EECS E6690: Statistical Learning for Biological and Information Systems
Lecture 5: Classification

Prof. Predrag R. Jelenković
Time: Tuesday 4:10-6:40pm
303 Seeley W. Mudd Building

Dept. of Electrical Engineering
Columbia University , NY 10027, USA
Office: 812 Schapiro Research Bldg.
Phone: (212) 854-8174
Email: predrag@ee.columbia.edu
URL: http://www.ee.columbia.edu/∼predrag

Last lecture: Dimension reduction

- Dimensionality reduction idea
  - Represent/approximate X with a vector Z having less dimensions
  - Then, apply regression to Z
- Many approaches for doing this
- Common approach: Principal Component Analysis (PCA)
  Finding the first principal component: projection of x onto ϕ

$$z_{1i} = \langle \boldsymbol{x}_i, \boldsymbol{\phi}_1 \rangle = \phi_{11} x_{1i} + \phi_{21} x_{2i} + \cdots + \phi_{p1} x_{pi}$$

  Assume x_i-s are centered ($\sum x_i = 0$)
- Look for $\boldsymbol{\phi}_1$ that has the largest sample variance, i.e.

$$\max_{\boldsymbol{\phi}_1} \frac{1}{n} \sum_{i=1}^{n} z_{1i}^2 = \max_{\boldsymbol{\phi}_1} \frac{1}{n} \sum_{i=1}^{n} (\phi_{11} x_{1i} + \phi_{21} x_{2i} + \cdots + \phi_{p1} x_{pi})^2$$

  subject to $\sum_{j=1}^{p} \phi_{j1}^2 = 1$ (i.e., $\boldsymbol{\phi}_1$ is a unit vector)

Last lecture: Second principal component

- Loading vector ϕ_1 represents the direction along which the data varies the most
- If we project $x_1, \ldots, x_n$ onto ϕ_1, the projected values are the PC scores z_{i1} since

$$z_{1i} =< \boldsymbol{x}_i, \boldsymbol{\phi}_1 >$$

Second and higher principal components

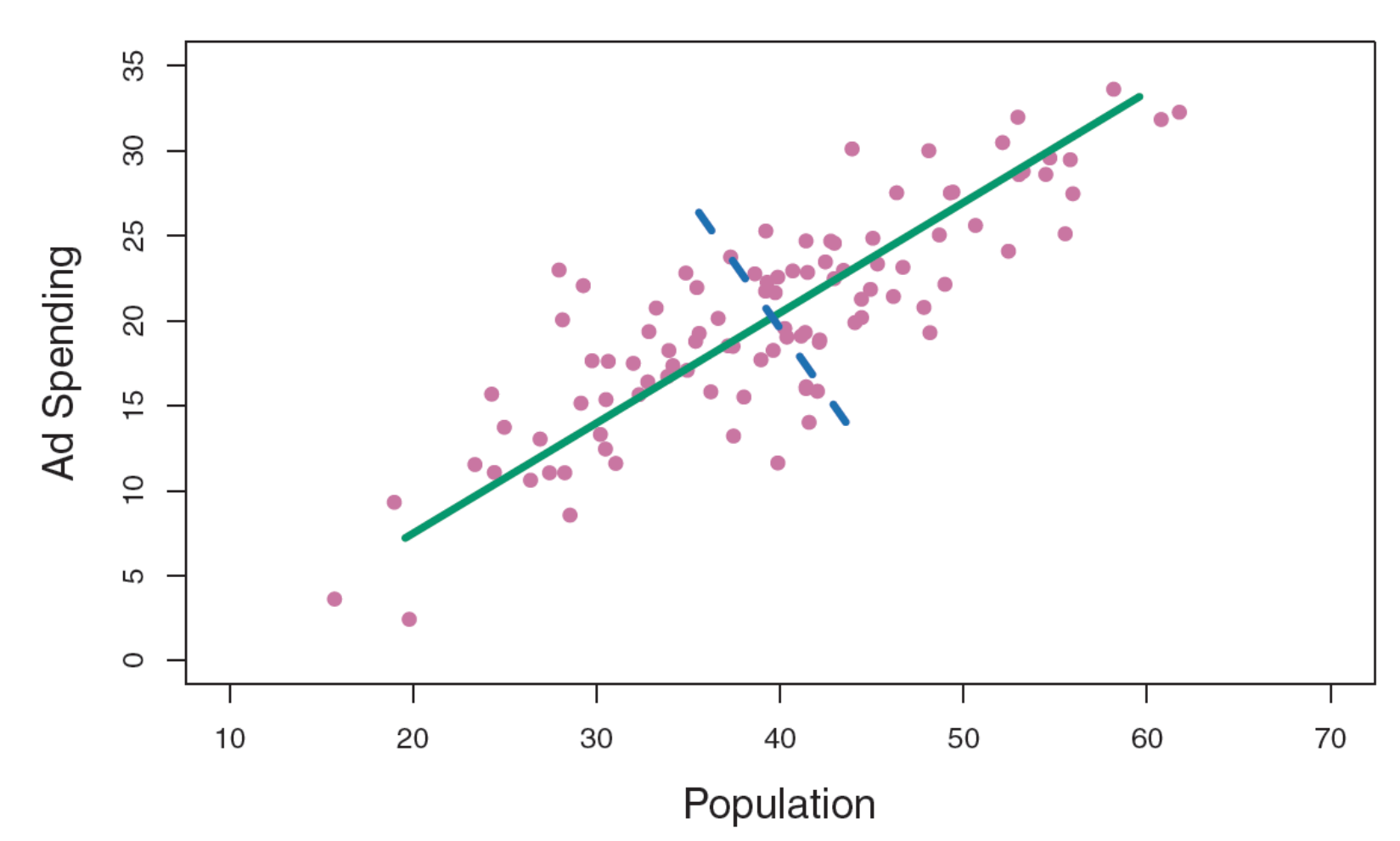
- After ϕ_1 has been determined, we look for ϕ_2 in a similar way, but with the additional constraint that ϕ_1, ϕ_2 are uncorrelated

$$< \boldsymbol{\phi}_1, \boldsymbol{\phi}_2 >= 0$$

- We continue this procedure until we find as many PC as we want
- This optimization problem can be solved via eigen-decomposition

PCA Example

Two PC-s: Solid line: First PC; Dashed line: Second PC



Last lecture: PCA as Eigenvalue-Eigenvector Decomposition

Consider finding $k \leq p$ principal components: $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_k$

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \cdots & \boldsymbol{\phi}_k \end{bmatrix}, \quad \boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_k.$$

Then, the projection of a data point $\boldsymbol{x}_i, 1 \leq i \leq n$ is given by

$$\hat{\boldsymbol{x}}_i = (\boldsymbol{x}_i \cdot \boldsymbol{\phi}_1)\boldsymbol{\phi}_1 + \cdots (\boldsymbol{x}_i \cdot \boldsymbol{\phi}_k)\boldsymbol{\phi}_k = \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{x}_i$$
$$\boldsymbol{z}_i = (z_{1i}, z_{2i}, \ldots, z_{ki}) = (\boldsymbol{x}_i \cdot \boldsymbol{\phi}_1, \boldsymbol{x}_i \cdot \boldsymbol{\phi}_2, \ldots, \boldsymbol{x}_i \cdot \boldsymbol{\phi}_k)$$

implying

$$\|\hat{\boldsymbol{x}}_i\|^2 = \boldsymbol{x}_i^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{x}_i = \boldsymbol{x}_i^\top \boldsymbol{U}U^\top \boldsymbol{x}_i.$$

Note that $\hat{\boldsymbol{x}}_i$ minimizes the distance $\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|$, and thus, finding k principle components is equivalent to finding $\boldsymbol{U}$ that maximizes

$$M = \max_{\boldsymbol{U}:\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_k} \sum_{i=1}^{n} \boldsymbol{x}_i^\top \boldsymbol{U}U^\top \boldsymbol{x}_i$$
$$= \text{trace}\left(\boldsymbol{U}^\top \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{U}\right), \quad (\text{using } \boldsymbol{x}^\top \boldsymbol{y} = \text{trace}(\boldsymbol{x}\boldsymbol{y}^\top))$$

Last lecture: PCA as Eigenvalue-Eigenvector Decomposition

Now, if $\boldsymbol{A} = \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top$, the optimization problem becomes

$$M = \max_{\boldsymbol{U}:\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_k} \text{trace}\left(\boldsymbol{U}^\top \boldsymbol{A} \boldsymbol{U}\right)$$

