Third Programming Project – Two-Dimensional Arrays

Larry Caretto
Computer Science 106

Computing in Engineering and Science

May 9, 2006

California State University Northridge

Outline

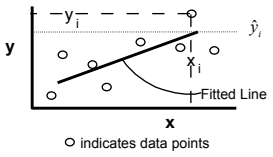
- Quiz three on Thursday for full lab period
  - See sample quiz on home page of course web site
- Project three
  - Multilinear regression
  - Equations to be solved
  - Data structures to be used
  - Library program use in separate code file

Files You Can Download

- Project instructions on home page and projects page
- Other files on projects page only
  - Data file for project
  - Library program for solving simultaneous linear equations
  - Excel file with answers
- This lecture on home page and laboratory presentations page

Exercise 8 Linear Regression

- Fit linear relationship to measured data pairs (x_i, y_i)
- $y = a + bx$
- Have equations to determine a and b
- Get goodness-of-fit measures



- Fitted value at x_i is $\hat{y}_i = a + bx_i$

Project Three Regression

- Have a result that depends on more than one variable
  - Example is emissions from diesel engine that depends on fuel properties
  - emissions = $b_0 + b_1$(cetane) + b_2(aromatics) + b_3(density)
- Use measured data on emissions, cetane, aromatics to find b_0, b_1, b_2, b_3

General Regression

- Use notation so that we can write code for any number of predictive variables
- Call predictive variables x_1, x_2, x_3, etc.
- Call response variable y
  - In previous example, x_1 = cetane, x_2 = aromatics, x_3 = density, and y = emissions
  - For three variables the equation is $y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3$

General Equation and Data

- In general we can have K predictive variables, x_1 to x_k
- General model equation: $y = b_0 + \sum_{j=1}^{K} b_j x_j$
- How do we represent the data?
  - Each data set consists of one value of y and one value for each of the x_j variables
  - For data set m, we can call the value of y, y_m, and we can call the value of x_j for data set m x_{jm}

California State University Northridge 7

Data Set with K = 3 and N = 8

m	y	x_1	x_2	x_3
0	2.55	3.00	440	500
1	1.95	3.47	350	400
2	1.89	3.14	440	540
3	2.24	3.46	350	370
4	2.31	3.59	450	480
5	1.74	1.75	200	320
6	1.87	3.03	310	470
7	0.83	3.18	290	400

y_0 x_{12} x_{35}

- Each data set, m, has a value for y and each x_j
- What are y_4, x_{23}, x_{31}?

California State University Northridge 8

Summary of Data

- We use K different variables (x_1 to x_K) to predict the value of another variable, y
- We have N sets of data
  - numbered from m = 0 to m = N-1
  - each data set has one value of y, called y_m, and one value of each x_j, called x_{jm}
  - all x and y values from file input
  - all data used to determine b_0 to b_K

California State University Northridge 9

How do we find b_j?

- Define x_{0m} = 1 for all m = 0, ... , N-1
  - Note that there is no x_0 in model
  - Setting x_{0m} = 1 used to simplify equations
- Values of b_0, ... , b_K found by solving set of K + 1 simultaneous linear equations
  - $A_{i0}\, b_0 + A_{i1}\, b_1 + A_{i2}\, b_2 + \ldots + A_{iK}\, b_K = c_i$
  - Compute A_{ij} and c_i from input data
  - Use library program to solve equations

California State University Northridge 10

Equations to be Used

- Compute the A_{ij} coefficients $A_{ij} = \sum_{m=0}^{N-1} x_{im} x_{jm}$
- Compute the c_i coefficients $c_i = \sum_{m=0}^{N-1} x_{im} y_m$
- Use the "library" routine provided to solve for b_j $\sum_{j=0}^{K} A_{ij} b_j = c_i \quad i = 0, \ldots, K$

California State University Northridge 11

Computing one A[i][j] (A[1][2])

m	y	x_0	x_1	x_2	x_3
0	2.55	1	3.00	440	500
1	1.95	1	3.47	350	400
2	1.89	1	3.14	440	540
3	2.24	1	3.46	350	370
4	2.31	1	3.59	450	480
5	1.74	1	1.75	200	320
6	1.87	1	3.03	310	470
7	0.83	1	3.18	290	400

$A_{ij} = \sum_{m=0}^{N-1} x_{im} x_{jm}$

$A_{12} = x_{10}x_{20} + x_{11}x_{21} + x_{12}x_{22} + x_{13}x_{23} + x_{14}x_{24} + x_{15}x_{25} + x_{16}x_{26} + x_{16}x_{26} + x_{17}x_{27} = A_{21}$

California State University Northridge 12

Computing A[1][2] = A[2][1]

m	y	x_0	x_1	x_2	x_3
0	2.55	1	3.00	440	500
1	1.95	1	3.47	350	400
2	1.89	1	3.14	440	540
3	2.24	1	3.46	350	370
4	2.31	1	3.59	450	480
5	1.74	1	1.75	200	320
6	1.87	1	3.03	310	470
7	0.83	1	3.18	290	400

$$A_{12} = \sum_{m=0}^{N-1} x_{1m} x_{2m}$$

A_{12} = A_{21} =
(3.00)(440) +
(3.47)(350) +
(3.14)(440) +
(3.46)(350) +
(3.59)(450) + …

Computing A[1][2] = A[2][1]

m	y	x_0	x_1	x_2	x_3
0	2.55	1	3.00	440	500
1	1.95	1	3.47	350	400
2	1.89	1	3.14	440	540
3	2.24	1	3.46	350	370
4	2.31	1	3.59	450	480
5	1.74	1	1.75	200	320
6	1.87	1	3.03	310	470
7	0.83	1	3.18	290	400

