LA-II Assignment 2 Report

Artem Matevosian | DSAI-03 | a.matevosian@innopolis.university

April 27, 2023

Source code

See the whole solution at GitHub. (https://github.com/ArtMGreen/LA-Assignments) **Note:** the repository is made private in order to avoid plagiarism. If you cannot access the repository, contact @artmgreen in Telegram.

```
int LA_Plotting(Matrix dataset, ColumnVector coefficients) {
    Gnuplot gp;

    double min_x = numeric_limits<double>::max();
    double max_x = numeric_limits<double>::min();
    double min_y = numeric_limits<double>::max();
    double max_y = numeric_limits<double>::min();

    for (int i = 0; i < dataset.getHeight(); i++) {
        double x = round(dataset.getElement(i, 0) * 10000.0) / 10000.0;
        if (x > max_x) max_x = x;
        if (x < min_x) min_x = x;
        double y = round(dataset.getElement(i, 1) * 10000.0) / 10000.0;
        if (y > max_y) max_y = y;
        if (y < min_y) min_y = y;
    }

    min_x -= 2; min_y -= 2; max_x += 2; max_y += 2;
    gp << "set xrange [" << min_x << ":" << max_x << "]\n";
    gp << "set yrange [" << min_y << ":" << max_y << "]\n";

    // constructing polynomial line
    string polynomial = "";
    for (int i = 0; i < coefficients.getHeight(); i++) {
        double coefficient = round(coefficients.getElement(i) * 10000.0) / 10000.0;
        polynomial += "+"+to_string(coefficient)+"*x**"+to_string(i);
    }

    gp << "plot " + polynomial + " , '-' using 1:2 with points\n";
    for (int i = 0; i < dataset.getHeight(); i++) {
        double x = round(dataset.getElement(i, 0) * 10000.0) / 10000.0;
        double y = round(dataset.getElement(i, 1) * 10000.0) / 10000.0;
        gp << to_string(x)+"\t"+ to_string(y)+"\n";
    }

    return 0;
}
```

```
int main() {
    int mLinesInDataset, nDegreesInPolynomial;
    cin >> mLinesInDataset;
    Matrix inputs = Matrix(mLinesInDataset, 2);
    cin >> inputs;
    cin >> nDegreesInPolynomial;
    AugmentedMatrix dataset = AugmentedMatrix(inputs);

    ColumnVector result = dataset.LSA(nDegreesInPolynomial);
    cout << "x~:" << endl;
    cout << fixed << setprecision(4) << result;

    return LA_Plotting(dataset, result);
}
```

Set of points (test case)

File `test_generator.py` generates the following test case to run the program with. The corresponding file at GitHub is `test_case.txt`.

```
50
-9.5594 2.6246
-9.3962 2.1521
-8.5692 2.4961
-8.2291 1.9754
-8.1709 2.0704
-7.2808 1.3244
-6.6749 1.4707
-6.6949 0.6827
-6.2107 0.2613
-6.3789 -0.8201
-5.1154 -0.6513
-5.1078 -1.2999
-5.0141 -1.6051
-4.4888 -2.1484
-3.4799 -2.8308
-3.2959 -2.9917
-3.2511 -3.8093
-2.6722 -3.6347
-1.98 -3.0382
-2.1954 -3.2513
-1.7803 -2.9693
-1.5065 -2.3828
-0.2966 -2.2393
-0.7556 -1.4766
-0.3293 -1.1087
0.9343 -0.9402
0.8362 0.0607
1.694 0.3028
1.7354 0.8346
1.7977 1.3839
2.1772 1.8882
2.5907 2.6755
3.5603 2.2059
3.8699 2.391
3.9609 2.4653
4.8195 1.9885
4.9946 1.6058
5.3695 1.6647
```

```
5.6295  1.0098
6.1878  0.9773
6.4836  −0.2844
6.5826  −0.7619
6.9494  −0.9376
7.6707  −1.6405
8.5579  −2.1081
8.5977  −3.076
8.4956  −3.0906
9.4747  −3.5155
9.4528  −3.8451
9.8807  −3.5071
4
```

Plotting & results

To execute `main.cpp`, compile it using `g++ -o LA_Plotting main.cpp -lboost_iostreams -lboost_system -lboost_filesystem` and copy the test case into the console.

The result should be as follows: