STAT 5630, Fall 2019

Linear Models for Regression

Xiwei Tang, Ph.D. <xt4yj@virginia.edu>

University of Virginia
September 10, 2019

Outline

- Linear Models for Regression

- Model Selection Criteria

Linear Models for Regression

Regression Models

- Observe a collection of i.i.d. training data

$$\mathcal{D}_n = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$$

  where each $\boldsymbol{x}_i$ is a p dimensional vector (prediction variables, covariates, features, inputs), i.e.

$$\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^\mathsf{T}$$

  and $y_i \in \mathbb{R}$ is a continuous response (outcome, output).
- Denote $\boldsymbol{X}_{\cdot j}$ is a n dimensional vector of the jth feature, i.e.

$$\boldsymbol{X}_{\cdot j} = (x_{1j}, x_{2j}, \ldots, x_{nj})^\mathsf{T}$$

- The design matrix $\mathbf{X}$ is $n \times p$ dimensional,

$$\mathbf{X} = (\boldsymbol{X}_{\cdot 1}, \boldsymbol{X}_{\cdot 2}, \ldots, \boldsymbol{X}_{\cdot p})$$

Loss and Risk functions

- To estimate $f(X)$, need to define a criterion for model fitting.
- A loss function L measures discrepancies between Y and $f(X)$.
- A commonly used loss function is the squared error loss:

$$L\big(Y, f(X)\big) = \big(Y - f(X)\big)^2.$$

- Risk is the expected loss over the entire population

$$\mathsf{R}(f) = \mathsf{E}\left[L\big(Y, f(X)\big)\right] = \mathsf{E}\left[\big(Y - f(X)\big)^2\right].$$

Minimizing the Empirical Risk

- With the training data $\mathcal{D}_n$, we can solve for a $f(x)$ such that the empirical risk is minimized, i.e., we replace the expectation with the average over n training samples:

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i - f(x_i)).$$

- Using the squared error, we search for a function $\widehat{f}$ to minimize the empirical risk on the training dataset

$$\widehat{f} = \arg\min_{f\in\mathcal{F}} \quad \sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2.$$

- $\mathcal{F}$ is a space of models that we consider.

Minimizing the Empirical Risk

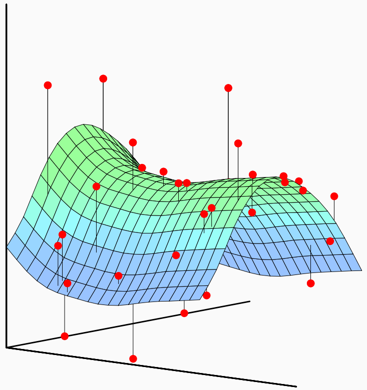


FIGURE 2.10. *Least squares fitting of a function of two inputs. The parameters of $f_\theta(x)$ are chosen so as to minimize the sum-of-squared vertical errors.*

Linear Regression

- To estimate f, we pose some restrictions/structures
- A linear regression model describes the dependence between X and Y by

$$\begin{aligned} Y =& X^{\mathsf{T}}\boldsymbol{\beta} + \epsilon \\ =& \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon \end{aligned}$$

  where we can set $X_1 = 1$ as the intercept if necessary, and ϵ is an independent error term.
- Given the training data $\mathcal{D}_n$, we express the regression model in the matrix form

$$\mathbf{y}_{n\times 1} = \mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1} + \mathbf{e}_{n\times 1}$$

  where $\mathbf{X}_{n\times p}$ is the design matrix with each row representing an input vector from one subject.

