Gaussian assumption.

(b) With two more classes, LDA is not the same as linear regression of the class indicator matrix.

8. **Quadratic Discriminant Analysis (QDA):** If we still assume that the class-conditional density functions are Gaussian but do *not* assume that all classes share the same covariance matrix $\mathbf{\Sigma}$ but rather possibly $\mathbf{\Sigma}_w \neq \mathbf{\Sigma}_u$ for different classes w and u, we obtain the *quadratic discriminant analysis (QDA).*

   The associated *quadratic discriminant function* is

$$\delta_w(\mathbf{x}) = -\frac{1}{2}\log|\mathbf{\Sigma}_w| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_w)^\top \mathbf{\Sigma}_w^{-1}(\mathbf{x}-\boldsymbol{\mu}_w) + \log \pi_w. \tag{15}$$

   The decision boundary between Classes w and u is described by a *quadratic* equation

$$\{\mathbf{x} \,|\, \delta_w(\mathbf{x}) = \delta_u(\mathbf{x})\}.$$

   Remark. When the parameters are *not* known, we can estimate them in a similar fashion as in LDA, but we need to treat the covariance matrix differently. In QDA, we estimate the covariance matrix for each class *separately.* However, if p is large, this means a dramatic increase in parameter estimation.

9. **Computations for QDA:** Let $\widehat{\mathbf{\Sigma}}_w$ be the estimate of the covariance matrix of Class w. By eigen-decomposition, we have $\widehat{\mathbf{\Sigma}}_w = \widehat{\mathbf{U}}_w \widehat{\mathbf{D}}_w \widehat{\mathbf{U}}_w^\top$, where $\widehat{\mathbf{U}}_w \in \mathbb{R}^{p\times p}$ is an orthogonal matrix and $\widehat{\mathbf{D}}_w$ is a diagonal matrix containing non-negative eigenvalues $d_{w,\ell}$. Then, the ingredients in $\delta_w(\mathbf{x})$ become

$$\log|\widehat{\mathbf{\Sigma}}_w| = \sum_{\ell} \log d_{w,\ell}$$

   and

$$(\mathbf{x}-\hat{\boldsymbol{\mu}}_w)^\top \widehat{\mathbf{\Sigma}}_k^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_w) = [\mathbf{U}_w^\top(\mathbf{x}-\hat{\boldsymbol{\mu}}_w)]^\top \mathbf{D}_w^{-1}[\mathbf{U}_w^\top(\mathbf{x}-\widehat{\boldsymbol{\mu}}_w)].$$

10. **Counting the Number of Parameters:**

   - In LDA, there are $(W-1)\times(p+1)$ parameters, since we only need the difference $\delta_w - \delta_W$ between the discriminant functions where W is the pre-chosen class, and each difference requires $(p+1)$ parameters;
   - In QDA, there are $(W-1)\times(\frac{1}{2}p(p+3)+1)$ parameters.

11. **Regularized Discriminant Analysis:**

   (a) *Approach 1:* The first approach is a compromise between LDA and QDA and shrinks the separate covariances of QDA toward a common covariance as in LDA. The regularized covariance matrices have the form

$$\widehat{\mathbf{\Sigma}}_w(\alpha) := \alpha \cdot \widehat{\mathbf{\Sigma}}_w + (1-\alpha)\cdot \widehat{\mathbf{\Sigma}}, \tag{16}$$

   where $\widehat{\mathbf{\Sigma}}$ is the pooled covariance matrix used in LDA. Here, $\alpha \in [0,1]$ is the tuning parameter that can be chosen based on the performance of the model on a validation set or by cross-validation and allows a continuum of models between LDA and QDA.

(b) *Approach 2:* The second approach allows $\widehat{\mathbf{\Sigma}}$ to be shrunk toward the scalar covariance matrix

$$\widehat{\mathbf{\Sigma}}(\gamma) = \gamma \cdot \widehat{\mathbf{\Sigma}} + (1-\gamma) \cdot \widehat{\sigma}^2 \mathbf{I} \tag{17}$$

for some $\gamma \in [0,1]$ and $\widehat{\sigma}^2 > 0$.

(c) *Approach 3:* The third approach replaces $\widehat{\mathbf{\Sigma}}$ in (16) by $\widehat{\mathbf{\Sigma}}(\gamma)$, leading to a more general family of covariance matrices $\widehat{\mathbf{\Sigma}}(\alpha, \gamma)$.

12. Observations from LDA Rule (9):

(a) The W centroids (each corresponding to one class) in p-dimensional input space lie in an *affine subspace* of dimensionality $\le W-1$.

This is because if $\mathbf{u}$ belongs to this affine subspace spanned by centroids $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_W$, we can find $\alpha_2, \cdots, \alpha_W$ such that

$$\begin{aligned} \mathbf{u} &= \left(1 - \sum_{i=2}^{W} \alpha_i\right) \boldsymbol{\mu}_1 + \alpha_2 \boldsymbol{\mu}_2 + \cdots + \alpha_W \boldsymbol{\mu}_W \\ &= \boldsymbol{\mu}_1 + \alpha_2(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \cdots + \alpha_W(\boldsymbol{\mu}_W - \boldsymbol{\mu}_1) \\ &= \boldsymbol{\mu}_1 + \alpha_2 \mathbf{d}_2 + \cdots + \alpha_W \mathbf{d}_W, \end{aligned}$$

where $\mathbf{d}_i := \boldsymbol{\mu}_i - \boldsymbol{\mu}_1$ for all $i = 2, 3, \cdots, W$.

(b) In locating the closest centroid, we can ignore distances orthogonal to this subspace of dimensionality $\le W-1$.

Let $\mathcal{M} \subseteq \mathbb{R}^p$ denote this affine subspace of dimensionality $\le W-1$. Then, for any $\mathbf{x} \in \mathbb{R}^p$, we have

$$\mathbf{x} = \mathcal{P}_{\mathcal{M}}\mathbf{x} + \mathcal{P}_{\mathcal{M}^\perp}\mathbf{x},$$

where $\mathcal{P}_{\mathcal{M}}\mathbf{x}$ and $\mathcal{P}_{\mathcal{M}^\perp}\mathbf{x}$ denote the projections of $\mathbf{x}$ onto $\mathcal{M}$ and $\mathcal{M}^\perp$, respectively. In addition, note that, for any $w = 1, 2, \cdots, W$,

$$\begin{aligned} (\mathbf{x} - \hat{\boldsymbol{\mu}}_w)^\top \widehat{\mathbf{\Sigma}}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_w) &= \|\tilde{\mathbf{x}} - \tilde{\boldsymbol{\mu}}_w\|_2^2 \\ &= \|\underbrace{\mathcal{P}_{\mathcal{M}}\mathbf{x} - \tilde{\boldsymbol{\mu}}_w}_{\in \mathcal{M}} + \underbrace{\mathcal{P}_{\mathcal{M}^\perp}\mathbf{x}}_{\in \mathcal{M}^\perp}\|_2^2 \\ &= \|\mathcal{P}_{\mathcal{M}}\mathbf{x} - \tilde{\boldsymbol{\mu}}_w\|_2^2 + \|\mathcal{P}_{\mathcal{M}^\perp}\mathbf{x}\|_2^2, \end{aligned}$$

where $\tilde{\mathbf{x}} = \widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top \mathbf{x} \in \mathbb{R}^p$ and $\tilde{\boldsymbol{\mu}}_w = \widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{U}}^\top \boldsymbol{\mu}_w \in \mathbb{R}^p$. Note that the term $\|\mathcal{P}_{\mathcal{M}^\perp}\mathbf{x}\|_2^2$ does *not* depend on w.

