Administration

- Please check web page regularly for updates

  `http://www.utstat.utoronto.ca/reid/414S10.html`
- Blackboard is used only for email and grades
- You should by now have have `R` on your PC, or be planning to go your own route re software
- Printing slides from web page (Acrobat: page setup (horizontal); expand to fit)
- Thursday: TA Li Li will answer your questions about `R`
- Project: check course information handout from last week
- More data sets: see *Applied Statistics* (Journal of the Royal Statistical Society, Series C); articles may have links to data sets used, at
- `http://www.blackwellpublishing.com/rss/default.htm`
- e.g. “Spatiotemporal smoothing and sulphur dioxide trends over Europe” by Bowman et al (December, 2009)

Polynomial regression

- See R code from last week
-

```
lm10 = lm ( y ~ x + I(x^2) + I(x^3) + ... + I(x^10) )
```

- i.e. $E(y) = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_{10} x^{10}$ $\quad y = X\beta + \epsilon$
- `fm10 = lm ( y ~ poly(x, 10) )`
- $E(y) = \alpha_0 + \alpha_1 P_1(x) + \cdots + \alpha_{10} P_{10}(x)$ $\quad y = X^* \alpha + \epsilon$
-

$$P_j(x) = a_{0j} + a_{1j} x + a_{2j} x^2 + \cdots + a_{jj} x^j$$

- coefficients a_{0j}, a_{1j}, etc. to be determined
- so that columns of X^* are orthogonal

... polynomial regression

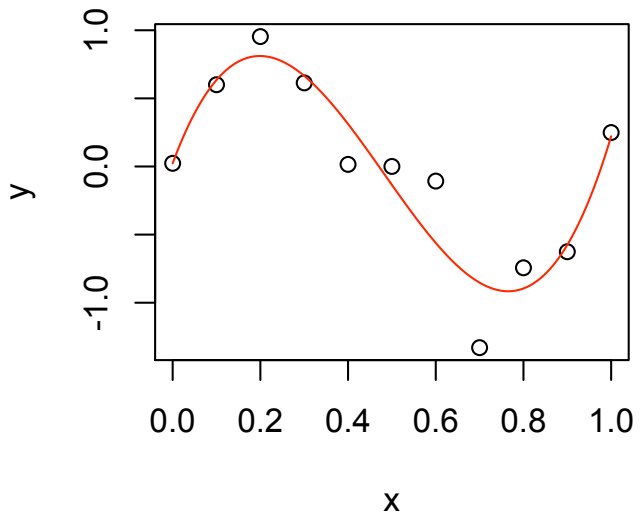
```
>x
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> model.matrix(lm10)
   (Intercept)   x I(x^2) I(x^3) I(x^4)  I(x^5)   I(x^6) ...
1            1 0.0   0.00  0.000 0.0000 0.00000 0.000000
2            1 0.1   0.01  0.001 0.0001 0.00001 0.000001
3            1 0.2   0.04  0.008 0.0016 0.00032 0.000064
4            1 0.3   0.09  0.027 0.0081 0.00243 0.000729
5            1 0.4   0.16  0.064 0.0256 0.01024 0.004096
6            1 0.5   0.25  0.125 0.0625 0.03125 0.015625
7            1 0.6   0.36  0.216 0.1296 0.07776 0.046656
8            1 0.7   0.49  0.343 0.2401 0.16807 0.117649
9            1 0.8   0.64  0.512 0.4096 0.32768 0.262144
10           1 0.9   0.81  0.729 0.6561 0.59049 0.531441
11           1 1.0   1.00  1.000 1.0000 1.00000 1.000000

> model.matrix(fm10)
   (Intercept) poly(x, degree)1 poly(x, degree)2 poly(x, degree)3 ...
1            1    -4.767313e-01       0.51209156    -4.580286e-01
2            1    -3.813850e-01       0.20483662     9.160572e-02
3            1    -2.860388e-01      -0.03413944     3.358876e-01
4            1    -1.906925e-01      -0.20483662     3.511553e-01
5            1    -9.534626e-02      -0.30725493     2.137467e-01
6            1    -1.323195e-17      -0.34139437     6.621275e-17
7            1     9.534626e-02      -0.30725493    -2.137467e-01
8            1     1.906925e-01      -0.20483662    -3.511553e-01
9            1     2.860388e-01      -0.03413944    -3.358876e-01
10           1     3.813850e-01       0.20483662    -9.160572e-02
11           1     4.767313e-01       0.51209156     4.580286e-01
```

