202A Fall 2017

Pro. Wu

Lecture 2
10/2/2017

Linear regression.

	Predictors 1	2	… j	p (genes)	outcome
1	x_{11}	x_{12}		x_{1p}	y_1
2	x_{21}	x_{22}		x_{2p}	y_2
⋮			x_{ij}	x_{ip}	y_i
n (people)					y_n

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i$$

$$\left(\beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j\right)$$

$\downarrow$ intersect $\qquad \beta_1$ = intersect

$$R(\beta) = \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2$$

$n \gg p$ classic

$p \gg n$ morden

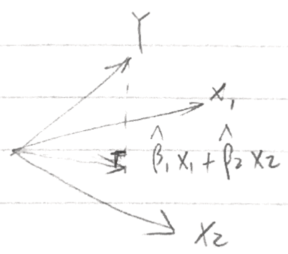
$n \; \underset{n\times p}{X} \; \underset{n\times 1}{Y}$ (X has p columns)

	1 . . . . . p
1	
2	
i	x_i^T
n	

$$x_i^T = \begin{pmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{pmatrix} \qquad \beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

	1	2	… j …	p	
1					
i	X_1	X_2	…	X_p	Y
n					

$$Y = \sum_{j=1}^{p} X_j \cdot \beta_j + \varepsilon \qquad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$



$$y_i = x_i^T\beta + \varepsilon_i$$

$$R(\beta) = \sum_{i=1}^{n}\left(y_i - x_i^T\beta_i\right)^2$$

least squares projection

$$R(\beta) = \left\| Y - \sum_{j=1}^{p} X_j\beta_j \right\|^2$$

$$\|Y - X\beta\|^2 + \lambda\|\beta\|$$

$$\|\beta\|_{\ell_1} = \sum_{j=1}^{p} |\beta_j|$$

$$\|\beta\|_{\ell_2}^2 = \sum_{j=1}^{p} \beta_j^2$$

$E[(X^TX)^{-1}X^T\varepsilon$

$n\times 1 - (n\times p)\times(p\times 1)$ $\quad n\times 1$

$$R(\beta) = |Y - X\beta|^2$$

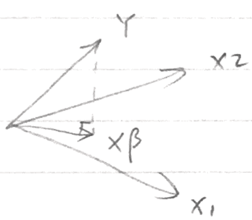
$$R'(\beta) = -2X^T(Y - X\beta) = 0$$

$p\times 1$? ✓ $\quad p\times n \quad n\times 1$

$$X^TX\beta = X^TY$$

$$\hat\beta = \boxed{(X^TX)^{-1}X^TY} \ ①$$

$E\left[\left[(X^TX)^{-1}X^T\varepsilon\right]^T\left[(X^TX)^{-1}X^T\varepsilon\right]\right]$



$X_j \perp Y - X\beta$

$X_j^T(Y - X\beta) = 0$

$X^T(Y - X\beta) = 0$

$Var(X) = E(X^2) - (E(X))^2$

$= E(X^2)$ $\quad 0$

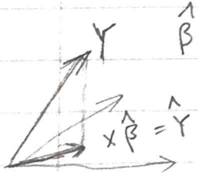
$(X^TX)^{-1}X^T\sigma^2 I_n (X^TX)^{-1}X^T$

$Var(\_\_ - E)^T(\_\_ - E)$

True model $\quad Y = X\beta_{true} + \varepsilon$

$\varepsilon \sim N(0, \sigma^2 I_n)$ $\quad Var[(\_\_ - \mu)^2]$

$E[\varepsilon] = 0 \quad Var(\varepsilon) = \sigma^2 I_n$



$$\hat\beta = (X^TX)^{-1}X^TY = (X^TX)^{-1}X^T(X\beta_{true} + \varepsilon)$$

$$= \beta_{true} + \underbrace{(X^TX)^{-1}X^T}_{a}\varepsilon \longrightarrow E[(X^TX)^{-1}X^T\varepsilon] = E(\varepsilon) = 0$$

$\therefore$

$\boxed{(X^TX)^{-1} = \sigma^2(X^TX)^{-1}}$ ②

$$X\hat\beta = X(X^TX)^{-1}X^TY$$

$$\hat Y = HY$$

$$e = Y - X\hat\beta$$

$$|e|^2 = \sum_{i=1}^{n}(y_i - x_i^T\hat\beta)^2$$

$$RSS = Y^TY - \hat Y^T\hat Y = Y^TY - Y^TH^THY = \boxed{Y^TY - Y^TX(X^TX)^{-1}X^TY} \ ③$$

$$\|Y - \hat Y\|^2 = (Y - \hat Y)^T(Y - \hat Y) = Y^TY - Y^T\hat Y - \hat Y^TY + \hat Y^T\hat Y$$?

Residual sum of squre

$(Y^T - \hat Y^T)(Y - \hat Y)$

$$= Y^TY - 2Y^T\hat Y + \hat Y^T\hat Y$$

$$= Y^TY - 2Y^T(X\hat\beta) + (\hat\beta X^T)(X\hat\beta)$$

$$= Y^TY - 2Y^TX\hat\beta + \hat\beta X^TX\hat\beta$$

Lecture 3

10/5/2017

Linear Regression

	p	
	X	Y

$Y = X\beta + \varepsilon$ ✓

$E(\varepsilon) = 0$ ✓

$Var(\varepsilon) = \sigma^2 I_n$ ✓

$R^2 = |Y - X\beta|^2$

$R'(\beta) = -2X^T(Y - X\beta) = 0$

$\hat{\beta}_{LS} = \boxed{(X^T X)^{-1} X^T Y}$

least square

$E(\hat{\beta}) = \beta_{\sigma^2}$

$Var(\hat{\beta}) = \sigma^2 \boxed{(X^T X)^{-1}}$

$e = Y - X\hat{\beta}$

$RSS = |e|^2 = \boxed{Y^T Y - Y^T \underbrace{X(X^T X)^{-1} X^T}_{H} Y}$

$\hat{\sigma}^2 = \frac{RSS}{n}$ or $\frac{RSS}{n-p}$ (over fitting)

Gauss-Jordan elimination

$$\begin{cases} x_1 + x_2 + x_3 = 5 \\ 2x_1 + 3x_2 + 5x_3 = 8 \\ 4x_1 \qquad + 5x_3 = 2 \end{cases}$$

$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 0 & 5 \end{bmatrix}$ $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ $b = \begin{pmatrix} 5 \\ 8 \\ 2 \end{pmatrix}$

$Ax = b$

$x = A^{-1} b.$

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 5 \\ 2 & 3 & 5 & 8 \\ 4 & 0 & 5 & 2 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 1 & 1 & 1 & 5 \\ 0 & 1 & 3 & -2 \\ 0 & -4 & 1 & -18 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 1 & 0 & -2 & 7 \\ 0 & 1 & 3 & -2 \\ & 0 & 13 & -26 \end{array}\right]$$

$$\rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & -2 \end{array}\right]$$

$$[A \mid b] \xrightarrow{GJ[1 \dots n]} [I \mid A^{-1}b]$$

$GJ[1 \dots n] = A^{-1}$

$A^{-1}[A \mid b] = [I \mid A^{-1}b]$

$[A \mid I] \xrightarrow{GJ[1 \dots n]} A^{-1}[A \mid I] = [I \mid A^{-1}]$

$a_{j1} \quad \frac{a_{ij} \cdot a_{ij}}{a_{11}}$

$\uparrow$

$\cancel{a}1 \quad \frac{a_{ij}}{a_{11}}$

$\uparrow$

$a_{11} \quad a_{ij}$

a_{j1}

$A = (a_{ij})_{n \times n}$

a_{11}

$a_{21} \quad a_{22} \quad a$

a_{31}

$$B = {}^{n}[A \mid I] \xrightarrow{GJ[1]} \begin{bmatrix} 1 & \dots \frac{a_{ij}}{a_{11}} & \Bigg| & \frac{1}{a_{11}} & \\ 0 & & & & \\ 0 & a_{ij} - \frac{a_{ij} a_{ij}}{a_{11}} & & -\frac{a_{i1}}{a_{11}} & \dots \end{bmatrix}$$

$X^TX \quad X^TY \quad m \times m$

$$\begin{bmatrix} A_{11} & A_{12} & \Big| & I_1 & 0 \\ A_{21} & A_{22} & & 0 & I_2 \end{bmatrix} \xrightarrow{GJ[1 \dots m]} \begin{bmatrix} I_1 & A_{11}^{-1}A_{12} & \Big| & A_{11}^{-1} & 0 \\ 0 & A_{22} - A_{21}A_{11}^{-1}A_{12} & & -A_{21}A_{11} & I \end{bmatrix}$$

B''

$Y^TX \qquad (n-m) \times (n-m) \quad Y^TY$

A_{11}^{-1}

$B = A + I$

$n \times (n$

$$Z = (\underset{n \times p}{X} \quad \underset{n \times 1}{Y})$$

$$Z^TZ = \begin{pmatrix} X^T \\ Y^T \end{pmatrix} (X \ Y) = \left[\begin{pmatrix} X^TX & X^TY \\ Y^TX & Y^TY \end{pmatrix} \Bigg| I \right]$$

$$\xrightarrow{GJ[1 \dots m]} = \left(\begin{matrix} I_1 & \hat{\beta} \\ 0 & RSS \end{matrix} \Bigg| \begin{matrix} \frac{1}{\sigma} & 0 \\ -\hat{\beta}^T & I_2 \end{matrix} \right]$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \downarrow \begin{bmatrix} I & \frac{a_{12}}{a_{11}} \\ a_{21} & a_{22} \end{bmatrix} \downarrow \begin{bmatrix} I & \frac{a_{12}}{a_{11}} \\ a_{21} - I \cdot a_{21} & a_{22} - \frac{a_{12}}{a_{11}} \cdot a_{21} \end{bmatrix}$$

a_{31}

$m < n$

$$n \begin{bmatrix} \overset{m \times m}{A_{11}} & A_{12} \\ (n-m) \times x \quad A & \\ A_{21} & A_{22} \end{bmatrix}$$

$m \quad n$

$m \times m$	$m \times (n-m)$
$(n-m) \times m$	$~~(n-m)~~ (n-m) \times (n-m)$

n

R

```
my Guss jordon <- function (A, m)

  n= dim(A) [1]
  B<- cbind (A, diag (rep(1, n)))
                        ↳ In

  for (k in 1:m)
  {
     a <- B [k,k]
     for (j in 1:(n*2)) B[k,j] <- B[k,j]/a

     for (i in 1:n)
        if (i != k)
        {
           a <- B [i, k]
           for (j in 1: (n*2))
           B [i,j] <- B[i,j] - b*B[k,j]
        }
  }
     return( B)
  }
```

```
my Guss jorelon <- funct(A, m)
n<
B<

for (k in 1:m)
 {
   B [k, ] <- B[k, ]/B[k, k]

for (i in 1:n)
   if (i != k)
   B[i, ] <- B[i, ] - B[k, ]*B[i, k]
```

Python:

```
def myGaussJordon (A, m):

        n = A shape [0]
        B = np. stack(( A  np. identity (n)))
        for K in range (m)
            a = [B, K]

        for j in range (n*2):
            B [K, j] = B [K, j]/a

        for i  in range (n):
            if i ! = K:
              b = B [I, K],
              for j in range (n*2):
                  B [i, j] = B [i, j] - B[K, j] * b;
      return B.
```

B[I, :] = B[I, :]
- B[K, :] * B[, K]

```
n = 100
p = 5

X = matrix (rnorm (n*p), n_row = n)

beta = matrix( 1:p, n_row = p)
Y = X %*% beta + rnorm (n)
lm (Y ~ X)
                ???
Z = cbind (rep(1,n), X, Y)
A = t(Z) %*% Z

S = myGaussJordon (A, p+1)
beta =
RSS =        V =
```

Lecture 4
10/10/2017

Guass-Jordan

$$\left[A_{n\times p} \mid I \right] \xrightarrow[A^{-1}]{GJ[1\ldots n]} \left[I \mid A^{-1} \right] \qquad A = \begin{pmatrix} A_{11\,m\times m} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}_{n\times n}$$

$$\left(\begin{array}{cc|cc} A_{11} & A_{12} & I_1 & 0 \\ A_{21} & A_{22} & 0 & I_2 \end{array}\right) \longrightarrow \left(\begin{array}{c|c|c|c} I & A_{11}^{-1}A_{12} & A_{11}^{-1} & 0 \\ 0 & A_{22}-A_{21}A_{11}^{-1}A_{12} & -A_{21}A_{11}^{-1} & I_2 \end{array}\right)$$

$$\begin{pmatrix} A_{11}^{-1} & 0 \\ -A_{21}A_{11}^{-1} & I \end{pmatrix} A = \begin{pmatrix} I & A_{11}^{-1}A_{12} \\ 0 & A_{22}-A_{21}A_{11}^{-1}A_{12} \end{pmatrix}$$

$\downarrow$

$GJ[A_{11}]$

1
$a_{21} - 1\cdot a_{21}$
$a_{31} - 1\cdot a_{31}$

Sweep Operator

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \xrightarrow{\text{swp}[1]} \begin{bmatrix} 1 & \frac{a_{12}}{a_{11}} & \cdots & \frac{a_{1n}}{a_{11}} & \frac{1}{a_{11}} & & \\ 0 & & & & \vdots & 1 & \\ \vdots & & a_{ij} - \frac{a_{i1}-a_{1j}}{a_{11}} & & -\frac{a_{i1}}{a_{11}} & & 1 \\ 0 & & & & & & 1 \end{bmatrix}$$

