

Student info

Problems

4.1

4.2

4.3

4.4

Extra

## Student info

---

Jingyi Zhuang

[jz2907@columbia.edu](mailto:jz2907@columbia.edu)

## Problems

---

### 4.1

Suppose that a person wants to determine the weight,  $m_j$ , of  $M = 40$  objects by weighing the first, and then weighing the rest in pairs: the first plus the second, the second plus the third, the third plus the fourth, and so on.

(A) What is the corresponding data kernel,  $\mathbf{G}$ ?

(B) Write a MatLab script that creates this data kernel and computes the covariance matrix,  $\mathbf{C}_m$ , assuming that the observations are uncorrelated and have a variance,  $\sigma_d^2 = 1\text{kg}^2$ .

(C) Make a plot of  $\sigma_{mj}$  a function of the object number,  $j$ , and comment on the results.

Solution:

(A) The data kernel  $\mathbf{G}$  is  $\mathbf{G}_{ij} = 1$  when  $i = j$  or  $i = j + 1$ .

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 1 \end{bmatrix}$$

with shape of  $(40 \times 40)$ .

(B) From the description, we know that  $\Sigma_d = \sigma_d^2 \mathbf{I}$

$\Sigma_m = \mathbf{M} \Sigma_d \mathbf{M}^T$ , while  $\mathbf{M} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T$

The Script is:

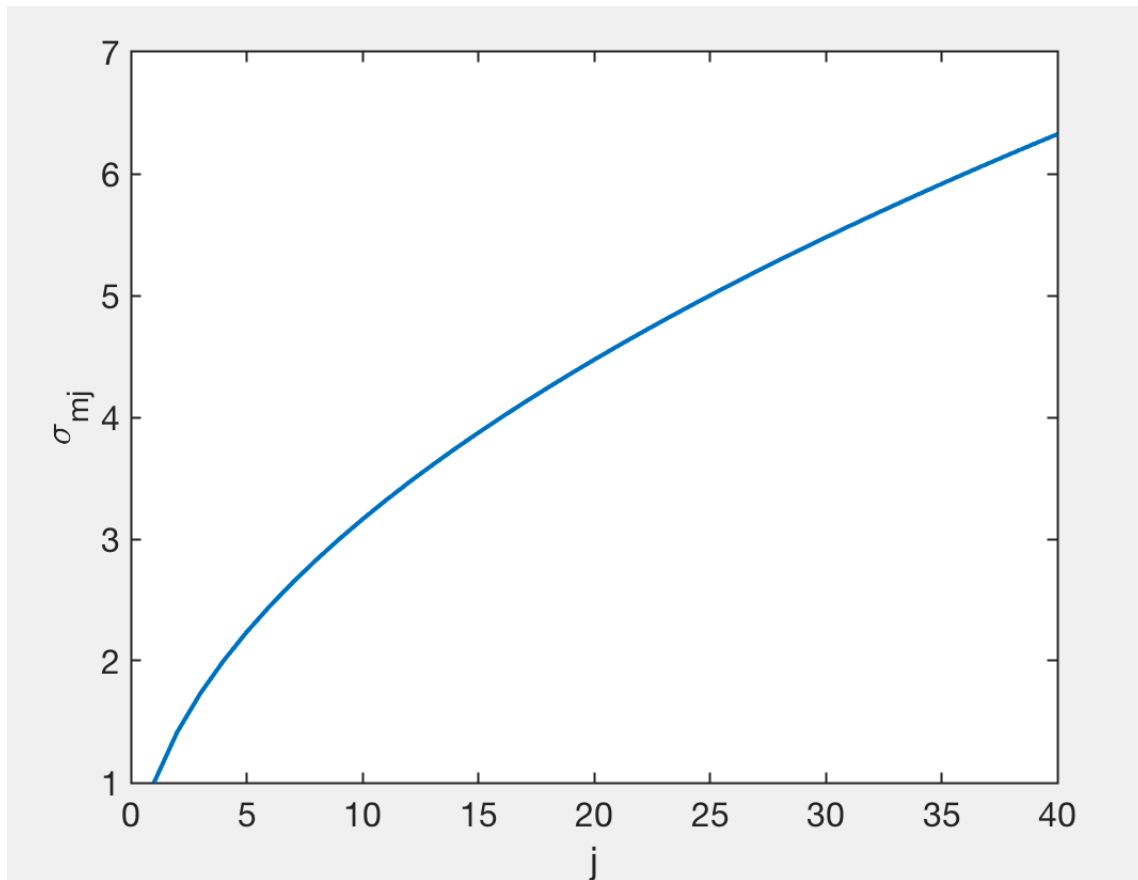
```
n = 40;
l = [1 zeros(1,n-1)];
c = [1 1 zeros(1,n-2)];
G = toeplitz(c,l);

sig_d = 1;
I = eye(n,n);

covar_d = sig_d ^ 2 * I;
M = (G' * G) \ G';
covar_m = M * covar_d * M';

disp(covar_m);
```

