情報工学実験II

-データ解析 (直線によるデータの当てはめ)-

055702B　　池野谷克俊

2007 年 1 月 22 日 月曜日

1　課題

- ３つの補足プログラムの動作を確認せよ.
- 線形回帰，最小二乗法について調べて簡潔にまとめよ.
- カイ 2 乗関数について調べて簡潔にまとめよ.
- 線形回帰当てはめプログラム (lin-reg.cpp) を完成させよ.
- octave (gnuplot 等でもよい) を用いて結果をプロットし考察せよ．なお 3 種類のデータを発生させ，それぞれグラフ化する事.

2　解答

2.1　３つの補足プログラムの動作を確認せよ

補足プログラムの動作を確認した。詳細は省略する。

2.2　線形回帰，最小二乗法について調べて簡潔にまとめよ

説明変数の情報を利用して目的変数の値を予測する手法を回帰分析という。(目的変数とは、回帰式を使って予測する変数のことであり、説明変数とは、刻的変数を予測する際に使用する変数のこと) 目的変数の値を予測するための式として、$y = a_1 + a_2 x$ のような目的変数の一次式を用いるものを線形回帰という。

最小二乗法とは、回帰分析で用いられる代表的な推計方法である。最小二乗法は、データ点 (x_i, y_i) と関数 $Y(x; \{a_j\})$ の線の距離を Δ_i と置き、この距離の二乗の和 $D(\{a_j\}) = \sum_{i=1}^{N} \Delta_i^2$ が最小になるような $\{a_j\}$ を求める方法。また、データには推定誤差範囲が付いてる場合が多い。

2.3　カイ 2 乗関数について調べて簡潔にまとめよ

カイ 2 乗関数とは、
$\chi^2 = \sum\{(\text{実数} - \text{期待度数})^2/(\text{期待度数})\}$
で表される関数のこと。
最小二乗法で用いられるカイ 2 乗関数は以下のものである。

$$\chi^2(\{a_j\}) = \sum_{i=1}^{N} \frac{[Y(x_i; \{a_j\}) - y_i]^2}{\sigma_i^2}$$

2.4 線形回帰当てはめプログラム (lin-reg.cpp) を完成させよ

```
// 線形回帰プログラム
// Inputs
//   x       独立変数
//   y       従属変数
//   sigma   y における推定誤差
// Outputs
//   a_fit   a(1):切片, a(2):傾き
//   sig_a   パラメータ a() における推定誤差
//   yy      推定値
//   chisqr  カイ二乗関数

#include "NumMeth.h"

void lin_reg(Matrix x, Matrix y, Matrix sigma,
     Matrix &a_fit, Matrix &sig_a, Matrix &yy, double &chisqr) {

  //* Evaluate various sigma sums
  int i, nData = x.nRow();
  double sigmaTerm;
  double s = 0.0, sx = 0.0, sy = 0.0, sxy = 0.0, sxx = 0.0;

  for( i=1; i<=nData; i++ ) {
sigmaTerm = sigma(i)*sigma(i);
        //式 (5.10) の計算を行う
s += 1/sigmaTerm;
sx += x(i)/sigmaTerm;
sy += y(i)/sigmaTerm;
sxy += (x(i)*y(i))/sigmaTerm;
sxx += (x(i)*x(i))/sigmaTerm;
      }

  //* 切片 a_fit(1) と 傾き a_fit(2) の計算

  //ここに式 (5.11) に対応する部分を書く
a_fit(1) = (sy*sxx - sx*sxy)/(s*sxx - sx*sx);
a_fit(2) = (s*sxy - sy*sx)/(s*sxx - sx*sx);

    //* 切片と傾きの誤差範囲を計算
sig_a(1) = sqrt(sxx/(s*sxx - sx*sx));
sig_a(2) = sqrt(s/(s*sxx - sx*sx));

    //* 推定値とカイ二乗関数の計算
    chisqr = 0.0;

    //ここに書く
 for( i=1; i<=nData; i++ )
{
  sigmaTerm = sigma(i)*sigma(i);
      chisqr += ((a_fit(1)+a_fit(2)*x(i)-y(i))*(a_fit(1)+a_fit(2)*x(i)-y(i)))/
  sigmaTerm;
      yy(i) = a_fit(1)+a_fit(2)*x(i);
    }
}
```