A_{12} = A_{21} =
(3.00)(440) +
(3.47)(350) +
(3.14)(440) +
(3.46)(350) +
(3.59)(450) +
(1.75)(200) +
(3.03)(310) +
(3.18)(290)

Computing one c[i] – c[1]

m	y	x_0	x_1	x_2	x_3
0	2.55	1	3.00	440	500
1	1.95	1	3.47	350	400
2	1.89	1	3.14	440	540
3	2.24	1	3.46	350	370
4	2.31	1	3.59	450	480
5	1.74	1	1.75	200	320
6	1.87	1	3.03	310	470
7	0.83	1	3.18	290	400

$$c_i = \sum_{m=0}^{N-1} x_{im} y_m$$

c_1 = $x_{10}y_0$ +
$x_{11}y_1$ + $x_{12}y_2$
$x_{13}y_3$ + $x_{14}y_4$ +
$x_{15}y_5$ + $x_{16}y_6$ +
$x_{16}y_6$ + $x_{17}y_7$ =
(3.00)(2.55) +
…

More Equations to be Used

- Compute estimated y values $\hat{y}_m = b_0 + \sum_{j=1}^{K} b_j x_{jm}$
- Compute the R^2 value

$$R^2 = 1 - \frac{\sum_{m=0}^{N-1} (y_m - \hat{y}_m)^2}{\left(\sum_{m=0}^{N-1} y_m^2 \right) - N (\bar{y})^2}$$

Required Code Steps

- Input data on K, N, y_m, and x_{jm}
- Set each x_{0m} = 1
- Compute A_{ij} and c_i (as A[i][j] and c[i])
- Use library program that takes A_{ij} and c_i as inputs and returns solution for b_j
- Use b_j to compute predicted values: $\hat{y}_m$
- Compute R^2 and output results

Getting the Input Data

- Download data file from course web site
- Data on file start with value of K followed by N sets of data
- Each data set starts with y_m followed by x_{1m}, x_{2m}, x_{3m}, etc.
- Can modify exercise eight input function
- Determine N by end of file condition
- Remember to set x_{0m} = 1

Arrays for Computing A_{ij} and c_i

$$A_{ij} = \sum_{m=0}^{N-1} x_{im} x_{jm} \qquad c_i = \sum_{m=0}^{N-1} x_{im} y_m$$

- A_{ij} is 2D array with both dimensions at least the number of variables plus two
- c_i is 1D array with same dimension as A_{ij}
- x_{im} is 2D array with different dimensions
  - First is the number of variables plus one
  - Second is maximum data points
- y_m is 1D array; dimension is maximum data points

Code for Computing A[i][j]

$$A_{ij} = \sum_{m=0}^{N-1} x_{im} x_{jm}$$

- Use typical code for computing a sum
- `A[i][j] += x[i][m] * x[j][m]`
- What is the for loop for this sum?
- `for ( m = 0; m < N; m++ )`
- How do we compute all the A[i][j]?
  - We need to place the computation of one A[i][j] inside two for loops over i and j

Library Function to Get b_j

$$\sum_{j=0}^{N-1} a_{ij} x_j = b_i \quad i = 0, \ldots, N-1$$

- Code variables use typical notation for system of linear equations shown above
  - a_{ij} are left-hand side coefficients
  - b_i are right-hand-side coefficients
  - x_j are unknowns
  - There are N equations to be solved

Library Function Definition

- Function prototype (second dimension required for passing 2D arrays to functions)

```
bool GaussianElim( double
  a[][MAX_VAR], double b[], double
  x[], int N_eqn );
```

`a[][MAX_VAR]` is left-hand side array (a_{ij})

`b[]` is right hand side array (b_i)

`x[]` is unknown array (x_j)

`N_eqn` is the number of equations (N)

Using `bool GaussianElim`

- Function returns an error flag in the function name
  - True indicates an error
  - False indicates a valid solution

```
if ( GaussianElim( A, c, b, K+1 ) )
{
    cout << "No solution for b[j]";
    return EXIT_FAILURE;
}
// remainder of code
```


Compare prototype and call

- Prototype

```
bool GaussianElim( double
  a[][MAX_VAR], double b[], double
  x[], int N_eqn );
```

Left-side

Unknowns

Right-side

Number of equations

```
if ( GaussianElim( A, c, b, K+1 ) )
{
    cout << "No solution for b[j]";
    return EXIT_FAILURE;
}
```

Passing the Second Dimension

- Use two files: your code and the library program
  - Place prototype for library program in your code
  - Make sure that the second dimension for a[i][j] (or A[i][j]) is the same for three cases
    - The header in the library program
    - The prototype in your code
    - Your declaration of the A[i][j] array

Project Three Code Summary

- Input data on K, N, y_m, and x_{jm}
- Set each $x_{0m} = 1$
- Compute A_{ij} and c_i
- Use library program that takes A_{ij} and c_i as inputs and returns solution for b_j
- Use b_j to compute predicted values: $\hat{y}_m$
- Compute R^2 and output results