+ "−"

$$\xrightarrow{\text{swp}[K]} \begin{bmatrix} \tilde{a}_{ij} \end{bmatrix} \qquad \begin{aligned} \tilde{a}_{KK} &= -\frac{1}{a_{KK}} \\ \tilde{a}_{Kj} &= \frac{a_{Kj}}{a_{KK}} & j \neq K \\ \tilde{a}_{jK} &= \frac{a_{jK}}{a_{KK}} & i \neq K \\ \tilde{a}_{ij} &= a_{ij} - \frac{a_{jK}a_{Kj}}{a_{KK}} & i, j \neq K \end{aligned}$$

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \xrightarrow[\text{swp}[A_1]]{\text{swp}[1\ldots m]} \begin{pmatrix} -A_{11}^{-1} & A_{11}^{-1}A_{12} \\ A_{11}A_{11}^{-1} & A_{22}-A_{21}A_{11}^{-1}A_{12} \end{pmatrix}$$

$$\begin{pmatrix} X^TX & X^TY \\ Y^TX & Y^TY \end{pmatrix} \xrightarrow{swp[X^TX]} \begin{pmatrix} -(X^TX)^{-1} & (X^TX)^{-1}X^TY \\ Y^TX(X^TX)^{-1} & Y^TY - Y^TX(X^TX)^{-1}XY \end{pmatrix}$$

$$= \begin{pmatrix} -\frac{Var(\hat{\beta})}{\sigma} & \hat{\beta} \\ \hat{\beta}^T & RSS \end{pmatrix}$$

```
mysweep <- function(A, m)
{
    n <- dim(A)[1]
    for (k in 1:m)
    {
        for ( i in 1:n)
        for ( j in 1:m) -> (indent)
                if ( i != k, & j != k)
                    A[i,j] <- A[ij] - A[i,k] * A[k,j]/A[k,k]

        for ( i in 1:n)
            if ( i != k)
              A[i,k] <- A[i,k]/A[k,k]

        for ( j in 1:n)
            if (j != k)
              A[k,j] <- A[k,j]/ A[k,k]

        A[k,k] <- -1/ A[k,k]
}
    return(A)
```

Gussian density:

$$f(x) = \frac{1}{(2\pi)^{\frac{p}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right]$$

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \xrightarrow{\underline{swp}} [x_2 | x_1] \sim N(\beta x, \tilde{\Sigma}) \sim N(\Sigma_{22}\Sigma_{11}^{-1}x, \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12})$$

10/12/2017

Lecture 5

```
# include <Rcpp.h>
using namespace Rcpp;

// [[ Rcpp::export]]        have to include before every functions.
double linearFunction 1 (double , x) {

        // set aside memory for output y
        double y = 0.0 ;

        // compute output
        y = x + 3 ;

        // Return output

        return y
}

// [[ Rcpp:: export]]
int linearFunction 2 ( int, x) {

      // print statement in Rcpp
      Rcout << "x is equal to" << x <<
```

Looping in Rcpp

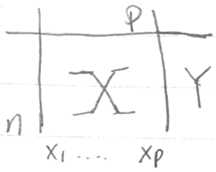
```
// [[R...  ]]
int loopexample (int n) {
      int x = 0
      for
```

```
// [[Rcpp.. ]]
double rbetaCCP( Function f, double x1 , double x2) {

        double output = 0.0

        // Create a list to store R function output.
        List temporary;

        // Call R function and store results into our list
        temporary = f(1, x1 , x2);

        // Access the 0th element of that list, aka our
        output = as <double> ( temporary [0]);

        return output;
}
```

Vector and Matrix

```
// [[ ... ]]
Numericvector vectorManiputation1( Numericvector x,

     double sumy = sum(y);
```

C++		python
A=clone (B)	~	A.copy (B)



$$Y = X\beta + \varepsilon$$

$$\hat{\beta} = \arg\min \| Y - X\beta \|^2$$

QR Decomposition

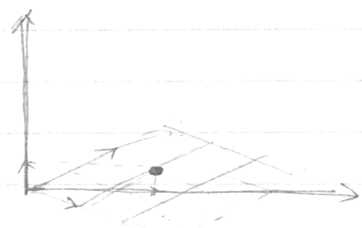
$$\underset{n\times p}{(X} \ \underset{n\times 1}{Y)} \xrightarrow{Q} \underset{n\times p}{(R,} \ \underset{n\times 1}{Y^*)} = \left(\begin{pmatrix} R_1^* \\ 0 \end{pmatrix}_{n\times p} \begin{pmatrix} y_1^* \\ y_2^* \end{pmatrix}_{n\times 1} \right)$$

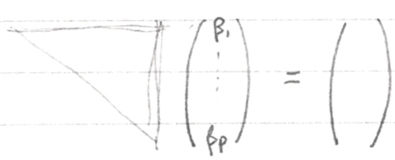
$$\hat{\beta} = \arg\min \| Y - X\beta \|^2$$

$$= \| Y^* - R\beta \|^2$$

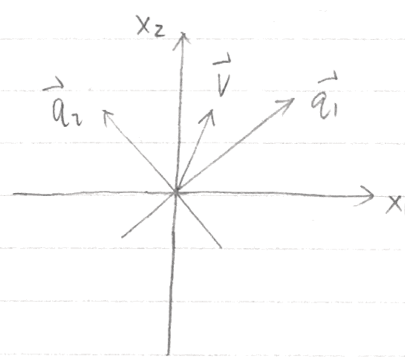
$$= \underbrace{\| Y_1^* - R_1^*\beta \|^2}_{\downarrow \ = 0} + \| Y_2^* \|^2$$

$$\hat{\beta} = R_1^{*-1} Y_1^*, \qquad RSS = \| Y_2^* \|^2$$





$$R_1^* \qquad \beta \quad = \quad Y_1^*$$



$$Q_{n\times m} = (q_1 \ q_2 \ \dots \ q_n)$$

$$\vec{V} = u_1 \cdot \vec{q}_1 + u_2 \cdot \vec{q}_2 + \dots + u_n \cdot \vec{q}_n$$

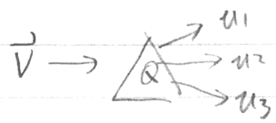
$$= (\vec{q}_1 \dots \vec{q}_n) \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = Q^T u$$

$$\begin{aligned} u_1 &= \langle \vec{V}, \vec{q}_1 \rangle = \vec{q}_1^{\,T} \cdot \vec{V} \\ u_2 &= \langle \vec{V}, \vec{q}_2 \rangle = \vec{q}_2 \cdot \vec{V} \\ &\ \vdots \\ u_n &= \langle \vec{V}, \vec{q}_n \rangle = \vec{q}_n \cdot \vec{V} \end{aligned} = \begin{pmatrix} q_1 \\ \vdots \\ q_n \end{pmatrix} \vec{V} = Q^T \vec{V}$$

$\vec{u} = Q^T \vec{v}$ analysis

$\vec{v} = Q \vec{u}$ synthesis

$QQ^T = Q^T Q = I$

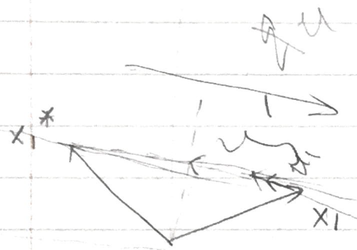


$X \longrightarrow X^*$

$(x_1, \dots, x_p) \longrightarrow (x_1^*, \dots, x_p^*)$

$$= \begin{pmatrix} x_1^* = \pm|x_1| & & \\ 0 & x_2^* & \\ \vdots & & \ddots \quad x_p^* \\ 0 & & \end{pmatrix}$$

$V = QR$



Householder reflection

$$\vec{u} = \frac{(x_1 - x_1^*)}{|x_1 - x_1^*|}$$

$$x_1^* = x_1 - 2\langle \vec{x}_1, \vec{u} \rangle \cdot \vec{u}$$

$$x_j^* = x_j - 2\langle \vec{x}_j, \vec{u} \rangle \cdot \vec{u} \qquad j = 1, \dots, p$$

$$= (x_j - 2 \cdot \vec{u}\,\vec{u}^T x_j)$$

$$= H_1 x_j$$

$$H_1 = I - 2\vec{u}\vec{u}^T$$

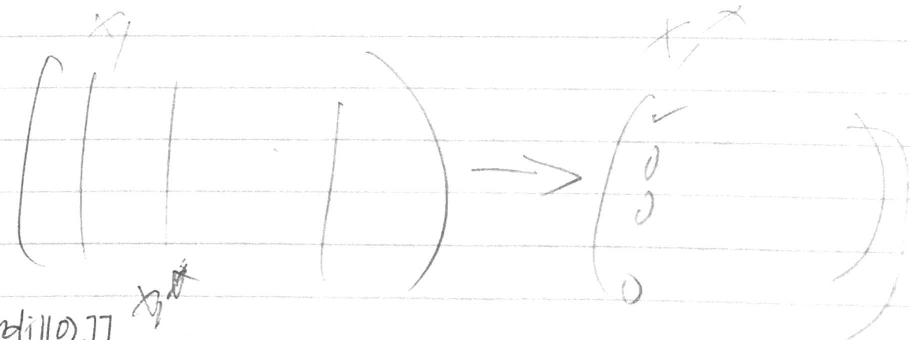
$$u = \frac{x_1 - x_1^*}{|x_1 - x_1^*|}$$

q_1

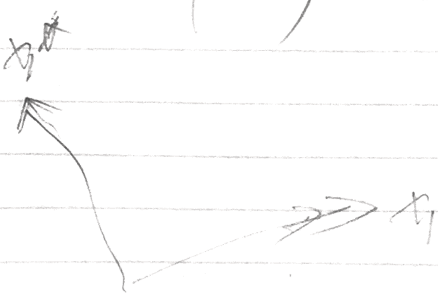
$\vec{x}_j \cdot \vec{u}$

$\Uparrow$

$\langle \vec{x}_j, \vec{u} \rangle$



```
// [[Rcpp::depends(RcppAramdillo)]]
# include <RcppAramdillo.h>
using namespace aram;
using namespace Rcpp;
```



10/17/2017

Lecture 6.

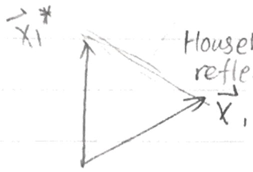
Review:

QR decomposition $\quad [X \mid Y] \xrightarrow{Q} [R \mid Y]$

$$\left[\begin{array}{c|c} R & Y_1^* \\ 0 & Y_2^* \end{array}\right] \qquad \hat{\beta} = R^{-1} Y_1^* \qquad RSS = |Y^*|^2$$

$$\begin{pmatrix} x_{11} & x_{12} & & x_{1p} \\ x_{21} & \boxed{x_{22}} & & x_{2p} \\ \vdots & & & \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \longrightarrow \begin{pmatrix} x_1^* & \\ 0 & \\ \vdots & \\ 0 & \end{pmatrix}$$

$\vec{x}_1 \qquad \vec{x}_2^* \qquad \vec{x}_1^*$



Householder reflection $\quad \vec{u} = \dfrac{\vec{x}_1 - \vec{x}_1^*}{|\vec{x}_1 - \vec{x}_1^*|} \to H_1 \qquad x_{11}^* = -\text{sign}(x_{11})\,|x_1|$

$$\begin{aligned} \vec{x}_1^* &= \vec{x}_1 - 2\langle \vec{x}, \vec{u}\rangle \vec{u} \\ &= \vec{x}_1 - 2 \cdot \vec{u} \cdot \vec{u}^T \cdot \vec{x}_1 \\ &= \underbrace{(I - 2\vec{u}\cdot\vec{u}^T)}_{H} \vec{x}_1 \end{aligned} \qquad \begin{aligned} \vec{x}_j^* &= \vec{x}_j - 2\langle \vec{x}_j, \vec{u}\rangle \cdot \vec{u} \\ &= H_1 \vec{x}_j \end{aligned}$$

$$X \xrightarrow[H_n \cdots H_2 H_1]{Q^T} R \qquad (X, I) \longrightarrow (R, Q^T)$$

$X = Q \cdot R$

$R = Q^T X$

$$\vec{v} = Q\vec{u} = (q_1\ q_2\ \ldots\ q_n)\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = u_1 \vec{q}_1 + \cdots + u_n \vec{q}_n$$

Eigen-decomposition.

$$\Sigma = Q \Lambda Q^T$$

orthogonal matrix (Q) $\quad$ diagonal matrix (Λ)

$\vec{u} = Q^T \vec{v} \qquad u_i = \langle \vec{v}_i, \vec{q}_i \rangle$

$$\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

ordering $|\lambda_1| > |\lambda_2| \geq \cdots > |\lambda_n|$

$\vec{V} = Q\vec{u}$ $\qquad\qquad \Sigma = Q\Lambda Q^T$

$u = Q^T \vec{V}$

$$\vec{w} = \Sigma \vec{V} = Q\Lambda Q^T Q \vec{u} = Q\underbrace{\boxed{\Lambda \vec{u}}}_{\vec{z}}$$

$$\vec{w} = \Sigma \vec{v}$$

$$\downarrow \qquad\qquad \downarrow$$

$$Q\vec{z} \qquad\qquad Q\vec{u}$$

$$Q\Lambda \cancel{Q^T Q}\vec{u}$$

$$\vec{z} = \Lambda \vec{u}$$

$$= \begin{pmatrix} \lambda_1 u_1 \\ \lambda_2 u_2 \\ \vdots \\ \lambda_n u_n \end{pmatrix}$$

$$\vec{w} = \Sigma^t \vec{v}$$

$$\downarrow \qquad\qquad \downarrow$$

$$\vec{z} = \Lambda^t \vec{u} = \begin{pmatrix} \lambda_1^t u_1 \\ \lambda_2^t u_2 \\ \vdots \\ \lambda_n^t u_n \end{pmatrix}$$

$$\Lambda^t = \begin{pmatrix} \lambda_1^t & & \\ & \ddots & \\ & & \lambda_m^t \end{pmatrix}$$

dominant.