... polynomial regression

- same $\hat{y} = X\hat{\beta} = X^*\hat{\alpha}$
-

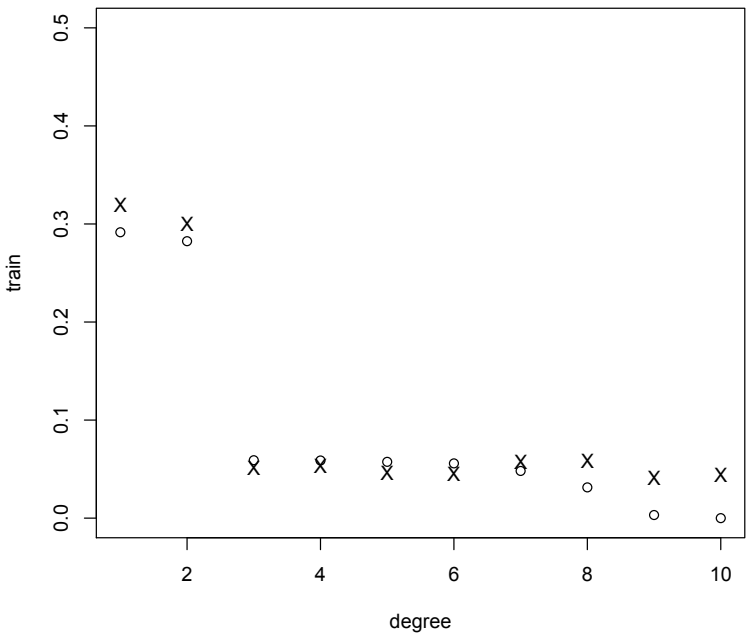
```
> lm10$fitted.values
          1           2           3           4           5
0.023170726 0.600194667 0.953585931 0.613647096 0.015437840
          6           7           8           9          10
0.000285572 -0.107110688 -1.329937671 -0.743709343 -0.625900416
         11
0.249038261
```

```
> fm$fitted.values
          1           2           3           4           5
0.023170726 0.600194667 0.953585931 0.613647096 0.015437840
          6           7           8           9          10
0.000285572 -0.107110688 -1.329937671 -0.743709343 -0.625900416
         11
0.249038261
```



Bias-variance trade-off

- choosing the degree of the polynomial equivalent to choosing a model
- choosing a high degree polynomial captures the data well
- but predicts new values of y poorly
- choosing a low degree polynomial captures the data less well but may give better predictions



... bias-variance trade-off

- true model $y = f(X) + \epsilon$; f is unknown
- fitted model $\hat{y} = \hat{f}(X)$
- measure error by least squares $\{\hat{y} - f(X)\}^2$
-

$$\begin{aligned} E_{\mathcal{D}}\{\hat{y} - f(X)\}^2 &= E_{\mathcal{D}}\{\hat{y} - E_{\mathcal{D}}(\hat{y}) + E_{\mathcal{D}}(\hat{y}) - f(X)\}^2 \\ &= E_{\mathcal{D}}\{\hat{y} - E_{\mathcal{D}}(\hat{y})\}^2 + \{E_{\mathcal{D}}(\hat{y}) - f(X)\}^2 \end{aligned}$$

- $E_{\mathcal{D}}$ over "all possible data sets of size n" (with same X)
- variance: 'sensitivity of $\hat{y}$ to the observed $\mathcal{D}$
- squared bias: 'systematic error in our prediction'
- "No Free Lunch": can't simultaneously drive variance and squared bias to zero
- NFL more general than this – see, e.g., Clarke et al. (2009)