Note that $\boldsymbol{A}$ is a symmetric matrix, and therefore orthogonally diagonalizable with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. If $\boldsymbol{U}$ is composed of k eigenvectors that correspond to the k largest eigenvalues, then

$$M \geq \sum_{i=1}^{k} \lambda_i.$$

On the other hand, we can diagonalize $\boldsymbol{A} = \boldsymbol{V}^\top \boldsymbol{\Lambda} \boldsymbol{V}$, where $\boldsymbol{V}$ is an orthogonal matrix, yielding

$$M = \max_{\boldsymbol{U}:\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_k} \text{trace}\left(\boldsymbol{U}^\top \boldsymbol{V}^\top \boldsymbol{\Lambda} \boldsymbol{V} \boldsymbol{U}\right) = \max_{\boldsymbol{B}:\boldsymbol{B}^\top \boldsymbol{B} = \boldsymbol{I}_k} \text{trace}\left(\boldsymbol{B}^\top \boldsymbol{\Lambda} \boldsymbol{B}\right)$$

$$= \sum_{i=1}^{p} \sum_{j=1}^{k} b_{ij}^2 \lambda_i = \sum_{i=1}^{p} \lambda_i \sum_{j=1}^{k} b_{ij}^2 \leq \sum_{i=1}^{k} \lambda_i \quad \left(\text{Note: } \sum_{i=1}^{p} b_{ij}^2 = 1, \sum_{j=1}^{k} b_{ij}^2 \leq 1\right)$$

$\Rightarrow M = \sum_{i=1}^{k} \lambda_i$: Hence, the first k principal components correspond to the eigenvectors of the k largest eigenvalues of $\boldsymbol{A}$.

Last lecture: Basis expansions

- Map data $\boldsymbol{x}$ into higher dimensional space $\mathbb{R}^d, d > p$: $\boldsymbol{\phi}(\boldsymbol{x}) : \mathbb{R}^p \to \mathbb{R}^d$, i.e.,

$$\phi(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_d(\boldsymbol{x})$$

- Fit linear regression on $\boldsymbol{\phi}(\boldsymbol{x})$ in the higher dimensional space

Example: Polynomial regression of degree q

- Map data $\boldsymbol{x} = (x_1, \ldots, x_p)$ into

$$\boldsymbol{\phi}(\boldsymbol{x}) = (x_1, x_1^2, \ldots, x_1^q, x_2, x_2^2, \ldots, x_2^q, \ldots, x_p, x_p^2, \ldots, x_p^q, \ldots)$$

- Fit linear regression in the higher dimensional, $\mathbb{R}^{pq}$, space
  - Exponential growth of dimensionality: if $p = 10$ and $q = 10$

Polynomial Kernel and Dual Solution for Ridge

Polynomial Kernel: dot products can have closed form

$$K(x_i, x_j) := \langle \phi(x_i), \phi(x_j) \rangle = (1 + \langle x_j, x_i \rangle)^d$$

K is $n \times n$, doesn't grow with d; recall, n is the number of data points x_i.

- Dual solution - recall dual form for $\hat{\boldsymbol{y}}$

$$\hat{\boldsymbol{y}} = \boldsymbol{K}\boldsymbol{\alpha},$$

- Dual Ridge: find $\boldsymbol{\alpha}$ that minimizes

$$\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_2^2 = \|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_2^2$$

- Solution

$$\boldsymbol{\alpha} = (\lambda \boldsymbol{I} + \boldsymbol{K})^{-1} \boldsymbol{y}$$

  which has computational complexity $O(n^3)$, and does not depend on the polynomial basis expansion parameter d.

Other basis: piecewise polynomial, Fourier, wavelets, etc.

Last lecture: Model validation

Training versus Test Error

- Select a statistical learning method, e.g.: linear model, polynomial, piecewise polynomial/splines, etc.
- Training error: the average error from using the method to predict the response on the observations used in its training

- **Test error**: the average error from using the method to predict the response on a **new observation**

- Ideally: a large designated test set – rarely available

Last lecture: Validation-set approach

- Randomly divide the available samples into:
  - training set
  - validation set
- Random split into two halves

- Fit a model using the training set
- Use the model to predict the responses in the validation set
- The validation-set error is an estimate of the test error

- Drawbacks
  - The error estimate can be variable – depends on the split
  - Only a subset of observations used to fit the model
  - Tends to overestimate the test error

Last lecture: K-fold cross-validation

- Popular approach
  - Pro: scales well with data size
  - Con: there is still randomness

- Procedure
  - Randomly divide observations into K equal-sized parts
  - Leave out part k, fit a model using the remaining $K - 1$ parts
  - Use the left-out part to estimate the error
  - Repat for all k
  - Combine results

Last lecture: K-fold cross-validation

- K parts: $C_1, C_2, \ldots, C_K$
- $\cup_k C_k = \{1, \ldots, n\}$
- n_k: the number of observations in part k
- Compute

$$\mathsf{CV}_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} \mathsf{MSE}_k$$

where

$$\mathsf{MSE}_k = \frac{1}{n_k} \sum_{i \in C_k} (y_i - \hat{y}_i)^2$$

and $\hat{y}_i$ is the prediction for observation i obtained from the data without part k

- $K = n$: leave-one out cross-validation (LOOCV)

Last lecture: LOOCV

Linear model example: we can compute CV error

- Pro: **No randomness** – all subsets of size $(n-1)$ considered
- Con: Doesn't scale with data size
- Linear regression
  - $\boldsymbol{X}$ and $\boldsymbol{y}$:
    - observation i: $\boldsymbol{X}_i$ and y_i
    - no observation i: $\boldsymbol{X}_{(i)}$ and $\boldsymbol{y}_{(i)}$
  - CV error

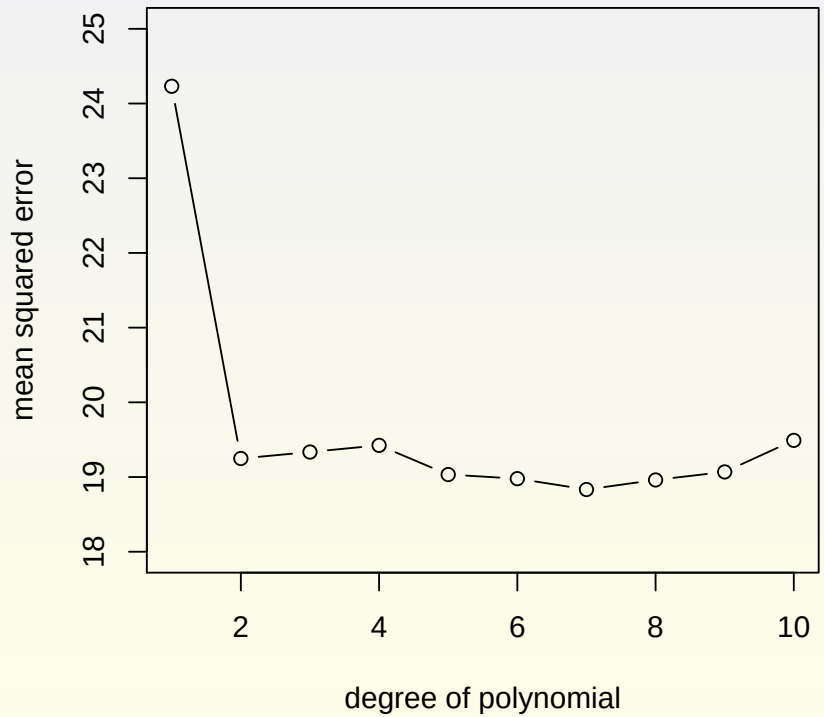
$$\begin{aligned}\text{CV}_{(n)} &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{X}_i\hat{\boldsymbol{\beta}}_{(i)})^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1-h_i}\right)^2\end{aligned}$$

    where $h_i = \boldsymbol{X}_i(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}_i^\top$
  - **Weighted sum of squared residuals**

