Rajshahi University of Engineering & Technology

CSE 2104: Sessional Based on CSE 2103

Lab Report 06

Dated: 01.04.18

Submitted to

Shyla Afroge

Assistant Professor

Dept. of Computer Science & Engineering

&

Instructor, CSE 2104

Submitted by

Fuad Al Abir

Roll: 1603021

Section: A

Dept. of Computer Science & Engineering

LEAST SQUARE CURVE FITTING PROCEDURE is a mathematical procedure for finding the best-fitting curve to a given set of points by minimizing the sum of the squares of the offsets of the points from the curve. The sum of the squares of the offsets is used instead of the offset absolute values because this allows the residuals to be treated as a continuous differentiable quantity. However, because squares of the offsets are used, outlying points can have a disproportionate effect on the fit, a property which may or may not be desirable depending on the problem at hand.

Problem#01: Least square curve fitting procedure for a linear system

```
#include <iostream>

using namespace std;

int main()
{
    int n;
    cout << "Enter the number of data: ";
    cin >> n;
    double x[n], y[n], sumX = 0, sumY = 0, sumXX = 0, sumXY =
0;

    for(int i = 0; i < n; i++) {
        cin >> x[i] >> y[i];
    }
    cout << "-----------------------------\nX\t  Y\tX^2\t
XY\n-----------------------------" << endl;
    for(int i = 0; i < n; i++) {
        cout << x[i] << "\t" << y[i] << "\t" << x[i]*x[i] <<
"\t" << x[i]*y[i] << endl;
        sumX = sumX + x[i];
        sumY = sumY + y[i];
        sumXX = sumXX + x[i]*x[i];
        sumXY = sumXY + x[i]*y[i];
    }
    cout << "-----------------------------\n" << sumX << "\t"
<< sumY << "\t" << sumXX << "\t" << sumXY << endl;

    double D = n*sumXX - sumX*sumX;
    double a0 = (sumY*sumXX - sumX*sumXY)/D;
    double a1 = (n*sumXY - sumY*sumX)/D;

    cout << endl << "a0 = " << a0 << "\ta1 = " << a1 << endl;
    double iX;

    cout << "Enter a value to check: ";
    cin >> iX;
```

```
    double dY = a0 + a1*iX;
    cout << iX << " = " << dY << endl;

    int dn;
    for(int i = 0; i < n; i++) {
        if(iX == x[i]) dn = i;
    }
    double ERR = (dY - y[dn])/dY*100;
    cout << "Persentage Error: " << ERR << endl;
}
```

OUTPUT:

```
Enter the number of data: 6
20 800.3
30 800.4
40 800.6
50 800.7
60 800.9
70 801.0
------------------------------
X          Y     X^2       XY
------------------------------
20       800.3   400      16006
30       800.4   900      24012
40       800.6   1600     32024
50       800.7   2500     40035
60       800.9   3600     48054
70       801     4900     56070
------------------------------
270      4803.9  13900    216201

a0 = 799.994    a1 = 0.0145714
Enter a value to check: 60
60 = 800.869
Persentage Error: -0.00392431
```

DISCUSSION: All the calculated constants were determined using the equation according to the system and therefore a value was taken to check the new evaluated curve fitting value and calculate the percentage of error as well.

Problem#02: Least square curve fitting procedure for a non-linear system

```
#include <iostream>

using namespace std;

int main()
{
    int n;
    cout << "Enter the number of data: ";
    cin >> n;
    double x[n], y[n], sumX = 0, sumY = 0, sumXX = 0, sumX3 = 0,
sumX4 = 0, sumXY = 0, sumXXY = 0;

    for(int i = 0; i < n; i++)
    {
        cin >> x[i] >> y[i];
    }
    cout << "-----------------------------------------------------
\nX\tY\tX^2\tX^3\tX^4\tXY\tX^2Y\n-----------------------------------
------------------" << endl;
    for(int i = 0; i < n; i++)
    {
        cout << x[i] << "\t" << y[i] << "\t" << x[i]*x[i] << "\t" <<
x[i]*x[i]*x[i] << "\t" << x[i]*x[i]*x[i]*x[i] << "\t" << x[i]*y[i]
<< "\t" << x[i]*x[i]*y[i] << endl;
        sumX = sumX + x[i];
        sumY = sumY + y[i];
        sumXX = sumXX + x[i]*x[i];
        sumX3 = sumX3 + x[i]*x[i]*x[i];
        sumX4 = sumX4 + x[i]*x[i]*x[i]*x[i];
        sumXY = sumXY + x[i]*y[i];
        sumXXY = sumXXY + x[i]*x[i]*y[i];
    }
    cout << "-----------------------------------------------------
-----" << endl;
    cout << sumX << "\t" << sumY << "\t" << sumXX << "\t" << sumX3
<< "\t" << sumX4 << "\t" << sumXY << "\t" << sumXXY << endl;

    double D = n * (sumX4 * sumXX - sumX3 * sumX3) - sumX * (sumX *
sumX4 - sumX3 * sumXX) + sumXX * (sumX * sumX3 - sumXX * sumXX);
    double a0 = (sumX * (sumX3 * sumXXY - sumXY * sumX4) - sumXX *
(sumXX * sumXXY  - sumXY * sumX3) + sumY * (sumXX * sumX4 - sumX3 *
sumX3))/D;
    double a1 = (n * (sumX3 * sumXXY - sumXY * sumX4) - sumXX *
(sumX * sumXXY - sumXY * sumXX) + sumY * (sumX * sumX4 - sumXX *
sumX3))/D;
    double a2 = (n * (sumXX * sumXXY - sumXY * sumX3) - sumX * (sumX
* sumXXY - sumXX * sumXY) + sumY * (sumX * sumX3 - sumXX *
sumXX))/D;
    a1 = a1 * -1;
    cout << endl << "a0 = " << a0 << "\ta1 = " << a1 << "\ta2 = " <<
a2 << endl;
```

```
    double iX;
    cout << "Enter a value to check: ";
    cin >> iX;

    double dY = a0 + a1*iX + a2*iX*iX;
    cout << iX << " = " << dY << endl;

    int dn;
    for(int i = 0; i < n; i++)
    {
        if(iX == x[i]) dn = i;
    }
    double ERR = (dY - y[dn])/dY*100;
    cout << "Persentage Error: " << ERR << endl;
}
```