(C)



Script:

```
n = 40;
l = [1 zeros(1,n-1)];
c = [1 1 zeros(1,n-2)];
G = toeplitz(c,l);

sig_d = 1;
I = eye(n,n);

covar_d = sig_d ^ 2 * I;
M = (G' * G) \ G';
covar_m = M * covar_d * M';
```

```

sig_m = sqrt(diag(covar_m));

figure(1);

plot(1:n,sig_m,"LineWidth",2);
xlabel("j");
ylabel(sprintf("\\sigma_{mj}"));
set(gca,'FontSize',16,'LineWidth',1);

```

Comment:

It is a square root function  $\sigma_{m_j} = \sqrt{j}$

---

## 4.2

Consider the equation  $d_i = m_1 \exp(-m_2 t_i)$ . Why cannot this equation be arranged in the linear form,  $\mathbf{d} = \mathbf{G}\mathbf{m}$ ? (A) Show that the equation can be linearized into the form,  $\mathbf{d}' = \mathbf{G}'\mathbf{m}'$ , where the primes represent new, transformed variables, by taking the logarithm of the equation. (B) What would have to be true about the measurement error in order to justify solving this linearized problem by least squares? (Notwithstanding your answer, this problem is often solved with least squares in a let's-hope-for-the-best mode).

Answer:

(A)

Take the logarithm of the equation:

$$\begin{aligned}
 \ln(d_i) &= \ln(m_1 \exp(-m_2 t_i)) \\
 &= \ln(m_1) - m_2 t_i
 \end{aligned}$$

Then we can get  $\mathbf{d}' = \mathbf{G}'\mathbf{m}'$ , where

$$\mathbf{d}' = \begin{bmatrix} \ln(d_1) \\ \ln(d_2) \\ \ln(d_3) \\ \vdots \\ \ln(d_N) \end{bmatrix}, \mathbf{G}' = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ 1 & t_3 \\ \vdots & \vdots \\ 1 & t_N \end{bmatrix}, \mathbf{m}' = \begin{bmatrix} \ln(m_1) \\ -m_2 \end{bmatrix}$$

(B)

We should get positive measurements.

The logarithm of measurement data error be normal distributed.

- It might be a good idea to avoid using least squares.

## 4.3

(A) What is the relationship between the elements of the matrix,  $\mathbf{G}^T \mathbf{G}$ , and the columns,  $\mathbf{c}^{(j)}$ , of  $\mathbf{G}$ ?

Answer:

We have  $\mathbf{G} = [\mathbf{c}^{(1)} \quad \mathbf{c}^{(2)} \quad \dots \quad \mathbf{c}^{(M)}]$ , so

$$\mathbf{G}^T \mathbf{G} = \begin{bmatrix} \mathbf{c}^{(1)T} \\ \mathbf{c}^{(2)T} \\ \vdots \\ \mathbf{c}^{(M)T} \end{bmatrix} [\mathbf{c}^{(1)} \quad \mathbf{c}^{(2)} \quad \dots \quad \mathbf{c}^{(M)}] = \begin{bmatrix} \mathbf{c}^{(1)T} \mathbf{c}^{(1)} & \mathbf{c}^{(1)T} \mathbf{c}^{(2)} & \dots & \mathbf{c}^{(1)T} \mathbf{c}^{(M)} \\ \mathbf{c}^{(2)T} \mathbf{c}^{(1)} & \mathbf{c}^{(2)T} \mathbf{c}^{(2)} & \dots & \mathbf{c}^{(2)T} \mathbf{c}^{(M)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}^{(M)T} \mathbf{c}^{(1)} & \mathbf{c}^{(M)T} \mathbf{c}^{(2)} & \dots & \mathbf{c}^{(M)T} \mathbf{c}^{(M)} \end{bmatrix}$$

The  $i^{th}$  row,  $j^{th}$  column element in  $\mathbf{G}^T \mathbf{G} = \mathbf{c}^{(i)T} \cdot \mathbf{c}^{(j)}$

(B) Under what circumstance is  $\mathbf{G}^T \mathbf{G}$  a diagonal matrix?

Answer:

Since  $\mathbf{G}^T \mathbf{G}$  is diagonal matrix, the  $i^{th}$  row,  $j^{th}$  column ( $i \neq j$ ) element of this matrix should be 0.

When  $\mathbf{c}^{(i)T} \cdot \mathbf{c}^{(j)} = 0$ , when  $i \neq j$ .

The column vectors  $\mathbf{c}^{(j)}$  of  $\mathbf{G}$  are orthogonal to each other.

(C) What is the form of the covariance matrix in this case?