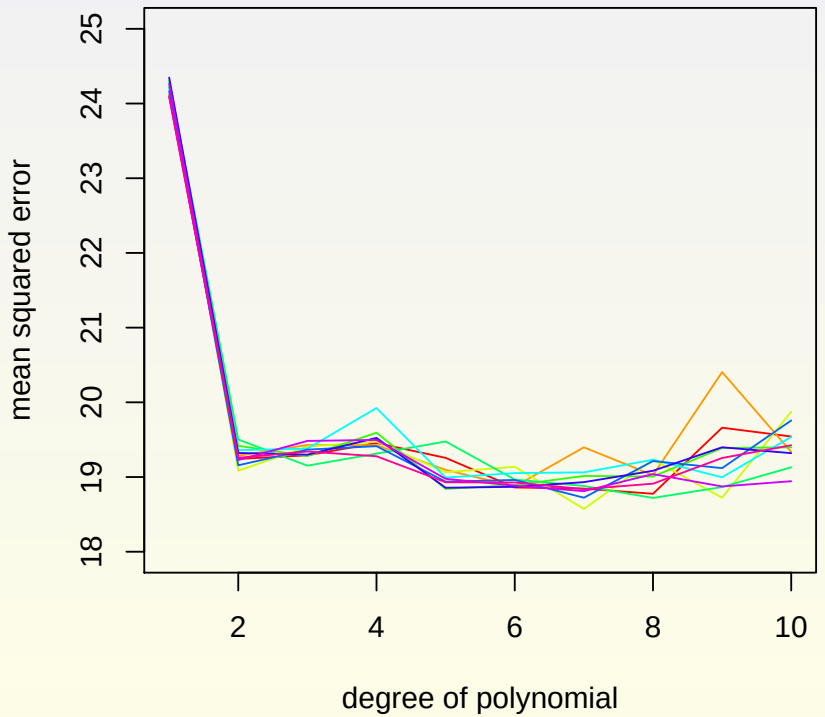
Example: Auto data set

```
> library(boot)
> cv.err<-rep(0,10)
> for(i in 1:10) {
+   glm.fit<-glm(mpg~poly(horsepower,i), data=auto)
+   cv.err[i]<-cv.glm(auto,glm.fit)$delta[1]
+ }

> set.seed(1)
> for(i in 1:10) {
+   glm.fit<-glm(mpg~poly(horsepower,i), data=auto)
+   cv.err[i]<-cv.glm(auto,glm.fit,K=10)$delta[1]
+ }
```

LOOCV



10−fold CV



Last lecture: Bootstrap

- Setup
  - Population model that produces an outcome Y
  - Observations $\boldsymbol{Z}$ from this population model
  - Statistic $T(\boldsymbol{Z})$
  - Distribution of $T(\boldsymbol{Z})$
- Idea
  - The distribution of $T(\boldsymbol{Z})$ can be estimated by sampling $\boldsymbol{Z}$ from the population model
  - Resample with replacement from $\boldsymbol{Z}$ to "approximate" sampling from the population model
- Why?
  - Only samples $\boldsymbol{Z}$ available
  - No information on the population model

Last lecture: Bootstrap

Basic algorithm

- Input
  - A sample of data $\boldsymbol{Z} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n)$
  - An estimation rule $\hat{T}$ for Statistic T
- Algorithm
  1. Generate bootstrap samples $\boldsymbol{Z}^{*1}, \boldsymbol{Z}^{*2}, \ldots, \boldsymbol{Z}^{*B}$
     - Create $\boldsymbol{Z}^{*b}$ by selecting n points from $\boldsymbol{Z}$
     - A particular $\boldsymbol{Z}_i$ can appear in $\boldsymbol{Z}^{*b}$ multiple times
  2. Evaluate the estimator on each $\boldsymbol{Z}^{*b}$:

$$\hat{T}_b = \hat{T}(\boldsymbol{Z}^{*b})$$

- The empirical distribution of $\{\hat{T}_1, \ldots, \hat{T}_B\}$ is an estimate of the distribution of $T(\boldsymbol{Z})$
- Bootstrap distribution
- Overlap between $\boldsymbol{Z}$ and $\boldsymbol{Z}^{*b}$?

Last lecture: Bootstrap regression modeling

- n observations, response $\boldsymbol{y}$, covariates $\boldsymbol{X}$
- Bootstrap standard errors for OLS coefficients using case resampling:
  - For $b = 1, \ldots, B$
  - Draw sample uniformly at random, with replacement, from observations $(\boldsymbol{X}, \boldsymbol{y})$. Let the ith outcome in the bth sample be $(\boldsymbol{X}_i^{*b}, y_i^{*b})$
  - Compute $\hat{\boldsymbol{\beta}}^{*b}$ given $(\boldsymbol{X}^{*b}, \boldsymbol{y}^{*b})$
- Bootstrap distribution of $\hat{\boldsymbol{\beta}}$ to compute standard errors

Few more words on regressions

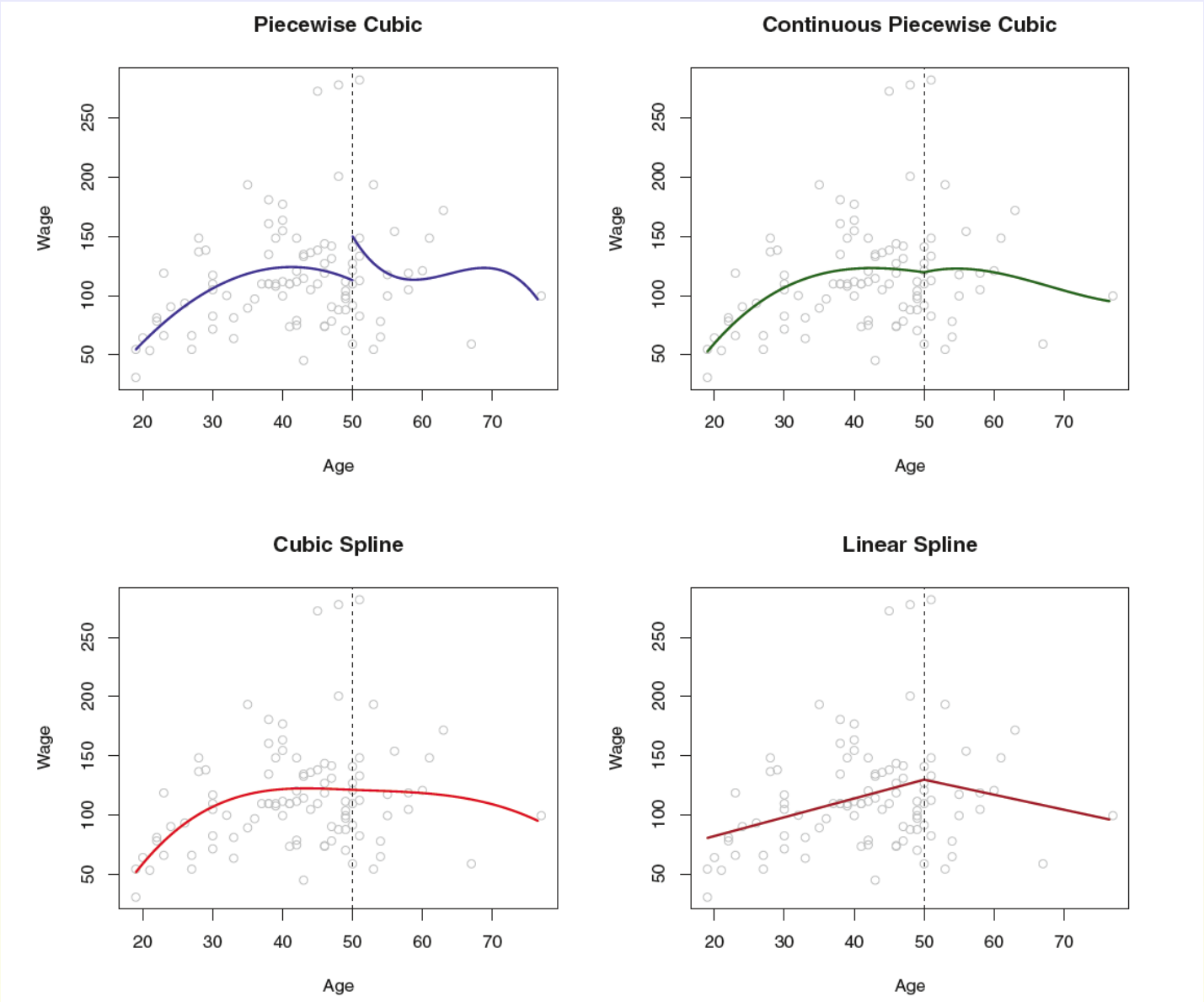
Regression: Estimate real valued/quantitative f

$$\boxed{Y = f(X)}$$

from training data $\{(x_i, y_i)\}$, and then use it for inference and prediction.