OUTPUT:

```
Enter the number of data: 7
1.0 1.1
1.5 1.2
2.0 1.5
2.5 2.6
3.0 2.8
3.5 3.3
4.0 4.1
-------------------------------------------------------------
X       Y       X^2     X^3     X^4     XY      X^2Y
-------------------------------------------------------------
1       1.1     1       1       1       1.1     1.1
1.5     1.2     2.25    3.375   5.0625  1.8     2.7
2       1.5     4       8       16      3       6
2.5     2.6     6.25    15.625  39.0625 6.5     16.25
3       2.8     9       27      81      8.4     25.2
3.5     3.3     12.25   42.875  150.062 11.55   40.425
4       4.1     16      64      256     16.4    65.6
-------------------------------------------------------------
17.5    16.6    50.75   161.875 548.188 48.75   157.275

a0 = 0.457143   a1 = 0.392857   a2 = 0.128571
Enter a value to check: 3.5
3.5 = 3.40714
Persentage Error: 3.14465
```

DISCUSSION: All the calculated constants were determined using the equation according to the system and therefore a value was taken to check the new evaluated curve fitting value and the percentage of error was printed also.

Problem#03: Least square curve fitting procedure for an exponential system

```
#include <iostream>
#include <cmath>

using namespace std;

int main()
{
    int n;
    cout << "Enter the number of data: ";
    cin >> n;
    double x[n], y[n], sumX = 0, lnY = 0, sumXX = 0, sumXY =
0;

    for(int i = 0; i < n; i++) {
        cin >> x[i] >> y[i];
    }
    for(int i = 0; i < n; i++) {
        cout << x[i] << "\t" << log(y[i]) << "\t" << x[i]*x[i]
<< "\t" << x[i]*log(y[i]) << endl;
        sumX = sumX + x[i];
        lnY = lnY + log(y[i]);
        sumXX = sumXX + x[i]*x[i];
        sumXY = sumXY + x[i]*log(y[i]);
    }
    cout << sumX << "\t" << lnY << "\t" << sumXX << "\t" <<
sumXY << endl;

    double D = n*sumXX - sumX*sumX;
    double a0 = (lnY*sumXX - sumX*sumXY)/D;
    double a1 = (n*sumXY - lnY*sumX)/D;

    double a = exp(a0), b = a1;

    cout << endl << "a0 = " << a0 << "\ta1 = " << a1;
    cout << endl << "a = " << a << "\tb = " << b << endl;

    double iX;

    cout << "\nEnter a value to check: ";
    cin >> iX;
    b = b * iX;
    double dY = a*exp(b);
    cout << iX << " = " << dY;

    int dn;
    for(int i = 0; i < n; i++) {
        if(iX == x[i]) dn = i;
    }
```

```
    double ERR = (dY - y[dn])/dY*100;
    cout << "\nPersentage Error: " << ERR << endl;

}
```

OUTPUT:

```
Enter the number of data: 5
2 4.077
4 11.084
6 30.128
8 81.897
10 222.62
2       1.40536 4       2.81072
4       2.4055  16      9.62201
6       3.40545 36      20.4327
8       4.40546 64      35.2437
10      5.40547 100     54.0547
30      17.0272 220     122.164

a0 = 0.405399   a1 = 0.500008
a = 1.4999      b = 0.500008

Enter a value to check: 10
10 = 222.624
Persentage Error: 0.00171407
```

DISCUSSION: All the calculated constants were determined using the equation according to the system and therefore a value was taken to check the new evaluated curve fitting value and the percentage of error was printed also.