Linear Regression

- Now, the estimation of f boils down to the estimating of $\boldsymbol{\beta}$
- By our previous definition of the empirical risk, we try to solve for $\boldsymbol{\beta}$ that minimizing the residual sum of squares (RSS)

$$
\begin{aligned}
\text{RSS} =& \sum_{i=1}^{n} \left(y_i - x_{i1}\beta_1 - \cdots , x_{ip}\beta_p \right)^2 \\
=& \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \\
=& \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)^{\mathsf{T}} \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)
\end{aligned}
$$

- The ordinary least squares estimator (OLS) is

$$
\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)^{\mathsf{T}} \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)
$$

Linear Regression

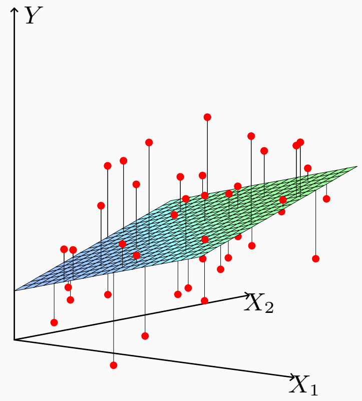


FIGURE 3.1. *Linear least squares fitting with $X \in \mathbb{R}^2$. We seek the linear function of X that minimizes the sum of squared residuals from Y.*

Estimate β

- To estimate $\boldsymbol{\beta}$, we set the derivative equal to 0

$$\begin{aligned} \frac{\partial \text{RSS}}{\partial \boldsymbol{\beta}} &= -2\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0 \\ \implies \quad \mathbf{X}^\top\mathbf{y} &= \mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} \end{aligned}$$

  which is commonly known as the normal equation.

- We then have, if $\mathbf{X}^\top\mathbf{X}$ is invertible,

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}.$$

- $\mathbf{X}$ full rank $\iff \mathbf{X}^\top\mathbf{X}$ invertible

Hat Matrix

- The fitted values (prediction at the observed data points) are

$$\widehat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \doteq \mathbf{H}_{n\times n}\mathbf{y}$$

- $\mathbf{H}$ (“hat matrix”) is a project matrix
  - symmetric: $\mathbf{H}^\top = \mathbf{H}$
  - idempotent: $\mathbf{H}\mathbf{H} = \mathbf{H}$

- The residual $\mathbf{r}_{n\times 1} = \widehat{\mathbf{e}} = \mathbf{y} - \widehat{\mathbf{y}} = (\mathbf{I} - \mathbf{H})\mathbf{y}$
- $\mathbf{r}$ can be used to estimate the error variance

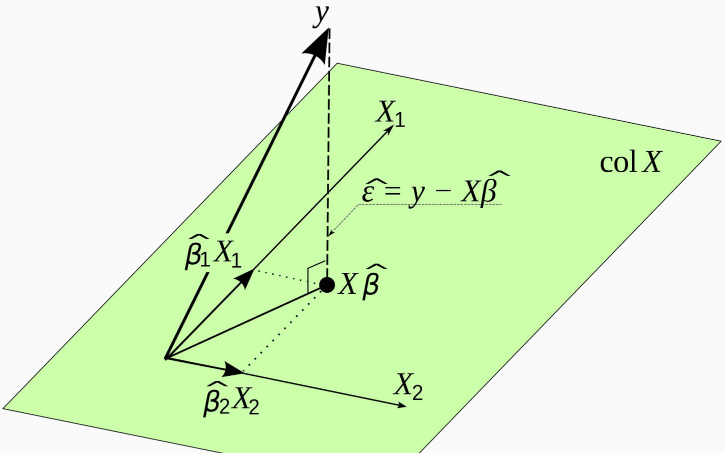
$$\widehat{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^{n} r_i^2 = \frac{\text{RSS}}{n-p}$$

Vector Space Interpretation

- The essence of LS is to decompose the data vector $\mathbf{y}$ into two orthogonal vectors

$$\mathbf{y} = \widehat{\mathbf{y}} + \mathbf{r}$$

- Note that the normal equations implies that $\mathbf{r}$ is orthogonal to each column of $\mathbf{X}$, i.e., $\mathbf{X}^{\top}\mathbf{r} = \mathbf{0}$