Linear Regression HTF §3.1, 3.2

- inputs $X = (X_1, \ldots, X_p)$: attributes, features, predictors, covariates
- output $Y \in R$: response (hence supervised learning)
- linear model $E(Y \mid X) = \beta_0 + \Sigma_{j=1}^{p} X_j \beta_j = f(X)$
- linear in β: X's can be quantitative, transformed, derived, basis expansions, dummy variables, interactions examples
- data $(x_i, y_i), i = 1, \ldots N$: instances
- $x_i^T = (x_{i1}, \ldots, x_{ip})$
- usual model for data:
  $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i, i = 1, \ldots, N$
  not assumed by HTF at this point: see p.45 "... intuitively satisfying..."
- implicit assumption for least squares: 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
- under the linear model $f(X)$ 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

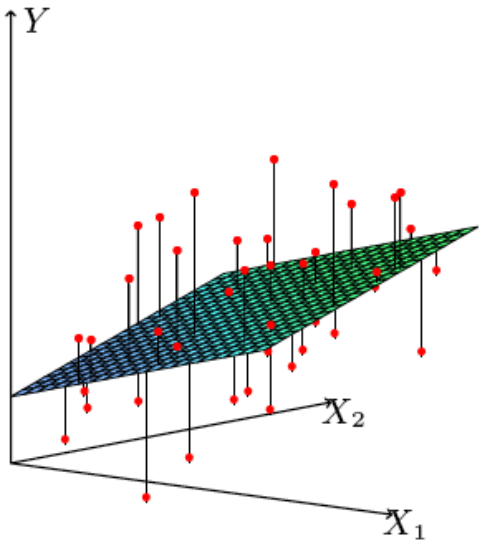


Figure 3.1: *Linear least squares fitting with* $X \in \mathbb{R}^2$.

Least squares

- $\min_\beta \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2$
-

$$\begin{aligned} RSS(\beta) &= \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2 \\ &= (y - X\beta)^T (y - X\beta) \end{aligned}$$

- X is $N \times (p+1)$: $X = (1 \quad x_1 \quad \ldots \quad x_p)$
- 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

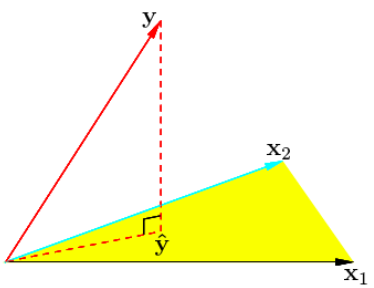


Figure 3.2: *The N-dimensional geometry of least squares regression with two predictors. The outcome*

Least squares fits

- fitted values (for training data)
  $\hat{y} = X\hat{\beta} = X(X^T X)^{-1} X^T y = Hy$
- $\hat{y} = Hy$: H is a *projection matrix*, projecting $y (\in R^N)$ onto the column space of X

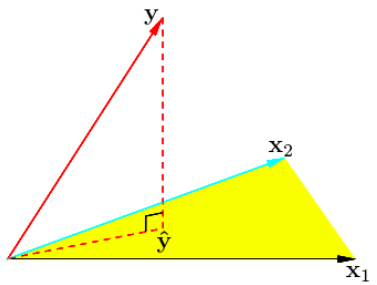


Figure 3.2: *The N-dimensional geometry of least squares regression with two predictors. The outcome vector* y *is orthogonally projected onto the hyperplane*

- if $X^T X$ 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

... LS fits

- what to do if $X^T X$ not invertible
- 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
- also guaranteed that $X^T X$ not invertible if $p > N$: smoothing or filtering (later)
- if $X^T X$ is only nearly singular, ...

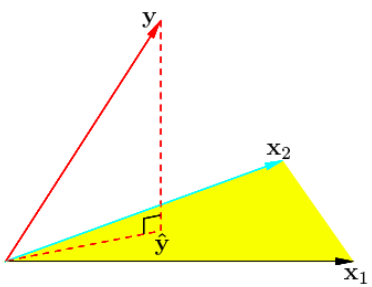


Figure 3.2: *The N-dimensional geometry of least squares regression with two predictors. The outcome vector* $\mathbf{y}$ *is orthogonally projected onto the hyperplane*

Predictions at a new value of the inputs

- sorting through some notation in HTF
- new set of inputs $(1 \ : \ x_0)^T$ (just above (3.7)): $x_0 = (x_{01}, \ldots, x_{0p})^T$
- $\hat{y}(x_0) = \hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_{01} + \cdots + \hat{\beta}_p x_{0p}$
- At the end of §3.2.2, $x_0^T = (1, x_{01}, \ldots, x_{0p})$ (also in §2.3.1, after (2.6))

- more important: Don't predict outside the range of the training data!!
- unless ...