Power method

Start from v $\quad \tilde{v} = \frac{\vec{v}}{|\vec{v}|} \quad = v = \Sigma \tilde{v}$

Start from V — $n \times n$ $\qquad \tilde{V} = \underline{\text{noralize}(V)}$ $\qquad V = Q \cdot R$

$V = \Sigma \tilde{V}$

$$\begin{pmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_n \\ | & | & & | \end{pmatrix} = \begin{pmatrix} q_1 & \cdots & q_n \end{pmatrix} \cdot \begin{pmatrix} R \end{pmatrix}$$

10/19/2017

Lecture 7.

QR Decomposition

$[X\ Y] \longrightarrow [R\ Y^*] \longrightarrow$

```
Import numpy as np
from scipy input linalg

def qr(A)

    n, m = A.shap
    R = A.copy()
    Q = np.eye(n)

    for k in range(m-1):
        x = np.zeor((n,1))
        x[k:, 0] = R[k:, k]
        v = x
        v[k] = x[k] + np.sign(x[k,0]) * np.linalg.norm(x)

        s = np.linalg.norm(v)
        u = v/s
        R -= 2* np.dot(u, np.dot(u.T, R))
        Q -= 2* np.dot(u, np.dot(u.T, Q)).

    Q = Q.T

    return Q, R.
```

```
n = 100
p = 5
X = np.random.random_sample((n,p))
beta = np.array(range(1, p+1))
Y = np.dot(X, beta) + np.random.standard_norm(n)
Z = np.hstack((np.ones(n).reshape((n,1)), X, Y.reshape((n,1))))
```

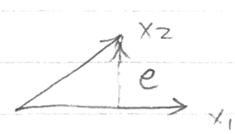
done on R

```
_, R = qr(Z)
R1 = R[:p+1, p+1]
Y1 = R[:p+1, p+1]

beta = np.linalg.solve(R1, Y1)

print beta
```

$X_{n\times n} = QR$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} = (q_1 \; q_2 \; \cdots \; q_n) \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & & \vdots \\ 0 & 0 & \ddots & \\ & 0 & & r_{nn} \end{pmatrix}$$

Gram-Schmite



$x_1 = r_{11} q_1$

$x_2 = r_{12} q_1 + r_{22} q_2$

$q_1 = \frac{x_1}{|x_1|} \qquad r_{11} = |x_1|$

$e = x_2 - \underbrace{\langle x_2, q_1 \rangle}_{r_{12}} q_1$

$q_2 = \frac{e}{|e|}$

$r_{22} = |e|$

Powell Method

$\Sigma = Q \Lambda Q^T$

Power method Python code:

```
def eigen-qr(A)

T=1000
A_copy = A.copy()
r,c = A_copy.shape

V = np.random.random_sample((r,r))

for i in range(T):
        Q,R = qr(V)
        V = np.dot(A_copy, Q)

return R.diagonal(), Q
```

matrix calculation

packaging $Q = (q_1, q_2, \quad q_n)$

$$Qu = (q_1, q_2, \dots, q_n)\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \sum q_i u_i$$

Representation / view point

$V = Q \cdot u$

$u = Q^T v$

$\Sigma = Q \Lambda Q^T$

$\Sigma Q = Q \Lambda$

$$\begin{aligned} w &= \Sigma v \\ &= \Sigma Q u \\ &= Q \Lambda Q^T Q u \\ &= Q \Lambda u \\ &= Q z \end{aligned}$$

Σ

$$\begin{tikzcd} v \arrow[r, "\Sigma"] \arrow[d, "Q"] & w \arrow[d, "Q"] \\ u \arrow[r, "\Lambda"] & z \end{tikzcd}$$

$$\begin{tikzcd} v \arrow[r, "\Sigma^t"] \arrow[d, "Q"] & w \arrow[d, "Q"] \\ u \arrow[r, "\Lambda^t"] & z \end{tikzcd} = \begin{pmatrix} \lambda_1^t u_1 \\ \vdots \\ \lambda_n^t u_n \end{pmatrix} \rightarrow$$

get q_1

$$\begin{tikzcd} V \arrow[r, "\Sigma^t"] \arrow[d, "Q"] & w \arrow[d, "Q"] & & q_1 \arrow[d, "Q"] \\ u \arrow[r, "\lambda^t"] & z = \begin{pmatrix} \lambda_1^t u_1 \\ \vdots \\ \lambda_n^t u_n \end{pmatrix} \arrow[rr, "\text{normlize}"] & & \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \end{tikzcd}$$

Could randomly choose a vector V which perpentiular to q_1

get q_2

$$\begin{tikzcd} V \perp q_1 \arrow[r, "\Sigma^t"] \arrow[d, "Q"] & w \arrow[d] & q_2 \\ u = \begin{pmatrix} 0 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \arrow[r, "\lambda^t"] & z = \begin{pmatrix} 0 \\ \lambda_2^t u_2 \\ \vdots \\ \lambda_n^t u_n \end{pmatrix} \arrow[r, "\text{normalize}"] & \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \arrow[u] \end{tikzcd}$$

get q_3

$$\begin{tikzcd} V \perp q_1 \perp q_2 \arrow[r] \arrow[d] & q_3 \\ u = \begin{pmatrix} 0 \\ 0 \\ u_3 \\ \vdots \\ u_n \end{pmatrix} \arrow[r, "\lambda^t"] & \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \arrow[u] \end{tikzcd}$$

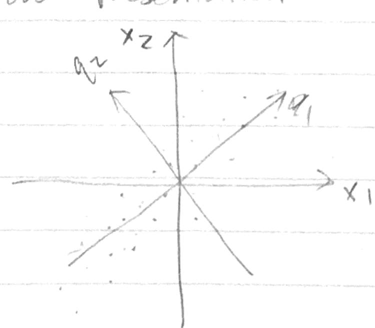
$$\Sigma Q = Q\Lambda Q^T Q = \underline{\underline{Q\Lambda}}$$

QR

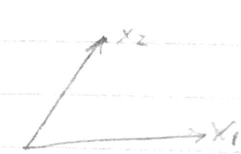
ex:

	1	2	Y	Z_1	Z_2
1	X_{11}	X_{12}		Z_{11}	Z_{12}
2	X_{21}	X_{22}		Z_{21}	Z_{22}
$\vdots$	$\vdots$	$\vdots$			
n	X_{n1}	X_{n2}			

row presentation



column reprentation



10/24/2017

Lecture 8

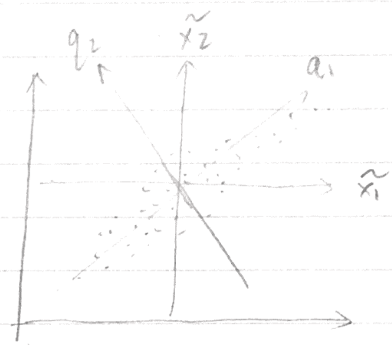
Principle component analysis

	1 2 j ... p	Y
1		
2		
⋮		
i		
⋮		
n		

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix}_{p\times 1} \sim f(x).$$

centerlize

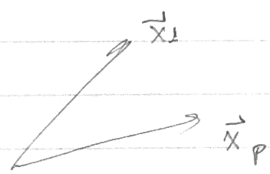
1	2		
		$x_{11}-\bar{x}_1$	$x_{12}-\bar{x}_2$
		⋮	
		$x_{i1}-\bar{x}_1$	$x_{i2}-\bar{x}_2$

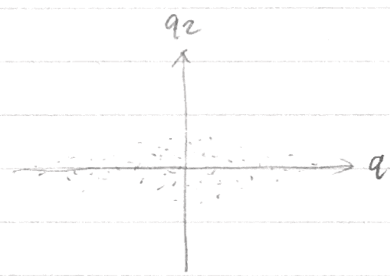


p-dim

Column picture

n-dim





$$\tilde{x}_i = z_{i1}q_1 + z_{i2}q_2$$
$$\approx z_{i1}q_1$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad A\bar{x} = \frac{1}{n}\sum_{i=1}^{n} Ax_i$$

$$\downarrow \qquad\qquad \downarrow$$

$$E[X] \qquad AE[X] = E[AX]$$

$$S = \frac{1}{n}\sum_{i=1}^{n}\underbrace{\underset{p\times 1}{(x_i-\bar{x})}\,\underset{1\times p}{(x_i-\bar{x})^T}}_{p\times p} \qquad \frac{1}{n}\sum_{i=1}^{n}(Ax_i - A\bar{x})(Ax_i - A\bar{x})^T = \frac{1}{n}ASA^T$$

$$\downarrow$$

$$\Sigma = E[(x-\mu)(x-\mu)^T] \qquad Var(AX) = A\,Var(x)\,A^T$$

$$\Sigma = Var(x) = Q\Lambda Q^T$$
$$Var(Q^Tx) = Q^T\Sigma Q = \Lambda$$

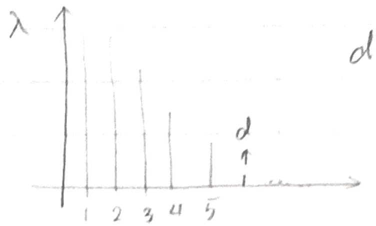
$$\begin{cases} Z = Q^T X \\ X = QZ \end{cases}$$

$$\begin{array}{ccc} X & \sim & (\mu, \Sigma) \\ Q \downarrow & & \downarrow \\ Z & \sim & (\quad \Lambda) \end{array}$$

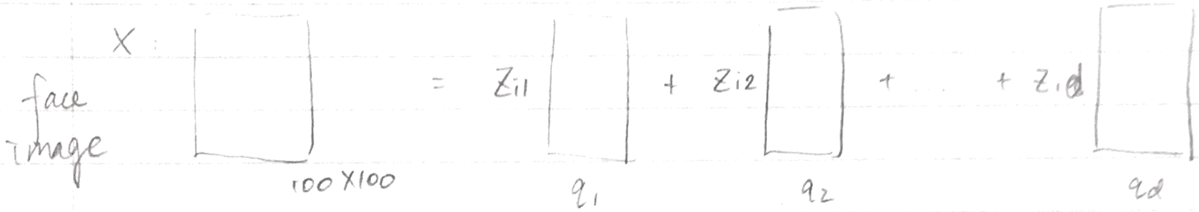
principle component.

$$X = z_1 q_1 + z_1 q_2 + \dots + z_p q_p \approx z_1 q_1 + \dots + z_d q_d$$

$$E[X] = E[Z] = 0$$



dimension reduction.



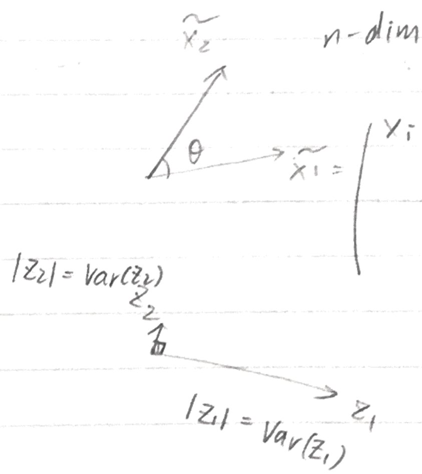
PCA : (1) Σ for $S = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T$

(2) $\Sigma = Q \Lambda Q^T$

(3) Cut off

	1	2	1	2
1				
2				
⋮				
i	$\tilde{x}_{i1}$	$\tilde{x}_{i2}$	z_{i1}	z_{i2}
⋮				
n				
	$\tilde{x}_1$	$\tilde{x}_2$	z_1	z_2

$$\tilde{X} Q = Z$$



n-dim

$\frac{1}{n} |\tilde{x}|^2 = \frac{1}{n} \sum_{i=1}^{n} (x - \bar{x})^2$

$\rightarrow var(x_i)$

$\frac{1}{n} |\tilde{x}_i|^2 \longrightarrow Var(\quad)$

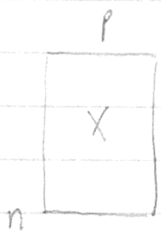
$\frac{1}{n} \langle \tilde{x}_1, \tilde{x}_2 \rangle = \frac{1}{n}$

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \qquad Var(X) = E[(X-\mu)(X-\mu)^T]$$

$$= E\left(\left[\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}\right]\left[\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}\right]^T \right)$$

$$= E\left(\begin{pmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{pmatrix} \begin{pmatrix} x_1-\mu_1 & x_2-\mu_2 \end{pmatrix} \right) = E\begin{bmatrix} (x_1-\mu_1)^2 & (x_1-\mu_1)(x_2-\mu_2) \\ \cdot & (x_2-\mu_2)^2 \end{bmatrix}$$

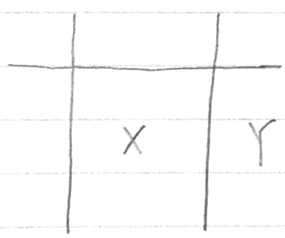
$$= \begin{pmatrix} Var(x_1) & Cov(x_1, x_2) \\ Cov(x_1, x_2) & Var(x_2) \end{pmatrix}$$



How to get $\bar{x}$? (vector)

$$\frac{1}{n}\underset{n\times 1}{(1, \ldots, 1)} \; \underset{n}{X}^{\,p} = \bar{X} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_p \end{pmatrix}$$

Multivariate Calculus



$$R(\beta) = |Y - X\beta|^2$$

$$\underset{p\times 1}{\dot{R}(\beta)} = -2\underset{p\times 1}{X^T}(Y - X\beta)$$

$$R(\beta) = \sum_{i=1}^{n}(y_i - x_i\beta)^2 \quad ③$$

$$= \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2$$

$$R'(\beta) = -2\sum_{i=1}^{n}(y_i - x_i^T\beta)\,x_i \quad ②$$

$$\left(R'(\beta)\right)_{j=1\ldots p} = -2\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)\left(x_{ij}\right)_{j=1,\ldots,p} \quad ①$$

2^{nd} derivative

$$R(\beta) = |\overset{e}{\overset{\shortparallel}{Y - X\beta}}|^2$$

$$\frac{\partial R}{\partial \beta^T} = \frac{\partial R}{\partial e^T} \cdot \frac{\partial e}{\partial \beta^T} = -2e^T X$$

$$R'(\beta) = \frac{\partial R}{\partial \beta} = -2X^T(Y - X\beta)$$

$$y_{1\times 1} = f(x_{n\times 1})$$

$$\frac{\partial^2 y}{\partial x \partial x^T} = \left(\frac{\partial^2 y}{\partial x_i \partial x_j} \right)_{n\times n} = \frac{\partial z}{\partial x^T}_{\ n\times n}$$

$$R''(\beta) = \frac{\partial z}{\partial \beta^T} = 2X^T X \geq 0$$

$A \geq 0$ positive definite

$$a^T A a \geq 0$$

$$\Sigma = Var(x) \geq 0$$

$$a^T \Sigma a = Var(a^T x) \geq 0 \qquad J \geq 0$$

10/26/2017
Lecture 9.

