STA 5106
Computational Methods in Statistics I

Department of Statistics
Florida State University

Class 1
August 27, 2019

Course Information

- Meeting time: T Th 11am – 12:15pm **(lecture)**
  Meeting location: 101 RBA

- Instructor: **Dr. Wei Wu**
  - Email: wwu@stat.fsu.edu
  - Office Hours: T Th 10:30-11am, 1:30-2pm
  - Office: 106B OSB

- Teaching Assistant: **Yue Mu**
  - Email: ym18e@my.fsu.edu
  - Office Hours: W 2-3pm
  - Office: 204 OSB

Course Objectives

- To gain an understanding of the techniques and ideas used in implementing mathematical/statistical formulations on computers, with a focus on common statistical approaches.

- Students will also obtain practice in using MATLAB.

- **Python is optional.**

- **Prerequisite:** basic probability and statistics, linear algebra, and advanced calculus.

Course Materials

- Textbook: No required textbook, but a good reference is: *Computational Statistics*, by G. H. Givens and J. A. Hoeting

- Class notes (in PDF version) will be provided.

- Software:
  - MATLAB (https://www.mathworks.com/academia/tah-portal/florida-state-university-731138.html)
  - Python (open-source)

- **Canvas class website**

Homework

- Weekly assigned, typically on Thursday
- Due by the following Thursday
- About 10 homework assignments over the semester
- You can work with other students, but each student must *independently* finish his/her own solutions.
- **No Late Homework!**

Projects

- One mid-term project and one final project
- The midterm project requires one report and one in-class presentation.
- The final project only requires one report.

Grading

- Final percentage points:

Homework	(50%)
Midterm project	(20%)
- report (15%), presentation (5%)	
Final project	(20%)
Attendance	(10%)

- For students who finish homework programming problems using both **Matlab** and **Python**, they can skip the final project and will automatically receive the perfect score for it.

Policy

- **Each class requires full attention**
  - **Come to the class on time**
  - Be quiet
  - Attend all classes
- **Lateness to the class is counted as absence**
- **No late homework**
- **Work for the grade you want**

Chapter 2

Numerical Linear Algebra

2.2 Multiple Regression Analysis

Multiple Linear Regression

- Let $x_1, x_2, \ldots, x_n$ be a set of independent variables and y be a variable of interest which depends upon the values of x_i' s.

- Furthermore, assume that the relationship between x_i' s and y is linear, that is,

$$y = \sum_{i=1}^{n} b_i x_i + \varepsilon$$

  where b_i' s are scalar constants and ε represents the residual error.

- Very often ε is assumed as a random variable and, in particular, a Gaussian random variable.

- The coefficients b_i' s are unknown and they have to be estimated using the observed values of x_i' s and y.

Matrix Form

- The observations may belong to different sample times $t_1, t_2, \ldots, t_m$ such that

$$y(t_i) = \sum_{j=1}^{n} b_j x_j(t_i) + \varepsilon(t_i)$$

- In a matrix form these equations can be restated as

$$y = Xb + \varepsilon$$

where

$$y = \begin{pmatrix} y(t_1) \\ y(t_2) \\ \vdots \\ y(t_m) \end{pmatrix} \quad X = \begin{pmatrix} x_1(t_1) & x_2(t_1) & & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & & x_n(t_2) \\ & & \ddots & \\ x_1(t_m) & x_2(t_m) & & x_n(t_m) \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad \varepsilon = \begin{pmatrix} \varepsilon(t_1) \\ \varepsilon(t_2) \\ \vdots \\ \varepsilon(t_m) \end{pmatrix}$$

Least Squares

- **Least Squares:** The goal is to find the weight vector b such that
$$\| y - Xb \|^2$$
is minimized.

