

# 連続系アルゴリズム レポート課題9

連絡先：mail@myuuuuun.com

平成 27 年 12 月 21 日

レポート内で使用したプログラムは GitHub にアップロードしています。 <https://github.com/myuuuuun/various/tree/master/ContinuousAlgorithm/HW9/>

## 問題 1

2次元データを最小二乗法を用いて直線にフィッティングすることを考える。正規方程式を書け。

$n$  個の観測値の組を  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  とする。ただし、 $x_1, \dots, x_n$  は相異なるとする。回帰直線を  $a + bx_i$  と書くことにすると、正規方程式は、

$$\begin{aligned} A^T A z &= A^T y \\ \begin{pmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_n \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} &= \begin{pmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_n \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \\ \begin{pmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} &= \begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix} \\ \begin{pmatrix} a \\ b \end{pmatrix} &= \begin{pmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix} \end{aligned}$$

となる。

## 問題 2

問 1 の正規方程式を解け。

逆行列の公式から、

$$\begin{aligned} \begin{pmatrix} a \\ b \end{pmatrix} &= \frac{1}{n \sum x_i^2 - (\sum x_i)^2} \begin{pmatrix} \sum x_i^2 & -\sum x_i \\ -\sum x_i & n \end{pmatrix} \begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix} \\ &= \frac{1}{n \sum x_i^2 - (\sum x_i)^2} \begin{pmatrix} (\sum x_i^2)(\sum y_i) - (\sum x_i)(\sum x_i y_i) \\ (-\sum x_i)(\sum y_i) + n(\sum x_i y_i) \end{pmatrix} \end{aligned}$$

となるので、これを元に  $a, b$  を求めればよい。

$x_i$  を -10 から 10 まで 0.1 おきに 201 個とりだし、それぞれの  $x_i$  に対して  $y_i$  を  $y_i = 3x_i + 2 + \epsilon_i$ ,  $\epsilon_i \sim N(0, 5^2)$  として、観測値を作成する。これに対して、最小二乗回帰を

行う。

ソースコード: hw9-1.cpp

```
#include <stdio.h>
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <iomanip>
#include <array>
#include <cmath>
const double EPS = 1.0e-12;
using namespace std;

template <size_t row, size_t col>
    using array_2d = std::array<std::array<double, col>, row>;

// 切片, 傾きの順で値を返す
template <size_t size>
std::array<double, 2> least_squared_2d
    (std::array<double, size> x_list, std::array<double, size> y_list){
    double x_sum=0, x2_sum=0, y_sum=0, xy_sum=0;
    for(int i=0; i<size; i++){
        x_sum += x_list[i];
        x2_sum += x_list[i]*x_list[i];
        y_sum += y_list[i];
        xy_sum += x_list[i] * y_list[i];
    }

    std::array<double, 2> result;
    double denom = size * x2_sum - pow(x_sum, 2);
    result[0] = (x2_sum*y_sum - x_sum*xy_sum) / denom;
    result[1] = (size*xy_sum - x_sum*y_sum) / denom;
    return result;
}

int main(void)
{
    FILE *fp;
    int ret;
    std::array<double, 201> x_list, y_list;

    fp = fopen("data.csv", "r");
    for(int i=0; i<201; i++){
        fscanf(fp, "%lf, %lf", &x_list[i], &y_list[i]);
    }
    fclose(fp);

    std::array<double, 2> result;
    result = least_squared_2d(x_list, y_list);

    cout << result[0] << ", " << result[1] << endl;
    return 0;
}
```

出力結果:

切片=1.67285, 傾き=2.99175

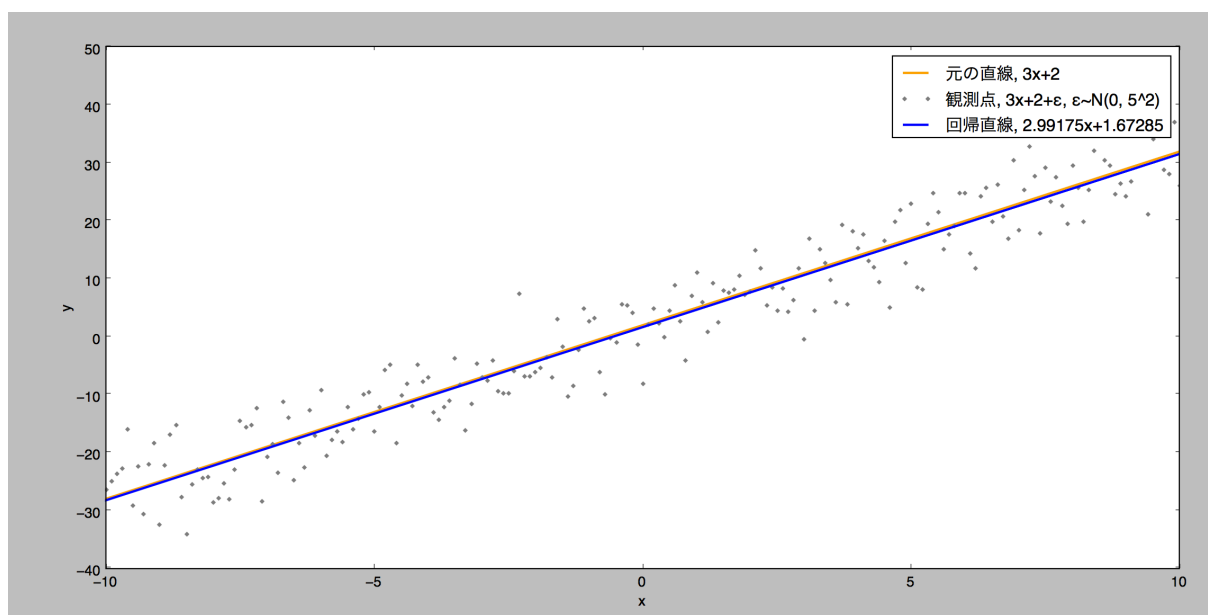


図 1: 観測値と回帰直線

となった (グラフ出力のソースコード: 9-2.py )