Университет ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.04 Программная инженерия
Дисциплина «Вычислительная математика»

Отчет
По лабораторной работе №4
Вариант 7

Выполнил:
Зинатулин А.В.
P32121

Преподаватель:
Наумова Н.А.

Санкт-Петербург, 2023 г.

Цель работы
найти функцию, являющуюся
наилучшим приближением заданной табличной функции по методу
наименьших квадратов

Рабочие формулы метода

$$\begin{vmatrix} n & \sum_{i=1}^{n} x_i & \dots & \sum_{i=1}^{n} x_i^m \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \dots & \sum_{i=1}^{n} x_i^{m+1} \\ & \dots \dots \dots & & \\ \sum_{i=1}^{n} x_i^m & \sum_{i=1}^{n} x_i^{m+1} & \dots & \sum_{i=1}^{n} x_i^{2m} \end{vmatrix} \cdot \begin{vmatrix} a_0 \\ a_1 \\ \dots \\ \dots \\ \dots \\ \dots \\ a_m \end{vmatrix} = \begin{vmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \dots \\ \sum_{i=1}^{n} x_i^m y_i \end{vmatrix}$$

Код программы
https://github.com/uvuv-643/Computational_Mathematics

Листинг кода:

```
//
// Created by artem on 16.04.2023.
//

#include "Metrics.h"

double Metrics::corr(CVector<CDouble>& x, CVector<CDouble>& y) {
    CDouble x_mean = x.mean();
    CDouble y_mean = y.mean();
    CVector<CDouble> tx = (x - x_mean);
    CVector<CDouble> ty = (y - y_mean);
    double s1 = CVector<CDouble>::apply(&tx, &ty, [] (double x, double y) { return x *
y; }).sum();
    double s2 = (x - x_mean).apply([](double t) { return pow(t, 2); }).sum();
    double s3 = (y - y_mean).apply([](double t) { return pow(t, 2); }).sum();
    return s1 / sqrt(s2 * s3);
}

double Metrics::mse(CVector<CDouble> &y_true, CVector<CDouble> &y_pred) {
    return (y_true - y_pred).apply([](double t) { return pow(t, 2); }).sum();
}

double Metrics::sd(CVector<CDouble> &y_true, CVector<CDouble> &y_pred) {
    size_t n = y_true.n;
    return sqrt(Metrics::mse(y_true, y_pred) / (double) n);
}

double Metrics::r2_score(CVector<CDouble> &y_true, CVector<CDouble> &y_pred) {
    CDouble y_pred_mean = y_pred.mean();
    double mse = Metrics::mse(y_true, y_pred);
    double variance = (y_true - y_pred_mean).apply([](double t) { return pow(t, 2);
}).sum();
```

```
    return 1 - mse / variance;
}
```

```
//
// Created by artem on 16.04.2023.
//

#include "PolynomialRegression.h"

CVector<CDouble> PolynomialRegression::performMethod(size_t number_of_points,
CVector<CDouble> x, CVector<CDouble> y) {

    size_t n = this->p + 1;
    Matrix<CDouble> a(n);
    CVector<CDouble> b(n);
    for (size_t i = 0; i < n; i++) {
        for (size_t j = 0; j < n; j++) {
            double t = 0;
            for (size_t k = 0; k < number_of_points; k++) {
                t += pow(x[k], (double)(i + j));
            }
            a[i][j] = t;
        }
    }
    for (size_t i = 0; i < n; i++) {
        double t = 0;
        for (size_t k = 0; k < number_of_points; k++) {
            t += y[k] * pow(x[k], (double) i);
        }
        b[i] = t;
    }

    CVector<CDouble> answer = GaussianSolver(a, b).solve();
    return answer;

}

PolynomialRegression::PolynomialRegression(size_t p) {
    this->p = p;
}

string PolynomialRegression::createDefinition(CVector<CDouble> answer) {
    string definition;
    for (int i = 0; i < (size_t) answer.n; i++) {
        switch (i) {
            case 0: {
                definition += to_string(answer[i]);
                break;
            }
            case 1: {
                definition += to_string(answer[i]) + " * x";
                break;
            }
            default: {
                definition += to_string(answer[i]) + " * x^" + to_string(i);
            }
        }
        if (i + 1 < (size_t) answer.n) {
            definition += " + ";
        }
    }
    return definition;
}
//
// Created by artem on 17.04.2023.
```

```cpp
//

#include "PowerRegression.h"

CVector<CDouble> PowerRegression::performMethod(size_t number_of_points,
CVector<CDouble>& x, CVector<CDouble>& y) {
    return PolynomialRegression::performMethod(number_of_points, x.apply(log),
y.apply(log));
}

PowerRegression::PowerRegression() : PolynomialRegression(1) {

}

string PowerRegression::createDefinition(CVector<CDouble> answer) {
    return "exp(" + to_string(answer[0]) + ") * x^" + to_string(answer[1]);
}



//
// Created by artem on 17.04.2023.
//

#include "LogRegression.h"

CVector<CDouble> LogRegression::performMethod(size_t number_of_points,
CVector<CDouble>& x, CVector<CDouble>& y) {
    return PolynomialRegression::performMethod(number_of_points, x.apply(log), y);
}

LogRegression::LogRegression() : PolynomialRegression(1) {

}

string LogRegression::createDefinition(CVector<CDouble> answer) {
    return to_string(answer[0]) + " + " + to_string(answer[1]) + "* log(x)";
}
//
// Created by artem on 17.04.2023.
//

#include "ExpRegression.h"

CVector<CDouble> ExpRegression::performMethod(size_t number_of_points,
CVector<CDouble>& x, CVector<CDouble>& y) {
    return PolynomialRegression::performMethod(number_of_points, x, y.apply(log));
}

ExpRegression::ExpRegression() : PolynomialRegression(1) {

}

string ExpRegression::createDefinition(CVector<CDouble> answer) {
    string definition = "exp(";
    definition += to_string(answer[0]) + " + " + to_string(answer[1]) + " * x";
    return definition + ")";
}
```

