Classification

- We have n observations; each of them consists of p measurements and they are from two classes[a].
- The goal is to find a classification rule which takes the p measurements as the input and outputs the class label.
- We would like our prediction rule makes small errors not only on the n samples, but also on future observations.
- Project 3: Lending Club Data
- Project 4: Sentiment Analysis

[a] Let's focus on binary classification at this moment.

How to Learn a Classifier

- Collect a training sample $(x_i, y_i)_{i=1}^n$ where $x_i \in \mathbb{R}^p$, $y_i \in \{0, 1\}$
- Pick a collection of functions

$$f : \mathbb{R}^p \longrightarrow \{0, 1\}.$$

- Pick a loss function: e.g., measure the performance of a classifier at (x, y) by a loss function $L(f(x), y)$, e.g., the 0–1 loss

$$L(f(x), y) = 0, \text{ if } y = f(x); \quad 1, \text{ if } y \neq f(x).$$

- Find an optimal classifier by minimizing

$$\min_f \frac{1}{n} \sum_{i=1}^n L(f(x_i), y_i).$$

The Optimal Classifier

Consider an ideal situation: we have infinite samples (or equivalently, we know how data (x, y) are generated), then

$$\frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i) \to \mathbb{E}_{X,Y} L(f(X), Y),$$

where the expectation is taken wrt the true data generating process $P(x, y)$.

Define $\text{Risk}[f] = \mathbb{E}_{X,Y} L(f(X), Y)$. The optimal classifier is given by

$$f^* = \arg\min_f \text{Risk}[f]$$

Suppose we adopt the 0–1 loss, and are allowed to use any function f. What's the optimal f^*?

$$
\begin{aligned}
\text{Risk}[f] &= \mathbb{E}_{X,Y} L(f(X), Y) = \int_{\mathcal{X}} \int_{\mathcal{Y}} L(y, f(x)) p(x, y) dy dx \\
&= \int_{\mathcal{X}} \int_{\mathcal{Y}} L(y, f(x)) p(y|x) p(x) dy dx \\
&= \int_{\mathcal{X}} \Big[\int_{\mathcal{Y}} L(y, f(x)) p(y|x) dy \Big] p(x) dx
\end{aligned}
$$

where

- $p(x)$ is the marginal distribution function of X;
- $p(y|x)$ is the conditional distribution function of Y given $X = x$;
- $p(x)p(y|x)$ is the joint distribution function of (X, Y).

$$\text{Risk}[f] = \int_{\mathcal{X}} \Big[\int_{\mathcal{Y}} L(y, f(x)) p(y|x) dy \Big] p(x) dx$$

The problem of finding f that minimizes $\text{Risk}[f] \Longrightarrow$ (reduced to) a series of sub-problems: for each given x, find the optimal value of $f(x)$ to minimize the inside integral.

$$f^*(x) = \arg\min_a \int_{\mathcal{Y}} L(y, a) p(y|x) dy \quad \text{☺}$$

We can do this for every x, and then the resulting f^* (of course, it may not be continuous) minimizes $\text{Risk}[f]$.

☺ turns out to be not difficult to solve. Note that Y is just a discrete random variable taking two possible values, whose pmf is given by

$$P(Y = 1|x) = \eta(x), \quad P(Y = 0|x) = 1 - \eta(x).$$

Given x, the integral over y (as a function of a) is given by

$$\begin{aligned}
\int_{\mathcal{Y}} L(y,a)p(y|x)dy &= L(1,a)\cdot P(Y=1|x) + L(0,a)\cdot P(Y=0|x) \\
&= L(1,a)\cdot \eta(x) + L(0,a)\cdot (1-\eta(x)) \\
&= \begin{cases} 1-\eta(x), & \text{if } a=1 \\ \eta(x), & \text{if } a=0 \end{cases}
\end{aligned}$$

So

$$f^*(x) = \arg\min_f R[f] = \begin{cases} 1, & \text{if } \eta(x) \geq 0.5 \\ 0, & \text{if } \eta(x) < 0.5 \end{cases}$$

The optimal classifier f^* is called the Bayes rule and the corresponding risk $R[f^*]$ is referred to as the Bayes risk or Bayes error (which is what you computed in `Coding Assignment 1`).

