Classification

We have \( n \) observations; each of them consists of \( p \) measurements and they are from two classes. The goal is to find a prediction 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.
In STAT/CS jargon: We have a training set of sample size $n \ (x_i, y_i)_{i=1}^n$ where $x_i \in \mathbb{R}^p$ and $y_i \in \{0, 1\}$ (binary classification). The goal is to find a classifier $f : \mathbb{R}^p \rightarrow \{0, 1\}$.

At $(x, y)$, the performance of a classifier can be measured by the following 0–1 loss

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

The optimal classifier is the one minimizing the risk

$$R[f] = \mathbb{E}_{X,Y} L(f(X), Y).$$
For 0–1 loss, we can show that the optimal classifier takes the following form

$$f^*(x) = \text{argmin}_f R[f] = \begin{cases} 
1 & \text{if } \eta(x) > 1/2 \\
0 & \text{otherwise},
\end{cases}$$

where

$$\eta(x) = \mathbb{P}(Y = 1|X = x).$$

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.
• Some classifiers $f(x)$ output numerical values $\in \mathbb{R}$ and we need to threshold $f(x)$ to obtain the classification result, e.g., $f(x) > c$, predict $y = 1$ and otherwise $y = 0$.

• Some classifiers $f(x)$ output probabilities, e.g., $f(x)$ is estimate of $\mathbb{P}(Y = 1 | X = x)$. Then we can threshold $f(x)$ at 0.5.

• 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 $x$. 
Binary vs Multi-Class

• For simplicity, we’ll focus on binary classifiers. Some binary classifiers can also handle multi-classes, such as discriminant analysis (LDA, QDA, NB), logistic regression, and $k$NN. But for some binary classifiers, the extension is not trivial, such as SVM.

• There are some default (although may not be optimal) ways to apply a binary classifier on a classification problem with $K > 2$ categories.
  
  – Train $K$ one-vs-other classifiers
  
  – Train $K(K - 1)/2$ pairwise classifiers

Then you combine the results to get a consensus prediction.
What’s Wrong with Linear Regression?

• For this discussion, let’s assume that we code $Y$ as $-1$ and $1$.

• Can we perform a linear regression of $Y$ on $X$ and then classify $Y = 1$ if $\hat{Y} > 0$? Actually that’s what we have done in HW1.

• Problems with Linear Regression (as a classifier)?
Discriminant Analysis

• “Here the approach is to model the distribution of $X$ in each of the classes separately, and then use Bayes theorem to flip things around and obtain $P(Y \mid X = x)$.”

• Quadratic Discriminant Analysis (LDA)

• Linear Discriminant Analysis (LDA) and its connection with Fisher Discriminant Analysis (FDA).

• Naive Bayes

• Importance of variable selection
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

\[ P(Y = k \mid X = x) = \frac{P(X = x \mid Y = k) \cdot P(Y = k)}{P(X = x)} = \frac{P(X = x \mid Y = k) \cdot P(Y = k)}{\sum_{l=1}^{K} P(X = x \mid Y = l) \cdot P(Y = l)} = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)} \]

- \( f_k(x) \): conditional density function of \( X \mid Y = k \) (we’ll model it as a normal distribution).
- \( \pi_k = P(Y = k) \): the marginal probability or prior probability for class \( k \).
We can write the decision function as

\[ f(x) = \arg \max_k \mathbb{P}(Y = k | X = x) = \arg \max_k \pi_k f_k(x). \]

The corresponding decision boundaries are

\[ \{ x : \pi_j f_j(x) = \pi_l f_l(x) = \arg \max_k \pi_k f_k(x), \ j \neq l. \}. \]
QDA

• Suppose $X|Y = k \sim N(\mu_k, \Sigma_k)$ with density function

$$f_k(x) = (2\pi)^{-p/2}|\Sigma_k|^{-1/2} \exp \left[ -\frac{1}{2} (x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k) \right],$$

where $\mu_k$ and $\Sigma_k$ are the mean vector and covariance matrix for class $k$. Simple calculations reveal that

$$P(Y = k | x) = \frac{e^{d_k(x)/2}}{\sum_l e^{d_l(x)/2}}$$

$$d_k(x) = 2 \left[ - \log f_k(x) - \log \pi_k \right] - \text{constant}$$

$$= (x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k) + \log |\Sigma_k| - 2 \log \pi_k,$$

where the first term is the so-called Mahalanobis distance between $x$ and $\mu_k$ and $\pi_k = P(Y = k)$. We can predict $x$ to class $k$ if $d_k(x)$ achieves the maximum among $(d_1(x), \ldots, d_K(x))$. 
• The classification rule above is called **quadratic discriminant analysis (QDA)**, since it leads to quadratic decision boundaries separating different classes.