2.5 octave (gnuplot 等でもよい) を用いて結果をプロットし考察せよ．なお 3 種類のデータを発生させ，それぞれグラフ化する事

1. c(1)=3, c(2)=2, c(3)=0, estimated error bar=100 の場合

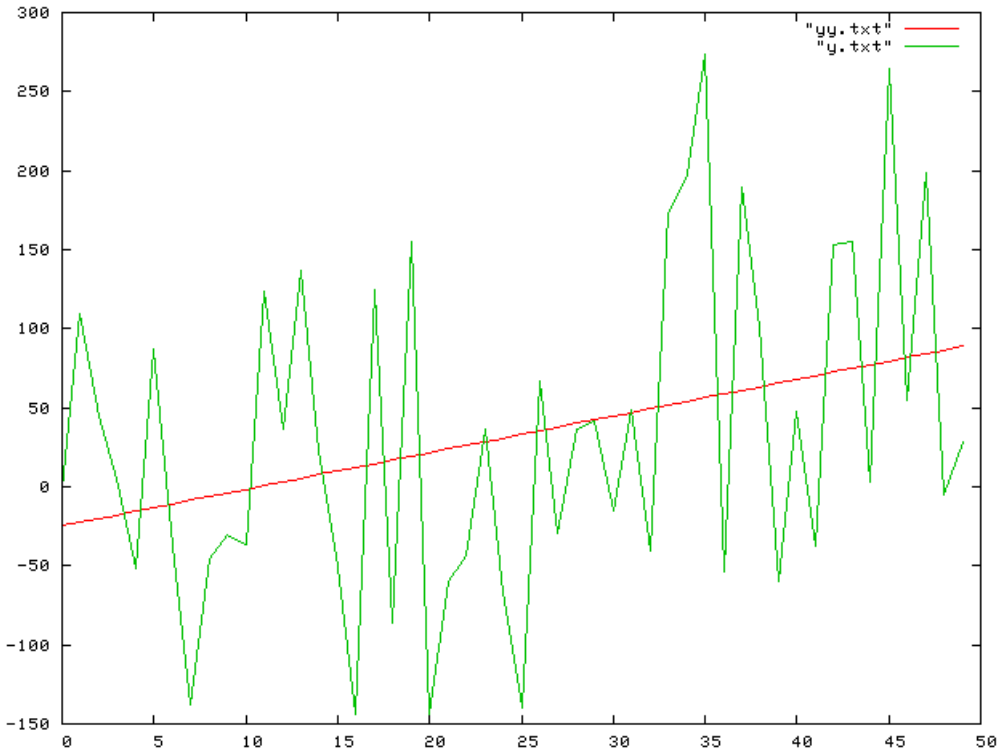

図 1: $y = 3 + 2x$,estimated error bar=100

2. c(1)=3, c(2)=2, c(3)=0, estimated error bar=1 の場合

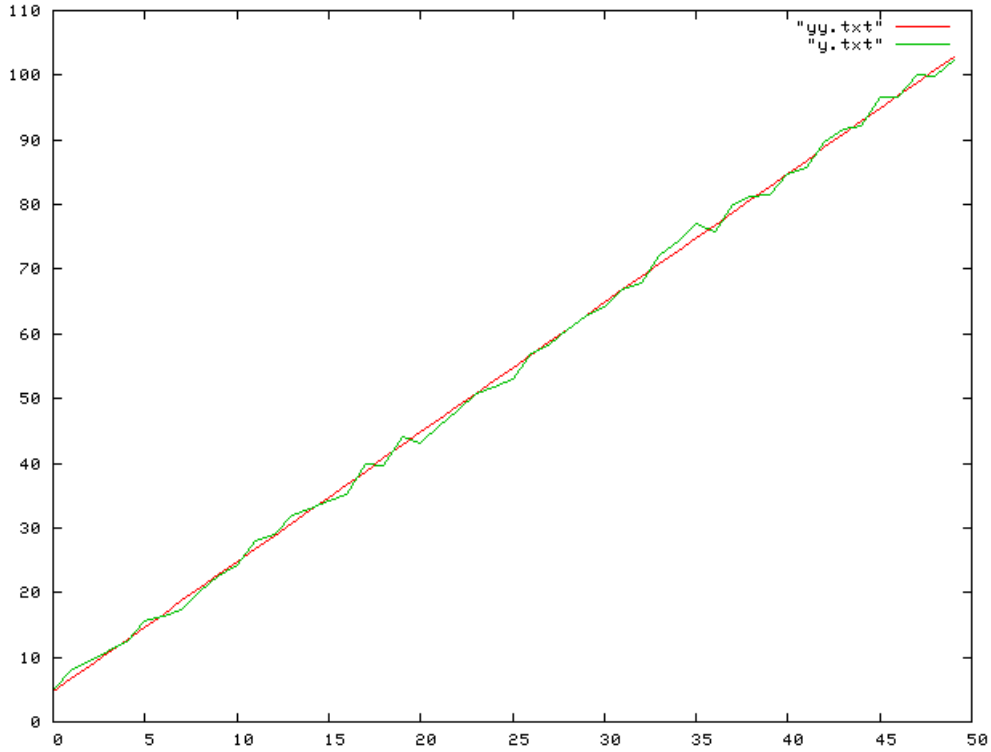

図 2: $y = 3 + 2x$,estimated error bar=1

3. c(1)=3, c(2)=2, c(3)=5, estimated error bar=1000 の場合

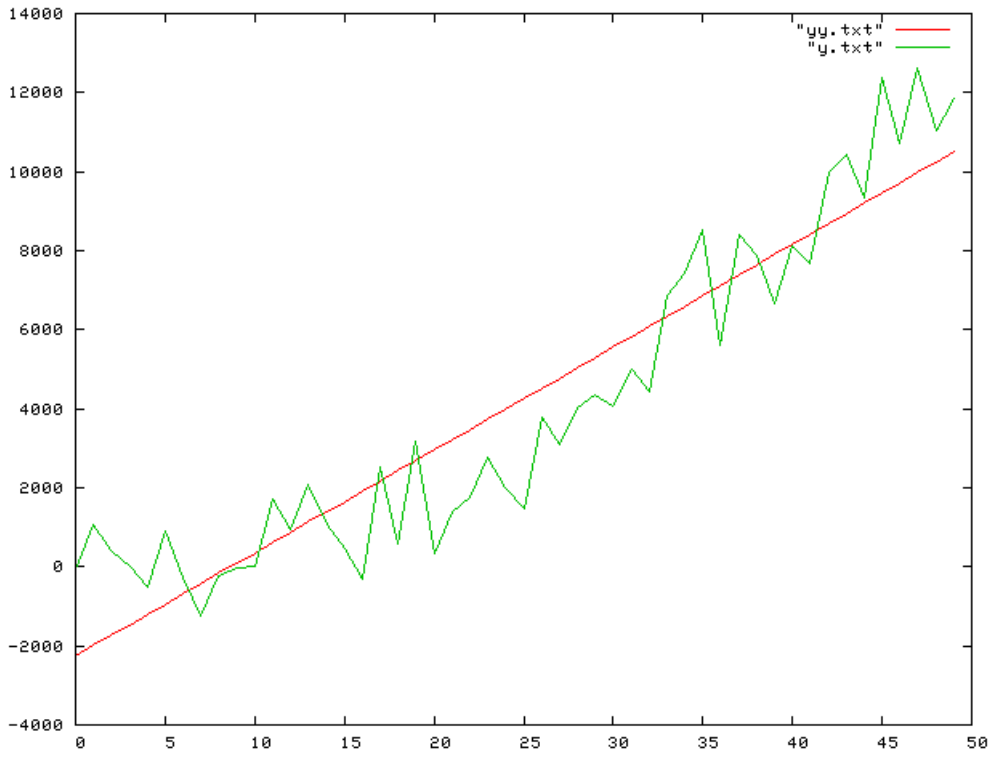

図 3: $y = 3 + 2x + 5x^2$,estimated error bar=100

図 1 と図 2 の結果から分かるように、estimated error bar の値が小さい方がデータと関数の誤差が小さい。estimated error bar の値を大きくするとデータの値のばらつきが大きくなることが分かる。このため、estimated error bar の値が小さい方がデータと関数の誤差が小さくなると考えられる。また、今回の場合、二次関数の時が一番傾きが大きかった。一次関数よりも二次関数の方が、y の値の上昇率が高いため、直線への当てはめを行った際に、傾きが大きくなると考えられる。

参考文献

[1] http://gibk26.bse.kyutech.ac.jp/ũeno/doc/seminar06.htm

[2] http://www.h7.dion.ne.jp/s̃hindan/dokusho04.html