- Introduction to classification
- Discriminant analysis: LDA, QDA, and NB
- Logistic regression

Classification

(In plain English) We have n observations; each of them consists of p measurements and they are from two classes[a]. 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.

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

(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

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

At (x, y), the performance of a classifier can be measured some loss function, e.g., 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[a]

$$R[f] = \mathbb{E}_{X,Y} L(f(X), Y).$$

[a] risk = integrated loss.

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[a] e.g., $f(x)$ is estimate of $\mathbb{P}(Y = 1 \mid X = x)$. 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.

[a]Hard decision classifiers: the ones that return just the label, 0 or 1; soft decision classifiers: the ones that return probabilities. For classifiers that return just a numerical value $\in \mathbb{R}$, we can transform them to hard or soft decision classifiers.

Loss Functions for Classification

- 0/1 Loss

$$L(f(x), y) = \mathbf{1}_{\{y=f(x)\}},$$

  where $f(x) = 0$ or 1.

- Log-loss, if $f(x)$ returns a probability $p(x) \in [0, 1]$

$$L(f(x), y) = \log \left[p(x)^{\{y=1\}} (1 - p(x))^{\{y=0\}} \right].$$

  Note that Log-loss does not work well for hard decision classifiers.

- AUC: area under the Receiver Operating Characteristic (ROC) curve. Will discuss it later.

Binary vs Multi-Class

- For simplicity, we'll focus on binary classifiers. Some binary classifiers can also handle multi-classes, but for some binary classifiers, the extension is not trivial.
- 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 a binary classification problem where $Y = 0$ or 1, can we perform a linear regression of Y on X and then classify $Y = 1$ if $\hat{Y} > 0.5$? Actually that's what we have done before.
- Problems with linear regression (as a classifier)? The square loss is not appropriate for classification, see the example on the R page.

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 $\mathbb{P}(Y \mid X = x)$."
- Quadratic Discriminant Analysis (QDA)
- 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

$$\begin{aligned}
\mathbb{P}(Y = k \mid X = x) &= \frac{\mathbb{P}(X = x \mid Y = k) \cdot \mathbb{P}(Y = k)}{\mathbb{P}(X = x)} \\
&= \frac{\mathbb{P}(X = x \mid Y = k) \cdot \mathbb{P}(Y = k)}{\sum_{l=1}^{K} \mathbb{P}(X = x \mid Y = l) \cdot \mathbb{P}(Y = l)} \\
&= \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}
\end{aligned}$$

- $f_k(x)$: conditional density function of $X|Y = k$ (we'll model it as a normal distribution).
- $\pi_k = \mathbb{P}(Y = k)$: 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

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

QDA

- Suppose $X|Y = k \sim \mathsf{N}(\mu_k, \Sigma_k)$ with density function

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

where μ_k and Σ_k are the mean vector and covariance matrix for class k. Simple calculations reveal that

$$\begin{aligned}
\mathbb{P}(Y = k \mid x) &= \frac{e^{d_k(x)/2}}{\sum_l e^{d_l(x)/2}} \\
d_k(x) &= 2\Big[-\log f_k(x) - \log \pi_k \Big] - \text{constant} \\
&= (x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k) + \log|\Sigma_k| - 2\log \pi_k,
\end{aligned}$$

where the first term is the so-called Mahabanobis distance between x and μ_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 we need to estimate π_k, μ_k, Σ_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:

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

LDA

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

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

- Estimate Σ by the pooled sample covariance matrix:

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

Reduced Rank LDA

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

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

- Note the feature vector x only appears in the 1st term which is the squared distance from x to the center of the kth class. Next we'll show that we can replace the squared distance in the original p-dim space by a square distance in a $(K-1)$-dim subspace, a subspace spanned by the K class centers (after removing the overall mean), $(\mu_1 - \bar{\mu}, \ldots, \mu_K - \bar{\mu})$.

- The K vectors, $(\mu_1 - \bar{\mu}, \ldots, \mu_K - \bar{\mu})$, form a $(K-1)$-dim subspace in $\mathbb{R}^p$. Denote this subspace by A and the corresponding projection matrix by P_A, and WLOG, assume $\bar{\mu} = \mathbf{0}$ (we care about the relative distance among points so we can always relabel $\bar{\mu}$ as the new origin without affecting the distance). For any point x in $\mathbb{R}^p$, denote its projection onto this subspace as $\tilde{x} = P_A x$. It can be shown that

$$\begin{aligned} \|x - \mu_k\|^2 &= \|P_A(x - \mu_k) + (\mathbf{I} - P_A)(x - \mu_k)\|^2 \\ &= \|\tilde{x} - \mu_k\|^2 + C, \quad k = 1, \ldots, K, \end{aligned}$$

  where the constant $C = \|(\mathbf{I} - P_A)x\|^2$ 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 first "normalize" x to x^*,

$$x \in \mathbb{R}^p \Longrightarrow x^* = \hat{\Sigma}^{-1/2} x \in \mathbb{R}^p.$$

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 $p \geq K$, 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 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 same good result cannot be reproduced on the test data, i.e., the prediction error could be large; a simple overfitting problem. See the analysis of the `digits` data on the R page.