- Let $\hat{b}$ denote the estimated b. Then the estimated response is
$$\hat{y} = X\hat{b}$$

- That is, the following sum of squares reaches minimum with $\hat{b}$
$$\begin{aligned}\left\| y - X\hat{b} \right\|^2 &= \left\| y - \hat{y} \right\|^2 \\ &= (y - \hat{y})^T (y - \hat{y}) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2\end{aligned}$$

Maximum Likelihood Estimation

- m-dimensional normal distribution $N(\mu, \Sigma)$: Density function

$$f(x;\mu,\Sigma) = (2\pi)^{-m/2}\det(\Sigma)^{-1/2}\exp\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\}$$

- **Maximum likelihood estimation** of b with model

$$y = Xb + \varepsilon$$

where $\varepsilon \sim N(0, \sigma^2 I)$:

$$\begin{aligned}\hat{b}_{ML} &= \arg\max_b P(y - Xb; 0, \sigma^2 I) \\ &= \arg\max_b \frac{1}{(2\pi\sigma^2)^{m/2}}\exp\{\frac{-1}{2\sigma^2}\| y - Xb \|^2\} \\ &= \arg\min_b \| y - Xb \|^2\end{aligned}$$

Least Squares Solution

- In order to make the system identifiable, we often have $m > n$.
- In case X^TX is non-singular, the solution is

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

The term $(X^TX)^{-1}X^T$ is called the **pseudo-inverse** of X.

- Computing inverse of a matrix is order $O(n^3)$ operation which is computationally expensive.

We seek an alternative approach.

- The sum of squares of error is given by:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \| y - X\hat{b} \|^2$$

Orthogonal Transformations

- We are interested in solving the problem

$$\hat{b} = \arg\min_{b} \| y - Xb \|^2$$

- Let Q be an $m \times m$ orthogonal matrix, i.e. $QQ^T = Q^TQ = I_m$, and

$$y^* = Qy = QXb + Q\varepsilon = X^*b + \varepsilon^*$$

- Multiplication of an orthogonal matrix does not change the length (2-norm) of a vector.

- Therefore,

$$\hat{b} = \arg\min_{b} \| y - Xb \|^2 = \arg\min_{b} \| y^* - X^* b \|^2$$

Upper Triangular Matrix

- If we can select a Q in such a way that X^* is an upper triangular matrix (zeros below the diagonal elements), then

$$\begin{pmatrix} y_1^* \\ y_2^* \end{pmatrix} = \begin{pmatrix} X_1^* \\ 0 \end{pmatrix} b + \begin{pmatrix} \varepsilon_1^* \\ \varepsilon_2^* \end{pmatrix}$$

- Therefore,

$$\| y^* - X^* b \|^2 = \| y_1^* - X_1^* b \|^2 + \| y_2^* \|^2$$

and

$$\hat{b} = \arg\min_b \| y^* - X^* b \|^2 = \arg\min_b \| y_1^* - X_1^* b \|^2$$

- X_1^* being upper triangular the solution can be found by the backward substitution.

Backward Substitution

- This is a technique to solve for the vector *b* in the linear equation:

$$\begin{pmatrix} y_1^* \\ y_2^* \\ \vdots \\ y_n^* \end{pmatrix} = \begin{pmatrix} x_{11}^* & x_{12}^* & & x_{1n}^* \\ 0 & x_{22}^* & & x_{2n}^* \\ & & \ddots & \\ 0 & 0 & & x_{nn}^* \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

- At first, $\hat{b}_n = y_n^* / x_{nn}^*$

- Then, $\hat{b}_{n-1} = (y_{n-1}^* - x_{n-1,n}^* \hat{b}_n) / x_{n-1,n-1}^*$

- In general, $\hat{b}_j = (y_j^* - \sum_{i=j+1}^{n} x_{j,i}^* \hat{b}_i) / x_{j,j}^*, \qquad j = n-1, n-2, \ldots, 1$

Algorithm

- Given an upper triangular matrix $X \in \mathbf{R}^{m\times n}$ and a vector $y \in \mathbf{R}^m$, find the least squares estimate of b.

- **Algorithm 8 (Backward Substitution)**

```
function b = backsub(X,y)
l = size(X);
n = l(2);
b(n, 1) = y(n, 1)/X(n, n);
for j = n - 1 : -1 : 1
    b(j, 1) = (y(j, 1) - X(j, j + 1 : n) * b(j + 1 : n, 1))/X(j, j);
end
```