- In the paragraph following (3.7), $\mathbf{x}_0 = \underbrace{(1, 1, \ldots, 1)^T}_{N}$
- we will also need $\mathbf{x}_j = (x_{1j}, \ldots, x_{Nj})^T$ the N observations on the jth input

Inference

- model $y = X\beta + \epsilon$
- assumption $\epsilon \sim \quad (0, \sigma^2 I)$
- note that $\hat{\beta}$ is linear in y
- $\text{var}(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$ (under the assumptions)
- $\hat{\sigma}^2 = \frac{1}{N-(p+1)} RSS(\hat{\beta})$
- $= \frac{1}{N-(p+1)} (y - \hat{y})^T (y - \hat{y})$
- $= \frac{1}{N-(p+1)} (y - X\hat{\beta})^T (y - X\hat{\beta})$
- $= \frac{1}{N-(p+1)} y^T (I - H) y$
- $E\hat{\sigma}^2 = \sigma^2$
- $N - p - 1$ called degrees of freedom for the residual

.. inference

- if $\epsilon \sim N(0, \sigma^2 I)$ then
- $\hat{\beta} \sim N(\beta, \sigma^2 (X^T X)^{-1})$
- $RSS(\hat{\beta})/\sigma^2 \sim \chi^2_{(N-p-1)}$
- $$\frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}(X^T X)^{-1}_{jj}} \sim t_{N-p-1}$$

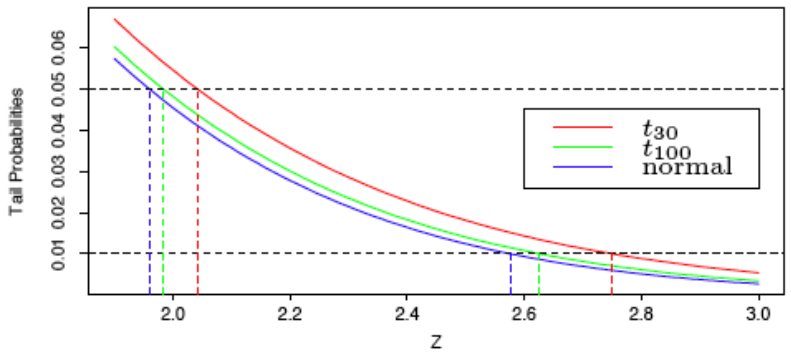


Figure 3.3: *The tail probabilities* $\Pr(|Z| > z)$ *for three*

confidence intervals; tests of $\beta_j = 0$

Example: Prostate data

	lcavol	lweight	age	lbph	svi	lcp	gleason	pgg45	lpsa	train
1	-0.579818495	2.769459	50	-1.38629436	0	-1.38629436	6	0	-0.4307829	T
2	-0.994252273	3.319626	58	-1.38629436	0	-1.38629436	6	0	-0.1625189	T
3	-0.510825624	2.691243	74	-1.38629436	0	-1.38629436	7	20	-0.1625189	T
4	-1.203972804	3.282789	58	-1.38629436	0	-1.38629436	6	0	-0.1625189	T
5	0.751416089	3.432373	62	-1.38629436	0	-1.38629436	6	0	0.3715636	T
6	-1.049822124	3.228826	50	-1.38629436	0	-1.38629436	6	0	0.7654678	T
7	0.737164066	3.473518	64	0.61518564	0	-1.38629436	6	0	0.7654678	F
8	0.693147181	3.539509	58	1.53686722	0	-1.38629436	6	0	0.8544153	T
9	-0.776528789	3.539509	47	-1.38629436	0	-1.38629436	6	0	1.0473190	F
10	0.223143551	3.244544	63	-1.38629436	0	-1.38629436	6	0	1.0473190	F
11	0.254642218	3.604138	65	-1.38629436	0	-1.38629436	6	0	1.2669476	T
12	-1.347073648	3.598681	63	1.26694760	0	-1.38629436	6	0	1.2669476	T
13	1.613429934	3.022861	63	-1.38629436	0	-0.59783700	7	30	1.2669476	T
14	1.477048724	2.998229	67	-1.38629436	0	-1.38629436	7	5	1.3480731	T
15	1.205970807	3.442019	57	-1.38629436	0	-0.43078292	7	5	1.3987169	F
16	1.541159072	3.061052	66	-1.38629436	0	-1.38629436	6	0	1.4469190	T
32	0.182321557	3.804438	65	1.70474809	0	-1.38629436	6	0	2.0082140	F