Результат выполнения программы при различных исходных данных

1)

Inputted data

#	X	Y
1	-3	-13.379298
2	-2.4545455	-10.427915
3	-1.9090909	-8.2945814
4	-1.3636364	-6.8897862
5	-0.81818181	-5.9361973
6	-0.27272728	-5.2906246
7	0.27272728	-4.7155981
8	0.81818181	-4.0635853
9	1.3636364	-3.1075892
10	1.9090909	-1.695087
11	2.4545455	0.43955114
12	3	3.4255447

Corr coefficient
0.973545

Linear function approximation

#	x	y_true	y_pred	error
1	-3	-13.379298	-11.801535	2.4893364
2	-2.4545455	-10.427915	-10.56391	0.018494777
3	-1.9090909	-8.2945814	-9.3262849	1.064412
4	-1.3636364	-6.8897862	-8.0886599	1.437298
5	-0.81818181	-5.9361973	-6.8510348	0.83692765
6	-0.27272728	-5.2906246	-5.6134097	0.10419023
7	0.27272728	-4.7155981	-4.3757846	0.11547321
8	0.81818181	-4.0635853	-3.1381596	0.85641275
9	1.3636364	-3.1075892	-1.9005345	1.4569813
10	1.9090909	-1.695087	-0.66290948	1.0653903
11	2.4545455	0.43955114	0.57471577	0.018269477
12	3	3.4255447	1.8123408	2.6024271

Squared function approximation

#	x	y_true	y_pred	error
1	-3	-13.379298	-11.787745	2.5330404
2	-2.4545455	-10.427915	-10.557642	0.01682921
3	-1.9090909	-8.2945814	-9.3260342	1.0638947
4	-1.3636364	-6.8897862	-8.0929222	1.4475361
5	-0.81818181	-5.9361973	-6.8583057	0.85028399
6	-0.27272728	-5.2906246	-5.622185	0.1099323
7	0.27272728	-4.7155981	-4.3845599	0.10958629
8	0.81818181	-4.0635853	-3.1454305	0.84300817
9	1.3636364	-3.1075892	-1.9047967	1.4467098
10	1.9090909	-1.695087	-0.66265875	1.065908
11	2.4545455	0.43955114	0.58098383	0.020003205
12	3	3.4255447	1.8261305	2.558126