• In practice need to estimate $\pi_k, \mu_k, \Sigma_k$.

  Sample frequency and mean for each class:

  \[
  \hat{\pi}_k = \frac{n_k}{n}, \quad \hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i, \quad k = 1 : K.
  \]

  Sample covariance matrix for each class:

  \[
  \hat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^t.
  \]
LDA

- If further assume $\Sigma_k = \Sigma$, we can simplify QDA as linear discriminant analysis (LDA) with

$$d_k(x) = (x - \mu_k)^t \Sigma^{-1} (x - \mu_k) + \log |\Sigma| - 2 \log \pi_k$$

$$= -x^t \Sigma^{-1} \mu_k + \left( \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k - \log \pi_k \right) + \text{constant.}$$

- Estimate $\Sigma$ by the pooled sample covariance matrix:

$$\hat{\Sigma} = \frac{1}{n - K} \sum_{k=1}^{K} \sum_{i: y_i = k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^t.$$
Reduced Rank LDA

• Suppose $\hat{\Sigma}$ is an identity matrix $I_p$. Then, we can write the discriminant function for LDA as

$$d_k(x) = \|x - \mu_k\|^2 + \log |\Sigma| - 2 \log \pi_k$$

where the 1st term is the squared distance from the data point $x$ to the center of the $k$th class.
Consider a \((K-1)\)-dim subspace in \(\mathbb{R}^p\), formed by the \(K\) centers, \((\mu_1, \ldots, \mu_K)\). Denote this subspace by \(A\). For any point \(x\) in \(\mathbb{R}^p\), denote its projection onto this subspace as \(\tilde{x}\). It can be shown that

\[
\|x - \mu_k\|^2 = \|\tilde{x} - \mu_k\|^2 + C, \quad k = 1, \ldots, K,
\]

where the constant \(C\) is the same for all \(k\).

For classification, \(d_k(x)\) or \(d_k(x) - C\) gives us the same result as long as \(C\) does not depend on \(k\). That is, we can operate LDA on this reduced space \(A\), which is the same as running LDA in the original \(p\)-dimensional space.
• If \( \hat{\Sigma} \) is not identity, then we can always “normalize” \( x \) to \( x^* \),

\[
x \in \mathbb{R}^p \implies x^* = \hat{\Sigma}^{-1/2} x \in \mathbb{R}^p.
\]

and then the sample covariance matrix of \( x^* \) will be identity.

Here we assume \( \hat{\Sigma}^{-1} \) exists, otherwise, we cannot run LDA. (Not difficult to resolve the singularity issue.)
• When $p > K$, this means that for LDA, one can project the data onto a lower-dimensional subspace, e.g., just one dimension for binary classification.

• As we will see that the same subspace $A$ 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 it's easy to get a good classification based on “just” the $(K - 1)$ (or even less) directions on the training data, but the prediction error may still be large; a simple overfitting problem.
Fisher’s Discriminant Analysis

- Find a direction $a \in \mathbb{R}^p$ such that the projection of data onto this direction is well separated.

- Denote the projection of an observation $x_i \in \mathbb{R}^p$ by $u_i = a^t x_i \in \mathbb{R}$.

- What’s being well separated? The group means of $u_i$’s are far apart from each other, and the within each group, the variation/spread is small, i.e., minimize the following ratio

\[
\frac{\text{Between group variation}}{\text{Within group variation}}.
\]
• The within-class and between-class sample covariance matrices

\[
W = \frac{1}{n - K} \sum_k \sum_j (x_{kj} - \bar{x}_k)(x_{kj} - \bar{x}_k)^t
\]

\[
B = \frac{1}{K - 1} \sum_k n_k (\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^t.
\]

Here we switch the notation and use double subscripts to index observations with \(x_{ij} \in \mathbb{R}^p\) being the \(j\)th observation from the \(i\)th group.