Orthogonal matrix Q new viewpoint $(q_1, q_2, \dots q_n)$

$$\begin{array}{ccc} u & \xrightarrow{\Sigma} & \Sigma u \\ Q \updownarrow & & \updownarrow Q \\ v & \xrightarrow{\Lambda} & \Lambda v \end{array}$$

$$\begin{array}{l} x_{p\times1} \sim N(0,\Sigma). \\ Q \downarrow \\ z_{p\times 1} \sim N(0,\Lambda) \end{array}$$

$\begin{cases} u = Qv \\ v = Q^T u \end{cases}$ $\Sigma = Q\Lambda Q^T$ $\begin{cases} x = Qz \\ z = Q^T x \end{cases}$ $\Sigma = E[(x-\mu)(x-\mu)^T] = Q\Lambda Q^T$

$|u| = 1$ max λ_1 $u^* = q_1$

$$\mathrm{Var}(\underset{p\times1}{u^T}\ \underset{p\times1}{x}) = u^T \Sigma u.$$

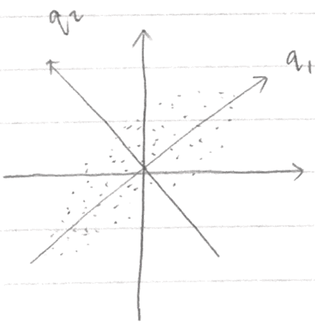
positive-definite.

$$\mathrm{Var}(v^T z) = v^T \Lambda v$$

$$= \sum_{j=1}^{p} \lambda_j \cdot v_j^2$$

$\downarrow$ max over v

$$\lambda_1, \quad v^* = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

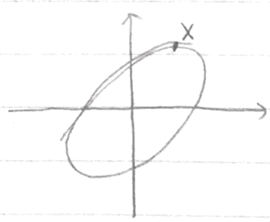


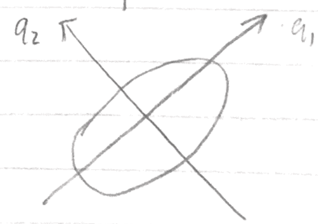
Hessian matrix

$$x^T \Sigma^{-1} x = \text{const}$$

$\downarrow u$

$$z^T \Lambda^{-1} z = \sum_{j=1}^{p} \frac{z_j^2}{\lambda_j} = \text{const}.$$





	1	2		p	
1					
2					
⋮					
i	x_{i1}	x_{i2}		x_{ip}	y_i
n					

$y_0 \in \{0,1\}$

understanding

classification

Linear Regression

$$\begin{cases} y_i \sim N(\mu_i, \sigma^2) \\ \mu_i = x_i^T\beta \end{cases}$$

Ligistive regression

$y_i \sim$ Bernalli (p_i)

$$p_i = \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}} \qquad P(y_i = 1 \mid x_i) = p_i$$

score $x_i^T\beta \to \infty$ $p_i = 1$

$\to 0$ $p_i = \frac{1}{2}$

$\to -\infty$ $p_i = 0$

$(\beta^{(1)} = \beta, \ \beta^{(2)} = 0)$

$y_i \in \{1, 2, \dots, K\}$

$$P(y_i = k) = p_{ik} \qquad \sum_{k=1}^{K} p_{ik} = 1. \qquad p_{ik} = \frac{e^{x_i^T\beta^{(t)}}}{\sum_{k=1}^{K} e^{x_i^T\beta^{(k)}}}$$

max liklihood — most plausible expectation.

$$L(\beta) = \prod_{i=1}^{n} P(y_i \mid x_i\beta) = \prod_{i=1}^{n} \frac{e^{y_i x^T\beta}}{1+e^{x^T\beta}}$$

$$P(y_i \mid x_i, \beta) = \left\{ \begin{array}{ll} p_i = \dfrac{e^{x_i^T\beta}}{1+e^{x_i\beta}} & y_i = 1 \\ 1 - p_i = \dfrac{1}{1+e^{x_i^T\beta}} & y_i = 0 \end{array} \right\} = \frac{e^{y_i x_i^T\beta}}{1+e^{x_i^T\beta}}$$

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

$$\left(\text{Recall } Y = AX \quad \frac{\partial y}{\partial x} = A \right)$$

$$\ell'(\beta) = \sum_{i=1}^{n} \left[x_i \cdot y_i - \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}} \cdot x_i \right]$$

$$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^{n} x_i (y_i - p_i)$$

$$= p_i(1-p_i) \quad \le 0$$

$$\frac{\partial l}{\partial\beta\cdot\partial\beta^T} = -\sum_{i=1}^{n} x_i x_i^T \boxed{\frac{e^{x_i^T\beta}}{(1+e^{x_i^T\beta})^2}}$$

$$* \quad \frac{\partial}{\partial\beta^T} \left(1-\frac{1}{1+e^{x_i^T\beta}}\right) = \frac{e^{x_i\beta}}{(1+e^{x_i^T\beta})^2} x_i^T$$

⊖ concave

obtain one max point.

---

talyor expasion.

$$l(\beta) = l(\beta_t) + \langle l'(\beta), \beta-\beta_t \rangle + \frac{1}{2}(\beta-\beta_t)^T \frac{\partial^2 l}{\partial\beta\,\partial\beta^T}\Big|_{\beta_t}(\beta-\beta_t)$$

$\tilde{l}(\beta)$

↓

max

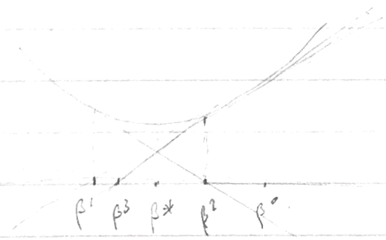
$$\frac{\partial \tilde{l}}{\partial\beta} = l'(\beta_t) + l''(\beta_t)(\beta-\beta_t) = 0 \quad \Rightarrow \quad \beta_{t+1} = \beta_t - l''(\beta_t)^{-1} l'(\beta_t)$$

$$l''(\beta_t)(\beta-\beta_t) = -l'(\beta_t)$$

Newtone - Method

$\boxed{\text{Newton - Raphson}}$

$$\beta-\beta_t = -[l''(\beta_t)]^{-1} l'(\beta_t)$$



10 - iterate.

more detailed:

$$\beta_{t+1} = \beta_t + \sum_{i=1}^{n} p_i(1-p_i) x_i x_i^T \sum_{i=1}^{n} x_i(y_i - p_i)$$

let $w_i = p_i(1-p_i)$

$$= \left(\sum_{i=1}^{n} w_i x_i x_i^T\right)^{-1} \cdot \sum_{i=1}^{n} \left(w_i x_i \underbrace{\left(x_i^T\beta_t + \frac{y_i - p_i}{w_i}\right)}_{z_i}\right)$$

$$= \left(\sum_{i=1}^{n} \tilde{x}_i \tilde{x}_i^T\right)^{-1} \left(\sum_{i=1}^{n} \tilde{x}_i \tilde{y}_i\right)$$

$\tilde{x}_i = \sqrt{w_i}\, x_i$

$\tilde{y}_i = \sqrt{w_i}\, z_i$

Iterate reweighted least squares

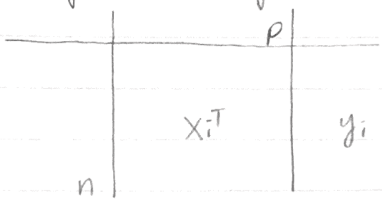
$\beta_t \rightarrow y = x^T\beta_t \rightarrow p_i = \frac{1}{1+e^{y_i}} \rightarrow w_i = p_i(1-p_i)$

$\beta_{t+1} \leftarrow$ linear Regression (QR) $\leftarrow \tilde{x} = \sqrt{n}\, x_i$, $\tilde{y} = \sqrt{n}\, z_i \leftarrow z_i = y_i + \frac{y_i - p_i}{w_i}$

10/31/2017

Lecture 10

Logistic Regression



generalized linear model (GLM)

$$\begin{cases} y_i \sim p(y_i \mid \theta_i) \\ \boxed{\theta_i = f(x_i^T\beta)} \end{cases}$$

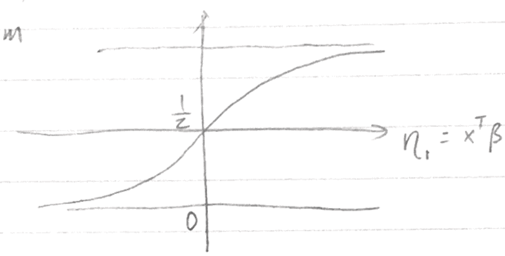
θ_i ↓ one dim, $x_i^T\beta$ ↓ η_i score

$\theta_i = F(x_i)$ ↓ p dim

$y_i \sim \text{Bernolli}(p_i)$

$$p_i = \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}} = \text{sigmoid}(x^T\beta)$$

$$= \frac{1}{1+e^{-x^T\beta}}$$

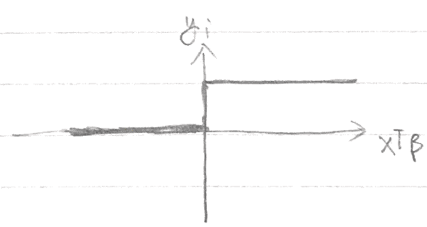


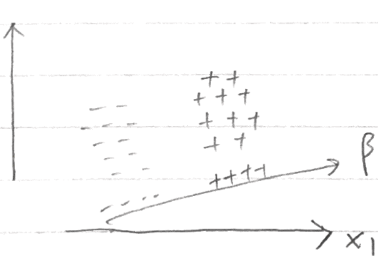
$$\text{logit}(p_i) = \log\frac{p_i}{1-p_i} \Leftrightarrow y_i = x_i^T\beta$$

perceptron.

$$y_i = \text{sign}(x_i^T\beta) = \begin{cases} 1 & x^T\beta \geq 0 \\ 0 & \text{o.w.} \end{cases}$$

or $y_i = \begin{cases} +1 \\ -1 \end{cases}$.





$\beta \propto x_+ - x_-$

what if $y = \begin{cases} +1 \\ -1 \end{cases}$

$$p(y_i = 1 \mid x_i, \beta) = \frac{1}{1+e^{-x_i^T\beta}} = p(y_i \mid x_i, \beta) = \frac{1}{1+e^{-y_i x_i^T\beta}}$$

$$p(y_i = -1 \mid x_i, \beta) = \frac{1}{1+e^{x_i^T\beta}}$$

maximum Likelihood.

$$L(\beta) = \text{Prob}(Y \mid X, \beta)$$

$$= \prod_{i=1}^{n} P(y_i \mid x_i, \beta)$$

$$= \prod_{i=1}^{n} p_i^{y_i} (1-p_i)^{1-y_i}$$

$y_i = 1,\ p_i$

$y_i = 0,\ 1-p_i$

$$= \prod_{i=1}^{n} \left(\frac{e^{x^T\beta}}{1+e^{x^T\beta}}\right)^{y_i} \left(\frac{1}{1+e^{x^T\beta}}\right)^{1-y_i}$$

$$= \prod_{i=1}^{n} \frac{e^{x^T\beta}}{1+e^{x^T\beta}} \longrightarrow P(y_i \mid x_i \beta)$$

what if $y = \begin{cases} +1 \\ -1 \end{cases}$

$$P(y_i = 1 \mid x_i, \beta) = \frac{1}{1+e^{-x^T\beta}}$$

$$P(y_i = -1 \mid x_i, \beta) = \frac{1}{1+e^{x^T\beta}}$$

$$P(y_i \mid x, \beta) = \frac{1}{1+e^{-y_i x^T\beta}}$$

$$l(\beta) = \text{Log } L(\beta) = \sum_{i=1}^{n} [\, y_i X^T\beta - \log(1+e^{x_i^T\beta})\,]$$

$$l'(\beta) = \sum_{i=1}^{n} y_i x_i - \underbrace{\boxed{\frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}}}_{p_i} \cdot x_i = \sum_{i=1}^{n} x_i (y_i - p_i).$$

Learning rule:

$$\beta_{t+1} = \beta_t + \gamma_t\, l'(\beta_t)$$

$$= \beta_t + \gamma_t \sum_{i=1}^{n} x_i (\underbrace{y_i - p_i}_{\text{error}}) \longrightarrow \text{Version 1}$$

β_t.