Cubed function approximation

#	x	y_true	y_pred	error
1	-3	-13.379298	-13.385234	3.5238571e-05
2	-2.4545455	-10.427915	-10.412416	0.00024021152
3	-1.9090909	-8.2945814	-8.3094501	0.00022107917
4	-1.3636364	-6.8897862	-6.8827032	5.0170053e-05
5	-0.81818181	-5.9361973	-5.9385393	5.4849866e-06
6	-0.27272728	-5.2906246	-5.2833237	5.3303675e-05
7	0.27272728	-4.7155981	-4.7234212	6.1201271e-05
8	0.81818181	-4.0635853	-4.065197	2.5975039e-06
9	1.3636364	-3.1075892	-3.1150157	5.515293e-05
10	1.9090909	-1.695087	-1.6792428	0.00025103855
11	2.4545455	0.43955114	0.43575763	1.4390752e-05
12	3	3.4255447	3.4236195	3.7064377e-06

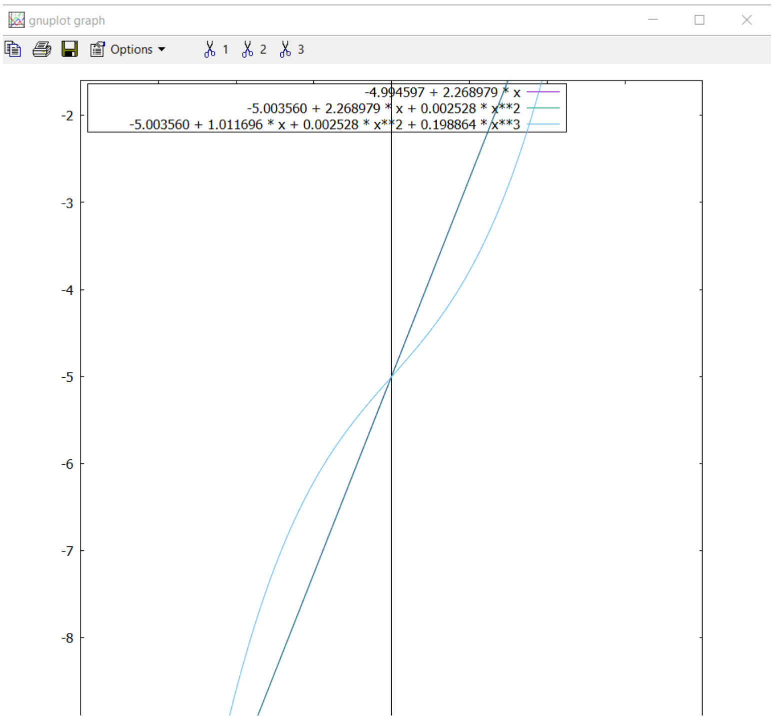
Unable to create power function approximation. There is negative x or y point

Unable to create exp function approximation. There is negative y point

Unable to create log function approximation. There is negative x point

Final answer

#	Function	MSE	SD	R2 score
1	-4.994597 + 2.268979 * x	12.065613	1.0027302	0.94779073
2	-5.003560 + 2.268979 * x + 0.002528 * x^2	12.064858	1.0026988	0.947794
3	-5.003560 + 1.011696 * x + 0.002528 * x^2 + 0.198864 * x^3	0.00099357543	0.009099338	0.9999957



2)

Inputted data

#	X	Y
1	-3	10.884967
2	-2.4545455	0.45761862
3	-1.9090909	-4.3206387
4	-1.3636364	-5.8495884
5	-0.81818181	-5.7641382
6	-0.27272728	-5.2865019
7	0.27272728	-4.7199588
8	0.81818181	-3.9382639
9	1.3636364	-2.1115682
10	1.9090909	2.2712705
11	2.4545455	11.332147
12	3	27.688623