Statistical Properties of $\widehat{\beta}$

- We often assume that the samples are generated from the model

$$Y = X^{\mathsf{T}}\boldsymbol{\beta} + \epsilon,$$

  where the errors ϵ_i are i.i.d. with $\mathsf{E}(\epsilon_i) = 0$ and $\mathsf{Var}(\epsilon_i) = \sigma^2$
- Then $\widehat{\boldsymbol{\beta}}$ is unbiased: $\mathsf{E}(\widehat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$
- Variance-covariance

$$\mathsf{Var}(\widehat{\boldsymbol{\beta}}) = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\sigma^2$$

- By the Gauss-Markov Theorem, $\widehat{\boldsymbol{\beta}}$ is the best linear unbiased estimator (BLUE)
- But it is not necessarily the minimum variance unbiased estimator (MVUE), unless the errors are generated from the Gaussian distribution

Two Computational Algorithms

- Consider the normal equation:

$$\mathbf{X}^\mathsf{T}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^\mathsf{T}\mathbf{y}$$

  We would like to avoid computing $(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$ directly.

- QR-decomposition of $\mathbf{X}^\mathsf{T}\mathbf{X}$
  - $\mathbf{X}^\mathsf{T}\mathbf{X} = \mathbf{Q}\mathbf{R}$ where $\mathbf{Q}$ is orthonormal and $\mathbf{R}$ is upper triangular

- Cholesky decomposition of $\mathbf{X}^\mathsf{T}\mathbf{X}$
  - $\mathbf{X}^\mathsf{T}\mathbf{X} = \mathbf{R}\mathbf{R}^\mathsf{T}$, where $\mathbf{R}$ is lower triangular

QR Decomposition

- The QR-decomposition is related to the Gram-Schmidt procedure (see appendix):

$$\mathbf{X}^\mathsf{T}\mathbf{X} = \mathbf{Z}\Gamma$$

  where Γ is an upper triangular matrix and $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_p]$ has orthogonal columns

- Standardize $\mathbf{Z}$ using $\mathbf{D} = \text{diag}\{\|\mathbf{z}_1\|, \ldots, \|\mathbf{z}_p\|\}$,

$$\mathbf{X}^\mathsf{T}\mathbf{X} = \mathbf{Z}\Gamma = \mathbf{Z}\mathbf{D}^{-1}\mathbf{D}\Gamma = \mathbf{Q}\mathbf{R}$$

  with $\mathbf{Q} = \mathbf{Z}\mathbf{D}^{-1}$ and $\mathbf{R} = \mathbf{D}\Gamma$

- The normal equation reduces to solving the linear system

$$\mathbf{R}\boldsymbol{\beta} = \mathbf{Q}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{y}$$

Cholesky Decomposition

- For any positive definite square matrix $\mathbf{A}$, we have

$$\mathbf{A} = \mathbf{R}\mathbf{R}^\mathsf{T}$$

  where $\mathbf{R}$ is a lower triangular matrix of full rank

- Factoring $\mathbf{X}^\mathsf{T}\mathbf{X} = \mathbf{R}\mathbf{R}^\mathsf{T}$
- Solve the triangular system $\mathbf{R}\mathbf{w} = \mathbf{X}^\mathsf{T}\mathbf{y}$
- Solve the triangular system $\mathbf{R}^\mathsf{T}\boldsymbol{\beta} = \mathbf{w}$ for $\boldsymbol{\beta}$

Variable Selection

Dealing with large p

- In many applications nowadays, we have many explanatory variables, i.e., p is large or even $p \gg n$.
  - There are more than 20,000 human protein-coding genes
  - About 10 million single nucleotide polymorphisms (SNPs)
  - Number of subjects, n, is usually in hundreds or thousands
- In some applications, the key question is to identify a subset of X variables that are most relevant to Y
- Inadequate to look at marginal effects (P-values)

Training vs. Testing error

- Training data $\mathcal{D}_n = \{x_i, y_i\}_{i=1}^n$
- Suppose $\{x_i, y_i^*\}_{i=1}^n$ is an independent (imaginary) testing dataset collected at the same location x_i's (aka, in-sample prediction
- Assume that the data are indeed from a linear model