$y_i = \begin{cases} +1 \\ -1 \end{cases}$

$$\frac{\partial}{\partial \beta} \log(1+e^{-y_0 x_i^T\beta})$$

$$= -\frac{e^{-y_i x^T\beta}}{1+e^{-y_i x^T\beta}}\, y_i x_i = -\frac{1}{1+e^{y_i x^T\beta}}\, y_i x_i \longrightarrow \text{Version 2}$$

Stochastic algorithom:

randomly select i $\qquad \beta_{t+1} = \beta_0 + r_i \, x_i (y_i - p_i)$

learn from error $\qquad r_i$ decrease.

re allignment:

$x_i \quad y_i = + \quad p_i \approx 0$

$\beta_{t+1} = \beta_t + r_t \cdot x_i$

$y_i = - \quad p_i \approx 1$

$\beta_{t+1} = \beta_t - r_t x_i$

Perceptron algorithm:

$\beta_0 = 0,$ randomly select i

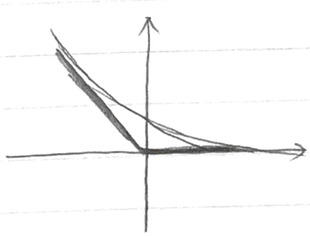
$$\beta_{i+1} = \beta_i + \begin{cases} x_i & \text{if } y_i = 1 \quad x_i^T\beta < 0 \\ -x_i & \quad\;\; = -1 \quad x_i^T\beta > 0 \end{cases}$$

$$= \beta_0 + 1(y_i \neq \text{sign}(x^T\beta_t))\, y_i x_i \rightarrow \text{Version 3}$$

Loss function $l(\beta) = \sum_{i=1}^{n} \max(0, -y_i x_i^T\beta)$

$\frac{\partial}{\partial \beta} \log(1 + e^{-y_i x_i^T \beta})$ $\quad y_i x_i^T\beta \to \infty \quad 0$

$y_i x_i^T\beta \to -\infty \quad -y_i x_i^T\beta$



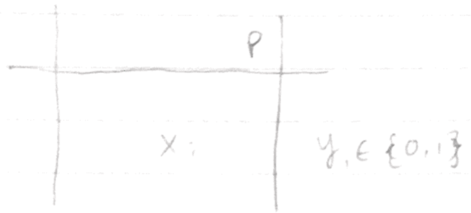
$y_i x_i^T\beta$.

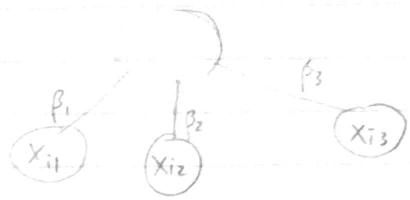
11/7/2017.
Lecture 12

Logistic regression

Linear structure $x_i^T\beta$

non-linear link sigmoid $(r) = \frac{1}{1+e^{-r}}$





linear model

$$R(\beta) = |Y - X\beta|^2 \Rightarrow \hat{\beta} = (X^T X)^{-1} X^T Y$$

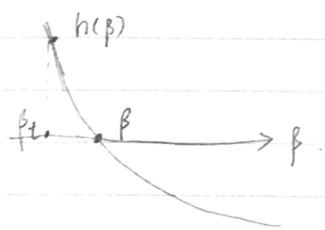
generalized linear model

$\ell(\beta) =$ iterated reweighted LS.

gradient ascent.

$$\beta_{t+1} = \beta_t + \gamma_t \ell'(\beta_t).$$

Newton Method



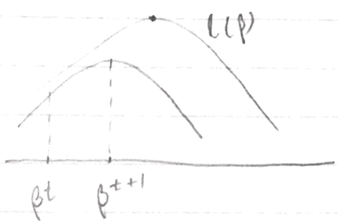
$\boxed{\text{1 - dim}}$ $h(\beta) = 0$

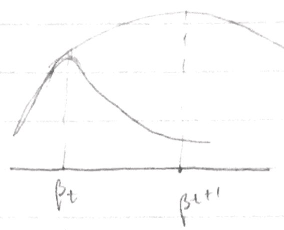
$$h(\beta) = h(\beta_t) + h'(\beta_t)(\beta - \beta_t)$$

$$\beta - \beta_t = -\frac{h(\beta_t)}{h'(\beta_t)}$$

$$\beta_{t+1} = \beta_t - \frac{h(\beta_t)}{h'(\beta_t)}$$

$\boxed{\text{2-dim}}$ $l(\beta) = l(\beta_t) + l'(\beta_t)\underbrace{(\beta-\beta_t)}_{\max} + \frac{1}{2} l''(\beta_t)(\beta-\beta_t)^2$

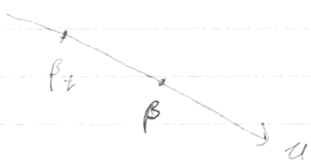




$$\beta_{t+1} = \beta_t - \frac{l'(\beta_t)}{l''(\beta_t)}$$

p-dim. $\beta_{t+1} = \beta_t - [l''(\beta_t)]^{-1}_{1\times p}\, l'(\beta_t)_{p\times 1}$

$$l(\beta) = l(\beta_t) + \langle l'(\beta_t), \beta-\beta_t\rangle + \frac{1}{2}(\beta-\beta_t)^T l''(\beta_t)(\beta-\beta_t).$$



$\beta = \beta_t + a\cdot\vec{u}$

$l(\beta) = l(\underbrace{\beta_t + a\cdot\vec{u}}_{\beta})$. $= f(a)$

$$u = \frac{\beta-\beta_t}{|\beta-\beta_t|}$$

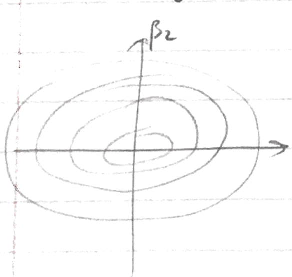
$$f(a) = f(0) + f'(0)\cdot a + \frac{1}{2} f''(0)\cdot a^2$$
$$= l(\beta_t) + \langle l'(\beta_t), \vec{u}\rangle\cdot a + u^T l''(\beta)\, u\cdot \frac{1}{2}a^2$$

$$\frac{df(a)}{da} = \frac{\partial l}{\partial a} = \frac{\partial l}{\partial \beta^T}\cdot\frac{\partial \beta}{\partial a} = \boxed{l'(\beta)^T\cdot u}$$
b

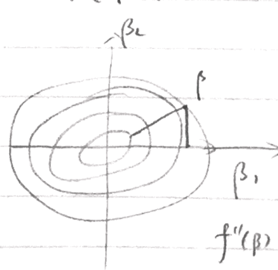
$$= \langle l'(\beta), u\rangle.$$

$$l''(\beta) = Q\lambda Q^T$$

Assume $l'' = \lambda$, $(\beta-\beta_t)^T l''(\beta_t)(\beta-\beta_t) = (\beta-\beta_t)^T\lambda(\beta-\beta_t) = \sum^{p}\lambda_j(\beta_j - \beta_{jt})^2$



$$f(\beta) = \sum_{j=1}^{p}\lambda_j\beta_j^2 \quad (\lambda_1<\lambda_2)$$



$$f'(\beta) = 2\begin{pmatrix}\lambda_1\beta\\ \lambda_2\beta\end{pmatrix}$$

$$f''(\beta) = 2\begin{pmatrix}\lambda_1 & \\ & \lambda_2\end{pmatrix}$$

$$f''(\beta)^{-1} f''(\beta)$$

Regularize learning. aviod overfitting.

accept for estimate error or uncertaintinty.

frequencies. $\hat{\beta} \sim N(\beta_{true}, V)$ — $\hat{\beta}$ random, β_{true} fixed

Bayesian $\beta_{true} \sim N(\hat{\beta}, V)$ — β_{true} fixed, $\hat{\beta}$ random.

$\rightarrow$ prior $\beta_{true} \sim P(\beta_{true})$

linear regression.

$\ell(\beta) = \|Y - X\beta\|^2 + \lambda \|\beta\|_{\ell_2}^2 \rightarrow$ ridge regression shrinkge

$+ \lambda \|\beta\|_{\ell_1} \rightarrow$ lasso Selection

LS Assume $X^TX = I$ $\quad Y = X\beta_{true} + \varepsilon = \sum_{j=1}^{p} x_j \beta_{true,j} + \varepsilon$.

$$\hat{\beta} = (X^TX)^{-1}X^TY = X^TY$$

$$\hat{\beta}_{LS,j} = \langle Y_j, X_j \rangle = \beta_{true,j} + \underbrace{\langle \varepsilon, x_j \rangle}_{\delta_j} = \beta_{true,j} + \delta_j \quad j = 1 \dots p$$

Training error: $|Y - X\hat{\beta}_{LS}|^2 = |\sum_j x_j \beta_{true,j} + \varepsilon - \sum x_j \hat{\beta}_{LS,j}|^2$.

$$= |\varepsilon - \sum_j x_j \delta_j|^2$$

$$= |\sum_{j=p+1}^{n} x_j \delta_j|^2 = \sum_{j=p+1}^{n} \delta_j^2 \qquad \varepsilon = x_1\delta_1 + x_2\delta_2 + \dots + x_p\delta_p + x_{p+1}\delta_{p+1} + \dots + x_n\delta_n.$$

$E(\text{training error}) = (n-p)\sigma^2$

$E(|\varepsilon|^2) = n\sigma^2$

we can do better than the true value.

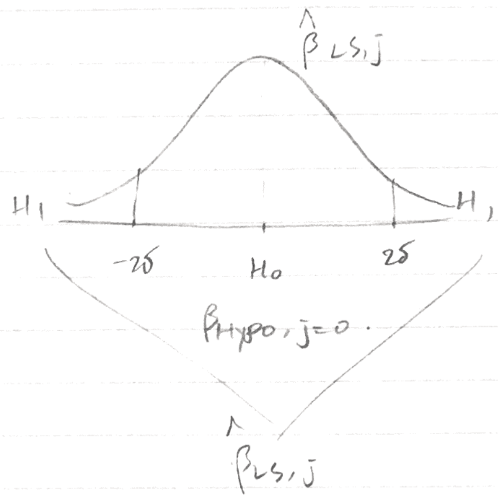
Testing error: $|\tilde{Y} - X\beta_{LS}|^2 = |X\beta_{true} + \tilde{\varepsilon} - X\beta_{LS}|^2 = |\tilde{\varepsilon} - \sum_{j=1}^{p} x_j\delta_j|^2$ $\quad (\tilde{\varepsilon} \perp \varepsilon)$

$$= |\tilde{\varepsilon}|^2 + |\sum_{j=1}^{p} x_j \delta_j|^2 - 2\langle \tilde{\varepsilon}, \sum_j x_j \delta_j \rangle$$

$$= n\sigma^2 + p\sigma^2 - 0 \doteq (n+p)\sigma^2.$$

11/9/2017
Lecture 14

That why we need to regularization:
How?



(1) Hypothistest. $H_0: \beta_{true,j}=0$

$H_1: \beta_{true,j} \neq 0$

(2) more smooth b/w null and alterative.

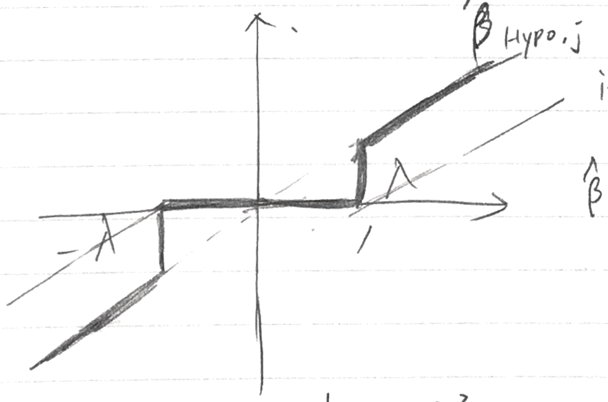
ridge estimator is shrinkage.

$$\hat{\beta}_{shrinakage,j} = \frac{\hat{\beta}_{LS,j}}{1+\lambda} \rightarrow \lambda > 0.$$

$$|Y - X\beta|^2 + \lambda|\beta|^2 \rightarrow l_2, \text{ redge regression.}$$

(3) Still from (1)

Lasso estimator is soft thresholding



it's a jump here.

$$\hat{\beta}_{Lasso} \begin{cases} 0 & \text{if } |\hat{\beta}_{LS,j}| \le \lambda \\ \hat{\beta}_{LS,j} - \lambda & \hat{\beta}_{LS,j} > \lambda \\ \hat{\beta}_{LS,j} + \lambda & \hat{\beta}_{LS,j} < -\lambda \end{cases}$$

$$\frac{1}{2}|Y - X\beta|^2 + \lambda|\beta|_{es}$$

$$\hat{\beta}_{j,Lasso,\lambda} = \text{sign}(\hat{\beta}_j)\max(0, |\hat{\beta}| - \lambda).$$

(4) Bayesian prior.

$\beta_{\text{true},j} \sim N(0, \underbrace{\tau^2}_{\text{small}})$

$\beta_{\text{true},j} \sim 99\% \, \delta_0 + 1\% \, N(0, \underbrace{r^2}_{\text{large}})$

spike/slab prior $\rightarrow$ selection

Mallow CP $\quad |Y - X\beta|^2 + 2p\sigma^2 = |Y - X\beta|^2 + 2\sigma^2 |\beta|_{\ell_0}$

$\hookrightarrow$ # of non-zero elements.

non-convex.

relax $\downarrow$

$|\beta|_{\ell_1} \quad |\beta|_{\ell_0}$

Ridge Regression.

$\ell(\beta) = |Y - X\beta|^2 + \lambda |\beta|^2$

how to minimize?

$$\frac{\partial \ell}{\partial \beta^T} = \frac{\partial \ell}{\partial e^T} \cdot \frac{\partial e}{\partial \beta^T} + \lambda 2 \beta^T \qquad e = Y - X\beta$$

$$= 2e^T(-X) + 2\lambda\beta^T$$

$$\ell'(\beta) = -2X^T(Y - X\beta) + 2\lambda\beta = 0$$

$$-X^TY + X^TX\beta + \lambda\beta = 0.$$

$$X^TX\beta + \lambda\beta = X^TY$$

$$(X^TX + \lambda)\beta = X^TY$$

$$\hat{\beta}_{\text{ridge}} = (X^TX + \lambda I)^{-1} X^TY$$

Code part: myRidge $\leftarrow$ function(X, Y, lambda).