Corr coefficient
0.437927

Linear function approximation

#	x	y_true	y_pred	error
1	-3	10.884967	-5.0794502	254.86261
2	-2.4545455	0.45761862	-3.8431265	18.496409
3	-1.9090909	-4.3206387	-2.6068025	2.9372343
4	-1.3636364	-5.8495884	-1.3704788	20.062422
5	-0.81818181	-5.7641382	-0.13415501	31.696711
6	-0.27272728	-5.2865019	1.1021688	40.815113
7	0.27272728	-4.7199588	2.3384926	49.821736
8	0.81818181	-3.9382639	3.5748163	56.446375
9	1.3636364	-2.1115682	4.8111402	47.923891
10	1.9090909	2.2712705	6.0474639	14.259636
11	2.4545455	11.332147	7.2837878	16.389209
12	3	27.688623	8.5201115	367.43185

Squared function approximation

#	x	y_true	y_pred	error
1	-3	10.884967	9.4863815	1.9560411
2	-2.4545455	0.45761862	2.7777066	5.3828083
3	-1.9090909	-4.3206387	-2.3419696	3.9151312
4	-1.3636364	-5.8495884	-5.872645	0.00053160607
5	-0.81818181	-5.7641382	-7.8143209	4.203249
6	-0.27272728	-5.2865019	-8.1669969	8.2972517
7	0.27272728	-4.7199588	-6.9306731	4.8872578
8	0.81818181	-3.9382639	-4.1053495	0.027917614
9	1.3636364	-2.1115682	0.30897404	5.8590248
10	1.9090909	2.2712705	6.3122968	16.329893
11	2.4545455	11.332147	13.904621	6.6176241
12	3	27.688623	23.085943	21.184665

Cubed function approximation

#	x	y_true	y_pred	error
1	-3	10.884967	7.8751054	9.0592659
2	-2.4545455	0.45761862	2.9241862	6.0839555
3	-1.9090909	-4.3206387	-1.3166121	9.0241758
4	-1.3636364	-5.8495884	-4.6519813	1.4342628
5	-0.81818181	-5.7641382	-6.8866165	1.2599574
6	-0.27272728	-5.2865019	-7.8252111	6.4450443
7	0.27272728	-4.7199588	-7.2724589	6.5152571
8	0.81818181	-3.9382639	-5.033054	1.1985653
9	1.3636364	-2.1115682	-0.91168968	1.4397085
10	1.9090909	2.2712705	5.2869392	9.0942578
11	2.4545455	11.332147	13.758141	5.8854505
12	3	27.688623	24.697219	8.9484989

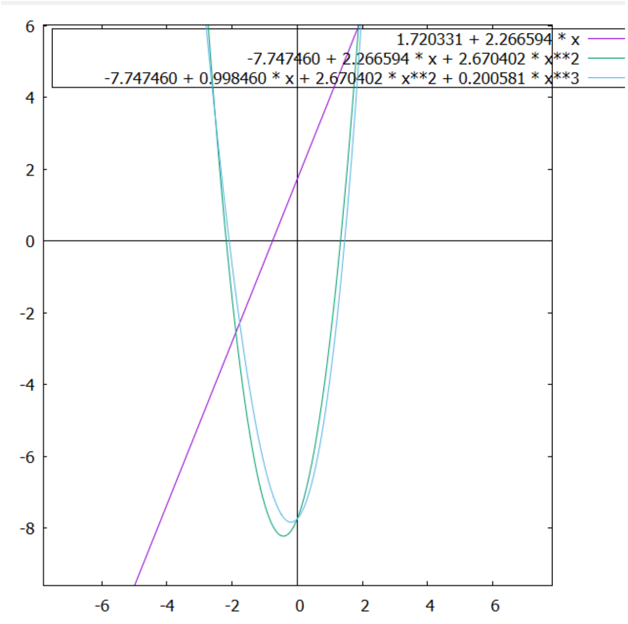
Unable to create power function approximation. There is negative x or y point

Unable to create exp function approximation. There is negative y point

Unable to create log function approximation. There is negative x point

Final answer

#	Function	MSE	SD	R2 score
1	1.720331 + 2.266594 * x	921.1432	8.7613888	0.19177987
2	-7.747460 + 2.266594 * x + 2.670402 * x^2	78.661396	2.5602961	0.93098171
3	-7.747460 + 0.998460 * x + 2.670402 * x^2 + 0.200581 * x^3	66.3884	2.3520984	0.94175016