Therefore,

- the LDA classification rule is *unchanged* if we project the points to be classified onto the affine subspace $\mathcal{M}$ of dimensionality at most $W-1$, since the distances orthogonal to $\mathcal{M}$ does *not* matter;

- there is a fundamental dimension reduction in LDA, and we only need to consider the data in a subspace of dimension at most $W - 1$.

13. **Reduce-Rank Linear Discriminant Analysis, Part I — Fisher's Approach:**

    (a) *Problem Formulation:* Fisher posed the problem:

    > *Find the linear combination $Z = \mathbf{a}^\top X$ such that the between-class variance is maximized relative to the within-class variance.*

    (b) *Review of the Law of Total Variance:* Recall that, for a random vector X, we have

    $$\mathrm{Var}[X] = \mathrm{Var}\big[\mathbb{E}[X|Y]\big] + \mathbb{E}\big[\mathrm{Var}[X|Y]\big].$$

    An interpretation of this result is that the total variance of X, $\mathrm{Var}[X]$, is the sum of the between-class variance, $\mathrm{Var}[\mathbb{E}[X|Y]]$, and the within-class variance, $\mathbb{E}[\mathrm{Var}[X|Y]]$.

    Now, letting $\mathbf{a} \in \mathbb{R}^p$ be a constant vector, we have

    $$\begin{aligned}\mathrm{Var}[\langle \mathbf{a}, X\rangle] &= \mathrm{Var}\big[\mathbb{E}[\langle \mathbf{a}, X\rangle|Y]\big] + \mathbb{E}\big[\mathrm{Var}[\langle \mathbf{a}, X\rangle|Y]\big] \\ &= \mathbf{a}^\top \mathbf{B}\mathbf{a} + \mathbf{a}^\top \mathbf{W}\mathbf{a},\end{aligned}$$

    where $\mathbf{B} := \mathrm{Var}[\mathbb{E}[X|Y]]$ and $\mathbf{W} := \mathbb{E}[\mathrm{Var}[X|Y]]$.

    (c) *Calculation of Variances:* Let $\bar{\boldsymbol{\mu}} = \sum_{w=1}^{W} \pi_w \boldsymbol{\mu}_w$, where π_w denotes the prior probability of Class w and $\boldsymbol{\mu}_w$ denotes the conditional mean of X of Class w.

    i. Between-class variance:

    $$\mathbf{B} = \sum_{w=1}^{W} \pi_w (\boldsymbol{\mu}_w - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_w - \bar{\boldsymbol{\mu}})^\top;$$

    Note that $\mathbf{B}$ is of rank at most $W - 1$.

    ii. Within-class variance:

    $$\mathbf{W} = \sum_{w=1}^{W} \pi_w \,\mathrm{Var}[X \mid G = w];$$

    iii. Total variance:

    $$\mathbf{T} = \mathbf{W} + \mathbf{B}.$$

    (d) *Mathematical Formulation of Fisher's LDA:* Fisher's LDA problem can be formulated as the following maximization problem

    $$\underset{\mathbf{a}_1}{\text{maximize}} \; \frac{\mathbf{a}_1^\top \mathbf{B}\mathbf{a}_1}{\mathbf{a}_1^\top \mathbf{W}\mathbf{a}_1},$$

or, equivalently,

$$\underset{\mathbf{a}_1}{\text{maximize}}\ \mathbf{a}_1^\top \mathbf{B}\mathbf{a}_1 \qquad \text{subject to } \mathbf{a}_1^\top \mathbf{W}\mathbf{a}_1 = 1. \tag{18}$$

This problem is a *generalized eigenvalue problem.* To find a solution, first note that

$$1 = \mathbf{a}_1^\top \mathbf{W}\mathbf{a}_1 = \mathbf{a}_1^\top \mathbf{W}^{\frac{1}{2}}\mathbf{W}^{\frac{1}{2}}\mathbf{a}_1 = (\mathbf{W}^{\frac{1}{2}}\mathbf{a}_1)^\top \mathbf{W}^{\frac{1}{2}}\mathbf{a}_1.$$

Letting $\mathbf{c}_1 := \mathbf{W}^{\frac{1}{2}}\mathbf{a}_1$, we have $\mathbf{a}_1 = \mathbf{W}^{-\frac{1}{2}}\mathbf{c}_1$ and can transform the problem (18) to

$$\underset{\mathbf{c}_1}{\text{maximize}}\ \mathbf{c}_1^\top \mathbf{W}^{-\frac{1}{2}}\mathbf{B}\mathbf{W}^{-\frac{1}{2}}\mathbf{c}_1 \qquad \text{subject to } \|\mathbf{c}_1\|_2^2 = 1. \tag{19}$$

The maximizer of (19) is the eigenvector of $\mathbf{W}^{-\frac{1}{2}}\mathbf{B}\mathbf{W}^{-\frac{1}{2}}$ associated with its largest eigenvalue, denoted by $\mathbf{c}_1^*$. It follows that the optimal solution of (18), denoted by $\mathbf{a}_1^*$, is

$$\mathbf{a}_1^* = \mathbf{W}^{-\frac{1}{2}}\mathbf{c}_1^*.$$

(e) *Additional Directions:* We aim to find additional directions $\mathbf{a}_2, \cdots, \mathbf{a}_{W-1}$ that maximize the ratio between the between-class variance and the within-class variance, under the constraint that $\mathbf{a}_w$ is orthogonal in $\mathbf{W}$ to $\mathbf{a}_{w-1}, \mathbf{a}_{w-2}, \cdots, \mathbf{a}_1$, for all $w = 2, \cdots, W-1$.