- The calculation can be easily extended to multi-class classification. For multi-class problems where $y \in \{1, \ldots, K\}$, the optimal rule is

$$f^*(x) = \arg\max_k P(Y = k | X = x).$$

- The classifiers that output $0/1$ divide the X-space into different regions and each region is assigned to either 1 or 0. Sometimes, we do not describe f, but the decision boundary.

- Linear classifiers refer to classification methods whose decision boundaries are linear function of x.

Discriminant Analysis

- Estimate the joint distribution $P(x, y) = P(x|y)p(y)$ by estimating
  - $p(x|y)$ (dist of X in each class) and
  - $p(y)$ (marginal frequency of each class),

  and then use Bayes theorem to flip things around and obtain $P(y|x)$.
- Quadratic Discriminant Analysis (QDA)
- Linear Discriminant Analysis (LDA) and its connection with Fisher Discriminant Analysis (FDA).
- Naive Bayes

Bayes Theorem for Classification

We have learned that the optimal classifier (i.e., the Bayes rule) does classification based on the conditional probability, which by the Bayes Theorem takes the following form

$$\begin{aligned} P(Y = k|X = x) &= \frac{P(X = x, Y = k)}{P(X = x)} \\ &= \frac{P(X = x|Y = k) \cdot P(Y = k)}{P(X = x)} \\ &= \frac{\pi_k f_k(x)}{P(X = x)} \propto \pi_k f_k(x) \end{aligned}$$

$f_k(x)$ denotes $P(X = x|Y = k)$ and $\pi_k = P(Y = k)$

- $f_k(x)$: conditional density function of $X|Y = k$
- $\pi_k = P(Y = k)$: the marginal probability or prior probability for class k.

We can write the decision function as

$$f(x) = \arg\max_k \pi_k f_k(x) = \arg\max_k \Big[\log \pi_k + \log f_k(x)\Big].$$

To construct a classifier, we just need to estimate π_k's and f_k's.

QDA

Given $Y = k$, let's model the p-dim feature $\mathbf{x}$ by a multivariate normal distribution with mean $\boldsymbol{\mu}_k$ and covariance matrix Σ_k.

$$\boldsymbol{\mu}_k = \begin{pmatrix} \mu_{k,1} \\ \mu_{k,2} \\ \vdots \\ \mu_{k,p} \end{pmatrix}_{p\times 1}, \quad \Sigma_k^{-1} = \begin{pmatrix} \theta_{k,11} & \cdots & \theta_{k,1p} \\ & & \\ \theta_{k,p1} & \cdots & \theta_{k,pp} \end{pmatrix}_{p\times p}$$

Its pdf is given by

$$f_k(\mathbf{x}) = \frac{1}{(\sqrt{2\pi})^p} \frac{1}{|\Sigma_k|^{1/2}} \exp\Big[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^t \Sigma_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)\Big]$$

$$(\mathbf{x}-\boldsymbol{\mu}_k)^t \Sigma_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k) = \sum_{j=1}^{p}\sum_{l=1}^{p} \theta_{k,jl}(x_j-\mu_{k,j})(x_l-\mu_{k,l})$$

$$
\begin{aligned}
\mathbb{P}(Y = k \mid \mathbf{x}) &\propto \pi_k f_k(\mathbf{x}) \propto e^{-d_k(\mathbf{x})/2} \\
d_k(\mathbf{x}) &= 2\Big[-\log f_k(\mathbf{x}) - \log \pi_k \Big] \\
&= (\mathbf{x} - \boldsymbol{\mu}_k)^t \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) + \log |\Sigma_k| - 2 \log \pi_k,
\end{aligned}
$$

where the first term is the so-called Mahabanobis distance between $\mathbf{x}$ and $\boldsymbol{\mu}_k$ and $\pi_k = P(Y = k)$ is the class frequency for the k-th class. We can predict $\mathbf{x}$ to class k if $d_k(\mathbf{x})$ achieves the minimum among $(d_1(\mathbf{x}), \ldots, d_K(\mathbf{x}))$.