3)

Inputted data

#	X	Y
1	-3	0.053634115
2	-2.4545455	0.092941895
3	-1.9090909	0.085362367
4	-1.3636364	0.1692235
5	-0.81818181	0.17257938
6	-0.27272728	0.33883101
7	0.27272728	0.45897889
8	0.81818181	0.67376602
9	1.3636364	0.95550823
10	1.9090909	1.3744467
11	2.4545455	2.1369615
12	3	2.9516544

Corr coefficient
0.880987

Linear function approximation

#	x	y_true	y_pred	error
1	-3	0.053634115	-0.45851808	0.26229987
2	-2.4545455	0.092941895	-0.23175893	0.10543063
3	-1.9090909	0.085362367	-0.0049997313	0.0081653089
4	-1.3636364	0.1692235	0.22175942	0.0027600223
5	-0.81818181	0.17257938	0.44851859	0.076142449
6	-0.27272728	0.33883101	0.67527775	0.11319641
7	0.27272728	0.45897889	0.90203692	0.19630042
8	0.81818181	0.67376602	1.1287961	0.20705236
9	1.3636364	0.95550823	1.3555553	0.16003762
10	1.9090909	1.3744467	1.5823144	0.043208963
11	2.4545455	2.1369615	1.8090736	0.10751045
12	3	2.9516544	2.0358328	0.83872935

Squared function approximation

#	x	y_true	y_pred	error
1	-3	0.053634115	0.23246388	0.031980085
2	-2.4545455	0.092941895	0.082323804	0.00011274386
3	-1.9090909	0.085362367	0.0075635589	0.0060526546
4	-1.3636364	0.1692235	0.0081831766	0.025933987
5	-0.81818181	0.17257938	0.084182645	0.0078139824
6	-0.27272728	0.33883101	0.23556196	0.010664497
7	0.27272728	0.45897889	0.46232113	1.117056e-05
8	0.81818181	0.67376602	0.76446014	0.0082254238
9	1.3636364	0.95550823	1.141979	0.034771354
10	1.9090909	1.3744467	1.5948777	0.048589802
11	2.4545455	2.1369615	2.1231563	0.00019058142
12	3	2.9516544	2.7268147	0.050552902

Cubed function approximation

#	x	y_true	y_pred	error
1	-3	0.053634115	0.024700165	0.00083717346
2	-2.4545455	0.092941895	0.1012114	6.8384786e-05
3	-1.9090909	0.085362367	0.13977684	0.0029609349
4	-1.3636364	0.1692235	0.16557993	1.3275586e-05
5	-0.81818181	0.17257938	0.20380418	0.00097498825
6	-0.27272728	0.33883101	0.27963305	0.0035043981
7	0.27272728	0.45897889	0.41825004	0.0016588396