The associated optimization problem can be collectively formulated as

$$\underset{\mathbf{a}_1,\cdots,\mathbf{a}_{W-1}}{\text{maximize}}\ \sum_{w=1}^{W-1} \mathbf{a}_w^\top \mathbf{B}\mathbf{a}_w \qquad \text{subject to } \mathbf{a}_w^\top \mathbf{W}\mathbf{a}_{w'} = \mathbb{1}(w = w') \text{ for all } w, w'. \tag{20}$$

To find the solution to (20), we first let $\mathbf{c}_w := \mathbf{W}^{\frac{1}{2}}\mathbf{a}_w$ for all $w = 1, \cdots, W-1$. The problem (20) then becomes

$$\underset{\mathbf{c}_1,\cdots,\mathbf{c}_{W-1}}{\text{maximize}}\ \sum_{w=1}^{W-1} \mathbf{c}_w^\top \mathbf{W}^{-\frac{1}{2}}\mathbf{B}\mathbf{W}^{-\frac{1}{2}}\mathbf{c}_w, \qquad \text{subject to } \mathbf{c}_w^\top \mathbf{c}_{w'} = \mathbb{1}(w = w'). \tag{21}$$

The optimal solution to (21) is the set of all eigenvectors of $\mathbf{W}^{-\frac{1}{2}}\mathbf{B}\mathbf{W}^{-\frac{1}{2}}$, denoted by $\mathbf{c}_1^*, \cdots, \mathbf{c}_{W-1}^*$. To obtain the optimal solution to (20), transform back as

$$\mathbf{a}_w^* = \mathbf{W}^{-\frac{1}{2}}\mathbf{c}_w^*, \qquad \text{for all } w = 1, \cdots, W-1.$$

Remark. These directions $\mathbf{a}_1^*, \cdots, \mathbf{a}_{W-1}^*$ are called the *discriminant directions.* There are at most $W-1$ of them, since $\mathbf{B}$ is of rank at most $W-1$.

(f) *Estimation:* All development above requires the knowledge of population quantities. In practice, when we only have access to data, we can estimate them as below:

i. $n_w := \sum_{i=1}^{n} \mathbb{1}(g_i = w)$, for all $w = 1, \cdots, W$;

ii. $\hat{\boldsymbol{\mu}}_w = \frac{1}{n_w} \sum_{\{i \,|\, g_i = w\}} \mathbf{x}_i$, for all $w = 1, \cdots, W$;

iii. $\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{w=1}^{W} n_w \hat{\boldsymbol{\mu}}_w = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$;

iv. $\widehat{\mathbf{B}} = \frac{1}{n} \sum_{w=1}^{W} n_w (\hat{\boldsymbol{\mu}}_w - \hat{\boldsymbol{\mu}})(\hat{\boldsymbol{\mu}}_w - \hat{\boldsymbol{\mu}})^\top$;

v. $\widehat{\mathbf{W}} = \frac{1}{n} \sum_{w=1}^{W} \sum_{\{i \,|\, g_i = w\}} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_w)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_w)^\top$.

In particular, we have the following decomposition

$$\widehat{\mathbf{B}} + \widehat{\mathbf{W}} = \widehat{\mathbf{S}},$$

where $\widehat{\mathbf{S}} := \frac{1}{n} \sum_{w=1}^{W} \sum_{\{i \,|\, g_i = w\}} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top$.

Source of Variations	df	Sum of Squares Matrix
Between classes	$W - 1$	$n\widehat{\mathbf{B}} = \sum_{w=1}^{W} n_w (\hat{\boldsymbol{\mu}}_w - \hat{\boldsymbol{\mu}})(\hat{\boldsymbol{\mu}}_w - \hat{\boldsymbol{\mu}})^\top$
Within classes	$n - W$	$n\widehat{\mathbf{W}} = \sum_{w=1}^{W} \sum_{\{i \,|\, g_i = w\}} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_w)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_w)^\top$
Total	$n - 1$	$n\widehat{\mathbf{S}} = \sum_{w=1}^{W} \sum_{\{i \,|\, g_i = w\}} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top$

14. **Reduce-Rank Linear Discriminant Analysis, Part II — Principal Component Subspace Approach:**

(a) *Main Idea:* We seek a $L < (W - 1)$ dimensional subspace $\mathcal{H}_L \subseteq \mathcal{H}_{W-1}$ that is *optimal* for LDA, where "optimal" means that the projected centroids were spread out as much as possible in terms of *variance.* This leads to finding principal component subspaces of the centroids themselves.

(b) *Procedure:*

i. Compute the $W \times p$ matrix of class centroids $\mathbf{M}$ and the common covariance matrix $\mathbf{W}$ (for *within-class* covariance);

ii. Compute $\mathbf{M}^* = \mathbf{M}\mathbf{W}^{-1/2}$ using the eigen-decomposition of $\mathbf{W}$;

iii. Compute $\mathbf{B}^*$, the covariance matrix of $\mathbf{M}^*$ ($\mathbf{B}$ for *between-class* covariance), and its spectral decomposition $\mathbf{B}^* = \mathbf{V}^* \mathbf{D}_B \mathbf{V}^{*\top}$. The columns $\mathbf{v}_\ell^*$ of $\mathbf{V}^*$ in sequence from first to last define the coordinates of the optimal subspaces;

iv. Combining all these operations the ℓ-th discriminant variable is given by $Z_\ell = \mathbf{v}_\ell^\top X$ with $\mathbf{v}_\ell = \mathbf{W}^{-1/2} \mathbf{v}_\ell^*$.

(c) *Remark:* This approach establishes the equivalence between the LDA and the PCA. The directions in the LDA are indeed the principal component directions of the feature variables standardized by the *within-class* covariance matrix.

15. **Connections between Fisher's Reduced-Rank Discriminant Analysis and Gaussian LDA:**

    - Gaussian LDA and Fisher's LDA are equivalent in the sense that the Gaussian LDA rule is simply the nearest centroid in the Fisher's LD space.
    - Gaussian LDA can be computed from the Fisher linear discriminant directions.

16. **Connections between Fisher's Reduced-Rank Discriminant Analysis and Regression of an Indicator Matrix:** It turns out that LDA amounts to the regression followed by a spectral decomposition of $\widehat{\mathbf{Y}}^\top \mathbf{Y}$. In particular, when $W = 2$, there is a single discriminant variable that is identical up to a scalar multiplication to either of the columns of $\widehat{\mathbf{Y}}$.

IV. Logistic Regression

1. **Introduction:** The logistic regression model arises from directly modeling the *posterior probabilities* of the W classes via *linear* functions in $\mathbf{x}$, while ensuring that they sum to one and remain in $[0, 1]$. The model is of the form

$$\begin{aligned}
\log \frac{\mathbb{P}(G = 1 \,|\, X = \mathbf{x})}{\mathbb{P}(G = W \,|\, X = \mathbf{x})} =& \beta_{1,0} + \boldsymbol{\beta}_1^\top \mathbf{x} \\
\log \frac{\mathbb{P}(G = 2 \,|\, X = \mathbf{x})}{\mathbb{P}(G = W \,|\, X = \mathbf{x})} =& \beta_{2,0} + \boldsymbol{\beta}_2^\top \mathbf{x} \\
&\vdots \\
\log \frac{\mathbb{P}(G = W-1 \,|\, X = \mathbf{x})}{\mathbb{P}(G = W \,|\, X = \mathbf{x})} =& \beta_{(W-1),0} + \boldsymbol{\beta}_{W-1}^\top \mathbf{x}.
\end{aligned}$$

By the model above, it is simple to obtain that

$$\begin{aligned}
\mathbb{P}(G = w \,|\, X = \mathbf{x}) =& \frac{\exp(\beta_{w,0} + \boldsymbol{\beta}_w^\top \mathbf{x})}{1 + \sum_{u=1}^{W-1} \exp(\beta_{u,0} + \boldsymbol{\beta}_u^\top \mathbf{x})}, \qquad \text{for all } w = 1, \cdots, W-1, \\
\mathbb{P}(G = W \,|\, X = \mathbf{x}) =& \frac{1}{1 + \sum_{u=1}^{W-1} \exp(\beta_{u,0} + \boldsymbol{\beta}_u^\top \mathbf{x})}.
\end{aligned}$$

It is easy to see that all probabilities above are positive and sum to 1.

