Lecture 23

General Linear Least Squares

- **General Linear Least Squares**

- Linear, polynomial, and multiple linear regression all belong to the general linear least-squares model:

$$y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m$$

where z_0, z_1 , …, z_m are a set of m+1 basis functions .

The basis functions z_i are given. Coefficients a_i are unknown

- The basis functions can be any functions, they can have a single or multiple independent variables.

- The equation:

$$y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m$$

can be written for each data point as:

$$y(x_i) = a_0 z_0(\mathbf{x}_i) + a_1 z_1(\mathbf{x}_i) + a_2 z_2(\mathbf{x}_i) + \cdots + a_m z_m(\mathbf{x}_i)$$

‘i’ is the index of data point. **x** could be a single value or a vector.

The purpose is to find the coefficients a_i

General Linear Least Squares

- **General Linear Least Squares**

- The residual at point i is,

$$e_i = y_i - y(\mathbf{x}_i) = y_i - \left(a_0 z_0(\mathbf{x}_i) + a_1 z_1(\mathbf{x}_i) + a_2 z_2(\mathbf{x}_i) + \cdots + a_m z_m(\mathbf{x}_i)\right)$$

- The formula can be expanded as:

$$\begin{gathered}
e_1 = y_1 - y(\mathbf{x}_1) = y_1 - \left(a_0 z_0(\mathbf{x}_1) + a_1 z_1(\mathbf{x}_1) + a_2 z_2(\mathbf{x}_1) + \cdots + a_m z_m(\mathbf{x}_1)\right) \\
e_2 = y_2 - y(\mathbf{x}_2) = y_2 - \left(a_0 z_0(\mathbf{x_2}) + a_1 z_1(\mathbf{x_2}) + a_2 z_2(\mathbf{x_2}) + \cdots + a_m z_m(\mathbf{x_2})\right) \\
e_3 = y_3 - y(\mathbf{x}_3) = y_3 - \left(a_0 z_0(\mathbf{x}_3) + a_1 z_1(\mathbf{x}_3) + a_2 z_2(\mathbf{x}_3) + \cdots + a_m z_m(\mathbf{x}_3)\right) \\
\cdots \\
e_n = y_n - y(\mathbf{x}_n) = y_n - \left(a_0 z_0(\mathbf{x}_n) + a_1 z_1(\mathbf{x}_n) + a_2 z_2(\mathbf{x_n}) + \cdots + a_m z_m(\mathbf{x}_n)\right)
\end{gathered}$$

- The residual can be re-written as a matrix equation:

$$\{e\} = \{y\} - [Z]\{a\}$$

General Linear Least Squares

- **General Linear Least Squares**

- The residual can be re-written as a matrix equation:

$$\{e\} = \{y\} - [Z]\{a\}$$

where

{y} is a column vector, contains the dependent data of the data points (known)

{a} is a column vector, contains the coefficients of the equation

{e} is a column vector, contains the error at each point

[Z] is a matrix of the calculated values of the basic functions at the measured values of the independent variable.

$$[Z] = \begin{pmatrix} z_{01} & z_{11} & \cdots & z_{m1} \\ z_{02} & z_{12} & \cdots & z_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ z_{0n} & z_{1n} & \cdots & z_{mn} \end{pmatrix}$$

where $z_{ij} = z_i(\mathbf{x}_j)$ is the value of basis function z_i at x_j

The subscripts of z do not represent the indices of rows and columns.
In the first column, i=0, the second column, i = 1 … i are the indices of basis function.
In the first row, j=1, j is the index of data point.

General Linear Least Squares

➢ **General Linear Least Squares**

- Generally, [Z] is not a square matrix, so simple inversion cannot be used to solve for the coefficients {a}. Instead the sum of the squares of the estimate residuals is minimized:

$$S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{m} a_j z_{ji}\right)^2$$

$$\boxed{\begin{aligned} &e_i \\ &= y_i \\ &- \big(a_0 z_0(\mathbf{x}_i) + a_1 z_1(\mathbf{x}_i) \\ &+ a_2 z_2(\mathbf{x}_i) + \cdots + a_m z_m(\mathbf{x}_i)\big) \end{aligned}}$$

- To find the minimum of Sr, we take the partial derivative of S_r with respect to a_j , and let the derivative be zero:

$$\frac{\partial S_r}{\partial a_j} = \frac{\partial}{\partial a_j}\left\{\sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{m} a_j z_{ji}\right)^2\right\} = 0$$

- The outcome of this minimization yields:

$$\boxed{\big[[Z]^T[Z]\big]\{a\} = \big\{[Z]^T\{y\}\big\}}$$ (This equation is in the form [A]{x} = {b})

[Z] is a matrix of the calculated values of the basic functions at the measured values of the independent variable.

General Linear Least Squares

➢ General Linear Least Squares

- Use general linear least square regression to fit the 3rd order polynomial

$y = a0 + a_1 * x + a_2 * x^2 + a_3 * x^3$

The values of data points are

X:	3	4	5	7	8	9	11	12
Y:	1.6	3.6	4.4	3.4	2.2	2.8	3.8	4.6

Compare the 3rd polynomial in the problem with the general function of y, we can find,
$z_0=1$, $z_1 = x$, $z_2 = x^2$, $z_3 = x^3$
Matrix Z will be

$$\begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{pmatrix}$$

➢ General Linear Least Squares

- The general linear least-squares model:

$$y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m$$

where $z_0, z_1, \ldots, z_m$ are a set of m+1 basis functions.