• Finally we can express the ratio as

\[
\frac{a^t B a}{a^t W a}.
\]
• Maximizing the ratio is equivalent to solving

\[
\max_a a^t B a \quad \text{subject to } a^t W a = 1.
\]

• The solution is given by the 1st eigen-vector of matrix \( W^{-1} B \).\(^a\)

• We can also solve for the 2nd optimal direction, among the ones orthogonal to the previous one.

• We can extract at most \((K - 1)\) direction, since the rank of \( B \) is \((K - 1)\).

\(^a\)The sample Cov matrix of the \( K \) points, which span \( A \), on p10, is \( W^{-1} B \).
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.
Discriminant Analysis in High Dimension

- **Singularity of the Covariance Matrix** When dimension $p$ is large, QDA/LDA may not be applicable, because the inverse of $\hat{\Sigma}$ for $\hat{\Sigma}_k$ doesn't exist. For example, if $p > n$, $\hat{\Sigma}$, the $p \times p$ covariance matrix for LDA, is of rank at most $n - 1$. So we cannot compute $\hat{\Sigma}^{-1}$.

A straightforward solution: using the generalized inverse of a matrix.

\[
\begin{pmatrix}
3 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0
\end{pmatrix}^{-1} = \begin{pmatrix}
\frac{1}{3} & 0 & 0 \\
0 & \frac{1}{2} & 0 \\
0 & 0 & 0
\end{pmatrix}.
\]

Singularity flags a warning: $n$ is small relatively to $p$, so watch out for 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 $= 0.5$).

A related issue with FDA: classes are well-separated on the training set could be meaningless for high-dimensional data. (Look at the FDA result for the digits data.)

• Regularization: sparse LDA, Naive Bayes, or RDA.
**Sparse LDA**

For simplicity, focus on binary LDA with discriminant function

\[
d_k(x) = -x^T \Sigma^{-1} \mu_k + \left( \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k - \log \pi_k \right), \quad k = 1, 2
\]

We only need to know the difference of \(d_1(x)\) and \(d_2(x)\), so eventually the quantity we need to compute is

\[
\left(x - \bar{\mu}\right)^T \Sigma^{-1} (\mu_1 - \mu_2) + (\cdot),
\]

where \(\bar{\mu} = (\mu_1 + \mu_2)/2\) and \((\cdot)\) is some quantity doesn’t depend on \(x\) nor \(\mu_1 - \mu_2\).
There is a collection of algorithms under the umbrella “Sparse LDA”.

- Assume $\Sigma$ is diagonal and the class difference $(\mu_1 - \mu_2)$ is sparse. For the latter, run two-sample $t$-test: if the $j$th feature is not significant, set $\mu_{1j} - \mu_{2j} = 0$.

- The Sample covariance matrix $\Sigma$ is sparse: estimate $\Sigma$ using sparse matrices.

- Both $\mu_1 - \mu_2$ and $\Sigma$ are sparse.
RDA (not covered)

- Regularized QDA with covariance matrix

\[(1 - \alpha)\hat{\Sigma}_k + \alpha\hat{\Sigma}.
\]

- Regularized LDA with covariance matrix

\[(1 - \alpha)\hat{\Sigma} + (1 - \alpha)\hat{\sigma}^2 I,
\]

where \(\hat{\sigma}^2 = tr(\hat{\Sigma})/p\).
RDA uses the following regularized covariance matrix

\[
\hat{\Sigma}_k(\lambda, \gamma) = (1 - \gamma)\hat{\Sigma}_k(\lambda) + \gamma \frac{1}{p} \text{tr}[\hat{\Sigma}_k(\lambda)] I_p,
\]

\[
\hat{\Sigma}_k(\lambda) \equiv (1 - \lambda)\hat{\Sigma}_k + \lambda \hat{\Sigma},
\]

with \( \lambda, \gamma \in [0, 1] \) \(^a\) Large values indicate higher degrees of regularization.