Remark. Note that in the formulation above, we use the last class, i.e., Class W, as the denominator in the log-odds ratio, but the choice of denominator is arbitrary.

2. **Notation:** Let $\boldsymbol{\theta} := \{\beta_{1,0}, \boldsymbol{\beta}_1^\top, \cdots, \beta_{(W-1),0}, \boldsymbol{\beta}_{W-1}^\top\}$ and the probabilities

$$\mathbb{P}(G = w \,|\, X = \mathbf{x}) =: p_w(\mathbf{x}; \boldsymbol{\theta}), \qquad \text{for all } w = 1, \cdots, W-1.$$

3. **Fitting Logistic Regression Models — Introduction:** We use the method of maximum likelihood to fit logistic regression models.

We use the *multinomial distribution* and the log-likelihood function for n observations is

$$\ell(\boldsymbol{\theta}) := \sum_{i=1}^{n} \log p_{g_i}(\mathbf{x}_i; \boldsymbol{\theta}).$$

4. **Newton-Raphson Algorithm for Two Classes:** We code the two classes g_i via a 0/1 response y_i and

$$y_i = \begin{cases} 1, & \text{if } g_i = 1 \\ 0, & \text{if } g_i = 2 \end{cases}.$$

Also, let $p_1(\mathbf{x}; \boldsymbol{\beta}) =: p(\mathbf{x}; \boldsymbol{\beta})$ and $p_2(\mathbf{x}; \boldsymbol{\beta}) = 1 - p(\mathbf{x}; \boldsymbol{\beta})$, where $\boldsymbol{\beta} := (\beta_{1,0}, \boldsymbol{\beta}_1^\top)^\top \in \mathbb{R}^{p+1}$. Then, including the constant term 1 into the $\mathbf{x}$ vector, the log-likelihood can be rewritten as

$$\begin{aligned} \ell(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \Big\{ y_i \cdot \log p(\mathbf{x}_i; \boldsymbol{\beta}) + (1 - y_i) \cdot \log(1 - p(\mathbf{x}_i; \boldsymbol{\beta})) \Big\} \\ &= \sum_{i=1}^{n} \Big\{ y_i \cdot \big(\boldsymbol{\beta}^\top \mathbf{x} - \log(1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i)) \big) + (1 - y_i) \cdot \big(- \log(1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i)) \big) \Big\} \\ &= \sum_{i=1}^{n} \Big\{ y_i \boldsymbol{\beta}^\top \mathbf{x}_i - \log(1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i)) \Big\}. \end{aligned} \tag{22}$$

To find the maximum, we first find the *score function*, the first-order derivative of ℓ with respect to $\boldsymbol{\beta}$, which is

$$\frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \mathbf{x}_i \left(y_i - \frac{\exp(\boldsymbol{\beta}^\top \mathbf{x}_i)}{1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i)} \right) = \sum_{i=1}^{n} \mathbf{x}_i \Big(y_i - p(\mathbf{x}_i; \boldsymbol{\beta}) \Big)$$

and set the preceding equation to $\mathbf{0}_{p+1}$, which are $(p+1)$ equations nonlinear in $\boldsymbol{\beta}$.

Since the first component of the vector $\mathbf{x}_i$ is 1 for all $i = 1, \cdots, n$, the first equation is

$$\sum_{i=1}^{n} \big(y_i - p(\mathbf{x}_i; \boldsymbol{\beta}) \big) = 0 \qquad \Longleftrightarrow \qquad \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} p(\mathbf{x}_i; \boldsymbol{\beta}),$$

which can be interpreted that the *expected* number of class ones matches the *observed* number.

To compute the maximizer of ℓ, we use the *Newton-Raphson algorithm*, which requires the second-order derivative of ℓ given by

$$\begin{aligned}\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} &= \sum_{i=1}^n \mathbf{x}_i \bigg(- \frac{\partial}{\partial \boldsymbol{\beta}^\top} \bigg(\frac{\exp(\boldsymbol{\beta}^\top \mathbf{x}_i)}{1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i)} \bigg) \bigg) \\ &= - \sum_{i=1}^n \mathbf{x}_i \bigg(\frac{\exp(\boldsymbol{\beta}^\top \mathbf{x}_i) \mathbf{x}_i^\top (1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i)) - \exp(\boldsymbol{\beta}^\top \mathbf{x}_i) \exp(\boldsymbol{\beta}^\top \mathbf{x}_i) \mathbf{x}_i^\top}{(1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i))^2} \bigg) \\ &= - \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \frac{\exp(\boldsymbol{\beta}^\top \mathbf{x}_i)}{(1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i))^2} \\ &= - \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \frac{\exp(\boldsymbol{\beta}^\top \mathbf{x}_i)}{1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i)} \cdot \frac{1}{1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i)} \\ &= - \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top p(\mathbf{x}_i; \boldsymbol{\beta})(1 - p(\mathbf{x}_i; \boldsymbol{\beta})).\end{aligned}$$

Then, given $\boldsymbol{\beta}^{\text{old}}$, the Newton-Raphson update is

$$\boldsymbol{\beta}^{(\text{new})} = \boldsymbol{\beta}^{(\text{old})} - \bigg(\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} \bigg)^{-1} \frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \bigg|_{\boldsymbol{\beta} = \boldsymbol{\beta}^{(\text{old})}}. \tag{23}$$

We write everything in the matrix form. Let

- $\mathbf{Y} \in \mathbb{R}^n$ denote the vector with values of y_i's,
- $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$ be the design matrix,
- $\mathbf{p} \in \mathbb{R}^n$ be the vector of fitted probabilities with the i-th element being $p(\mathbf{x}_i; \boldsymbol{\beta}^{(\text{old})})$, and
- $\mathbf{W} \in \mathbb{R}^{n \times n}$ be the diagonal matrix with the i-th element being $p(\mathbf{x}_i; \boldsymbol{\beta}^{(\text{old})}) \cdot (1 - p(\mathbf{x}_i; \boldsymbol{\beta}^{(\text{old})}))$.