...

p. 3 of 2nd edition (footnote)
see web page

... prostate data

```
> library(ElemStatLearn)
> data(prostate)
> prostate[32,]
      lcavol lweight age     lbph svi       lcp gleason pgg45
32 0.1823216 6.10758  65 1.704748   0 -1.386294       6     0
       lpsa train
32 2.008214 FALSE
```

```
> prostate[32,2] = 3.804438  # should save this in a local directory for later
```

```
> save(prostate,file="myfile")
> rm(prostate)
> load("myfile")
> ls()
[1] "pr.std"   "prostate"
> prostate[32,]
      lcavol  lweight age     lbph svi       lcp gleason pgg45
32 0.1823216 3.804438  65 1.704748   0 -1.386294       6     0
       lpsa train
32 2.008214 FALSE
> attach(prostate)

> ## standardize the data (but not the lpsa score and the training indicator)

> pr.std = data.frame(cbind(apply(prostate[,1:8],2,scale)),lpsa,train)

> pr.lm = lm(lpsa~.-train, subset=train, data=pr.std)
```

... prostate data

```
> options(digits=4)

> prostate[1:2,]
   lcavol lweight age   lbph svi    lcp gleason pgg45    lpsa train
1 -0.5798   2.769  50 -1.386   0 -1.386       6     0 -0.4308  TRUE
2 -0.9943   3.320  58 -1.386   0 -1.386       6     0 -0.1625  TRUE
> pr.std[1:2,]
  lcavol lweight     age   lbph     svi     lcp gleason   pgg45    lpsa train
1 -1.637  -2.006 -1.8624 -1.025 -0.5229 -0.8632  -1.042 -0.8645 -0.4308  TRUE
2 -1.989  -0.722 -0.7879 -1.025 -0.5229 -0.8632  -1.042 -0.8645 -0.1625  TRUE

> summary(pr.lm)

Call:
lm(formula = lpsa ~ . - train, data = pr.std, subset = train ==
    TRUE)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6487 -0.3415 -0.0542  0.4494  1.4868

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.4649     0.0893   27.60  < 2e-16 ***
lcavol        0.6795     0.1266    5.37  1.5e-06 ***
lweight       0.2631     0.0956    2.75   0.0079 **
age          -0.1415     0.1013   -1.40   0.1681
lbph          0.2101     0.1022    2.06   0.0443 *
svi           0.3052     0.1236    2.47   0.0165 *
lcp          -0.2885     0.1545   -1.87   0.0670 .
gleason      -0.0213     0.1452   -0.15   0.8839
pgg45         0.2670     0.1536    1.74   0.0875 .
```

Notes on example

- estimated coefficients in Table 3.2 of HTF
- Each $\mathbf{x}_k$ was centered and standardized to have mean 0, variance 1: on the full data set
- interpretation of coefficients?
- categorical coefficients?
- standardizing x's is needed for subset selection methods in §3.4
- §3.2.2 – on the training data, $\hat{\beta}$ has the smallest variance among all unbiased estimators of β
- Model selection: do we need all the features
- Would fewer lead to better prediction error on test data?