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{e} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$$
$$\mathbf{y}^* = \boldsymbol{\mu} + \mathbf{e}^* = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}^*$$

  where both $\mathbf{y}$ and $\mathbf{y}^*$ are $n \times 1$ response vectors, $\mathbf{e}$ and $\mathbf{e}^*$ are i.i.d. error terms with mean 0 and variance σ^2.
- The true model is indeed linear

Training vs. Testing error

$$\begin{aligned}
\mathsf{E}[\text{Test Err}] &= \mathsf{E}\|\mathbf{y}^* - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2 \\
&= \mathsf{E}\|(\mathbf{y}^* - \mathbf{X}\boldsymbol{\beta}) + (\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\widehat{\boldsymbol{\beta}})\|^2 \\
&= \mathsf{E}\|\mathbf{y}^* - \boldsymbol{\mu}\|^2 + \mathsf{E}\|\mathbf{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|^2 \\
&= \mathsf{E}\|\mathbf{e}^*\|^2 + \text{Trace}\big(\mathbf{X}^\mathsf{T}\mathbf{X}\text{Cov}(\widehat{\boldsymbol{\beta}})\big) \\
&= n\sigma^2 + p\sigma^2
\end{aligned}$$

$$\begin{aligned}
\mathsf{E}[\text{Train Err}] &= \mathsf{E}\|\mathbf{y} - \widehat{\mathbf{y}}\|^2 = \mathsf{E}\|(\mathbf{I} - \mathbf{H})\mathbf{y}\|^2 \\
&= \mathsf{E}\|(\mathbf{I} - \mathbf{H})\mathbf{e}\|^2 \\
&= \text{Trace}\big((\mathbf{I} - \mathbf{H})^\mathsf{T}(\mathbf{I} - \mathbf{H})\text{Cov}(\mathbf{e})\big) \\
&= (n - p)\sigma^2
\end{aligned}$$

So the testing error increase with p and training error decreases with p. When p gets large, this is a big trouble...

Variable Selection

- Hence, it is necessary to select a set of relevant variables, especially when p is large.
- Variable selection may improve
  - Prediction accuracy
  - Interpretability
- This is a difficult task
  - No natural ordering of importance for the variables
  - The role of a variable needs be measured conditioning on others, high correlation causes trouble
  - It is essential to check all possible combinations, however, this may be computationally expansive

Model Selection Criteria

- Model selection is usually done in the following way
  1. Give each model a score
  2. Design an algorithm to find the model with the best (smallest) score
- The score of a model fitting takes the the form

Goodness-of-fit + Complexity-Penalty

- The first term will decrease as the model gets more complicated (recall 1NN)
- From last week's lecture, the second term increases with the number of predictor variables, which prefers "smaller" model

Model Selection Criteria

- Popular choices of scores:
    - Mallows' C_p (Mallows 1973): RSS + $2\widehat{\sigma}^2_{\text{full}} \cdot p$
    - AIC (Akaike 1970): -2 Log-likelihood $+ 2 \cdot p$
    - BIC (Schwarz, 1978): -2 Log-likelihood $+ \log n \cdot p$
- When n is large, adding an additional predictor costs a lot more in BIC than AIC (or C_p). So AIC tends to pick a larger model than BIC.
- C_p performs similarly to AIC.

Stepwise Regression

- Greedy algorithms: fast, but only return a local optimal solution (which might be good enough in practice).
  - Backward: start with the full model and sequentially delete predictors until the score does not improve.
  - Forward: start with the null model and sequentially add predictors until the score does not improve.
  - Stepwise: consider both deleting and adding one predictor at each stage.

