Administration

- Friday this week (1–2) in LM 158: fitting linear models in R
- Please check web page regularly for updates
- You should by now have a Cquest account, or have R or Splus on your PC, or be planning to go your own route re software
- Homework 1 coming next week
- Printing slides from web page (Acrobat: page setup (horizontal); expand to fit)

Linear Regression, à la HTF

inputs $X = (X_1, \dots, X_p)$: attributes, features, predictors, covariates

output $Y \in R$: response

data $(x_i, y_i), i = 1, \dots N$: instances

linear model $E(Y \mid X) = \beta_0 + \Sigma_{j=1}^{p} X_j \beta_j$

model for data: $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i, i = 1, \dots, N$

ϵ_i independent, $E(\epsilon_i) = 0$, $\mathrm{var} \epsilon_i$ constant

Learning the model: finding $f(X)$ to describe $E(Y|X)$, or other properties of the distribution of Y

Here we assume $f(X)$ known up to $p+1$ unknown parameters; just need to estimate these parameters

Want ‘good’ estimates, possibly defined via a loss function on the training data, possibly defined by prediction error on the test data

Least squares

$$\min_\beta \sum_{i=1}^N (y_i - \beta_0 \beta_z x_{i1} - \cdots - \beta_p x_{ip})^2$$

$$RSS(\beta) = \sum_{i=1}^N (y_i - \beta_0 \beta_z x_{i1} - \cdots - \beta_p x_{ip})^2$$
$$= (y - X\beta)^T (y - X\beta)$$

X is $N \times p + 1$: $X = (\underline{1}, \underline{x}_1, \ldots, \underline{x}_p)$

β is $p + 1 \times 1$: $\beta = (\beta_0, \ldots, \beta_p)^T$

solution is

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

assuming ...

fitted values (for training data)

$\hat{y} = X\hat{\beta} = X(X^T X)^{-1} X^T y = Hy$

Notes

$\hat{y} = Hy$: H is a *projection matrix*, projecting $y(\in R^N)$ onto the column space of X

if X^TX is not invertible, then the column space has dimension less than $p+1$, but we can still project y onto this space

we can remove redundant columns, or equivalently use a generalized inverse

most usual situation is when several columns of X serve to code levels of a factor

most packages detect and remove redundant columns in this case, but the convention for removing differs among packages

if X^TX is only nearly singular, ...

Properties of $\hat{\beta}$

$$\mathrm{var}\hat{\beta} = \sigma^2(X^TX)^{-1} \text{ (under the assumptions)}$$

$$\begin{aligned}\hat{\sigma}^2 = \tfrac{1}{N-(p+1)}RSS(\hat{\beta}) \\ = \tfrac{1}{N-(p+1)}(y-\hat{y})^T(y-\hat{y}) \\ = \tfrac{1}{N-(p+1)}(y-X\hat{\beta})^T(y-X\hat{\beta}) \\ = \tfrac{1}{N-(p+1)}y^T(I-H)y\end{aligned}$$

if $\epsilon \sim N(0, \sigma^2 I)$ then $\hat{\beta} \sim N(\beta, \sigma^2(X^TX)^{-1})$

$$\frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}(X^TX)^{-1}_{jj}} \sim t_{N-(p+1)}$$

```
> summary(pr.z.lm)

Call:
lm(formula = lpsa ˜ lcavol.z + lweight.z + age.z + lbph.z + svi.z +
    lcp.z + gleason.z + pgg45.z, data = pr.z.train)

Residuals:
     Min       1Q   Median       3Q      Max
-1.64870 -0.34147 -0.05424  0.44941  1.48675

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.47951    0.08963  27.665  < 2e-16 ***
lcavol.z     0.67953    0.12663   5.366 1.47e-06 ***
lweight.z    0.30494    0.11086   2.751  0.00792 **
age.z       -0.14146    0.10134  -1.396  0.16806
lbph.z       0.21015    0.10222   2.056  0.04431 *
svi.z        0.30520    0.12360   2.469  0.01651 *
lcp.z       -0.28849    0.15453  -1.867  0.06697 .
gleason.z   -0.02131    0.14525  -0.147  0.88389
pgg45.z      0.26696    0.15361   1.738  0.08755 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7123 on 58 degrees of freedom
Multiple R-Squared: 0.6944,Adjusted R-squared: 0.6522
F-statistic: 16.47 on 8 and 58 DF,  p-value: 2.042e-12

##much better is:
pr.std<-data.frame(cbind(apply(pr,2,std),pr$lpsa,pr$train))
names(pr.std)[9]<-"lpsa"
names(pr.std)[10]<-"train"
lm(lpsa˜.-train,subset=train==1,data=pr.std)
```

```
>  x=cbind(pr.z.test$lcavol.z,pr.z.test$lweight.z,pr.z.test$age.z,
+  pr.z.test$lbph.z, pr.z.test$svi.z,pr.z.test$lcp.z,pr.z.test$gleason.z,
+ pr.z.test$pgg45.z)
> dim(x)
[1] 30  8
> test.fitted = x %*% coef(pr.z.lm)
Error in x %*% coef(pr.z.lm) : non-conformable arguments
> coef(pr.z.lm)
(Intercept)    lcavol.z   lweight.z       age.z      lbph.z       svi.z
 2.47951205  0.67952814  0.30494116 -0.14146483  0.21014656  0.30520060
      lcp.z   gleason.z     pgg45.z
-0.28849277 -0.02130504  0.26695576
> x=cbind(rep(1,30),x)
> dim(x)
[1] 30  9
> test.fitted = x %*% coef(pr.z.lm)
> sum((lpsa-test.fitted)^2)
[1] 17.58988
> .Last.value/30
[1] 0.5863292
> sum((lpsa-2.47951205)^2)/30
[1] 1.052896

#this can be done better using predict.lm
```

Notes on example:

estimated coefficients in Table 3.2 of HTF

Each $\underline{x}_k$ was centered and standardized (on the full data set) to have mean 0, var 1

this makes interpretation very difficult, although emphasis here is on prediction

standardizing *x*'s is needed for subset selection methods in Section 3.4

on the training data, $\hat{\beta}$ has the smallest variance among all *unbiased* estimators of β

two questions: Can we do better on training data by allowing biased estimators?

Does this lead to better prediction error on test data?

Geometric view of least squares fitting

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

$$\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$$

$\hat{\beta}_p$ can be obtained by a series of regressions (projections) as outlined in algorithm 3.1 on p.52

regress x_1 on 1, get coefficient $\hat{\gamma}_{01}$, form residual $z_1 = x_1 - \hat{x}_1$

regress x_2 on 1, z_1, get coefs $\hat{\gamma}_{02}, \hat{\gamma}_{12}$, form residual $z_2 = x_2 - \hat{\gamma}_{02}1 - \hat{\gamma}_{12} z_1$

$\vdots$

regress x_p on z_{p-1} to get $z_p - x_p - \hat{x}_p$

regress y on z_p to get $\hat{\beta}_p$

obtain each $\hat{\beta}_j$ by a similar process, hence interpretation at top of p.53

note effect of correlations among columns of X

illustration on prostate training data