- Assume something on $\hat{f}$, e.g.: linear or polynomial
  Many other options for the approximation function $\hat{f}$, e.g.
  - Piecewise-polynomial/splines, e.g., piecewise-constant/linear; see Sec. 5.2 in ESL, Sec. 7.2-74 in ISL
  - Smoothing splines: impose smoothness at the boundaries, e.g. continuity, continuous derivatives, etc.; see Sec. 5.4 in ESL, Sec. 7.5 in ISL
- After selecting a class of approximation functions, we go through all the steps we did before:
  - Training (fitting)
  - Simplifying: model selection, regularization (e.g., Ridge, Lasso)
  - Testing: analytical, cross-validation, bootstrap
    keep an eye on overfitting

Example: Piecewise-polynomial, i.e., splines



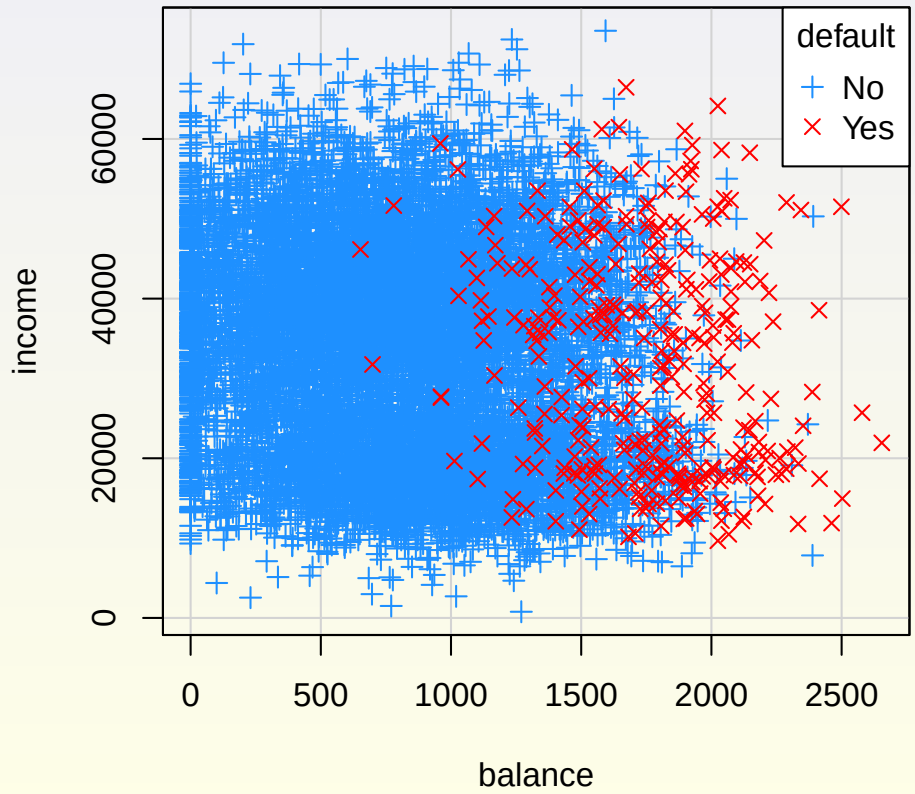
Classification

- Regression: real-valued/quantitative response
- Classification: categorical response
- Probability that a data point belongs to a class $c \in C$
- Example: Medical diagnosis
  - cancer, stroke, drug overdose, epileptic seizure
  - unordered set

Default data set

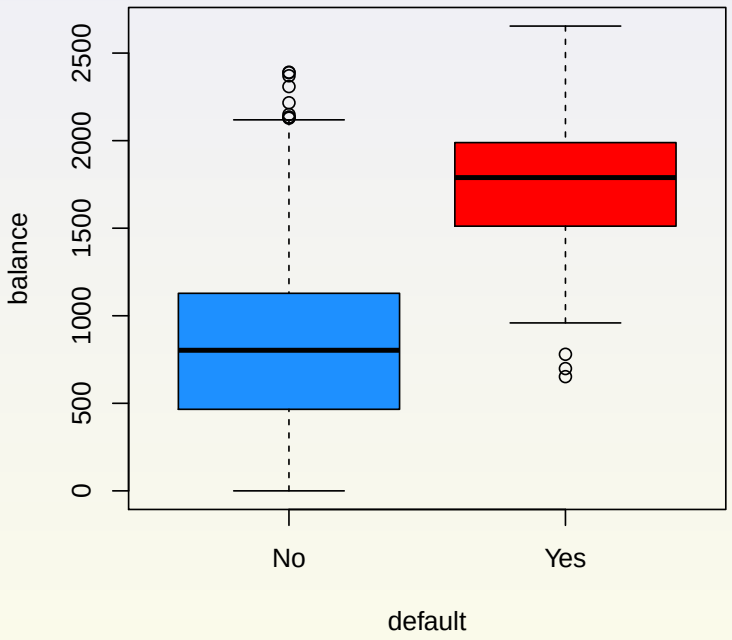
- Available in the ISLR package

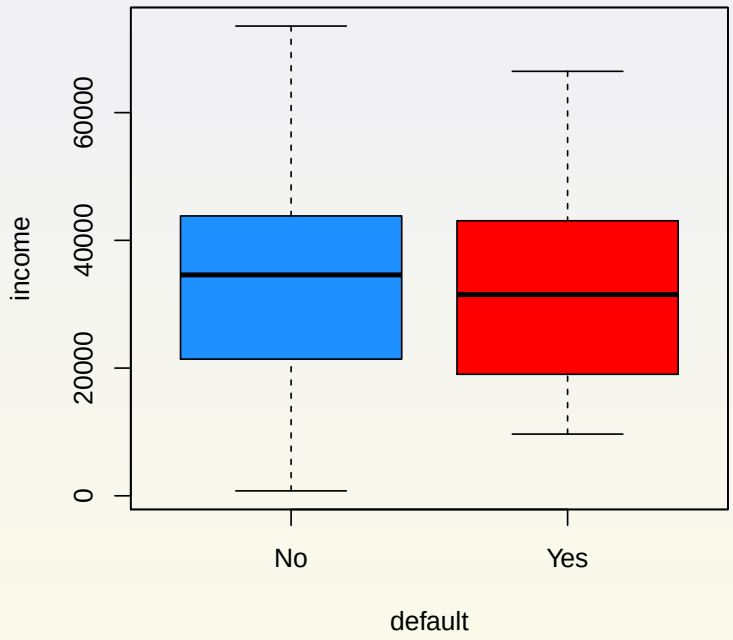
```
> summary(Default)
 default    student       balance           income
 No :9667   No :7056   Min.   :   0.0   Min.   :  772
 Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340
                       Median : 823.6   Median :34553
                       Mean   : 835.4   Mean   :33517
                       3rd Qu.:1166.3   3rd Qu.:43808
                       Max.   :2654.3   Max.   :73554
```



Default data set

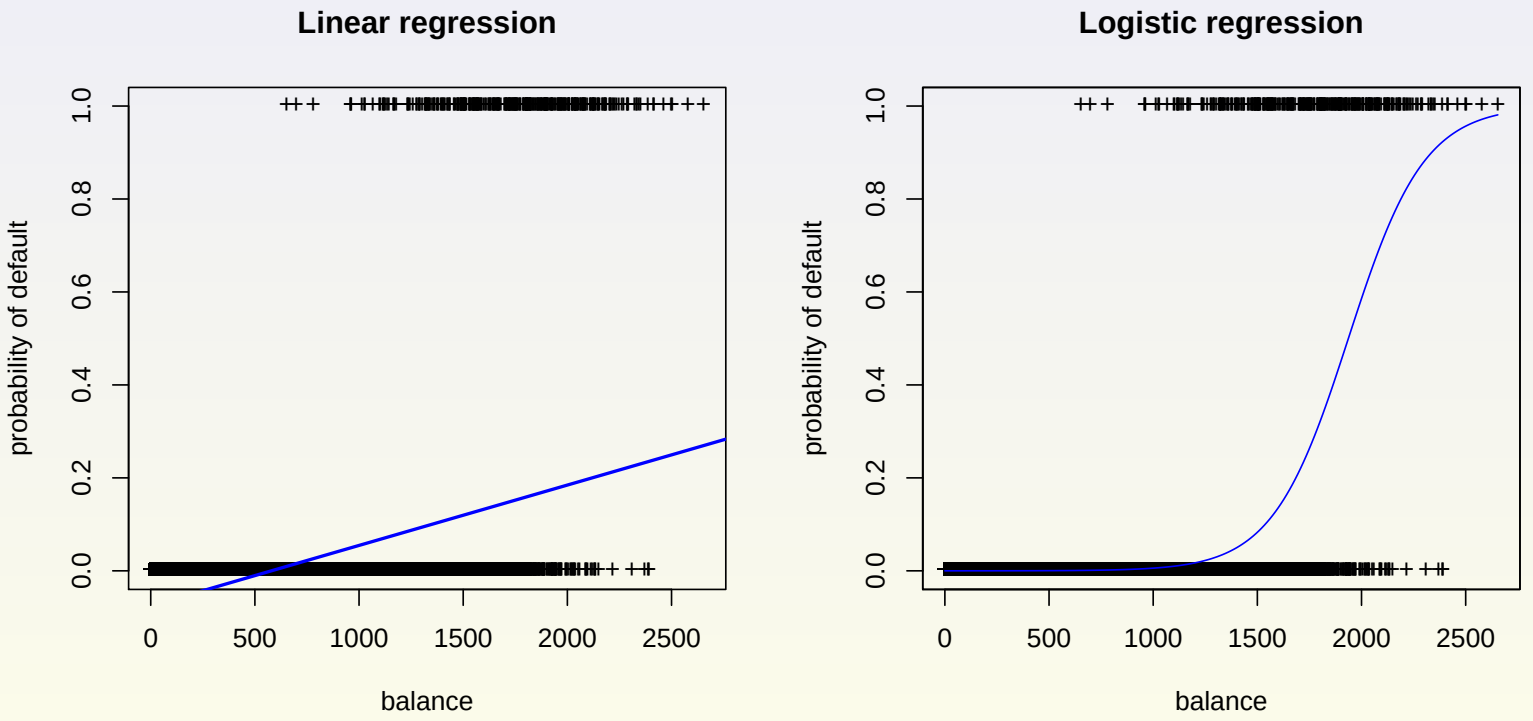
```
> par(mfrow=c(1,2))
> boxplot(balance~default,data=Default,col=c("dodgerblue","red"),xlab="default",ylab="balance")
> boxplot(income~default,data=Default,col=c("dodgerblue","red"),xlab="default",ylab="income")
```