Best Subset Selection

- Best subset selection is a level-wise search algorithm, which returns the global optimal solution for a given model size.
- Only feasible for p not very large (< 50)
- Algorithm:
  1. For each $k = 1, \ldots, p$, check 2^k possible combinations, and find the model with smallest RSS
     - The penalty term is the same for models with the same size
  2. To choose the best k, use model selection criteria
- Note: if RSS$(X_1, X_2) <$ RSS(X_3, X_4, X_5, X_6) then we do not need to visit any size 2 or 3 sub-models of (X_3, X_4, X_5, X_6), which can be leaped over.
- Implemented in $\mathbb{R}$ contributed package "*leaps*", using the leaps and bounds algorithm (Furnival and Wilson, 1974)

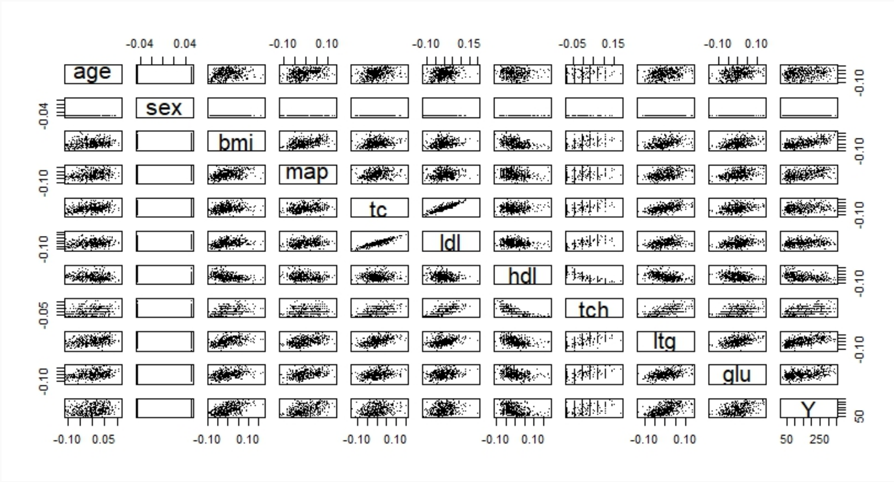
Diabetes Data Analysis

- The Diabetes Data (Efron et al, 2004) contains ten baseline variables from 442 subjects: age, sex, body mass index, average blood pressure, and six blood serum measurements
- The goal is to model a quantitative measure of disease progression one year after baseline
- Data can be loaded from the R package "lars"
- We perform model selections on this dataset (see R code from course material)

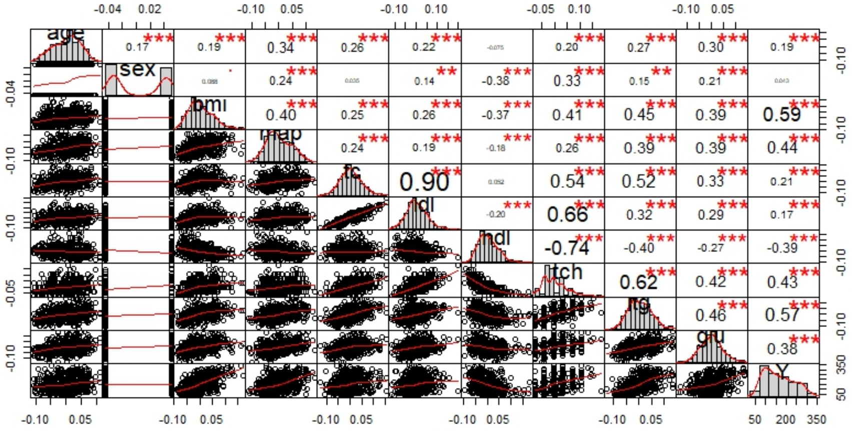
Diabetes Data

```r
# Get the Diabetes Data from the package "lars"
install.packages("lars")
library(lars)
data(diabetes)
diab = data.frame(cbind(diabetes$x, "Y" = diabetes$y))

#some data summarization
pairs(diab, pch=".")  # produce pair-wise scatter plots. Caution
    : this is a big figure.

# a fancier plot, requires another package
install.packages("PerformanceAnalytics")
library(PerformanceAnalytics)
suppressWarnings(chart.Correlation(diab, col = "purple", pch = "
    *"))
```