- The classification rule above is called quadratic discriminant analysis (QDA), since it leads to quadratic decision boundaries.

- In practice we need to estimate $\pi_k, \boldsymbol{\mu}_k, \Sigma_k$.

  Sample frequency and mean for each class:

$$\hat{\pi}_k = \frac{n_k}{n}, \quad \hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i:y_i=k} \mathbf{x}_i, \quad k = 1 : K.$$

  Sample covariance matrix for each class:

$$\hat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{i:y_i=k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)_{p\times 1} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^t_{1\times p}.$$

  What if $\hat{\Sigma}_k^{-1}$ does not exist? Replace it by $(\hat{\Sigma}_k + \eta \mathbf{I}_p)^{-1}$ where η is a small number, e.g., $\eta = 0.01$.

Summary of QDA

- Training
  - `Input`: $(\mathbf{x}_i, y_i)_{i=1}^n$
  - `Output`: $(\pi_k, \boldsymbol{\mu}_k, \Sigma_k)_{k=1}^K$ (see formulae on previous slide.)
- Make Prediction
  - `Input`: test point $\mathbf{x}^*$ and output from Training
  - For `k = 1:K`, compute

$$d_k(\mathbf{x}^*) = (\mathbf{x}^* - \boldsymbol{\mu}_k)^t \Sigma_k^{-1} (\mathbf{x}^* - \boldsymbol{\mu}_k) + \log|\Sigma_k| - 2\log \pi_k.$$

  - `Output`: $\arg\min_k d_k(\mathbf{x}^*)$.

LDA

- If further assume $\Sigma_k = \Sigma$, all we need to compute is

$$d_k(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu}_k)^t \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) + \log |\Sigma| - 2 \log \pi_k \qquad (1)$$

  which is a linear function of $\mathbf{x}$:

$$\begin{aligned} & (\mathbf{x} - \boldsymbol{\mu}_k)^t \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \\ = \quad & \mathbf{x}^t \Sigma^{-1} \mathbf{x} - 2\mathbf{x}^t \Sigma^{-1} \boldsymbol{\mu}_k + \boldsymbol{\mu}_k^T \Sigma^{-1} \boldsymbol{\mu}_k \end{aligned}$$

- Estimate Σ by the pooled sample covariance matrix:

$$\hat{\Sigma} = \frac{1}{n - K} \sum_{k=1}^{K} \sum_{i: y_i = k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^t.$$

What if $\hat{\Sigma}^{-1}$ does not exist? Replace it by $(\hat{\Sigma} + \eta \mathbf{I}_p)^{-1}$ where η is a small number, or compute $\hat{\Sigma}^{-1}$ as follows (assume $p = 3$)

$$\hat{\Sigma} = U_{p\times 3} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} U^t, \quad \hat{\Sigma}^{-1} = U \begin{pmatrix} 1/d_1 & 0 & 0 \\ 0 & 1/d_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} U^t.$$

Reduced Rank LDA

- Suppose Σ is an identity matrix $\mathbf{I}_p$. Then, we can write the discriminant function for LDA as

$$d_k(\mathbf{x}) = \|\mathbf{x} - \boldsymbol{\mu}_k\|^2 - 2\log \pi_k. \tag{2}$$

  The feature vector $\mathbf{x}$ only appears in the 1st term which is the squared distance from $\mathbf{x}$ to $\boldsymbol{\mu}_k$, the center of the kth class.

- Two points determine a line; three points determine a plane; K class centers determine a $(K-1)$-dim subspace.

- Next we'll show that we can replace the squared distance $\|\mathbf{x} - \boldsymbol{\mu}_k\|^2$ in the original p-dim space by a square distance in a $(K-1)$-dim subspace. That is, LDA naturally leads to dimension reduction from p to $K-1$. (Of course, here we assume $(K-1) < p$.