Linear regression

- Binary variables
- Predicted values not always in $[0, 1]$



Classification setting

- Loss function: Error rate

$$\frac{1}{n}\sum_{i=1}^{n} 1_{\{y_i \neq \hat{y}_i\}}$$

- Bayes Classifier - Optimal: Assign an observation x_0 to a class j for which the conditional probability

$$\mathbb{P}[Y = j | X = x_0]$$

  is the largest. We will prove this later.

- Bayes error rate

$$1 - \mathbb{E}(\max_j \mathbb{P}[Y = j | X])$$

- We look for ways to approximate the conditional probabilities They are difficult to estimate/compute from data. Why?

Logistic regression

- Model the conditional probability of $\mathbb{P}[Y = j | X = x]$
- Example: $\mathbb{P}[\text{default=Yes} \,|\, \text{balance}] = p(\text{balance})$
- Need a function with values in $[0, 1]$

- Logistic function (for binary variables, can be extended)

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- Estimate $\boldsymbol{\beta}$ via Maximum Likelihood Estimation (MLE)
- Odds:

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X}$$

- A unit increase in X multiplies odds by e^{β_1}
- $\operatorname{logit} p(X) = \beta_0 + \beta_1 X$

Example

```
> glm1a<-glm(default~balance,data = Default,family = binomial())
> summary(glm1a)

Call:
glm(formula = default ~ balance, family = binomial(), data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2697  -0.1465  -0.0589  -0.0221   3.7589

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
        (Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
balance      5.499e-03  2.204e-04   24.95   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1596.5  on 9998  degrees of freedom
AIC: 1600.5

Number of Fisher Scoring iterations: 8
```

- Example:

$$\hat{\mathbb{P}}[\text{default} = \text{Yes} \,|\, \text{balance} = 1000] = \frac{e^{-10.65+0.0055\cdot 1000}}{1+e^{-10.65+0.0055\cdot 1000}} = 0.006$$

Example

```
> glm1b<-glm(default~student,data = Default,family = binomial())
> summary(glm1b)

Call:
glm(formula = default ~ student, family = binomial(), data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.2970  -0.2970  -0.2434  -0.2434   2.6585

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.50413    0.07071  -49.55  < 2e-16 ***
studentYes   0.40489    0.11502    3.52 0.000431 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 2908.7  on 9998  degrees of freedom
AIC: 2912.7

Number of Fisher Scoring iterations: 6
```

- Example:

$$\hat{\mathbb{P}}[\text{default} = \text{Yes} \,|\, \text{student} = \text{Yes}] = \frac{e^{-3.504+0.405\cdot 1}}{1+e^{-3.504+0.405\cdot 1}} = 0.043$$

MLE

- The probability of observed data under the model specified by $\boldsymbol{\beta}$ is given by the likelihood function

$$\ell(\boldsymbol{\beta}) = \prod_{i:\, y_i=1} p(x_i) \prod_{i:\, y_i=0} (1 - p(x_i))$$

- MLE: select a model that maximizes likelihood of data

$$\max_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta})$$

or equivalently

$$\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left\{ y_i(\beta_0 + \beta_1 x_i) - \ln\left(1 + e^{\beta_0 + \beta_1 x_i}\right) \right\}$$

- First-order conditions
- Newton's method

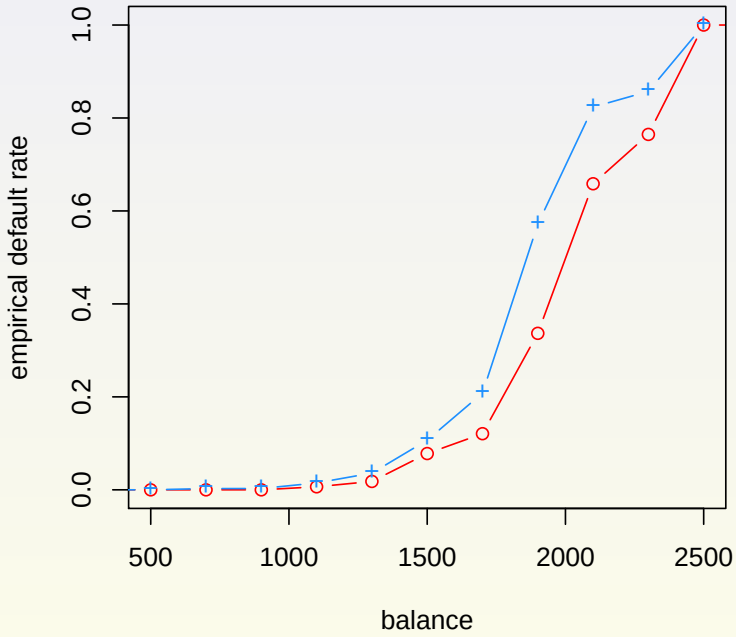
Multiple logistic regression

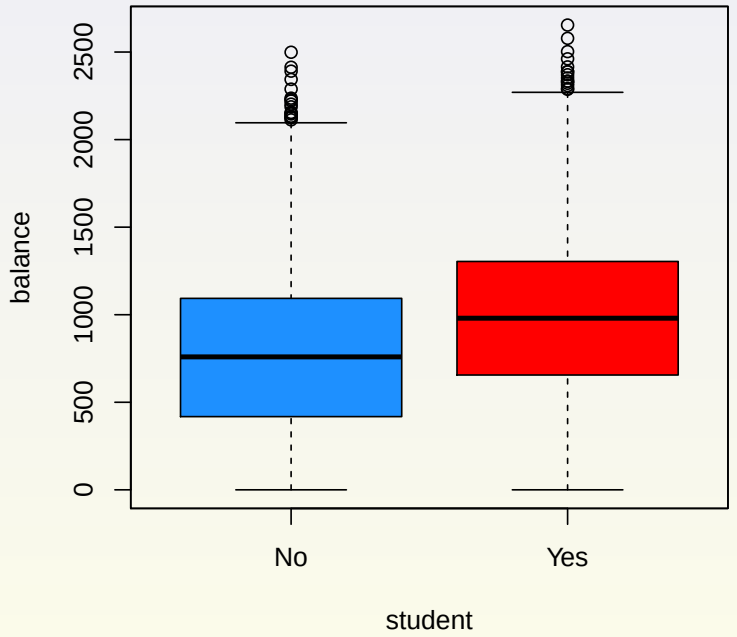
$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}$$

```
> glm2<-glm(default~balance+income+student,data = Default,family = binomial())
> summary(glm2)

Call:
glm(formula = default ~ balance + income + student, family = binomial(),
    data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4691  -0.1418  -0.0557  -0.0203   3.7383

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
income       3.033e-06  8.203e-06   0.370  0.71152
studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1571.5  on 9996  degrees of freedom
AIC: 1579.5

Number of Fisher Scoring iterations: 8
```