Diabetes Data



Diabetes Data



Diabetes Data

```
# Run linear regression
lmfit=lm(Y~., data=diab)
names(lmfit)  # What have been returned by "lm"?
lmfit$coef    # 11 regression coefficients

summary(lmfit)
round(summary(lmfit)$coef,3)  # coefficients and the p-values

>
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   152.133      2.576  59.061    0.000
age           -10.012     59.749  -0.168    0.867
sex          -239.819     61.222  -3.917    0.000
bmi           519.840     66.534   7.813    0.000
map           324.390     65.422   4.958    0.000
tc           -792.184    416.684  -1.901    0.058
ldl           476.746    339.035   1.406    0.160
hdl           101.045    212.533   0.475    0.635
tch           177.064    161.476   1.097    0.273
ltg           751.279    171.902   4.370    0.000
glu            67.625     65.984   1.025    0.306
```

Diabetes Data

```
# Step-wise Variable Selection
?step # or help(step)
step(lmfit, direction="both")                   # AIC
step(lmfit, direction="both", trace=0)    # do not print
     intermediate results

step(lmfit, direction="backward")
step(lm(Y~1, data=diab), scope=list(upper=lmfit, lower=~1),
     direction="forward")

#n = nrow(diab)
step(lmfit, direction="both", k=log(n))  # BIC (the default
     value for k=2, which corresponds to AIC)

>Call:
lm(formula = Y ~ sex + bmi + map + tc + ldl + ltg, data = diab)
Coefficients:
(Intercept)   sex      bmi      map      tc       ldl     ltg
152.1        -226.5   529.9    327.2    -757.9   538.6   804.2
```

Diabetes Data

```
# Best subset model selection (Cp, AIC, and BIC): leaps
install.packages("leaps")
library(leaps)
RSSleaps=regsubsets(as.matrix(diab[,-11]),diab[,11], nvmax=10)
summary(RSSleaps,matrix=T)
>1 subsets of each size up to 10
Selection Algorithm: exhaustive
          age sex bmi map tc  ldl hdl tch ltg glu
1  ( 1 ) " " " " "*" " " " " " " " " " " " " " "
2  ( 1 ) " " " " "*" " " " " " " " " " " "*" " "
3  ( 1 ) " " " " "*" "*" " " " " " " " " "*" " "
4  ( 1 ) " " " " "*" "*" "*" " " " " " " "*" " "
5  ( 1 ) " " "*" "*" "*" " " " " "*" " " "*" " "
6  ( 1 ) " " "*" "*" "*" "*" "*" " " " " "*" " "
7  ( 1 ) " " "*" "*" "*" "*" "*" " " "*" "*" " "
8  ( 1 ) " " "*" "*" "*" "*" "*" " " "*" "*" "*"
9  ( 1 ) " " "*" "*" "*" "*" "*" "*" "*" "*" "*"
10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

Appendix

Gram-Schmidt Procedure (Successive Orthogonalization)

— In general, to solve for OLS fit (with an intercept), the Gram-Schmidt procedure:

1 Initialize $\mathbf{z}_0 = \mathbf{x}_0 = \mathbf{1}$

2 For $j = 1, \ldots, p$, project $\mathbf{x}_j$ on orthogonal basis $\{\mathbf{z}_0, \ldots, \mathbf{z}_{j-1}\}$, and let the residual vector be $\mathbf{z}_j$. More precisely, let

$$\widehat{\gamma}_{kj} = \frac{\langle \mathbf{x}_j, \mathbf{z}_k \rangle}{\langle \mathbf{z}_k, \mathbf{z}_k \rangle} \quad \text{for} \quad k = 0, \ldots, j-1,$$

$$\text{and} \quad \mathbf{z}_j = \mathbf{x}_j - \sum_{k=0}^{j-1} \widehat{\gamma}_{kj} \mathbf{z}_k$$

3 Regress $\mathbf{y}$ on the residual $\mathbf{z}_p$ (the last covariate) to get

$$\widehat{\beta}_p = \frac{\langle \mathbf{y}, \mathbf{z}_p \rangle}{\langle \mathbf{z}_p, \mathbf{z}_p \rangle}$$

4 Compute β_j successively for $j = p-1, \ldots, 0$.

Ozone data example for Gram-Schmidt