- \((\gamma = 0, \lambda = 0)\): QDA (individual cov for each class).
- \((\gamma = 0, \lambda = 1)\): LDA (shared cov matrix).
- \((\gamma = 1, \lambda = 0)\): Variables are conditionally independent with equal class-specific variance; similar to Naive Bayes.
- \((\gamma = 1, \lambda = 1)\): Nearest centroid (objects are assigned to group with nearest mean with euclidean distance).

\(^a\)In practice, the tuning parameters are selected by CV.
Naive Bayes

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

\[ f^*(x) = \arg \max_k \mathbb{P}(Y = k | X = x) = \arg \max_k \pi_k f_k(x). \]

• Approximate \( f_k(x) \) by

\[ f_k(x) \approx \prod_{j=1}^p f_{kj}(x_j), \]

i.e., each dim of \( x \) is approximately independent (independence assumption may not hurt too much 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.
Logistic Regression

• As we have learned before, the best classifier depends on

\[ \eta(x) = \mathbb{P}(Y = 1 | X = x). \]

One type of approaches for classification is to directly model or estimate \( \eta(x) \), for example, by a linear model. Since \( \eta(x) \) is constrained to between 0 and 1, it is not realistic to assume \( \eta(x) \) takes a linear form. Instead we assume its transformation (or referred to as a link function) is a linear function,

\[ g(\eta(x)) = x^t \beta. \]

Here we merge the intercept into the \( x \)-vector.

• For Logistic regression, we use the so-called logit link function

\[ \log \frac{\eta(x)}{1 - \eta(x)} = x^t \beta, \quad \eta(x) = \frac{\exp(x^t \beta)}{1 + \exp(x^t \beta)}. \]
How does $\eta(x)$ relate to the response variable $Y$? Well,

$$Y | X = x \sim \text{Bern}(\eta(x)),$$

i.e. $p(y|x) = \eta(x_i)^{y_i}[1 - \eta(x_i)]^{1-y_i}$

So given a set of training data $(x_i, y_i)_{i=1}^n$, we can estimate $\beta$ by maximizing the log-likelihood

$$\ell(\beta) = \sum_{i=1}^{n} \log \left[ \eta(x_i) \right]^{y_i} \left[ 1 - \eta(x_i) \right]^{1-y_i}$$

$$= \sum_{i=1}^{n} y_i \log \frac{\eta(x_i)}{1 - \eta(x_i)} + \log [1 - \eta(x_i)]$$

$$= \sum_{i=1}^{n} y_i x_i^t \beta - \log [1 + \exp(x_i^t \beta)].$$
Another link function: **Probit Model**

- **Probit link function**

  \[
  \Phi^{-1}(\eta) = \alpha + x^t \beta, \quad \eta = \Phi(\alpha + x^t \beta),
  \]

  where \( \Phi(\cdot) \) is the CDF of \( N(0, 1) \).

- **Its equivalent form: Gaussian latent model**

  \[
  Z = \alpha + x^t \beta + \epsilon, \quad \epsilon \sim N(0, 1),
  \]

  and \( Y = 1 \) if \( Z > 0 \).
Back to Logistic...

- The MLE $\hat{\beta}$ can be obtained by the following Reweighted LS Algorithm:

  – Start with some initial values $\beta^0$

  – Calculate the corresponding $p_i^0 = \eta^0(x_i)$ for $i = 1, \ldots, n$; define $W = \text{diag}(p_i^0(1 - p_i^0))_{i=1}^n$.

  – Calculate $z = X\beta^0 + W^{-1}(y - p^0)$.

  – Update $\beta^0 \leftarrow \beta^1$ with $\beta^1 = (X^tWX)^{-1}X^tWz$.

  and iterative the above steps until convergence.
Variable Selection for Logistic

Penalized likelihood

\[- \log \text{Likelihood}(\beta) + \lambda \sum_j Pen(\beta_j),\]

where \(Pen(\beta_j)\) denotes the penalty function on \(\beta_j\).

For example, \(Pen(\beta_j) = \|\beta_j\|_0\) (\(L_0\) norm) corresponds to AIC/BIC, and \(Pen(\beta_j) = \beta_j^2\) corresponds to Ridge, and \(Pen(\beta_j) = |\beta_j|\) corresponds to Lasso.

- AIC/BIC with step-wise/forward/backward selection, or
- Ridge and Lasso using glmnet.