Confounding

- Dependency among predictors
- Similar to colinearity





South African heart disease data set

```
> head(SAheart)
  sbp tobacco  ldl adiposity famhist typea obesity alcohol age chd
1 160   12.00 5.73     23.11 Present    49   25.30   97.20  52   1
2 144    0.01 4.41     28.61  Absent    55   28.87    2.06  63   1
3 118    0.08 3.48     32.28 Present    52   29.14    3.81  46   0
4 170    7.50 6.41     38.03 Present    51   31.99   24.26  58   1
5 134   13.60 3.50     27.78 Present    60   25.99   57.34  49   1
6 132    6.20 6.47     36.21 Present    62   30.77   14.14  45   0
```

```
> summary(SAheart)
      sbp             tobacco            ldl              adiposity      famhist        typea      
 Min.   :101.0   Min.   : 0.0000   Min.   : 0.980   Min.   : 6.74   Absent :270   Min.   :13.0  
 1st Qu.:124.0   1st Qu.: 0.0525   1st Qu.: 3.283   1st Qu.:19.77   Present:192   1st Qu.:47.0  
 Median :134.0   Median : 2.0000   Median : 4.340   Median :26.11                 Median :53.0  
 Mean   :138.3   Mean   : 3.6356   Mean   : 4.740   Mean   :25.41                 Mean   :53.1  
 3rd Qu.:148.0   3rd Qu.: 5.5000   3rd Qu.: 5.790   3rd Qu.:31.23                 3rd Qu.:60.0  
 Max.   :218.0   Max.   :31.2000   Max.   :15.330   Max.   :42.49                 Max.   :78.0  
    obesity          alcohol            age              chd        
 Min.   :14.70   Min.   :  0.00   Min.   :15.00   Min.   :0.0000  
 1st Qu.:22.98   1st Qu.:  0.51   1st Qu.:31.00   1st Qu.:0.0000  
 Median :25.80   Median :  7.51   Median :45.00   Median :0.0000  
 Mean   :26.04   Mean   : 17.04   Mean   :42.82   Mean   :0.3463  
 3rd Qu.:28.50   3rd Qu.: 23.89   3rd Qu.:55.00   3rd Qu.:1.0000  
 Max.   :46.58   Max.   :147.19   Max.   :64.00   Max.   :1.0000  
```

- Prevalence in the region (not the data set) ≈ 0.05
- Adjust β_0 :

$$\hat{\beta}_0^* = \hat{\beta}_0 + \log \frac{\pi}{1-\pi} - \log \frac{\tilde{\pi}}{1-\tilde{\pi}}$$

Example

```
> heartfit<-glm(chd~.,data=SAheart,family = binomial())
> summary(heartfit)

Call:
glm(formula = chd ~ ., family = binomial(), data = SAheart)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7781  -0.8213  -0.4387   0.8889   2.5435

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -6.1507209  1.3082600  -4.701 2.58e-06 ***
sbp             0.0065040  0.0057304   1.135 0.256374
tobacco         0.0793764  0.0266028   2.984 0.002847 **
ldl             0.1739239  0.0596617   2.915 0.003555 **
adiposity       0.0185866  0.0292894   0.635 0.525700
famhistPresent  0.9253704  0.2278940   4.061 4.90e-05 ***
typea           0.0395950  0.0123202   3.214 0.001310 **
obesity        -0.0629099  0.0442477  -1.422 0.155095
alcohol         0.0001217  0.0044832   0.027 0.978350
age             0.0452253  0.0121298   3.728 0.000193 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 596.11  on 461  degrees of freedom
Residual deviance: 472.14  on 452  degrees of freedom
AIC: 492.14

Number of Fisher Scoring iterations: 5
```

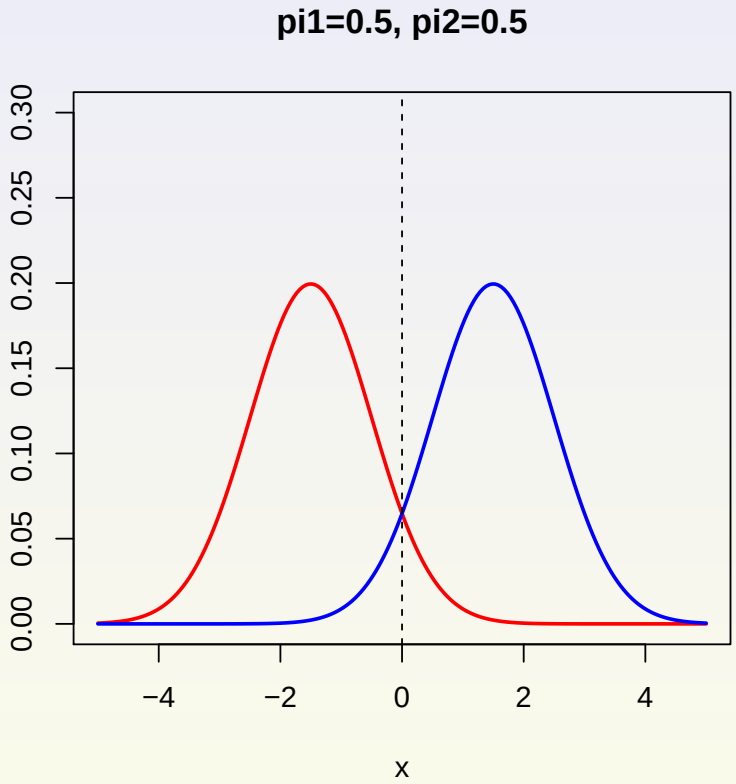
Discriminant classification

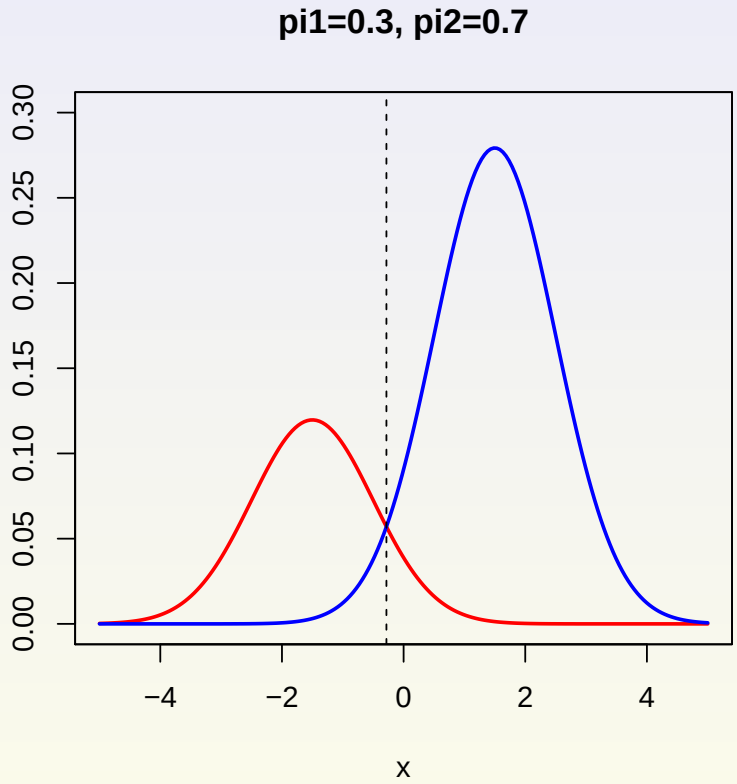
- Model the distribution of X for each class separately
  - Will use Gaussian distribution
  - Leads to linear or quadratic discriminant analysis
  - Possible to use other distributions
- Bayes theorem

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

  where
  - f_k is the density of X in class k
  - π_k is the prior probability for class k
- Classify to the most likely class – the highest $\pi_k f_k(x)$

Example


pi1=0.5, pi2=0.5
0.00
0.05
0.10
0.15
0.20
0.25
0.30
−4
−2
0
2
4
x


pi1=0.3, pi2=0.7
0.00
0.05
0.10
0.15
0.20
0.25
0.30
−4
−2
0
2
4
x

Discriminant analysis

- Start with $p = 1$
- Gaussian density (mean μ_k, variance σ_k^2):

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

- Discriminant function δ_k is quadratic (in x):

$$p_k(x) \propto \delta_k(x) = -x^2 \frac{1}{2\sigma_k^2} + x\frac{\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} - \log \sigma_k + \log \pi_k$$

- Probabilities:

$$\mathbb{P}[Y = k \,|\, X = x] = \frac{e^{\delta_k(x)}}{\sum_{l=1}^{K} e^{\delta_l(x)}}$$

Linear discriminant analysis

- Special case: $\sigma_1 = \sigma_2 = \ldots = \sigma_K = \sigma$
- Discriminant function δ_k is linear (in x):

$$p_k(x) \propto \delta_k(x) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k$$