Unable to create power function approximation. There is negative x or y point

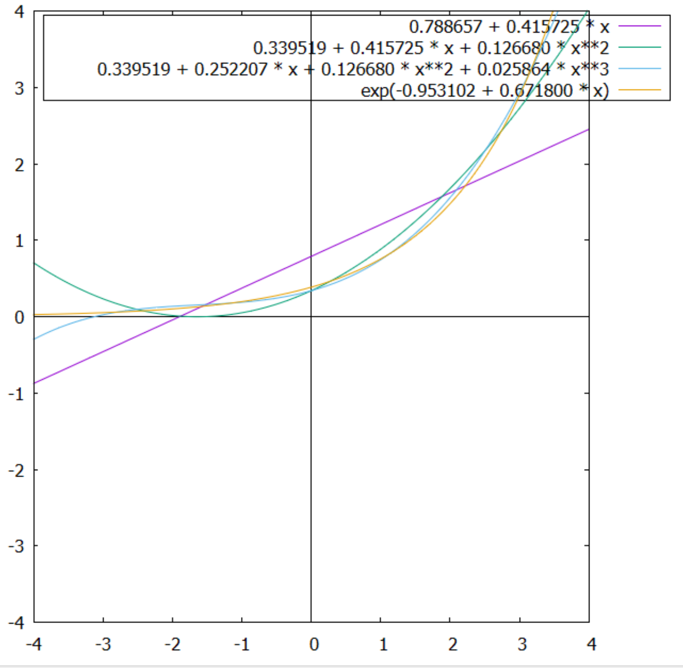
Exp function approximation

#	x	y_true	y_pred	error
1	-3	0.053634115	0.051380246	5.0799213e-06
2	-2.4545455	0.092941895	0.074120347	0.00035425068
3	-1.9090909	0.085362367	0.10692487	0.00046494159
4	-1.3636364	0.1692235	0.15424816	0.00022426082
5	-0.81818181	0.17257938	0.22251602	0.0024936678
6	-0.27272728	0.33883101	0.32099816	0.00031801044
7	0.27272728	0.45897889	0.46306699	1.6712527e-05
8	0.81818181	0.67376602	0.66801327	3.3094112e-05
9	1.3636364	0.95550823	0.96366562	6.654301e-05
10	1.9090909	1.3744467	1.3901691	0.00024719179
11	2.4545455	2.1369615	2.0054365	0.017298809
12	3	2.9516544	2.8930116	0.0034389779

Unable to create log function approximation. There is negative x point

Final answer

#	Function	MSE	SD	R2 score
1	0.788657 + 0.415725 * x	2.1208339	0.42039999	0.77613821
2	0.339519 + 0.415725 * x + 0.126680 * x^2	0.22489918	0.13689996	0.97626107
3	0.339519 + 0.252207 * x + 0.126680 * x^2 + 0.025864 * x^3	0.02084285	0.041676182	0.99779996
4	exp(-0.953102 + 0.671800 * x)	0.024961539	0.045608423	0.99736574



Вычислительная часть лабораторной работы

Таблица 1 – Линейная аппроксимация

#	x	Y_true	Y_pred	error
1	-2	1	1.3	0.090
2	-1.8	1.314	1.544	0.052
3	-1.6	1.696	1.788	0.008
4	-1.4	2.121	2.03	0.008
5	-1.2	2.534	2.276	0.067
6	-1	2.875	2.519	0.126
7	-0.8	3.104	2.763	0.115
8	-0.6	3.225	3.007	0.047
9	-0.4	3.273	3.251	0.0005
10	-0.2	3.284	3.494	0.044
11	0	3.285	3.738	0.205

Таблица 2 – Квадратичная аппроксимация

#	x	Y_true	Y_pred	error
1	-2	1	0.869	0.017
2	-1.8	1.314	1.371	0.003
3	-1.6	1.696	1.817	0.014
4	-1.4	2.121	2.204	0.007
5	-1.2	2.534	2.534	0
6	-1	2.875	2.807	0.004
7	-0.8	3.104	3.022	0.006
8	-0.6	3.225	3.180	0.002
9	-0.4	3.273	3.280	0
10	-0.2	3.284	3.322	0.001
11	0	3.285	3.307	0

Линейная функция:
3.738801 + 1.219055 * x

$$MSE = \sum (y_i - \phi_i)^2 = 0.7665$$

$$SD = \sqrt{\frac{MSE}{n}} = 0.2639$$

$$R2 = 1 - \frac{SS_p}{SS_t} = 1 - \sum \frac{MSE}{(y_i - \overline{\phi})^2} = 0.895$$

Квадратичная функция:
3.307450 - 0.218781 * x - 0.718918 * x^2

$$MSE = \sum (y_i - \phi_i)^2 = 0.057$$

$$SD = \sqrt{\frac{MSE}{n}} = 0.072$$

$$R2 = 1 - \frac{SS_p}{SS_t} = 1 - \sum \frac{MSE}{\left(y_i - \overline{\phi}\right)^2} = 0.992$$

Вывод

В ходе выполнения лабораторной работы я ознакомился с методом наименьших квадратов, ознакомился с деталями реализации данного метода на C++. Повторил основные определения математической статистики, мне понравилось.