Then, we have

$$\frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \mathbf{X}^\top (\mathbf{Y} - \mathbf{p}), \qquad \text{and} \qquad \frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = - \mathbf{X}^\top \mathbf{W} \mathbf{X}.$$

With these ingredients defined, the Newton-Raphson update becomes

$$\begin{aligned}\boldsymbol{\beta}^{\text{new}} &= \boldsymbol{\beta}^{(\text{old})} - \bigg(\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} \bigg)^{-1} \frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \bigg|_{\boldsymbol{\beta} = \boldsymbol{\beta}^{(\text{old})}} \\ &= \boldsymbol{\beta}^{(\text{old})} + (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top (\mathbf{Y} - \mathbf{p}) \\ &= (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} (\mathbf{X} \boldsymbol{\beta}^{(\text{old})} + \mathbf{W}^{-1} (\mathbf{Y} - \mathbf{p})) \\ &= (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \mathbf{z},\end{aligned}$$

where we re-express this update as a weighted least squares update and define

$$\mathbf{z} := \mathbf{X}\boldsymbol{\beta}^{\text{old}} + \mathbf{W}^{-1}(\mathbf{Y} - \mathbf{p}),$$

known as the *adjusted response* or *working response*. Then, the equations can be solved iteratively.

Remarks.

(a) This algorithm is referred to as *iteratively reweighted least squares*, abbreviated as *IRLS*, since each iteration solves the following weighted least squares problem

$$\boldsymbol{\beta}^{\text{new}} \leftarrow \arg\min_{\boldsymbol{\beta}} (\mathbf{z} - \mathbf{X}\boldsymbol{\beta})^{\top}\mathbf{W}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}).$$

(b) One can choose a starting point arbitrarily, but convergence is *never* guaranteed. *Typically* the algorithm does converge as the log-likelihood is concave, but *overshooting* can occur.

5. **Fitting Algorithm for More Than Two Classes:** Consider the case when there are more than 2 classes, i.e., $W \geq 3$.

- The Newton-Raphson algorithm can also be expressed as an iteratively reweighted least squares algorithm, but with a vector of $W - 1$ responses and a non-diagonal weight matrix per observation;
- Another choice for algorithm is the coordinate-descent methods;
- The `R` package `glmnet` can fit very large logistic regression problems efficiently, both in n and p.

6. **Quadratic Approximations and Inference:**

(a) The maximum likelihood estimates $\widehat{\boldsymbol{\beta}}$ satisfy a *self-consistency* relationship: they are the coefficients of a weighted least squares fit with the responses

$$z_i = \mathbf{x}_i^{\top}\widehat{\boldsymbol{\beta}} + \frac{y_i - \hat{p}_i}{\hat{p}_i(1 - \hat{p}_i)}$$

and the weights are $w_i = \hat{p}_i(1 - \hat{p}_i)$, both of which depend on $\widehat{\boldsymbol{\beta}}$ itself.

(b) The *weighted residual sum-of-squares* is the Pearson chi-square statistic

$$\sum_{i=1}^{n} \frac{(y_i - \hat{p}_i)^2}{\hat{p}_i(1 - \hat{p}_i)},$$

a quadratic approximation to the deviance.

(c) Asymptotically, if the model is correct, then $\widehat{\boldsymbol{\beta}}$ is consistent.

(d) By the central limit theorem, as $n \to \infty$, the distribution of $\widehat{\boldsymbol{\beta}}$ converges to $\text{Normal}(\boldsymbol{\beta}, (\mathbf{XWX})^{-1})$.

7. **L_1 Regularized Logistic Regression:** The L_1 penalty used in the lasso for variable selection and estimation can be applied to logistic regression. We maximize a penalized likelihood function

$$\underset{\beta_0,\boldsymbol{\beta}}{\text{maximize}} \left\{ \sum_{i=1}^{n} \left[y_i \cdot (\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i) - \log(1 + \exp(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i)) - \lambda \sum_{j=1}^{p} |\beta_j| \right] \right\}, \tag{24}$$

where we do *not* penalize the intercept. To compute the maximizer, one has several options:

- using nonlinear programming method due to the concavity of the problem;
- using the quadratic approximations and applying a weighted lasso algorithm to (24). In this case, it turns out that the variables with non-zero coefficients have the form

$$\mathbf{x}_j^\top (\mathbf{Y} - \mathbf{p}) = \lambda \cdot \text{sign}(\beta_j);$$

  the active variables are tied in their *generalized correlation* with the residuals.

8. **Comparisons between LDA and Logistic Regression:**

   (a) *Similarities:*

   i. Both methods attempt to approximate the Bayes' rule classifier

$$\begin{aligned} \widehat{G} &= \underset{w \in \mathcal{W}}{\arg\max} \left\{ \mathbb{P}(G = w \,|\, X = \mathbf{x}) \right\} \\ &= \underset{w \in \mathcal{W}}{\arg\max} \left\{ \mathbb{P}(G = w) \mathbb{P}(X = \mathbf{x} \,|\, G = w) \right\}; \end{aligned}$$

   ii. The log-posterior odds take on similar forms:

   - Under the Gaussian and common covariance matrix assumptions of the LDA, the log-posterior odds between Classes w and W is of the form

$$\begin{aligned} &\log \frac{\mathbb{P}(G = w \,|\, X = \mathbf{x})}{\mathbb{P}(G = W \,|\, X = \mathbf{x})} \\ &= \log\left(\frac{\pi_w}{\pi_W}\right) - \frac{1}{2}(\boldsymbol{\mu}_w + \boldsymbol{\mu}_W)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_w - \boldsymbol{\mu}_W) + \mathbf{x}^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_w - \boldsymbol{\mu}_W) \\ &= \alpha_{w,0} + \boldsymbol{\alpha}_w^\top \mathbf{x}. \end{aligned}$$

   - Under the logistic construction, the log-odds is modeled as

$$\log \frac{\mathbb{P}(G = w \,|\, X = \mathbf{x})}{\mathbb{P}(G = W \,|\, X = \mathbf{x})} = \beta_{w,0} + \boldsymbol{\beta}_w^\top \mathbf{x}.$$

   (b) *Differences:*

   - The logistic regression model is more general and has fewer assumptions;

- The way the linear coefficient vectors are estimated are different;
- The logistic regression model assumes a model for $\mathbb{P}(G = w \mid X = \mathbf{x})$ directly, but the LDA assumes a model for $\mathbb{P}(X \mid G = w)$ and use the Bayes' rule to model $\mathbb{P}(G = w \mid X)$;
- The joint density of G and X is

  $$\mathbb{P}(X, G = w) = \mathbb{P}(X)\mathbb{P}(G = w \mid X),$$

  where $\mathbb{P}(X)$ denotes the marginal density of the inputs X. For both LDA and logistic regression, the second term on the RHS has the logit-linear form

  $$\mathbb{P}(G = w \mid X = \mathbf{x}) = \frac{\exp(\beta_{w,0} + \boldsymbol{\beta}_w^\top \mathbf{x})}{1 + \sum_{u=1}^{W-1} \exp(\beta_{u,0} + \boldsymbol{\beta}_u^\top \mathbf{x})}.$$

  - In *logistic regression*, the marginal density of X can be an *arbitrary* density function $\mathbb{P}(X)$ and we fit the parameters of $\mathbb{P}(G \mid X)$ by maximizing the conditional likelihood. Here, we ignore $\mathbb{P}(X)$.
  - In *LDA*, we fit parameters by maximizing the full log-likelihood based on the joint density

    $$\mathbb{P}(X, G = w) = \phi(X; \boldsymbol{\mu}_w, \boldsymbol{\Sigma}) \cdot \pi_w,$$

    where $\phi(\,\cdot\,; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the Gaussian density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Even though these parameters are unknown, they can be estimated from data. Here, the density of X does play a role and is a mixture of Gaussian densities

    $$p(\mathbf{x}) = \sum_{w=1}^{W} \pi_w \cdot \phi(\mathbf{x}; \boldsymbol{\mu}_w, \boldsymbol{\Sigma}).$$

- By putting additional model assumptions, the approach by LDA is more efficient and has lower variance;
- LDA may *not* be very robust to gross outliers, but outliers are reduced in importance in logistic regression.