- Example: $K = 2$, $\pi_1 = \pi_2$ – decision boundary is at

$$x = \frac{\mu_1 + \mu_2}{2}$$

- Parameter estimation:

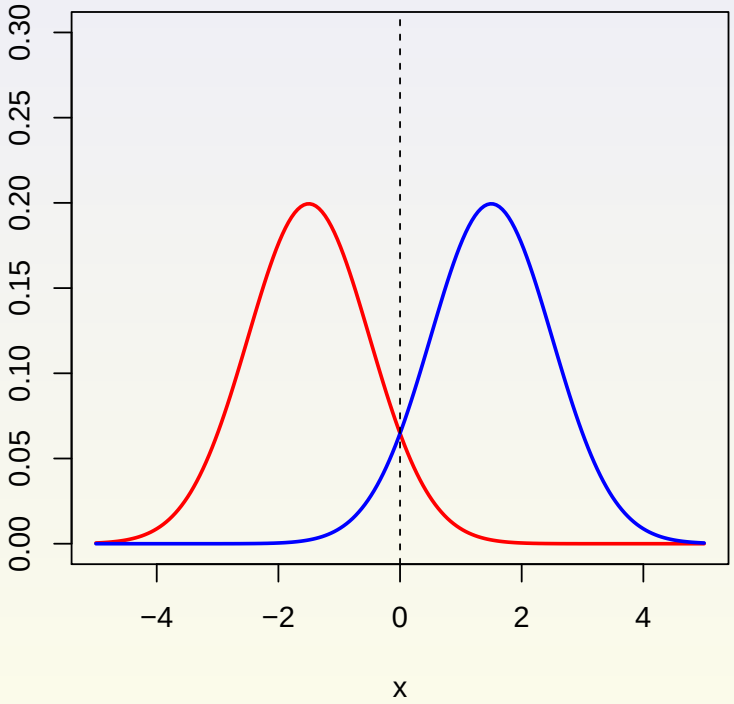
$$\hat{\pi}_k = \frac{n_k}{n}$$

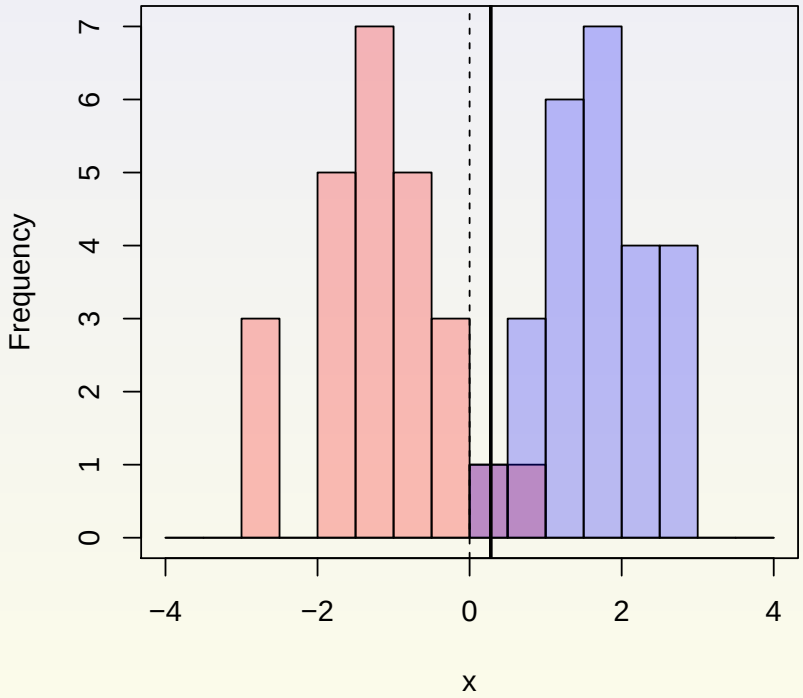
$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:\, y_i = k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n - K} \sum_{k=1}^{K} \sum_{i:\, y_i = k} (x_i - \hat{\mu}_k)^2$$

Example

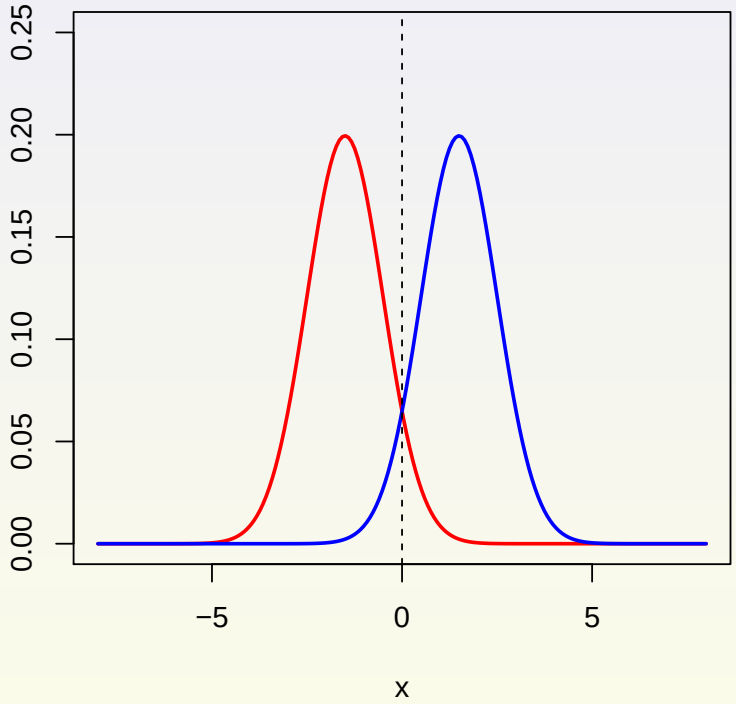
In practice - finite number of samples = deviations

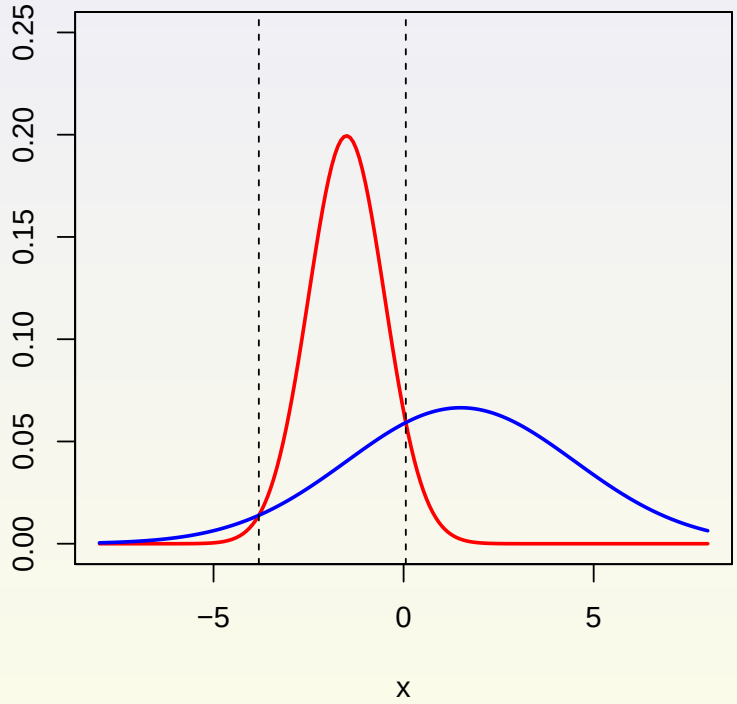




Linear vs. quadratic

Decision region more complicated





Discriminant analysis for $p > 1$

Quadratic - sigma unequal

- Density:

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)}$$

- Linear discriminant function (equal $\boldsymbol{\Sigma}_k$):