- WLOG, assume $\bar{\boldsymbol{\mu}} = \mathbf{0}$ (we care about the relative distance among points so we can always move the origin to $\bar{\boldsymbol{\mu}}$ without affecting the distance).
- The K vectors, $(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$, form a $(K-1)$-dim subspace in $\mathbb{R}^p$. Denote this subspace by $\mathcal{A}$. For any vector $\mathbf{x}$ in $\mathbb{R}^p$,

$$\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2, \quad \mathbf{x}_1 \in \mathcal{A}, \quad \mathbf{x}_2 \in \mathcal{A}^c.$$

  where $\mathbf{x}_1$ is the projection of $\mathbf{x}$ onto the $(K-1)$-dim subspace $\mathcal{A}$, and $\mathbf{x}_2$ is the projection of $\mathbf{x}$ onto the $(p-K+1)$-dim subspace that is orthogonal to $\mathcal{A}$.
- How to get this decomposition? You can run a regression of $\mathbf{x}$ against $B = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{K-1})_{p \times (K-1)}$, and $\mathbf{x}_1 = \mathbf{H}\mathbf{x} =$ the fitted vector, $\mathbf{x}_2 =$ the residual vector, where $\mathbf{H} = B(B^tB)^{-1}B^t$.

- Due to the orthogonality of $\mathbf{x}_1$ and $\mathbf{x}_2$, we have

$$\begin{aligned} \|\mathbf{x}\|^2 &= \|\mathbf{x}_1 + \mathbf{x}_2\|^2 = (\mathbf{x}_1 + \mathbf{x}_2)^t(\mathbf{x}_1 + \mathbf{x}_2) \\ &= \mathbf{x}_1^t\mathbf{x}_1 + \mathbf{x}_1^t\mathbf{x}_2 + \mathbf{x}_2^t\mathbf{x}_1 + \mathbf{x}_2^2\mathbf{x}_2 = \|\mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2. \end{aligned}$$

  The norm-square (length-square) of a vector = norm-square of its projection onto $\mathcal{A}$ + norm-square of its projection onto $\mathcal{A}^c$.

- Now let's look at the square distance from a point $\mathbf{x}$ to $\boldsymbol{\mu}_k$, the center of the kth class.

$$\begin{aligned} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2 &= \|(\mathbf{x}_1 - \boldsymbol{\mu}_k) + (\mathbf{x}_2 - \mathbf{0})\|^2 \\ &= \|\mathbf{x}_1 - \boldsymbol{\mu}_k\|^2 + C, \quad k = 1, \ldots, K, \end{aligned}$$

  where the constant $C = \|\mathbf{x}_2\|^2$ is the same for all k. Note that the projection of $\boldsymbol{\mu}_k$ onto $\mathcal{A}^c$ is zero.

Note that for classification, all we care is the magnitude of $d_k(\mathbf{x})$:

$$\begin{aligned} d_k(\mathbf{x}) &= \|\mathbf{x} - \boldsymbol{\mu}_k\|^2 - 2\log \pi_k \\ &= \|\mathbf{x}_1 - \boldsymbol{\mu}_k\|^2 - 2\log \pi_k + \text{Const.} \end{aligned}$$

So we can operate LDA on this reduced $(K-1)$-dim space $\mathcal{A}$: replace data $\mathbf{x}$ by $\mathbf{x}_1$ (its projection onto $\mathcal{A}$) and compute

$$d_k(\mathbf{x}) = \|\mathbf{x}_1 - \boldsymbol{\mu}_k\|^2 - 2\log \pi_k,$$

which is the same as running LDA in the original p-dimensional space. (We assume $\Sigma = \mathbf{I}$ in our derivation.)