(type on computer).

how to do in QR?

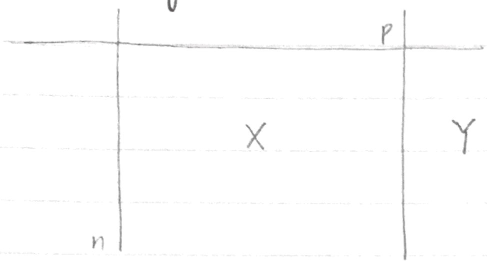
$$\ell(\beta) = |Y - X\beta|^2 + |0 + \sqrt{\lambda} I \cdot \beta|^2 = \left| \begin{pmatrix} Y \\ 0 \end{pmatrix} - \begin{pmatrix} X \\ \sqrt{\lambda} I \end{pmatrix} \beta \right|^2$$

$$\begin{array}{c|c} X & Y \\ \hline \begin{matrix} \sqrt{\lambda} & & \\ & p \times p & \\ & & \sqrt{\lambda} \end{matrix} & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \end{array}$$

11/14/2017.

Lecture 15.

Overfitting.



n big

p also big, sometimes $p \gg n$.

high dimension statistics.

Ridge:

$$|Y - X\beta|^2 + \lambda |\beta|_{\ell_2}^2$$

Spline:

$$y = \beta_0 + \sum_{j=1}^{p} \beta_j \max(0, x - k_j)$$

→ change of slope

$\lambda |\beta|_{\ell_2}^2 \longrightarrow$ Smooth.

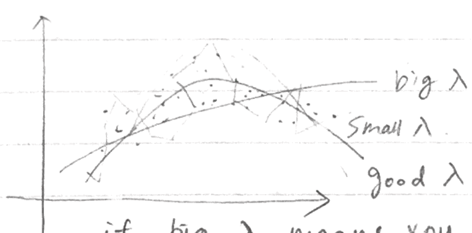
$$\hat{\beta}_\lambda = (X^TX + \lambda I_{p \times p})^{-1} X^TY$$

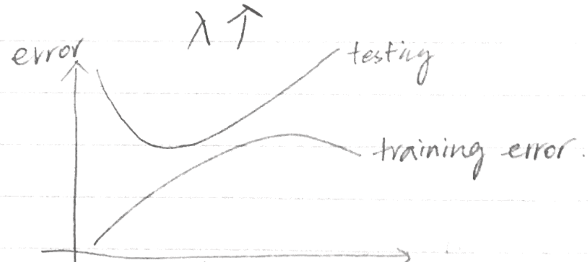
special case: $X^TX = I_{p\times p}$

$$\hat{\beta}_{j\lambda} = \frac{\langle x_j, Y \rangle}{1+\lambda} = \frac{\hat{\beta}_{j,LS}}{1+\lambda}$$

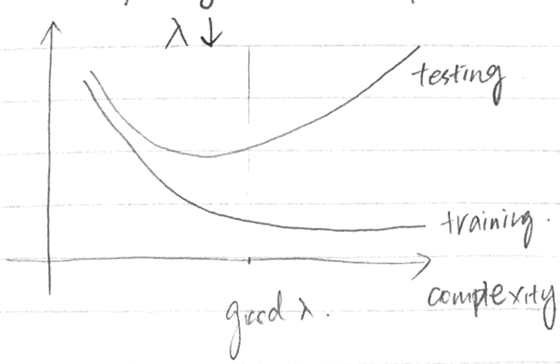
↓ shrinkage estimator.

$H_0: \beta_{j,true} = 0 \qquad H_1: \beta_{j,true} \neq 0$





if big λ means you have simple model

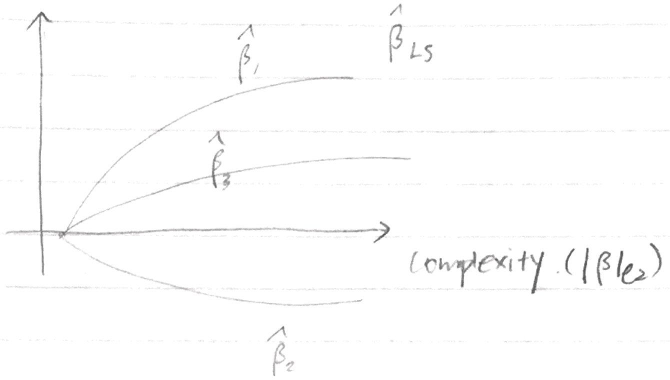


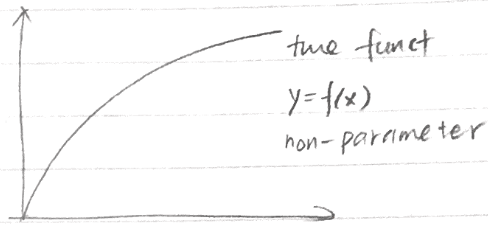
cross-validation.

training → $\frac{4}{5}$ "training"

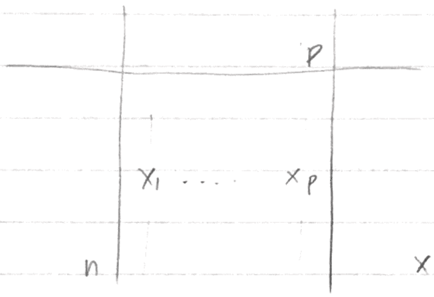
training → $\frac{1}{5}$ "testing"

Solution path.

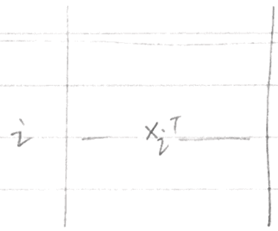




$Y = f(x) + \varepsilon.$



$X^TX = I_p$



$y_i = f_i + \varepsilon$

$f(x_i)$

$\underset{n\times1}{Y} = \underset{n\times1}{f} + \underset{n\times1}{\varepsilon}$

$\hat\beta_{LS} = X^TY$

$= X^T(f+\varepsilon)$

$= \underbrace{X^Tf}_{\text{signal}} + \underbrace{X^T\varepsilon}_{\text{noise } \delta}$

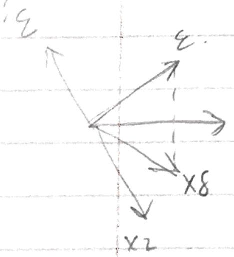
$\beta_{best} \to$ to approx the true signal.

what about training / test error?

training error. $|Y - X\hat\beta_{LS}|^2 = |f + \varepsilon - X(\beta_{best} + \delta)|^2$

$= |f - X\beta_{best} + \varepsilon - X\delta|^2$

$= |f - X\beta_{best}| + |\varepsilon - X\delta|^2 + 2 < f - X\beta_{best}, \varepsilon - X\delta >$

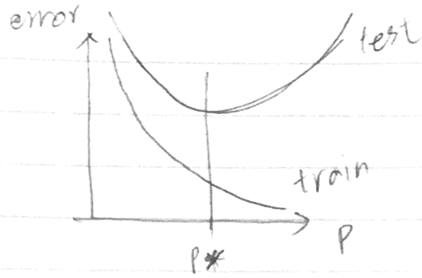


$E(\text{train}) = \underbrace{|f - X\beta_{true}|^2}_{\text{model bias.}} + \underbrace{(n-p)\sigma^2}_{\text{overfitting.}}$

test error: $|\tilde Y - X\hat\beta_{LS}|^2 = |f + \tilde\varepsilon - X(\beta_{best} + \delta)|^2$

$E(\text{testing}) = |f - X\beta_{best}|^2 + \underbrace{E(|\tilde\varepsilon - X\delta|^2)}_{(n+p)\sigma^2}$

$E(\text{test} - \text{train}) = 2p\sigma^2$



we do not do least square but regulization.
if you increase the complex of model
model bias ↓
overfitting ↓
but $E(|\tilde{\varepsilon} - X\delta|^2) \uparrow$.

Ridge: $\hat{\beta}_\lambda = (X^TX + \lambda I_p)^{-1} X^T Y$.

the randomness come from

training: $|Y - X\hat{\beta}_\lambda|^2 = |f + \varepsilon - X\hat{\beta}_\lambda|^2$

testing: $|\tilde{Y} - X\hat{\beta}_\lambda|^2 = |f + \tilde{\varepsilon} - X\hat{\beta}_\lambda|^2$

$E = 0$

$E[\text{testing} - \text{training}] = E(2\langle \varepsilon, X\hat{\beta}\rangle - 2\langle \tilde{\varepsilon}, X\hat{\beta}_\lambda\rangle)$
$= 2E\langle \varepsilon, X\hat{\beta}_\lambda\rangle - 2\langle \tilde{\varepsilon}, X\hat{\beta}_\lambda\rangle)$
$= 2E\langle \varepsilon, X\hat{\beta}_\lambda\rangle = 2\hat{P}_\lambda \delta^2$

$\hat{P}_\lambda = \dfrac{E\langle \varepsilon, X\hat{\beta}_\lambda\rangle}{\delta^2}$ → effective degree of freedom.

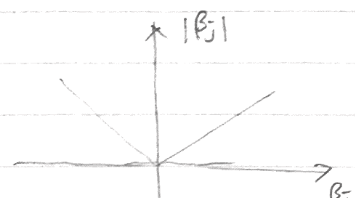
Recall LS. $E[\text{testing} - \text{training}] = 2p\delta^2$

spline: 1000 knots. $p = 1000$ $\hat{P}_\lambda = 10$

Lasso (Least absolute shrinkage selection operator)

$R(\beta) = \frac{1}{2}|Y - X\beta|^2 + \lambda|\beta|_{\ell_1}$

$|\beta|_{\ell_1} = \sum_{j=1}^{p} |\beta_j|$



$R'(\beta) = -X^T(Y - X\beta) + \lambda S = 0$

$S_j = \begin{cases} 1 & \beta_j > 0 \\ -1 & \beta_j < 0 \\ [-1,1] & \beta_j = 0 \end{cases}$

minimize fix β_j $\forall j \neq k$
Solve β_k, $R(\beta_k) = \frac{1}{2}|\underbrace{Y - \sum_{j\neq k} X_j\beta_j}_{R} - X_k\beta_k| + \lambda\sum_{j\neq k}|\beta_j| + \lambda|\beta_k|$

$R(\beta_K) = \frac{1}{2} |R - X_K\beta_K|^2 + \lambda|\beta_K| + \text{const}$

$R'(\beta_K) = -X_R^T(R - X_K\beta_K) + \lambda S_K = 0$

$= -\langle R, X_K\rangle + |X_K|^2\beta_K + \lambda S_K = 0$

$|X_K|^2 \cdot \beta_K = \langle R, X_K\rangle - \lambda S_K$

$$\beta_K \begin{cases} \dfrac{\langle R, X_K\rangle - \lambda}{|X_K|^2}, & \text{if } \langle R, X_K\rangle > \lambda \\ \dfrac{\langle R, X_K\rangle + \lambda}{|Y_K|} & \text{if } \langle R, X_K\rangle < -\lambda \\ = 0 & \langle R, X_K\rangle \in [-\lambda, \lambda] \end{cases}$$

$\beta_K = \text{sign}(\hat{\beta}_{R,LS}) \max(0, |\hat{\beta}_{R,LS}| - \frac{\lambda}{|X_K|^2})$

$= \text{soft thresholding}(\hat{\beta}_{R,LS})$ $\qquad \hat{R}_{R,LS} = \dfrac{\langle R, X_K\rangle}{|X_K|^2}$

$\begin{cases} H_0 : & R_{K\,\text{true}} = 0 \\ H_1 : & \beta_{K,\text{true}} \neq 0 \end{cases}$

Coordinate descent:

$R(\beta)$ for $(K = 1, \dots, P)$ find β_K given $\beta_{j \neq K} = (\beta_j, j \neq K)$ minimize

change

$K(\beta_1 \ldots \beta_K \ldots \beta_P)$

fixed

n = 50

P = 200

S = 10

T = 10

lambda_all = (100:1)*10

L = length(lambda_all)

x = matrix((n*p), nnrom=n)

beta_true[1:S] = 1:S

Y = X%*% beta_true + rnorm(n)

$Y = X\beta_{true} + \varepsilon$

$\begin{pmatrix} 1 \\ \vdots \\ 5 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

from λ big to small

all 0 → β_K start to get non-zero value → more value

11/16/2017.

Lecture 16

Lasso:

$$[R, \beta] = \frac{1}{2} \|Y - X\beta\|^2 + \lambda |\beta|$$

$$R(\beta) = -X^T(Y - X\beta) + \lambda S(\beta) = 0$$

$$S = \begin{pmatrix} s_1 \\ \vdots \\ s_p \end{pmatrix} \qquad S_p = \begin{cases} 1 & \beta > 0 \\ -1 & \beta < 0 \\ (-1, 1) & \beta = 0 \end{cases}$$

Calculate

$\beta(\beta_1, \ldots, \beta_i, \ldots \beta_p)$.

for (t = 1 ... T)

for (k = 1 ... p)

min $k(\beta_1 \ldots \beta_k)$.

Solution path: λ = big $\beta = 0$.

gradually reduce λ.