(c) *Conclusion:* In general, logistic regression model is a safer and more robust approach comparing to the LDA model. Nevertheless, both approaches provide similar results in most cases.

V. Separating Hyperplanes

1. **Main Idea:** The separating hyperplane classifiers construct linear decision boundaries that *explicitly* try to separate the data into different classes as well as possible.

2. **Setup:** We consider the case of $W = 2$ and let the class label be $\mathcal{W} = \{+1, -1\}$.

3. **Separating Hyperplane Classifier:** Separating hyperplane classifiers are in the form of a linear combination of the input features, i.e.,

$$\left\{ \mathbf{x} := (x_1, \cdots, x_p)^\top \in \mathbb{R}^p \,\middle|\, \hat{\beta}_0 + \sum_{i=1}^{p} \hat{\beta}_i x_i = 0 \right\}.$$

4. **A Brief Review of Linear Algebra:** Let L be a hyperplane or affine set in $\mathbb{R}^p$.

   - Any points $\mathbf{x} \in L \subset \mathbb{R}^p$ can be characterized as

   $$\left\{ \mathbf{x} \in \mathbb{R}^p \,\middle|\, f(\mathbf{x}) := \beta_0 + \boldsymbol{\beta}^\top \mathbf{x} = 0 \right\},$$

   where $\beta_0 \in \mathbb{R}$ and $\boldsymbol{\beta} \in \mathbb{R}^p$ are fixed.
   - For any two point $\mathbf{x}_1$ and $\mathbf{x}_2$ lying in L, we must have $\boldsymbol{\beta}^\top(\mathbf{x}_1 - \mathbf{x}_2) = 0$, and hence, $\boldsymbol{\beta}^* := \frac{\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|_2}$ is the unit vector normal to the surface of L;
   - For any point $\mathbf{x}_0 \in L$, we have $\beta_0 = -\boldsymbol{\beta}^\top \mathbf{x}_0$;
   - The signed distance of any point $\mathbf{x}$ to L is given by

   $$\boldsymbol{\beta}^{*\top}(\mathbf{x} - \mathbf{x}_0) = \frac{1}{\|\boldsymbol{\beta}\|_2}(\boldsymbol{\beta}^\top x + \beta_0) = \frac{1}{\|\nabla f(\mathbf{x})\|_2} f(\mathbf{x}).$$

   Thus, $f(\mathbf{x})$ is proportional to the signed distance from $\mathbf{x}$ to the hyperplane defined by $f(\mathbf{x}) = 0$.

5. **Perceptron:** A separating hyperplane classifier that returns the sign is called *perceptron.*

6. **Rosenblatt's Perceptron Learning Algorithm:**

   (a) *Main Idea:* The perceptron learning algorithm attempts to find a separating hyperplane by *minimizing* the distance of misclassified points to the decision boundary.

   (b) *Observations:* Note that

      i. if $y_i = +1$ and if y_i is misclassified, we have $\beta_0 + \boldsymbol{\beta}^\top \mathbf{x} < 0$;
      ii. if $y_i = -1$ and if y_i is misclassified, we have $\beta_0 + \boldsymbol{\beta}^\top \mathbf{x} > 0$.

   (c) *Mathematical Formulation:* We minimize

   $$D(\boldsymbol{\beta}, \beta_0) := -\sum_{i \in \mathcal{M}} y_i \cdot (\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0), \tag{25}$$

   where $\mathcal{M}$ indexes the set of misclassified points.

   (d) *Properties of D:*

   - $D(\boldsymbol{\beta}, \beta_0) \geq 0$;
   - $D(\boldsymbol{\beta}, \beta_0)$ is *proportional* to the distance of the misclassified points to the decision boundary defined by $\boldsymbol{\beta}^\top \mathbf{x} + \beta_0 = 0$.

(e) *Algorithm:* Assume that $\mathcal{M}$ is *fixed*. the gradient of D is

$$\frac{\partial D(\boldsymbol{\beta}, \beta_0)}{\partial \boldsymbol{\beta}} = - \sum_{i \in \mathcal{M}} y_i \mathbf{x}_i, \qquad \text{and} \qquad \frac{\partial D(\boldsymbol{\beta}, \beta_0)}{\partial \beta_0} = - \sum_{i \in \mathcal{M}} y_i.$$

The algorithm uses *stochastic gradient descent*, and a step is taken after *each* observation is visited. The parameters are updated by

$$\begin{pmatrix} \boldsymbol{\beta} \\ \beta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \boldsymbol{\beta} \\ \beta_0 \end{pmatrix} + \rho \begin{pmatrix} y_i \mathbf{x}_i \\ y_i \end{pmatrix},$$

where $\rho > 0$ is the *learning rate* (also known as the step size).

(f) *Convergence Property:* If the classes are *linearly separable*, the algorithm converges to a separating hyperplane in a finite number of steps.

(g) *Problems of This Algorithm:*

- When data are separable, there are many solutions, and which one is found depends on the starting values;
- The "finite" number of steps can be very large — the smaller the gap, the longer the time to find it;
- When the data are *not* separable, the algorithm will *not* converge, and cycles develop. The cycles can be long and therefore hard to detect.

7. Optimal Separating Hyperplane:

(a) *Main Idea:* The *optimal separating hyperplane* separates the two classes and *maximizes* the distance to the closest point from either class.

(b) *Advantages:* This approach

- provides a *unique* solution to the separating hyperplane problem; and
- maximizes the margin between the two classes on the training data, leading to better classification performance on test data.

(c) *Mathematical Formulation:*

$$\begin{aligned} &\underset{\beta_0, \boldsymbol{\beta}, \|\boldsymbol{\beta}\|_2 = 1}{\text{maximize}} \, M, \\ &\qquad \text{subject to } y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq M, \text{ for all } i = 1, \cdots, n. \end{aligned} \tag{26}$$

The conditions ensure all the points are at least a signed distance M from the decision boundary determined by β and β_0.

(d) *Equivalent Formulation 1:* We can get rid of the constraint $\|\boldsymbol{\beta}\|_2 = 1$ by replacing the conditions with

$$\frac{1}{\|\boldsymbol{\beta}\|} y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq M,$$

or equivalent,

$$y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq M \|\boldsymbol{\beta}\|_2.$$

Note that β_0 here may be different from the one appearing in (26).