Suppose Σ is not identity. Assume its SVD as $\Sigma = UD^2U^t$ where D is diagonal with non-negative entries and U is an orthonormal (i.e., rotation) matrix with $UU^t = \mathbf{I}_p$, then we first transform $\mathbf{x}$ to $\tilde{\mathbf{x}}$,

$$\mathbf{x} \in \mathbb{R}^p \Longrightarrow \tilde{\mathbf{x}} = \Sigma^{-1/2}\mathbf{x} \in \mathbb{R}^p, \quad \Sigma^{-1/2} := UD^{-1}U^t{}^{\text{a}}.$$

Then the covariance matrix of $\tilde{\mathbf{x}}$ is identity:

$$\begin{aligned} \text{Cov}(\tilde{\mathbf{x}}) &= \text{Cov}(\Sigma^{-1/2}\mathbf{x}) = \Sigma^{-1/2}\text{Cov}(\mathbf{x})(\Sigma^{-1/2})^t \\ &= UD^{-1}U^t\Sigma UD^{-1}U^t = UD^{-1}U^tUD^2U^tUD^{-1}U^t = \mathbf{I}_p. \end{aligned}$$

---

[a] How we scale a 1-dim variable to have unit variance? We multiple that variable by $1/\sqrt{\sigma^2}$. You can view $\Sigma^{-1/2}$ as the inverse of the square-root of Σ in the multi-dimensional setting. Note $\Sigma^{-1/2}$ is symmetric and $(\Sigma^{-1/2})(\Sigma^{-1/2}) = \Sigma$

Summary of Reduced Rank LDA

- Training
  - `Input`: $(\mathbf{x}_i, y_i)_{i=1}^n$
  - `Output`: $(\pi_k, \boldsymbol{\mu}_k)_{k=1}^K$ and Σ.

- Make Prediction
  - `Input`: test point $\mathbf{x}^*$ and output from Training
  - Compute SVD of $\Sigma = UD^2U^t$, and $A = UD^{-1}U^t$
  - Compute the projection matrix $\mathbf{H}$ based on $(A\boldsymbol{\mu}_1, \ldots, A\boldsymbol{\mu}_K)$

$$B_{p\times(K-1)} = \Big[A(\boldsymbol{\mu}_1 - \bar{\boldsymbol{\mu}}), \ldots, A(\boldsymbol{\mu}_{K-1} - \bar{\boldsymbol{\mu}})\Big], \quad \mathbf{H} = B(B^tB)^{-1}B.$$

  - Dimension reduction: $\tilde{\mathbf{x}}^* = \mathbf{H}A\mathbf{x}^*$
  - For `k = 1:K`, compute

$$d_k(\mathbf{x}^*) = \|\tilde{\mathbf{x}}^* - \boldsymbol{\mu}_k\|^2 - 2\log\pi_k.$$

  - `Output`: $\arg\min_k d_k(\mathbf{x}^*)$.

- When $p \geq K$, this means that for LDA, one can project the data onto a lower-dimensional subspace (dim $= K - 1$), e.g., just one dimension for binary classification.

- As we will see that the same subspace also arises in a dimension reduction method called Fisher discriminant analysis (FDA), though FDA is motivated from a slightly different aspect.

- Caution: Each of the $(K - 1)$ directions are linear combinations of the original p dimensions, and the weights are all learned from the data, so overfitting can occur: good separation on "just" the $(K - 1)$ (or even less) directions on the training data, but the same good result cannot be reproduced on the test data. See the analysis of the `digits` data on the R page.

Fisher's Discriminant Analysis

- Find a direction $\mathbf{a} \in \mathbb{R}^p$ such that the projection of data onto this direction is well separated.
- Denote the projection of an observation $\mathbf{x}_i \in \mathbb{R}^p$ by $u_i = \mathbf{a}^t \mathbf{x}_i \in \mathbb{R}$.
- What's being well separated? The group means of u_i's are far apart from each other, and within each group, the variation/spread is small, i.e., minimize the following ratio

$$\frac{\text{Between group variation}}{\text{Within group variation}} = \frac{\mathbf{a}^t B \mathbf{a}}{\mathbf{a}^t W \mathbf{a}}.$$