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

$$\begin{aligned} W &= \frac{1}{n-K}\sum_k \sum_j (x_{kj} - \bar{x}_{k\cdot})(x_{kj} - \bar{x}_{k\cdot})^t \\ B &= \frac{1}{K-1}\sum_k n_k(\bar{x}_{k\cdot} - \bar{x}_{\cdot\cdot})(\bar{x}_{k\cdot} - \bar{x}_{\cdot\cdot})^t. \end{aligned}$$

  Here we switch the notation and use double subscripts to index observations with $x_{ij} \in \mathbb{R}^p$ being the jth observation from the ith group.

- Finally we can express the ratio as

$$\frac{a^t B a}{a^t W a}.$$

- Maximizing the ratio is equivalent to solving

$$\max_a a^t B a \text{ subject to } a^t W a = 1.$$

- 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 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

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

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 $\mu_1 - \mu_2$.

Below is a brief summary of the types of algorithms under the umbrella "Sparse LDA". For details, carry out a literature review on this topic.

- Assume Σ is diagonal and the class difference $(\mu_1 - \mu_2)$ is sparse. For the latter, e.g., we can run a two-sample t-test for each feature: if the jth feature is not significant, set $\mu_{1j} - \mu_{2j} = 0$.
- The sample covariance matrix Σ is sparse: estimate Σ using sparse matrices.
- Both $\mu_1 - \mu_2$ and Σ 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 $\hat{\sigma}^2 = \mathsf{tr}(\hat{\Sigma})/p$.

RDA uses the following regularized covariance matrix

$$\begin{aligned}\hat{\Sigma}_k(\lambda,\gamma) &= (1-\gamma)\hat{\Sigma}_k(\lambda) + \gamma\frac{1}{p}\text{tr}\left[\hat{\Sigma}_k(\lambda)\right]\mathbf{I}_p, \\ \hat{\Sigma}_k(\lambda) &\equiv (1-\lambda)\hat{\Sigma}_k + \lambda\hat{\Sigma},\end{aligned}$$

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,

$$Y|X = x \sim \text{Bern}(\eta(x)), \text{ i.e. } p(y|x) = \eta(x_i)^{y_i}[1-\eta(x_i)]^{1-y_i}$$

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

$$\begin{aligned}
\ell(\beta) &= \sum_{i=1}^n \log \Big[\eta(x_i)\Big]^{y_1} \Big[1-\eta(x_i)\Big]^{1-y_i} \\
&= \sum_{i=1}^n y_i \log \frac{\eta(x_i)}{1-\eta(x_i)} + \log[1-\eta(x_i)] \\
&= \sum_{i=1}^n y_i x_i^t \beta - \log[1+\exp(x_i^t\beta)].
\end{aligned}$$

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 $\mathsf{N}(0, 1)$.

- Its equivalent form: Gaussian latent model

$$Z = \alpha + x^t\beta + \epsilon, \quad \epsilon \sim \mathsf{N}(0, 1),$$

  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 β^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
$$\mathbf{z} = X\beta^0 + W^{-1}(y - \mathbf{p}^0).$$
  - Update $\beta^0 \leftarrow \beta^1$ with
$$\beta^1 = (X^t W X)^{-1} X^t W \mathbf{z}.$$
    and iterative the above steps until convergence.

Variable Selection for Logistic

Penalized likelihood

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

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

More on Logistic Regression

- Convergence issue with logistic regression when data are well-separated
- Multinomial logistic regression
- Move beyond linear decision boundary: add quadratic terms to logistic regression
- Retrospect sampling (both LDA and Logistic can handle this)

We usually assume that data (x_i, y_i)'s are collected as iid samples from a population of interest. However, in some applications, we may have data collected retrospectively. For example, in a study on cancer, instead of taking a random sample of 100 people from the whole population (then the number of cancer patients will be too small), we may draw 50 samples from the cancer group and 50 from the control group.

Let Z indicate whether an individual is sampled or not. Using retrospective samples, we are estimating $P(Y = 1|Z = 1, X = x)$, while what we care is $P(Y = 1|X = x)$. Assume the logit of the latter follows a linear model, that is,

$$P(Y = 1|X = x) = \frac{\exp(\alpha + x^t\beta)}{1 + \exp(\alpha + x^t\beta)}, \tag{2}$$

where we isolate the intercept α from other coefficients β. Then the question is whether we can estimate α or β in (2) based on a retrospective sample.

It turns out that for logistic models, the coefficients estimated from a retrospective sample is roughly the same as the one from an ordinary random sample. Again, let Z indicate whether an individual is sampled or not and denote the sample proportions (in each class) by

$$r_0 = \mathbb{P}(Z = 1|Y = 0), \quad r_1 = \mathbb{P}(Z = 1|Y = 1).$$

Then

$$\begin{aligned}
\mathbb{P}(Y = 1|Z = 1, X = x) &= \frac{\mathbb{P}(Z = 1|Y = 1, x)\mathbb{P}(Y = 1|x)}{\mathbb{P}(Z = 1, Y = 1|x) + \mathbb{P}(Z = 1, Y = 0|x)} \\
&= \frac{r_0 \exp(\alpha + x^t\beta)}{r_1 + r_0 \exp(\alpha + x^t\beta)} \\
&= \frac{\exp(\alpha^* + x^t\beta)}{1 + \exp(\alpha^* + x^t\beta)}.
\end{aligned}$$

Although the data have been sampled retrospectively, the logistic model continues to apply with the same coefficients β but a different intercept.