(e) *Equivalent Formulation 2:* Note that *for any $\boldsymbol{\beta}$ and β_0 satisfying the inequalities above, any positively scaled multiple satisfies them too. We can arbitrarily set* $\|\boldsymbol{\beta}\|_2 = \frac{1}{M}$, and rewrite the original optimization problem (26) as

$$\begin{aligned} &\underset{\beta_0, \boldsymbol{\beta}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{\beta}\|_2^2, \\ &\quad \text{subject to } y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq 1 \text{ for all } i = 1, \cdots, n. \end{aligned} \tag{27}$$

This means that the constraints define an *empty margin* around the linear decision boundary of thickness $1/\|\boldsymbol{\beta}\|_2$, and we choose $\boldsymbol{\beta}$ and β_0 to maximize its thickness.

Remark. Note that (27) is a convex optimization problem with a quadratic objective function and linear constraints.

(f) *Characterizing the Solution of (27):* The primal Lagrangian function to be minimized with respect to $\boldsymbol{\beta}$ and β_0 is

$$L_P(\boldsymbol{\beta}, \beta_0) := \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 - \sum_{i=1}^n \alpha_i \big[y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) - 1\big]. \tag{28}$$

We set the first-order derivatives to be 0, and obtain

$$\begin{aligned} \frac{\partial L_P(\boldsymbol{\beta}, \beta_0)}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = 0 \quad &\Longleftrightarrow \quad \boldsymbol{\beta} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \\ \frac{\partial L_P(\boldsymbol{\beta}, \beta_0)}{\partial \beta_0} = -\sum_{i=1}^n \alpha_i y_i = 0 \quad &\Longleftrightarrow \quad \sum_{i=1}^n \alpha_i y_i = 0. \end{aligned}$$

Substituting back to L_P, we obtain the *dual function*

$$L_D(\alpha_1, \cdots, \alpha_n) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n \sum_{k=1}^n \alpha_i \alpha_k y_i y_k \mathbf{x}_i^\top \mathbf{x}_k.$$

Letting $\boldsymbol{\alpha} := (\alpha_1, \cdots, \alpha_n)^\top$, the corresponding dual problem to (27) is

$$\begin{aligned} &\underset{\boldsymbol{\alpha}}{\text{maximize}} \ L_D(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n \sum_{k=1}^n \alpha_i \alpha_k y_i y_k \mathbf{x}_i^\top \mathbf{x}_k \\ &\quad \text{subject to } \alpha_i \geq 0 \text{ for all } i = 1, \cdots, n, \quad \sum_{i=1}^n \alpha_i y_i = 0. \end{aligned} \tag{29}$$

(g) *KKT Conditions for (27):* The *Karush-Kuhn-Tucker (KKT) conditions* for (27) are

i. Primal stationarity:

$$\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i - \boldsymbol{\beta} = 0, \qquad \text{and} \qquad \sum_{i=1}^n \alpha_i y_i = 0,$$

ii. Dual feasibility: $\alpha_i \geq 0$ for all $i = 1, 2, \cdots, n$,

iii. Complementary slackness:

$$\alpha_i [y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) - 1] = 0, \qquad \text{for all } i = 1, \cdots, n.$$

(h) *Implications from KKT Conditions:*

i. From complementary slackness:

- if $\alpha_i > 0$, then $y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) = 1$, and $\mathbf{x}_i$ is on the boundary of the margin;
- if $y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) > 1$, $\mathbf{x}_i$ is *not* on the boundary and $\alpha_i = 0$.

ii. From primal stationary, $\boldsymbol{\beta}$ is determined in terms of a linear combination of the support points $\mathbf{x}_i$, the points that are on the decision boundary so that $\alpha_i > 0$. This is how the name "*support vector machine*" comes from.

(i) *Optimal Separating Hyperplane and Classification Rule:* The *optimal separating hyperplane* produces a function

$$\hat{f}(\mathbf{x}) = \mathbf{x}^\top \widehat{\boldsymbol{\beta}} + \hat{\beta}_0 = \mathbf{x}^\top \left(\sum_{i=1}^n \hat{\alpha}_i y_i \mathbf{x}_i \right) + \hat{\beta}_0 = \sum_{i=1}^n \hat{\alpha}_i y_i \mathbf{x}_i^\top \mathbf{x} + \hat{\beta}_0, \tag{30}$$

for classifying new observations, i.e.,

$$\widehat{G}(\mathbf{x}) = \text{sign}\big(\hat{f}(\mathbf{x})\big).$$

In (30), $(\widehat{\boldsymbol{\beta}}, \hat{\beta}_0)$ is the minimizer of L_P in (28), and $(\hat{\alpha}_1, \hat{\alpha}_2, \cdots, \hat{\alpha}_n)^\top$ is the maximizer of L_D in (29).

(j) *On* $\|\widehat{\boldsymbol{\beta}}\|_2$*:* Let $\widehat{\boldsymbol{\beta}}$ be the minimizer of L_P in (28). We derive $\|\widehat{\boldsymbol{\beta}}\|_2$. Using the complementary slackness condition, we have

$$\begin{aligned}
0 &= \sum_{i=1}^n \hat{\alpha}_i [y_i(\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}} + \hat{\beta}_0) - 1] \\
&= \sum_{i=1}^n \hat{\alpha}_i \left[y_i \left(\mathbf{x}_i^\top \sum_{j=1}^n \hat{\alpha}_j y_j \mathbf{x}_j + \hat{\beta}_0 \right) - 1 \right] \\
&= \sum_{i=1}^n \sum_{j=1}^n \hat{\alpha}_i \hat{\alpha}_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + \hat{\beta}_0 \sum_{i=1}^n \hat{\alpha}_i y_i - \sum_{i=1}^n \hat{\alpha}_i \\
&= \sum_{i=1}^n \sum_{j=1}^n \hat{\alpha}_i \hat{\alpha}_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j - \sum_{i=1}^n \hat{\alpha}_i.
\end{aligned}$$

As a consequence, we have

$$\|\widehat{\boldsymbol{\beta}}\|_2^2 = \sum_{i=1}^n \sum_{j=1}^n \hat{\alpha}_i \hat{\alpha}_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j = \sum_{i=1}^n \hat{\alpha}_i.$$

Thus, the optimal hyperplane has the maximum margin $2/\|\widehat{\boldsymbol{\beta}}_2\|$, where

$$\frac{1}{\|\widehat{\boldsymbol{\beta}}\|_2} = \left(\sum_{i=1}^{n} \hat{\alpha}_i\right)^{-\frac{1}{2}}.$$

In addition, since $\hat{\alpha}_i = 0$ if the i-th observation does *not* lie on the boundary of the margin (i.e., $\mathbf{x}_i$ is not the support vectors), we have

$$\frac{1}{\|\widehat{\boldsymbol{\beta}}\|_2} = \left(\sum_{i\in \text{SV}} \hat{\alpha}_i\right)^{-\frac{1}{2}},$$

where $\text{SV} \subset \{1, 2, \cdots, n\}$ denotes the subset of indices that identify the support vectors.