The within-class and between-class sample covariance matrices

$$W_{p\times p} = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(\mathbf{x}_i-\hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i-\hat{\boldsymbol{\mu}}_k)^t$$

$$B_{p\times p} = \frac{1}{K-1}\sum_{k=1}^{K} n_k(\hat{\boldsymbol{\mu}}_k-\bar{\hat{\boldsymbol{\mu}}})(\hat{\boldsymbol{\mu}}_k-\bar{\hat{\boldsymbol{\mu}}})^t$$

The following equalities are trivial.

$$\begin{aligned}
\mathbf{x}_1 &= \mathbf{x}_1 - \hat{\boldsymbol{\mu}}_{y_1} + \hat{\boldsymbol{\mu}}_{y_1}\\
\mathbf{x}_2 &= \mathbf{x}_2 - \hat{\boldsymbol{\mu}}_{y_2} + \hat{\boldsymbol{\mu}}_{y_2}\\
\ldots &= \ldots\\
\mathbf{x}_n &= \mathbf{x}_n - \hat{\boldsymbol{\mu}}_{y_n} + \hat{\boldsymbol{\mu}}_{y_n}
\end{aligned}$$

You can view W as the sample covariance matrix over $\mathbf{x}_i - \hat{\boldsymbol{\mu}}_{y_i}$ (same as the pooled sample covariance matrix $\hat{\Sigma}$ in LDA), and B as the sample covariance matrix over the K class centers $\hat{\boldsymbol{\mu}}_{y_i}$.

Generalized Eigenvalue Problem

$$\max_{\mathbf{a}} \frac{\mathbf{a}^t B\mathbf{a}}{\mathbf{a}^t W\mathbf{a}} \Longrightarrow \max_{\mathbf{a}} \mathbf{a}^t B\mathbf{a} \text{ subj to } \mathbf{a}^t W\mathbf{a} = 1.$$

Assume $W = UD^2U^t$. Define $\mathbf{b} = W^{1/2}\mathbf{a}$, where $W^{1/2} := UDU^t$ is symmetric and write its inverse as $W^{-1/2} = UD^{-1}U^t$.

$$\begin{aligned} \mathbf{a}^t B\mathbf{a} &= \mathbf{a}^t W^{1/2} W^{-1/2} B W^{-1/2} W^{1/2} \mathbf{a} \\ &= (W^{1/2}\mathbf{a})^t W^{-1/2} B W^{-1/2} (W^{1/2}\mathbf{a}) \\ &= \mathbf{b}^t W^{-1/2} B W^{-1/2} \mathbf{b}, \quad \text{subj to } \|\mathbf{b}\|^2 = 1. \end{aligned}$$

The optimization above is a classical eigenvalue problem!

We can solve the directions sequentially as follows

- $\mathbf{b}_1 =$ the 1st eigen-vector of matrix $W^{-1/2}BW^{-1/2}$, then solve $\mathbf{a}_1 = W^{-1/2}\mathbf{b}_1$.
- $\mathbf{b}_2 =$ the 2nd eigen-vector of matrix $W^{-1/2}BW^{-1/2}$, then solve $\mathbf{a}_2 = W^{-1/2}\mathbf{b}_2$.
- Note that although $\mathbf{b}_j$'s are orthonormal, but $\mathbf{a}_j$'s are not:

$$\mathbf{b}_j^t\mathbf{b}_j = 1, \quad \mathbf{a}_j^t W\mathbf{a}_j = 1$$

$$\mathbf{b}_j^t\mathbf{b}_l = 0, \quad \mathbf{a}_j^t W\mathbf{a}_l = 0.$$

- We can extract at most $(K-1)$ directions, since the rank of B is $(K-1)$. The $(K-1)$ directions span exactly the same space as the one from the reduced rank LDA.

LDA vs FDA

- LDA: a classifier.
- FDA: a dimension reduction method, i.e., the output of FDA is a set of directions, but not a classification rule.
- The normal assumption is never mentioned in FDA, but why the space from FDA is similar to the reduced space from LDA?

  FDA implicitly assumes the data from each group follows or approximately follows a normal distribution with the same covariance matrix.