```
> library(ElemStatLearn) # for ozone data
> # fit linear regression to the full data
> fullmodel = lm(ozone ~ radiation + temperature + wind, data=
    ozone)
> round(summary(fullmodel)$coef, dig=3)

              Estimate Std. Error t value Pr(>|t|)
(Intercept)    -64.232     23.042  -2.788    0.006
radiation        0.060      0.023   2.580    0.011
temperature      1.651      0.253   6.516    0.000
wind            -3.338      0.654  -5.105    0.000

> # get the residual of wind after projection on others
> wind.res = lm(wind ~ radiation + temperature, data=ozone)$res
> # get the coefficient of wind
> parmodel=lm(ozone ~ -1 + wind.res, data=ozone)
> round(summary(parmodel)$coef, dig=3)

         Estimate Std. Error t value Pr(>|t|)
wind.res   -3.338      1.631  -2.046    0.043
```

Justification of Mallows' C_p

- Recall our previous analysis of the training and testing errors with $\mathbf{y}$ and $\mathbf{y}^*$
- Now, lets assume that the model is not necessarily a linear model, i.e.,

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{e}$$
$$\mathbf{y}^* = \boldsymbol{\mu} + \mathbf{e}^*$$

We assume mean 0 and variance σ^2 for the two error vectors, but we don't have $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$. However, we still perform linear regression regardless. This will introduce bias of the estimations.

Justification of Mallows' C_p

$$\begin{aligned}
\mathsf{E}[\text{Test Err}] &= \mathsf{E}\|\mathbf{y}^* - \mathbf{X}\widehat{\beta}\|^2 = \|\mathbf{y}^* - \mathbf{H}\mathbf{y}\|^2 \\
&= \mathsf{E}\|(\mathbf{y}^* - \boldsymbol{\mu}) + (\boldsymbol{\mu} - \mathbf{H}\boldsymbol{\mu}) + (\mathbf{H}\boldsymbol{\mu} - \mathbf{H}\mathbf{y})\|^2 \\
&= \mathsf{E}\|\mathbf{y}^* - \boldsymbol{\mu}\|^2 + \mathsf{E}\|\boldsymbol{\mu} - \mathbf{H}\boldsymbol{\mu}\|^2 + \mathsf{E}\|\mathbf{H}\boldsymbol{\mu} - \mathbf{H}\mathbf{y}\|^2 \\
&= \mathsf{E}\|\mathbf{e}^*\|^2 + \mathsf{E}\|\boldsymbol{\mu} - \mathbf{H}\boldsymbol{\mu}\|^2 + \mathsf{E}\|\mathbf{H}\mathbf{e}\|^2 \\
&= n\sigma^2 + \text{Bias}^2 + p\sigma^2
\end{aligned}$$

$$\begin{aligned}
\mathsf{E}[\text{Train Err}] &= \mathsf{E}\|\mathbf{y} - \widehat{\mathbf{y}}\|^2 = \mathsf{E}\|(\mathbf{I} - \mathbf{H})\boldsymbol{\mu} + (\mathbf{I} - \mathbf{H})\mathbf{e}\|^2 \\
&= \mathsf{E}\|(\mathbf{I} - \mathbf{H})\boldsymbol{\mu}\|^2 + \mathsf{E}\|(\mathbf{I} - \mathbf{H})\mathbf{e}\|^2 \\
&= \text{Bias}^2 + (n - p)\sigma^2
\end{aligned}$$

Hence, Test Err is approximately Train Err $+ 2\sigma^2 p$, which justifies Mallows' C_p.