(k) *Remarks:*

- A large margin on the training data will lead to good separation on the test data;
- The fact that the solution depends *only* on *support points* suggests that the optimal hyperplane focuses more on the points that count, and is *more robust* to model misspecification. The LDA solution, on the other hand, depends on *all* of the data, even points far away from the decision boundary;
- When the data are *not* separable, there will be no feasible solution to this problem, and an alternative formulation is needed.

VI. Evaluating a Classifier

1. Goal: We discuss how to evaluate the performance of a classifier in this section.

Note that the metrics discussed here are *not* restricted to the linear classification methods, but are applicable to all kinds of classifiers, including both

(a) the linear ones introduced in this chapter, and

(b) the non-linear ones that will be introduced in later chapters.

Remark. We focus on binary classifiers only.

2. True Positive, False Positive, True Negative, and False Negative:

(a) *Scenario:* Consider the cancer screening example. For each person tested, there are

i. a true label of whether this person has cancer or not, and

ii. a predicted label made by the classifier.

(b) *True Positive:* If this person actually has cancer and the classifier predicts this person has cancer, the prediction is called a *true positive.*

(c) *False Positive:* If this person actually does *not* have cancer but the classifier predicts this person has cancer, the prediction is called a *false positive.*

(d) *True Negative:* If this person actually does *not* have cancer and the classifier predicts this person does *not* have cancer, the prediction is called a *true negative.*

(e) *False Negative:* If this person actually has cancer but the classifier predicts this person does *not* have cancer, the prediction is called a *false negative.*

Remark. The false positives are also known as *type 1 errors*, and the false negatives are called *type 2 errors.*

3. **Confusion Matrix:** If we let

(a) n be the total number of people taking the test,

(b) n_{TP} be the number of true positives,

(c) n_{FP} be the number of false positives,

(d) n_{TN} be the number of true negatives, and

(e) n_{FN} be the number of false negatives,

then

$$n = n_{\text{TP}} + n_{\text{FP}} + n_{\text{TN}} + n_{\text{FN}},$$

and they can be represented in a *confusion matrix* as

		Predicted	
		Positive	Negative
Actual	Positive	n_{TP}	n_{FN}
	Negative	n_{FP}	n_{TN}

4. **Accuracy:** *Accuracy* is measured by the fraction of correct classifications and is given by

$$\text{Accuracy} = \frac{n_{\text{TP}} + n_{\text{TN}}}{n_{\text{TP}} + n_{\text{FP}} + n_{\text{TN}} + n_{\text{FN}}} = \frac{n_{\text{TP}} + n_{\text{TN}}}{n}.$$

Remark. When there are strongly imbalanced classes, accuracy can be misleading.

For example, if there are 1,000 patients in total and is only 1 person with cancer, a naive classifier that simply decides that nobody has cancer will achieve 99.9% accuracy, which is completely useless.

5. **Precision:** *Precision* is an estimate of the probability that a person who has a positive prediction does indeed belong to the positive class, and is given by

$$\text{Precision} = \frac{n_{\text{TP}}}{n_{\text{TP}} + n_{\text{FP}}}.$$

6. **Recall:** *Recall* is an estimate of the probability that a person who actually belongs to the positive class is correctly detected by the test, and is given by

$$\text{Recall} = \frac{n_{\text{TP}}}{n_{\text{TP}} + n_{\text{FN}}}.$$

7. **False Positive Rate:** The *false positive rate* is an estimate of the probability that a person who actually belongs to the negative class is predicted to belong to the positive class, and is given by

$$\text{False Positive Rate} = \frac{n_{\text{FP}}}{n_{\text{TN}} + n_{\text{FP}}}.$$

8. **False Discovery Rate:** The *false discovery rate* is an estimate of the probability that a person who is predicted to belong to the positive class does actually belong to the negative class, and is given by

$$\text{False Discovery Rate} = \frac{n_{\text{FP}}}{n_{\text{TP}} + n_{\text{FP}}}.$$

9. **F-Score:** A measure combining precision and recall together is the *F-score*, which is the geometric mean of precision and recall and is defined as

$$F = \frac{1}{\frac{1}{2}(\text{Precision}^{-1} + \text{Recall}^{-1})}.$$

Remark. More generally, we can define the F_β-score as

$$F_\beta = (1 + \beta^2)\frac{1}{\beta^2 \times \text{Precision}^{-1} + \text{Recall}^{-1}}.$$

10. **Receiver Operating Characteristic Curve:**

(a) *Motivation:* A probabilistic classifier can be converted to a class decision by setting a threshold. As the value of the threshold varies, we can reduce type 1 errors at the expense of increasing type 2 errors, or vice versa.

We can plot the *receiver operating characteristic* (ROC) curve to better understand this trade-off.

(b) *ROC Curve:* As the decision boundary varies from $-\infty$ to $+\infty$, the ROC curve can be generated by plotting

- the cumulative fraction of correct detection of the positive class on the vertical axis against
- the cumulative fraction of incorrect detection of the positive class on the horizontal axis.

Remark. A specific confusion matrix represents one point on the ROC curve.

(c) *Special Points on the ROC Curve Plot:*

i. *Top Left Point:* If the top left point $(0, 1)$ appears on the ROC curve, this is the best possible classifier with no misclassification at all.

ii. *Top Bottom Point:* The bottom left corner $(0, 0)$ represents a simple classifier that assigns every point to the negative class and therefore has no true positives but also no false positives.

iii. *Top Right Point:* The top right corner $(1, 1)$ represents a classifier that assigns everything to the cancer class and therefore has no false negatives but also no true negatives.

(d) *Diagonal Line on the ROC Curve Plot:* We can consider a random classifier that simply assigns each data point to the positive class with probability ρ and to the negative class with probability $1 - \rho$. As we vary the value of ρ, it will trace out an ROC curve given by a diagonal straight line.

Remark. Any classifier below the diagonal line performs worse than random guessing.

11. Area Under the Curve: To generate a single number that characterizes the whole ROC curve, we use the *area under the curve* (AUC), i.e., the area under the ROC curve.

Remark. A value of 0.5 for the AUC represents random guessing, whereas a value of 1.0 represents a perfect classifier.

References

Bishop, Christopher Michael and Hugh Bishop (2023). *Deep Learning — Foundations and Concepts*. Ed. by Springer Cham. 1st ed. ISBN: 978-3-031-45468-4.

Burges, Christopher J. C. (2010). “Dimension Reduction: A Guided Tour”. In: *Foundations and Trends in Machine Learning* 2.4, pp. 275–365. ISSN: 1935-8237. DOI: 10.1561/2200000002. URL: http://dx.doi.org/10.1561/2200000002.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. Vol. 1. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Izenman, Alan J (Mar. 2009). *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. en. Springer Science & Business Media.