LDA & QDA in High Dimension

- Singularity of the Covariance Matrix When dimension p is large, the inverse of $\hat{\Sigma}$ for $\hat{\Sigma}_k$ may not exist. For example, if $p > n$, $\hat{\Sigma}$, the $p \times p$ covariance matrix for LDA, is of rank less than n. So we cannot compute $\hat{\Sigma}^{-1}$.

  But singularity is not a serious issue, and we have discussed how to fix singularity for LDA and QDA. A more serious issue is overfitting.

- When dimension p is large, even LDA could end up overfitting the data: one can show that when p gets large, LDA could behave like random guessing (i.e., classification error $= 0.5$).

- Regularization: restrict matrices/vectors to be sparse (e.g., sparse LDA or regularization DA), or restrict features to be independent (NaiveBayes).

Naive Bayes

- Recall: for multi-class problems the optimal decision rule is

$$\arg\max_k \mathbb{P}(Y = k|X = \mathbf{x}) = \arg\max_k \pi_k f_k(\mathbf{x}).$$

- Require $f_k(\mathbf{x})$ to be

$$f_k(\mathbf{x}) = f_{k1}(x_1) \times f_{k2}(x_2) \cdots \times f_{kp}(x_p),$$

  i.e., each dim of $\mathbf{x}$ is independent (You can view independence as regularization for high-dimensional problems).

- Then each density f_{kj} $(j = 1 : p, k = 1 : K)$ is estimated separately within each class. E.g., discrete features via histograms; numerical features via kernel density estimates (nonparametric NB) or normal densities (parametric NB).

Summary of Parametric Naive Bayes

- Training
  - `Input`: $(\mathbf{x}_i, y_i)_{i=1}^n$
  - `Output`: $\left(\pi_k, (\mu_{kj}, \sigma_{kj}^2)_{j=1:p}\right)_{k=1}^K$
- Make Prediction
  - `Input`: test point $\mathbf{x}^*$ and output from Training
  - For `k = 1:K`, compute

$$\begin{aligned} d_k(\mathbf{x}^*) &= -2\log\left[\pi_k \frac{1}{\sqrt{\sigma_{k1}^2}} e^{-\frac{(x_1^*-\mu_{k1})^2}{2\sigma_{k1}^2}} \cdots \frac{1}{\sqrt{\sigma_{kp}^2}} e^{-\frac{(x_p^*-\mu_{kp})^2}{2\sigma_{kp}^2}}\right] \\ &= -2\log\pi_k + \sum_{j=1}^p \left[\log\sigma_{kj}^2 + \frac{(x_j^*-\mu_{kj})^2}{\sigma_{kj}^2}\right] \end{aligned}$$

  - `Output`: $\arg\min_k d_k(\mathbf{x}^*)$.

Summary

- For classification, the ultimate goal is to estimate $P(Y = k|X = \mathbf{x})$.
- In Discriminant Analysis (DA), we estimate the joint

$$P(X = \mathbf{x}, Y = k) = P(X = \mathbf{x}|Y = k) \times P(Y = k),$$

  and then obtain $P(Y = k|X = \mathbf{x})$.
- DA is conceptually simple and works for some low-dimensional problems, but not an effective way of building classifiers.
- For example, for binary LDA with discriminant function (1):

$$d_k(\mathbf{x}) = -2\mathbf{x}^t\Sigma^{-1}\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^T\Sigma^{-1}\boldsymbol{\mu}_k - 2\log\pi_k$$

  What matters is the decision boundary which is a linear function has

$(p+1)$ parameters:

$$d_1(\mathbf{x}) - d_2(\mathbf{x}) = -2\mathbf{x}^t\Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \beta_0 = \mathbf{x}^t\boldsymbol{\beta} + \beta_0.$$

However, we estimate $(\boldsymbol{\beta}, \beta_0)$ by learning a much larger collection of parameters such as Σ, $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and π_1.

- Next we'll discuss how to directly learn $P(Y = k|X = \mathbf{x})$ (e.g., logistic regression, tree models) or directly learn the decision boundary (e.g., SVM).