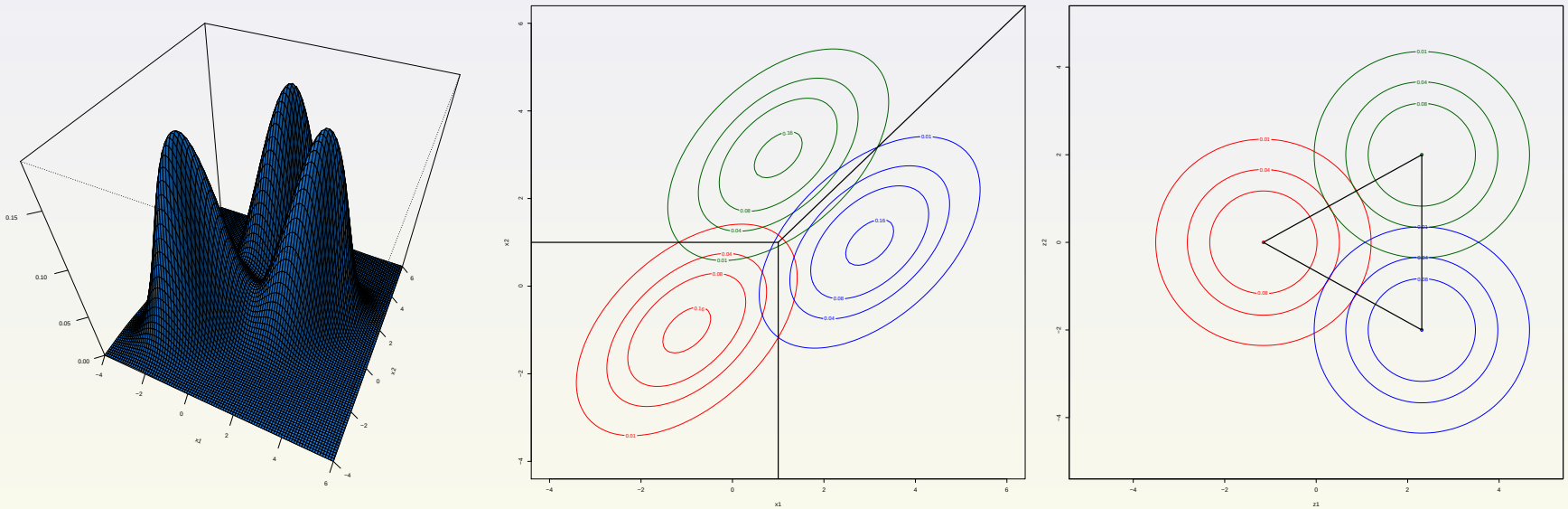
$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log \pi_k$$

- Quadratic discriminant function (different $\boldsymbol{\Sigma}_k$):

$$\delta_k(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log \pi_k$$

Example

- $p = 2$, $K = 3$, $\pi_1 = \pi_2 = \pi_3 = 1/3$
- Coordinate transformation



- Sufficient to consider a $(K - 1)$-dimensional hyperplane

Logistic regression vs. LDA

- Two classes
- Logistic regression

$$\log \frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})} = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

- LDA

$$\log \frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})} = \left(\log \frac{\pi_1}{\pi_2} - \frac{1}{2} \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 + \frac{1}{2} \boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 \right) + \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 \right)$$

- Same linear form
- Different way to estimate parameters

Naïve Bayes

- Assumes features are independent in each class
- Covariance matrices $\boldsymbol{\Sigma}_k$ are diagonal:

$$\pi_k f_k(\boldsymbol{x}) = \pi_k \prod_{i=1}^{p} f_{ki}(x_i) = \pi_k \prod_{i=1}^{p} \frac{1}{\sqrt{2\pi}\sigma_{ki}} e^{-\frac{(x_i-\mu_{ki})^2}{2\sigma_{ki}^2}}$$

and

$$\delta_k(\boldsymbol{x}) = -\sum_{i=1}^{p} \left[\frac{(x_i - \mu_{ki})^2}{2\sigma_{ki}^2} + \log \sigma_{ki} \right] + \log \pi_k$$

- Advantages
  - much easier to estimate parameters for $p \gg 1$
  - can use both qualitative and categorical features (use PMFs instead of PDFs)
  - often produces good results

Naïve Bayes: Example

Name	Over 170cm	Eye	Hair length	Sex
Drew	No	Blue	Short	Male
Claudia	Yes	Brown	Long	Female
Drew	No	Blue	Long	Female
Drew	No	Blue	Long	Female
Alberto	Yes	Brown	Short	Male
Karin	No	Blue	Long	Female
Nina	Yes	Brown	Short	Female
Sergio	Yes	Blue	Long	Male

Sex	Name	$\hat{p}$
Male	Drew	$1/3$
	!Drew	$2/3$
Female	Drew	$2/5$
	!Drew	$3/5$

Sex	Over 170cm	$\hat{p}$
Male	Yes	$2/3$
	No	$1/3$
Female	Yes	$2/5$
	No	$3/5$

Sex	Eye	$\hat{p}$
Male	Blue	$2/3$
	Brown	$1/3$
Female	Blue	$3/5$
	Brown	$2/5$

Sex	Hair length	$\hat{p}$
Male	Short	$2/3$
	Long	$1/3$
Female	Short	$1/5$
	Long	$4/5$

$$\{\texttt{Name} = \texttt{Drew}, \texttt{Over 170cm} = \texttt{Yes}, \texttt{Eye} = \texttt{Blue}, \texttt{Hair length} = \texttt{Long}\} = ?$$

Proof of optimality of the Bayes Classifier

- Consider the case of two classes $Y \in \{0, 1\}$ and general X, $X \in \mathbb{R}^p$
- The proof is not needed for the grade.

Definition (Bayes Classifier)

Let $\eta(x) = \mathbb{P}[Y = 1 \,|\, X = x]$ and let the Bayes Classifier be defined as

$$f^*(x) = \begin{cases} 1, & \text{if } \eta(x) \geq 1/2 \\ 0, & \text{otherwise,} \end{cases}$$

i.e., it assigns x to a class k for which $\mathbb{P}[Y = k \,|\, X = x]$ has maximum value.

Theorem (Optimality)

For any classifier $g(x) \in \{0, 1\}$,

$$\mathbb{P}[g(X) \neq Y] \geq \mathbb{P}[f^*(X) \neq Y],$$

i.e., the Bayes classifier is optimal.

Proof of optimality of the Bayes Classifier

Proof. We will actually prove a stronger statement that

$$\mathbb{P}[g(X) \neq Y | X = x] \geq \mathbb{P}[f^*(X) \neq Y | X = x],$$

which by taking the expectation with respect to X yields the theorem.

$$\begin{aligned}
&\mathbb{P}[g(X) \neq Y | X = x] = 1 - \mathbb{P}[g(X) = Y | X = x] \\
&= 1 - (\mathbb{P}[Y = 1, g(X) = 1 | X = x] + \mathbb{P}[Y = 0, g(X) = 0 | X = x]) \\
&= 1 - \left(\mathbb{E}[1_{\{Y=1\}} 1_{\{g(X)=1\}} | X = x] + \mathbb{E}[1_{\{Y=0\}} 1_{\{g(X)=0\}} | X = x]\right) \\
&= 1 - \left(1_{\{g(x)=1\}} \mathbb{E}[1_{\{Y=1\}} | X = x] + 1_{\{g(X)=0\}} \mathbb{E}[1_{\{Y=0\}} | X = x]\right) \\
&= 1 - \left(1_{\{g(x)=1\}} \mathbb{P}[Y = 1 | X = x] + 1_{\{g(x)=0\}} \mathbb{P}[Y = 0 | X = x]\right) \\
&= 1 - \left(1_{\{g(x)=1\}} \eta(x) + 1_{\{g(x)=0\}} (1 - \eta(x))\right)
\end{aligned}$$

Proof of optimality of the Bayes Classifier

Next, consider the difference

$$\begin{aligned}
&\mathbb{P}[g(X) \neq Y | X = x] - \mathbb{P}[f^*(X) \neq Y | X = x] \\
&= \eta(x) \left(1_{\{f^*(x)=1\}} - 1_{\{g(x)=1\}}\right) + (1 - \eta(x)) \left(1_{\{f^*(x)=0\}} - 1_{\{g(x)=0\}}\right) \\
&= (2\eta(x) - 1) \left(1_{\{f^*(x)=1\}} - 1_{\{g(x)=1\}}\right),
\end{aligned}$$

where the last equality follows from $1_{\{g(x)=0\}} = 1 - 1_{\{g(x)=1\}}$.
Finally, we show that the last expression is nonnegative. To this end, consider the following two cases:

1. $f^*(x) = 1 \Leftrightarrow \eta(x) \geq 1/2$, and therefore

$$(2\eta(x)-1) \left(1_{\{f^*(x)=1\}} - 1_{\{g(x)=1\}}\right) = (2\eta(x)-1) \left(1 - 1_{\{g(x)=1\}}\right) \geq 0$$

2. $f^*(x) = 0 \Leftrightarrow \eta(x) < 1/2$, which also imples

$$(2\eta(x)-1) \left(1_{\{f^*(x)=1\}} - 1_{\{g(x)=1\}}\right) = (2\eta(x)-1) \left(0 - 1_{\{g(x)=1\}}\right) \geq 0$$

□

Reading:

ISL: Read Chapter 4

ESL: Read Chapter 4

Homework 2: Due Fri, Oct 7th, by 11:59pm.

Midterm planned for Oct 25th