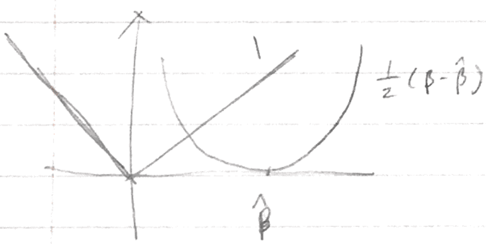
$X_{n \times p}$ $Y_{n \times 1}$

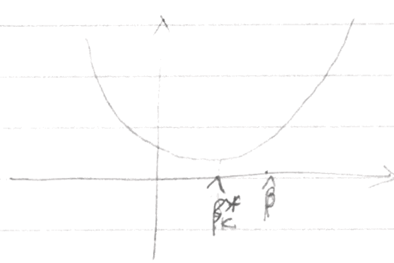
```
length_all = (100 : 1) * 10.
L = Length ( length_all )

R = Y
ss = rep(n,p).
for(j in 1:p).
   ss[j] = Sum (X[,j]^2)

for ( l in 1:L)
{ lambda = lambda_all [ l ].
    for ( t in 1:(T) ) :  → try different.
      { for( k in 1:p) :
          {
            db = sum [ R * Y[,k]) / ss[k ]
            b  =  beta [k] + db
```

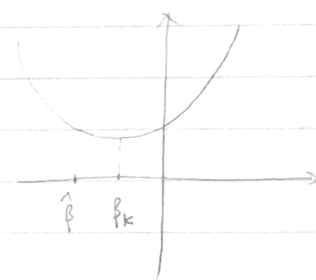
no intercept →

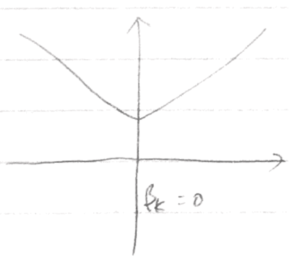
```
b = sign[b] * max[0, d[b] - lamda/ss[K]]
db = b - beta[K]
R = R - X[,K] * db.
beta[R] = b.
}
}
beta_all[,l] = beta
}
```

Solve $\frac{1}{2}(\beta-\hat{\beta})^2 + \lambda|\beta| = R(\beta)$

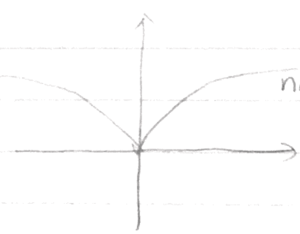


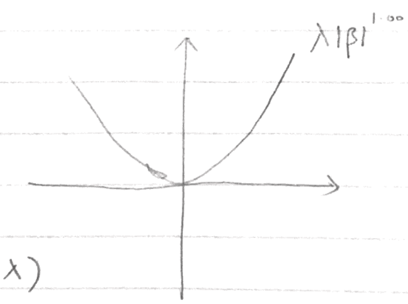






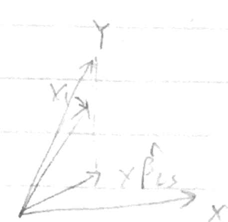
big λ, slope $+\lambda \rightarrow -\lambda$.





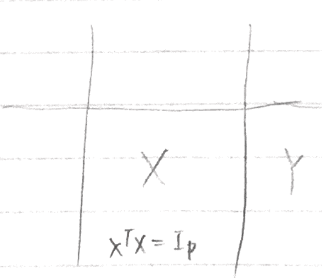
$\beta_t = \text{sign}(\hat{\beta})\ \max(0, |\hat{\beta}|-\lambda)$

$\begin{cases} H_0: \beta = 0 \\ H_1: \beta \neq 0 \end{cases}$



$\frac{1}{2}|Y - X\beta|^2 + \lambda|\beta|_{l_1}$

$= \frac{1}{2}|Y - X\hat{\beta}_{LS}|^2 + \frac{1}{2}|\beta - \hat{\beta}|^2 + \lambda|\beta|_{l_1}$



$\hat{\beta}_K = \text{sign}(\hat{\beta}_{LS,K})\max(0, |\beta_{LS,K}| - \lambda)$

Lasso: $\hat{\beta}_{Lasso,K} = \text{Soft-threshold}(\hat{\beta}_{LS,K}, \lambda)$

$\hat{\beta}_{ridge,K} = \text{shrinkage}(\hat{\beta}_{LS,K}, \lambda) = \frac{\hat{\beta}_{LS,K}}{1+\lambda}$

when $X^TX = I$

$E[\text{testing error}]$

$= |f - X\beta_{best}|^2 + n\sigma^2 + \boxed{E(|\hat{\beta} - \beta_{best}|^2)}$

$|f - X\beta_{best}|^2$ → model bias

$n\sigma^2$ → noise

$E(|\hat{\beta} - \beta_{best}|^2)$ → $|E(\hat{\beta}) - \beta_{best}|^2 + Var(\hat{\beta})$

$|E(\hat{\beta}) - \beta_{best}|^2$ → estimation bias

$Var(\hat{\beta})$ → variance

testing error:

$|Y - X\hat{\beta}|^2 = |f + \vec{\varepsilon} - X\hat{\beta}|^2$

$= |f - X\hat{\beta}|^2 + |\varepsilon|^2 \rightarrow n\sigma^2$

$= |f - X\beta_{best}|^2 + |\hat{\beta} - \beta_{best}|^2$

sten estimator

$$\hat{\beta}_{sten} = \left(1 - \frac{(p-2)\sigma^2}{|\hat{\beta}_{LS}|^2}\right)\hat{\beta}$$

$$= \hat{\beta}_{ridge, \lambda sten}$$

☆ $E(|\hat{\beta}_{sten} - \beta_{best}|^2) \leq E(|\hat{\beta}_{LS} - \beta_{best}|^2)$

Linear Regression

$|Y - X\beta|^2 + \begin{cases} \lambda|\beta|^2_{\ell_2} \rightarrow \text{Ridge} \rightarrow \text{shrinkage: good for smooth} \\ \lambda|\beta|_{\ell_1} \rightarrow \text{Lasso} \rightarrow \text{selection: good for sparsity} \end{cases}$

Logistic Regression

$Loss(Y, X, \beta) + \begin{cases} \lambda|\beta|^2_{\ell_2} \rightarrow \text{hinge loss} \quad \sum_{i=1}^{n} \max(0, 1 - y_i x_i^T\beta) + \lambda|\beta|^2_{\ell_2} \rightarrow \text{SVM} \\ \lambda|\beta|_{\ell_1} \rightarrow \text{exponential loss} + \text{coordinate descent} \rightarrow \text{Adaboost.} \end{cases}$

neural network.

11/28/2007

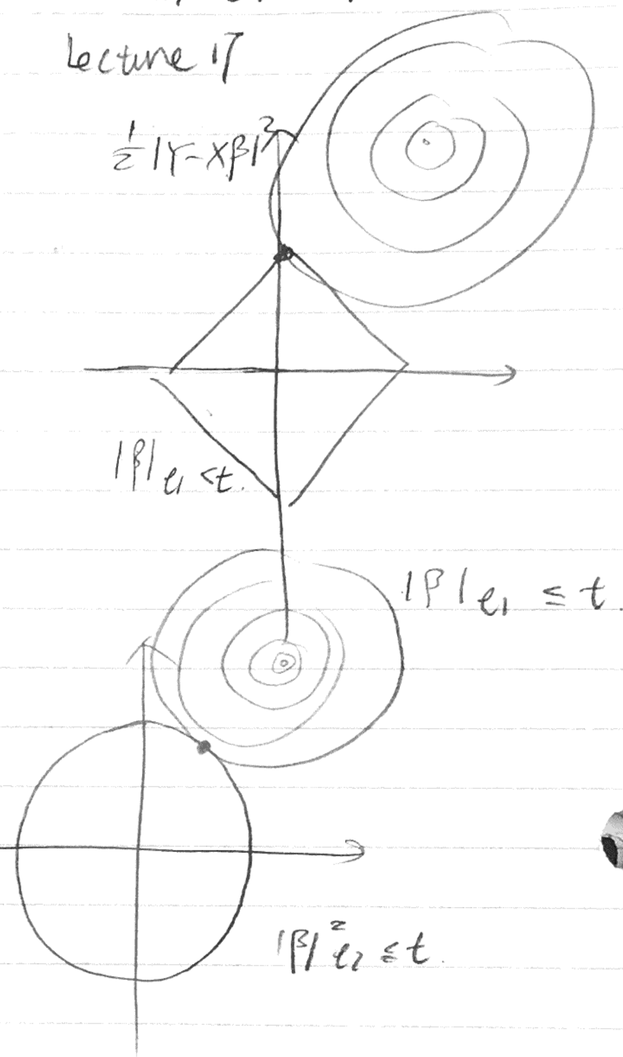
Lecture 17

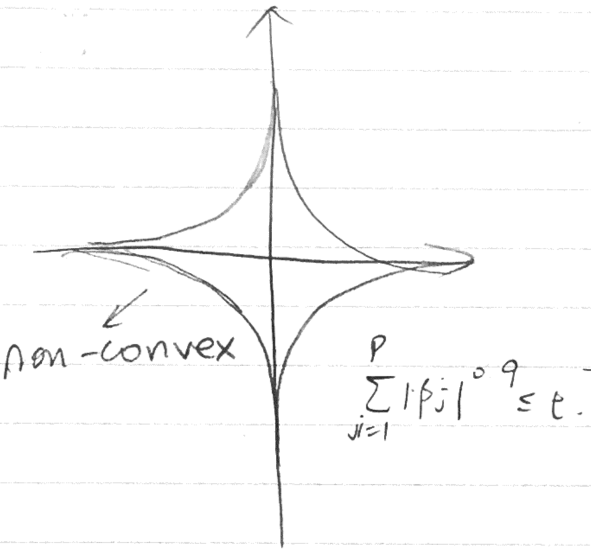
Lasso.

$$\frac{1}{2}|Y - X\beta|_{\ell_2}^2 + \lambda|\beta|_{\ell_1}$$

principle error: $\min \frac{1}{2}|Y - X\beta|_{\ell_2}^2$

s.b. $\|\beta\|_{\ell_1} < t$





Forward regression. (matching pursit).

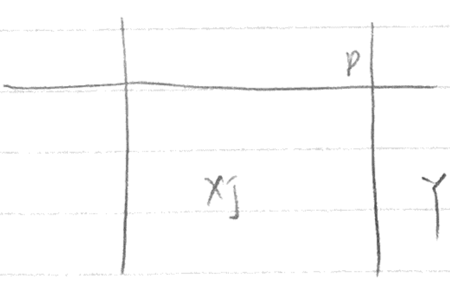
selection.

$$R(\beta) = \frac{1}{2}|Y - X\beta|^2$$

Gradient descent from $\beta = 0$.

$$\frac{\partial R}{\partial \beta} = -X^T(Y - X\beta)$$

$$\frac{\partial R}{\partial \beta_j} = -\langle X_j, Y - X\beta\rangle$$



choose j over $\langle X_j, Y - X\beta\rangle$.

$$R = \frac{1}{2}|Y - X\beta - X_j \Delta\beta_j|^2$$

$$\Delta\beta_j = \frac{\langle Y - X\beta, X_j\rangle}{|X_j|^2}$$

Connection to Lasso.

$$\min \frac{1}{2}\|Y - X\beta\|^2$$

$\Delta\beta_j = \varepsilon$

$$|\beta|_{\ell_1} \le t \rightarrow |\beta|_{\ell_1} \le \underset{\text{choose } j}{t} + \underset{\varepsilon}{\Delta t}.$$

ε - boosting $\longrightarrow \approx$ Lasso.

$$\begin{array}{c|ccc|l}
 & & j & p & \\
\hline
i & x_{11} & x_{ij} & x_{ip} & y_i \in \{+1, -1\} \\
 & & & & \text{weak classifier.} \\
n & & & &
\end{array}$$

committe.

$$y_i \approx \text{sign}\Big\{\underbrace{\sum_{j=1}^{p} x_j\beta_j}_{\text{score}}\Big\}.$$

$$\text{margin of } i: \quad y_i \sum_{j=1}^{p} x_j \beta_j.$$

$$L(\beta) = \sum_{i=1}^{n} e^{-y_i \sum_{j=1}^{p} x_{ij}\beta_j}.$$

$$\frac{\partial L}{\partial \beta_k} = \sum_{i=1}^{n} \underbrace{e^{-y\sum_{j=1}^{p} x_j\beta_j}}_{w_i}(-y\, x_{ik})$$

① find k to max $\sum_{i=1}^{n} w_i \; y_i x_{ik}$.

pay attention for examples. not classify well.

choose classifier k perform the best on (x_i, y_i, w_i)

$$L(\Delta\beta_k) = \sum_{i=1}^{n} e^{-y_i\left(\sum_{j=1}^{p} x_j\beta_j + x_{ik}\Delta\beta_k\right)}$$

+ correct

- incorrect

$$= \sum_{i=1}^{n} w_i \cdot e^{-\boxed{y_i x_{ik}}\Delta\beta_k}.$$

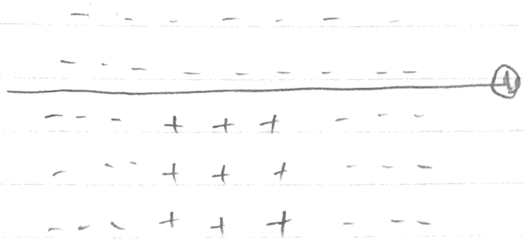
② After find k, min $L(\Delta\beta_k)$.