- To find the coefficients, we need to solve the equation:

$$[[Z]^T[Z]]\{a\} = \{[Z]^T\{y\}\}$$

$$[Z] = \begin{pmatrix} z_{01} & z_{11} & \cdots & z_{m1} \\ z_{02} & z_{12} & \cdots & z_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ z_{0n} & z_{1n} & \cdots & z_{mn} \end{pmatrix}$$

where $z_{ij} = z_i(x_j)$

General Linear Least Squares

➢ General Linear Least Squares

- Use general linear least square regression to fit the 3rd order polynomial

  $y = a0 + a_1 * x + a_2 * x^2 + a_3 * x^3$

X: 3 4 5 7 8 9 11 12

Y: 1.6 3.6 4.4 3.4 2.2 2.8 3.8 4.6

```
clc
clear

x = [3   4   5   7   8   9   11   12]';
y = [1.6   3.6   4.4   3.4   2.2   2.8   3.8   4.6]';

n = length(x);

Z = [ones(n,1)   x   x.^2   x.^3];     % key step

A = Z'*Z;
b = Z'*y;

a = A\b;     % same as a = inv(A)*b

xx = linspace(3,12,50);
yy = a(1) + a(2)*xx + a(3)*xx.^2 + a(4)*xx.^3;
plot(x,y,'r*',xx,yy)
```

➢ General Linear Least Squares

- The general linear least-squares model:

$$y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m$$

where $z_0, z_1, \ldots, z_m$ are a set of m+1 basis functions.

- To find the coefficients, we need to solve the equation:

$$[[Z]^T[Z]]\{a\} = \{[Z]^T\{y\}\}$$

$$[Z] = \begin{pmatrix} z_{01} & z_{11} & \cdots & z_{m1} \\ z_{02} & z_{12} & \cdots & z_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ z_{0n} & z_{1n} & \cdots & z_{mn} \end{pmatrix}$$

where $z_{ij} = z_i(x_j)$

$$\boxed{z_0 = 1,\ z_1 = x,\ z_2 = x^2,\ z_3 = x^3}$$

Nonlinear Regression (Not Covered in Exams)

- **Nonlinear Regression**

- As seen in the previous chapter, not all fits are linear equations of coefficients and basis functions.
- One method to handle this is to transform the variables and solve for the best fit of the transformed variables. There are two problems with this method:
  - ✓ Not all equations can be transformed easily
  - ✓ The best fit line represents the best fit for the transformed variables, not the original variables.
- Another method is to perform nonlinear regression to directly determine the least-squares fit.
- To perform nonlinear regression in MATLAB, write a function that returns the sum of the squares of the estimate residuals for a fit and then use MATLAB's ***fminsearch*** function to find the values of the coefficients where a minimum occurs.
- The arguments to the function to compute *Sr* should be the coefficients, the independent variables, and the dependent variables.

Nonlinear Regression

➢ Nonlinear Regression

- Given dependent force data F for independent velocity data v, determine the coefficients for the fit:

$$F = a_0 v^{a_1}$$

- First - write a function program called fSSR.m containing the following:

```
function f = fSSR(a, xm, ym)
    yp = a(1)*xm.^a(2);
    f = sum((ym-yp).^2);
```

where **a** is the vector of unknown coefficients and **f** is the sum of the squares of the estimate residuals.

- Then, use ***fminsearch*** in the command window or a program to obtain the values of **a** that minimize fSSR:

```
a = fminsearch(@fSSR, [1, 1], [], xm, ym)
```

where [1, 1] is an initial guess for the [a0, a1] vector, and [] is a placeholder for the options.

Nonlinear Regression

➢ **Nonlinear Regression**

- Use nonlinear regression to fit the following nonlinear function

  $y = \alpha x e^{\beta x}$

X: 0.1 0.2 0.4 0.6 0.9 1.3 1.5 1.7 1.8

Y: 0.75 1.25 1.45 1.25 0.85 0.55 0.35 0.25 0.1

```
function f = fSSR(a, xm, ym)
yp = a(1)*xm.*exp(a(2).*xm);
f = sum((ym-yp).^2);
```

```
clc
clear

x = [0.1  0.2  0.4  0.6  0.9  1.3  1.5  1.7  1.8]';
y = [0.75 1.25 1.45 1.25 0.85 0.55 0.35 0.25 0.1]';

 a = fminsearch(@fSSR, [1, 1], [], x, y);

xx = linspace(0.1, 1.8, 50);
yy = a(1)*xx.*exp(a(2).*xx);

plot(x,y,'r*', xx, yy)
```

➢ **Nonlinear Regression**

- First - write a function called fSSR.m containing the following:

  function f = fSSR(a, xm, ym)

  yp = a(1)*xm.^a(2);

  f = sum((ym-yp).^2);

  where **a** is the vector of unknown coefficients and **f** is the sum of the squares of the estimate residuals.

- Then, use ***fminsearch*** in the command window to obtain the values of **a** that minimize fSSR:

  a = fminsearch(@fSSR, [1, 1], [], xm, ym)

  where [1, 1] is an initial guess for the [a0, a1] vector, and [] is a placeholder for the options.

- **Homework on Canvas**