Aside: re Homework and Project

Acceptable:

TABLE 3.2. *Linear model fit to the prostate cancer data. The Z score is the coefficient divided by its standard error (3.12). Roughly a Z score larger than two in absolute value is significantly nonzero at the $p = 0.05$ level.*

Term	Coefficient	Std. Error	Z Score
`Intercept`	2.46	0.09	27.60
`lcavol`	0.68	0.13	5.37
`lweight`	0.26	0.10	2.75
`age`	−0.14	0.10	−1.40
`lbph`	0.21	0.10	2.06
`svi`	0.31	0.12	2.47
`lcp`	−0.29	0.15	−1.87
`gleason`	−0.02	0.15	−0.15
`pgg45`	0.27	0.15	1.74

Aside: re Homework and Project

Not:

```
> summary(pr.lm)

Call:
lm(formula = lpsa ~ . - train, data = pr.std, subset = train ==
    TRUE)

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)  2.46493    0.08931  27.598  < 2e-16 ***
lcavol       0.67953    0.12663   5.366 1.47e-06 ***
lweight      0.26305    0.09563   2.751  0.00792 **
age         -0.14146    0.10134  -1.396  0.16806
lbph         0.21015    0.10222   2.056  0.04431 *
svi          0.30520    0.12360   2.469  0.01651 *
lcp         -0.28849    0.15453  -1.867  0.06697 .
gleason     -0.02131    0.14525  -0.147  0.88389
pgg45        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
```

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.54

  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 $z_{p-1}, z_{p-2}, \ldots, z_1, 1$ 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.55
- note effect of correlations among columns of X
- illustration on prostate training data

Mean squared error of prediction in linear models

Let $\tilde{\beta} = \tilde{\beta}(y)$ be a competing estimator of β (not $\hat{\beta}$, the LS estimator). Using $\tilde{\beta}$ for prediction would give $\tilde{y}_0 = x_0^T \tilde{\beta}$, where $x_0^T = (1, x_{01}, \ldots, x_{0p})$ is the new value of the inputs. The expected prediction error is

$$
\begin{aligned}
E(\tilde{y}_0 - y_0)^2 &= E(x_0^T \tilde{\beta} - y_0)^2 \quad \text{(by definition)} \\
&= E(y_0 - x_0^T \beta + x_0^T \beta - x_0^T \tilde{\beta})^2 \\
&= \text{var}(y_0) + E(x_0^T \tilde{\beta} - x_0^T \beta)^2 \quad \text{(why is cross prod 0?)} \\
&= \sigma^2 + E\{x_0^T (\tilde{\beta} - \beta)(\tilde{\beta} - \beta)^T x_0\} \\
&= \sigma^2 + x_0^T E\{(\tilde{\beta} - \beta)(\tilde{\beta} - \beta)^T\} x_0 \\
&= \sigma^2 + x_0^T E\{(\tilde{\beta} - E\tilde{\beta} + E\tilde{\beta} - \beta)(\tilde{\beta} - E\tilde{\beta} + E\tilde{\beta} - \beta) x_0 \\
&= \sigma^2 + x_0^T [E\{(\tilde{\beta} - E\tilde{\beta})(\tilde{\beta} - E\tilde{\beta})^T\} + (E\tilde{\beta} - \beta)(E\tilde{\beta} - \beta)^T] x_0 \\
&= \sigma^2 + x_0^T \{\text{cov}(\tilde{\beta}) + \text{bias}^2(\tilde{\beta})\} x_0
\end{aligned}
$$

The first term, σ^2, is unavoidable. The next two terms together are the Mean Squared Error (MSE) of the prediction $\tilde{y}_0$, and are shown here to be a function of x_0 and the MSE of $\tilde{\beta}$. If $\tilde{\beta}$ is unbiased, i.e. $E\tilde{\beta} = \beta$, then we only need to worry about the covariance terms. A key question is whether by allowing possibly biased estimators, we can have a smaller covariance term, and in sum, reduce the MSE of prediction.

Next week

- MSE and prediction error (eq. (3.21) and (3.22))
- Algorithm 3.1 and the QR decomposition of X (§3.2.3)
- Subset selection (§3.3)
- Shrinkage methods: ridge regression and lasso (§3.4.2)
- Principal components (§3.4.1)