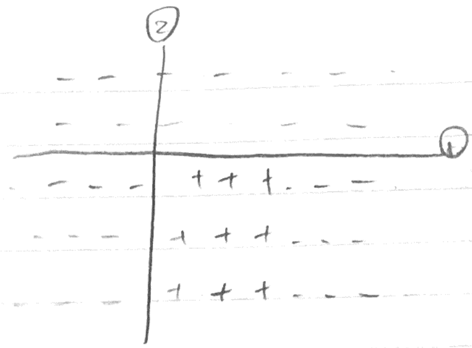
$$= \sum_{y_\beta = x_{\Delta k}} w_i e^{-\Delta\beta_k} + \underbrace{\sum_{y_i \ne x_k} w_i}_{\varepsilon} e^{\Delta\beta_k}$$

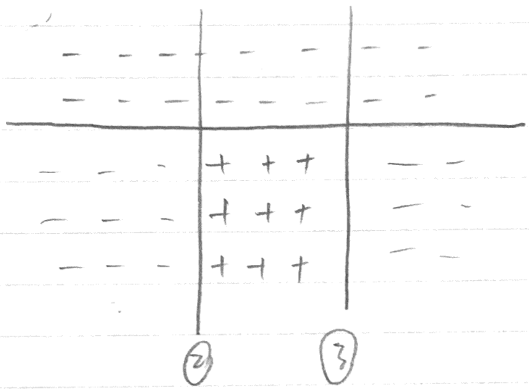
$$\sum_{i=1}^{n} w_i = 1 \qquad a+b \ge 2\sqrt{ab} \qquad \ge 2\sqrt{\varepsilon(1-\varepsilon)}$$

$$(\sqrt{a}-\sqrt{b})^2 \ge 0. \qquad (1-\varepsilon)e^{-\Delta\beta_k} = \varepsilon\, e^{\Delta\beta_k}$$

$$\Delta\beta_k = \frac{1}{2}\log\frac{1-\varepsilon}{\varepsilon}$$







Adapting brifling

↓

$\beta_k = \log \frac{1-\varepsilon}{\varepsilon}$.

```
MyAdaboosting ← function ( x_train, Y_train, x_test, Y_test, num_iterations )
{ n, p
    for ( i in 1: num_iterations).
    {
        w ← w/ sum(wi)
        a = matrix( rep(1,n) , nrow=1 ) %k% (matrix( w * Y_train, n,p) * X_train).
        e = (1-n)/2
        K = whch.min(e).
        db= log(1-e[k]/e[k]))/2

        beta [k] ← beta [k] + db
        w ← w * exp(-y_train * x [, k] * db)
    }.
```

11/30/2017.

Lecture 18.

Adaboost.

$x_1 \cdots x_j$		$y_i \in \{+1,-1\}$
$x_j \in \{+1,-1\}$		

$$y_i = \text{sign}\left(\sum_{k=1}^{d} \beta_k h_k(X_i)\right)$$

$$L(\beta) = \sum_{i=1}^{n} \boxed{e^{-y_i \sum_{j=1}^{p} \beta_j x_{ij}}}$$ exp loss

$$e^{-y_i x_i^T \beta} = e^{-r}$$

restric situation

X_i : raw observ.

$\downarrow$

$h_{i1}, h_{i2}, \ldots, h_{ik}, \ldots h_{id}$ weak classifiers

$h_{ik}(X_i) \in \{+1,-1\}$.

$w_i = \frac{1}{n}$ $\quad i = 1, \ldots, n$.

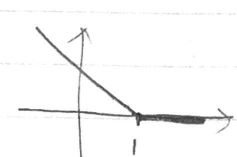
Iterate: correct committee $= \sum_{\beta=1}^{d} \beta_k h_k(X_k) \xrightarrow{\text{boost}}$ correct $+ \beta_{new} h_{new}(x_i)$.

$$w_i = e^{-y_i \sum_{k=1}^{d} \beta_k h_k(x)}$$

$$w_i \leftarrow \frac{w_i}{\sum_{i=1}^{n} w_i}$$ $h_{new} \leftarrow$ weak learn $(\{x_i, y_i, w_i\})$

$\Sigma = \sum_{\text{mistakes of } h_{new}} w_i$ $\quad y_i \neq h_{new}(x_i)$ $\quad \beta_{new} = \frac{1}{2} \log \frac{1-\varepsilon}{\varepsilon}$.

<u>Supported Vector Machine</u>: hinge loss



x_{ij} : continuous. $\quad L(\beta) = \sum_{i=1}^{n} \max(0, 1 - y X_i^T \beta) + \frac{\lambda}{2} |\beta|_{\ell_2}^2$

$\hookrightarrow$ better beyond 1.
then you won't suffer the loss

$$L'(\beta) = -\sum_{i=1}^{n} 1(y_i x_i^T \beta \le 1)\, y_i X_i + \lambda \beta_i$$

↓

support vectors

```
my-SVM <- function (X-train, Y_train, X-test, Y_test, lamda = 0.1,
                    num-iterations = 1000, learning_rate = .1)

for ( it in 1: num_iterations)
{
   score <- X_train %*% beta
   delta <- score * Y_train < 1
   dbeta <- matrix (rep(1,n), nrow) %*% ((matrix(delta * Y, n, P) *
                                                   X-train)) /n

   beta <- beta + learning_rate * t(dbeta)
   beta [2:p] <- beta [2:p] -lamda + beta [2:P]
}
```

Logistic Regression

$$P(y_i = +1) = \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}} = \text{sigmoid}(x_i\beta) = \frac{1}{1+e^{-x_i^T\beta}} \Big\} \Rightarrow P(y_i \mid x_i^T\beta)$$

$$= \frac{1}{1+e^{-y_i + x_i^T\beta}}$$

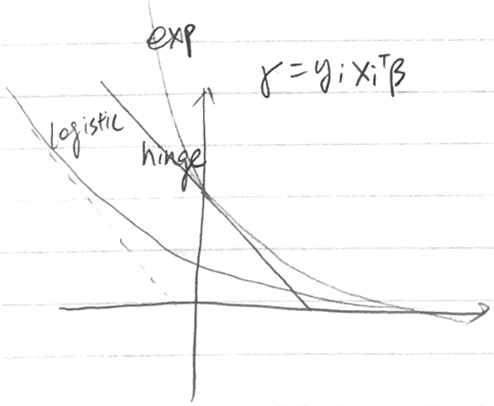
$$P(y_i = -1) = \frac{1}{1+e^{x_i^T\beta}}$$

$$-\log P(y_i \mid x_i^T\beta) = \log(1+e^{-y_i x^T\beta}) \quad \text{lostigic loss.}$$

$$= \log(1+e^{-r})$$

$r \to \infty$ e^{-r}

$r \to -\infty$ $-r$.



$y_i = 1, \quad x_i^T\beta \geq 1$

$y_i = -1 \quad x_i^T\beta \leq -1$

$\vec{u} = \frac{\beta}{|\beta|_{\ell_2}}$ $\quad x_i$: marginal margin

$\langle x_i, \vec{u} \rangle = \langle x_i, \frac{\beta}{|\beta|_{\ell_2}} \rangle = \frac{x^T\beta}{|\beta|_{\ell_2}}$

$= \frac{1}{|\beta|_{\ell_2}}$

$\min \frac{1}{2}|\beta|^2_{\ell_2}$

subject to $\quad y_i x_i^T\beta \geq 1 \quad i = 1, \dots, n$

line-seperable $\quad$ slack-variable ξ_i

$\min \frac{1}{2}|\beta|^2_{\ell_2} + C \#(\xi_i > 0)$.

s.t. $\quad y_i x^T\beta \geq 1 - \xi_i \quad \hookrightarrow C \sum_{i=1}^n \xi_i$

$\xi_i = \max(0, 1 - y_i x_i^T\beta)$ $\quad \xi_i \geq 0$.

$X_+ = \sum_{i \in +} c_i x_i \quad c_i \geq 0 \quad \sum_{i=1}^n c_i$

$X_- = \sum_{i \in -} c_i x_i \quad c_i \geq 0 \quad \sum_{i=1}^n c_i$

Condition

$\min |x_+ - x_-|^2$:
$c = \{c_i\}$

$|x_+ - x_-|^2 = |\sum_{i=1}^n c_i y_i x_i|^2$.

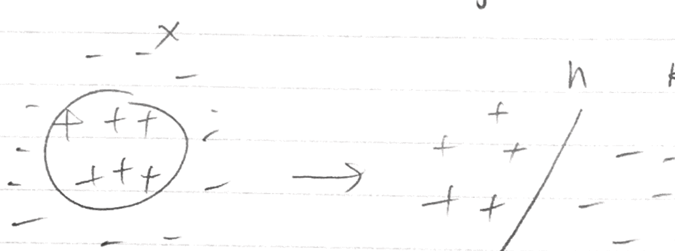
$= \sum_{i,j} c_i c_j y_i y_j \langle x_i, x_j \rangle$.

realistic situation.

x_i : raw observ.

↓

$(h_1, \dots h_k \dots h_d) = h(x)$



$K(x_i, x_j) = e^{-(x_i - x_j)^2 / 2\sigma^2}$

sequential marginal optimial (SMO)

each time change 2 c_i.

$c_i > 0 \longrightarrow$ S.V's $\quad \langle h(x_i), h(x_j) \rangle$

Kernal trick $\quad = K\langle x_i, x_j \rangle$

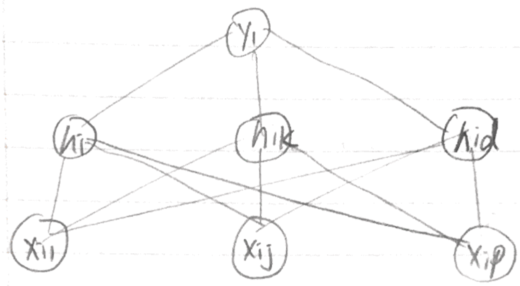
y_i

h_{1k} h_{2k} . . $= h(x)$.

$$\begin{cases} y_i \sim \text{logistic regression } (h_{i1}, \dots, h_{id}). \\ h_{ik} \sim \text{logistic regression } (x_{i1}, x_{i2}, \dots, x_{ip}). \end{cases}$$

12/5/2017.

Lecture 19



Adaboost: h_{ik}: weak classies hypothesis.

Selected sequentially

SVM: $h_i = \begin{pmatrix} h_i \\ \vdots \\ h_{ik} \\ \vdots \\ h \end{pmatrix}$ $h_i = h(x_i)$ h_i: implicit. $d = \infty$

$y_i = \text{sign}(\langle h(X), \beta \rangle)$

$\uparrow$

kernal tink: reproducing kernal $\langle h(x_i), h(x_j) \rangle = K(x_i, x_j)$

$$\beta_{t+1} = \beta_t + \underset{\text{learning rate}}{\gamma_t} \cdot \sum_{i=1}^{n} 1(\langle h(x_i), \beta \rangle < 1) \cdot y_i h(x_i)$$

$$\longrightarrow \quad y = \text{sign}(\langle h(x), \beta \rangle) = \text{sign}(\langle h(x_i), \sum_{i=1}^{n} \alpha_i h(x) \rangle)$$

$$= \text{sign}\left(\sum_{i=1}^{n} \alpha K(x, x_i)\right)$$

Neural network:

y_i: logistic regression on $(h_{i1}, \dots, h_{id})$. cross entropy

$$P(y_i = 1 \mid h_i) = \text{sigmoid}\left(h_i^T \beta = \sum_{k=1}^{d} \beta_k \cdot h_{ik}\right)$$

$$h_{ik} = \text{sigmoid}\left(\underbrace{x_i^T \alpha_k = \sum_{j=1}^{p} \alpha_{kj} \cdot x_{ij}}_{\gamma_{ik}}\right)$$

maximum liklihood: $L(\alpha, \beta) = \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1-y}$

$$= \prod_{i=1}^{n} \left(\frac{e^{h_i^T \beta}}{1 + e^{h_i^T \beta}}\right)^{y_i} \left(\frac{1}{1 + e^{h_i^T \beta}}\right)^{1-y_i}.$$

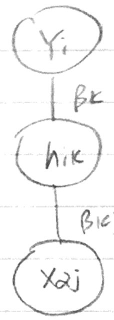
$$= \prod_{i=1}^{n} \frac{e^{y_i + h_i^T\beta}}{1 + e^{y_i + h_i^T\beta}}$$

$$\ell(\alpha, \beta) = \log L(\alpha, \beta) = \sum_{i=1}^{n} [\, y_i \, h_i^T\beta - \log(1 + e^{h_i^T\beta})]$$

$$\frac{\partial \ell}{\partial \beta_k} = \sum_{i=1}^{n} \left(y_i h_{ik} - \frac{e^{h_i^T\beta}}{1 + e^{h_i^T\beta}} \cdot h_{ik} \right) = \sum_{i=1}^{n} \underbrace{(y_i - p_i)}_{error} \, h_{ik}$$

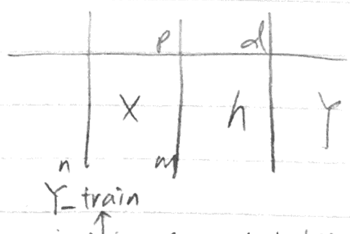
$$\frac{\partial \ell}{\partial \alpha_{kj}} = \sum_{i=1}^{n} \frac{\partial \ell}{\partial h_{ik}} \cdot \frac{\partial h_{ik}}{\partial \alpha_{kj}} = \sum_{i=1}^{n} (y_i \beta_k - p_i \beta_k) \cdot \frac{d \, sgmoid}{d \gamma_{ik}} \, x_{ij}$$

$$= \sum_{i=1}^{n} \underbrace{(y_i - p_i)}_{error} \beta_k \cdot \boxed{h_{ik}(1 - h_{ik})} \, x_{ij}$$



error back preparation

```
My_NN <- function (X_train, Y_train, X_test, Y_test, num_hidden = 20,
                   num_iterations = 1000, learning rate = .. )
{  n, p
   ..

   for(it in 1:num_iterations)
   {
      h <- 1/(1+ exp(1 + X_train %*% alpha)
                         nxp           pxd
                         nxp
      p <- 1/(1+exp( -h %*% beta))
      dbeta <- matrix(rep(1,n), nrow=1) %*% ((matrix(y - p, n, d) * h))/n
```



Rectified linear unit.

$h_k = \max(0, x_i^T \alpha_k)$

sgmoid