NA HW#9 Linear Data Fitting

2013012148 이재일

Linear Least Square

x′ = a1x + a2y + a3

y′ = a4x +a5y + a6

이 linear mapping model에서 best set of parameter a1~a6을 구해야 한다.

Data fitting pdf 7p를 참조하면

$$S(a,b) = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} [y_i - (a + bx_i)]^2$$

a와b의 함수

S(a1,a2,..,a6)가 최소가 되는 a1,a2,..a6를 구하면 된다. 해당 값이 최소가 되려면 각 parameter별로 편미분을 한 S값이 0이 되어야 한다.

$$\frac{\partial S}{\partial a} = 2\sum_{i=1}^{N} [y_i - a - bx_i](-1) = 0$$

$$\frac{\partial S}{\partial b} = 2\sum_{i=1}^{N} [y_i - a - bx_i](-x_i) = 0$$

이런식으로 a1,a2,..,a6까지 편미분을 해 주어서 행렬을 만들게 되면

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum 1 & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix}$$

위와 같이 행렬을 구성할 경우

$$\begin{matrix} a1 \\ a2 \\ a3 \end{matrix} = \left| \begin{matrix} \sum x_i & \sum y_i & \sum 1 \\ \sum x_i y_i & \sum y^2 & \sum y_i \\ \sum x^2 & \sum x_i y_i & \sum x_i \end{matrix} \right| \left| \begin{matrix} \sum x_i' \\ \sum x_i x_i' \\ \sum y_i x_i' \end{matrix} \right|$$ 형식으로 원하는 답을 구할 수 있게 된다.

$$\begin{matrix} a4 \\ a5 \\ a6 \end{matrix} = \begin{matrix} \sum x_i & \sum y_i & \sum 1 & \sum y_i' \\ \sum x_i y_i & \sum y^2 & \sum y_i & \sum x_i y_i' \\ \sum x^2 & \sum x_i y_i & \sum x_i & \sum y_i y_i' \end{matrix}$$

Code

```
while (!feof(fp)) {
    fscanf_s(fp, "%f %f %f %f", &x, &y, &xp, &yp);
    nsum += 1.0;
    xsum += x;
    ysum += y;
    x2sum += x*x;
    y2sum += y*y;
    xysum += x*y;
    xpsum += xp;
    xpxsum += xp*x;
    xpysum += xp*y;
    ypsum += yp;
    ypxsum += yp*x;
    ypysum += yp*y;
}
```

```
a[1][1] = xsum;
a[1][2] = ysum;
a[1][3] = nsum;
a[2][1] = xysum;
a[2][2] = y2sum;
a[2][3] = ysum;
a[3][1] = x2sum;
a[3][2] = xysum;
a[3][3] = xsum;
b[1][1] = xpsum;
b[2][1] = xpxsum;
b[3][1] = xpysum;
```

역행렬을 구해야 할 행렬의 기본 원소들을 구한다.

```
for (int i = 1; i <= 3; i++) {
    for (int k = 1; k <= 3; k++) ai[k][i] = a[k][i];
    for (int k = 1; k <= 1; k++) r[i][k] = b[i][k];
}
printf("₩nOriginal matrix a : ₩n");
for (int k = 1; k <= 3; k++) {
    for (int i = 1; i <= 3; i++) printf("%12.6f₩t", a[k][i]);
    printf("₩n");
}

gaussj(ai, 3, r, 3);
printf("₩nInverse of matrix a : ₩n");
for (int k = 1; k <= 3; k++) {
    for (int i = 1; i <= 3; i++) printf("%12.6f", ai[k][i]);
    printf("₩n");
}

printf("₩nsolving the equation₩n");
for (int i = 1; i <= 3; i++){
    sol[i][1] = 0.0;
    for (int j = 1; j <= 3; j++){
        sol[i][1] += ai[i][j] * b[j][i];
    }
    printf("a%d is %12.6f₩n", i,sol[i][1]);
}
```

gaussj 함수를 사용하여서 역행렬을 구한다

```
b[1][1] = ypsum;
b[2][1] = ypxsum;
b[3][1] = ypysum;


for (int i = 1; i <= 3; i++){
    sol[i][1] = 0.0;
    for (int j = 1; j <= 3; j++){
        sol[i][1] += ai[i][j] * b[j][1];
    }
    printf("a%d is %12.6f\n", i+3, sol[i][1]);
}
```

같은 방법으로 a4,a5,a6도 구해준다.

Result

data1

```
Original matrix a :
100622.273438   86491.265625      104.000000
81907232.000000 75205056.000000 86491.265625
105695008.000000        81907232.000000 100622.273438

Inverse of matrix a :
   -0.000192    0.000000    0.000000
   -0.000358    0.000000    0.000000
    0.493415   -0.000358   -0.000192

solving the equation
a1 is    -0.405864
a2 is 154386.343750
a3 is -212721488.000000
a4 is    -0.591734
a5 is 154386.343750
a6 is -212721488.000000
```

data2

```
Original matrix a :
81978.429688    75607.195313      121.000000
54041372.000000 50425356.000000 75607.195313
57280968.000000 54041372.000000 81978.429688

Inverse of matrix a :
   -0.000013   -0.000000    0.000000
   -0.000184    0.000000   -0.000000
    0.132640   -0.000184   -0.000013

solving the equation
a1 is    -1.610640
a2 is 79485.367188
a3 is -57162396.000000
a4 is    -0.655446
a5 is 79485.367188
a6 is -57162396.000000
계속하려면 아무 키나 누르십시오 . . .
```

data3

```
Original matrix a :
156062.609375   91507.421875      185.000000
87645080.000000 55219692.000000 91507.421875
152732768.000000        87645080.000000 156062.609375

Inverse of matrix a :
   -0.000032   -0.000000    0.000000
   -0.000016    0.000000   -0.000000
    0.040385   -0.000016   -0.000032

solving the equation
a1 is   -10.537248
a2 is  6861.288086
a3 is -17409508.000000
a4 is    -6.927316
a5 is  6861.288086
a6 is -17409508.000000
```