Answer:

$$\mathbf{C}_m = \sigma_d^2 [\mathbf{G}^T \mathbf{G}]^{-1}$$

From (B) we know  $\mathbf{G}^T \mathbf{G}$  is a diagonal matrix, and the inverse of a diagonal matrix is still diagonal matrix,

$$[\mathbf{G}^T \mathbf{G}]^{-1} = \begin{bmatrix} \frac{1}{\mathbf{c}^{(1)T} \mathbf{c}^{(1)}} & 0 & \dots & 0 \\ 0 & \frac{1}{\mathbf{c}^{(2)T} \mathbf{c}^{(2)}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\mathbf{c}^{(M)T} \mathbf{c}^{(M)}} \end{bmatrix}$$

And this diagonal matrix times the variance is still a diagonal matrix.

$\mathbf{C}_m$  is also a diagonal form.

(D) What is the form least-squares solution in this case? Is it harder or easier to compute than the case where  $\mathbf{G}^T \mathbf{G}$  is not diagonal?

Answer:

$$\begin{aligned}
\mathbf{m}^{est} &= [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} \\
&= \begin{bmatrix} \frac{1}{\mathbf{c}^{(1)T} \mathbf{c}^{(1)}} & 0 & \dots & 0 \\ 0 & \frac{1}{\mathbf{c}^{(2)T} \mathbf{c}^{(2)}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\mathbf{c}^{(M)T} \mathbf{c}^{(M)}} \end{bmatrix} \begin{bmatrix} \mathbf{c}^{(1)T} \\ \mathbf{c}^{(2)T} \\ \vdots \\ \mathbf{c}^{(M)T} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{\mathbf{c}^{(1)T} \mathbf{c}^{(1)}} & 0 & \dots & 0 \\ 0 & \frac{1}{\mathbf{c}^{(2)T} \mathbf{c}^{(2)}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\mathbf{c}^{(M)T} \mathbf{c}^{(M)}} \end{bmatrix} \begin{bmatrix} \mathbf{c}^{(1)T} \mathbf{d} \\ \mathbf{c}^{(2)T} \mathbf{d} \\ \vdots \\ \mathbf{c}^{(M)T} \mathbf{d} \end{bmatrix}
\end{aligned}$$

If we assume  $\mathbf{c}^{(j)} = \begin{bmatrix} c_{1j} \\ c_{2j} \\ \vdots \\ c_{Nj} \end{bmatrix}$

$$\begin{aligned}
\mathbf{m}^{est} &= \begin{bmatrix} \frac{1}{\sum_1^N c_{i1}^2} & 0 & \dots & 0 \\ 0 & \frac{1}{\sum_1^N c_{i2}^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\sum_1^N c_{iM}^2} \end{bmatrix} \begin{bmatrix} \sum_1^N c_{i1} d_i \\ \sum_1^N c_{i2} d_i \\ \vdots \\ \sum_1^N c_{iM} d_i \end{bmatrix} \\
&= \begin{bmatrix} \frac{\sum_1^N c_{i1} d_i}{\sum_1^N c_{i1}^2} \\ \frac{\sum_1^N c_{i2} d_i}{\sum_1^N c_{i2}^2} \\ \vdots \\ \frac{\sum_1^N c_{iM} d_i}{\sum_1^N c_{iM}^2} \end{bmatrix}
\end{aligned}$$

It's easier to compute!

(E) Examine the straight line case in this context?

Answer:

For straight line case we have:

$$\mathbf{G} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}$$

So,  $\mathbf{G}^T \mathbf{G}$  will be:

$$\mathbf{G}^T \mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_1 & x_2 & x_3 & \dots & x_N \end{bmatrix} \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} = \begin{bmatrix} N & \sum_1^N x_i \\ \sum_1^N x_i & \sum_1^N x_i^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \sum_1^N x_i^2 \end{bmatrix}$$

The column vectors  $\mathbf{c}^{(j)}$  of  $\mathbf{G}$  are:

$$\mathbf{c}^{(1)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \mathbf{c}^{(2)} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix},$$



If  $\mathbf{G}^T \mathbf{G}$  is a diagonal matrix, we should have  $\sum_1^N x_i = 0$ , while  $\mathbf{c}^{(1)T} \mathbf{c}^{(2)} = \sum_1^N x_i$ , so that means  $\mathbf{c}^{(1)T} \mathbf{c}^{(2)} = 0$ , when  $\mathbf{G}^T \mathbf{G}$  is a diagonal matrix.

Also, the covariance matrix will be

$$\mathbf{C}_m = \sigma_d^2 [\mathbf{G}^T \mathbf{G}]^{-1} = \sigma_d^2 \begin{bmatrix} \frac{1}{N} & 0 \\ 0 & \frac{1}{\sum_1^N x_i^2} \end{bmatrix}$$

So we can find that the covariance is also a diagonal matrix.

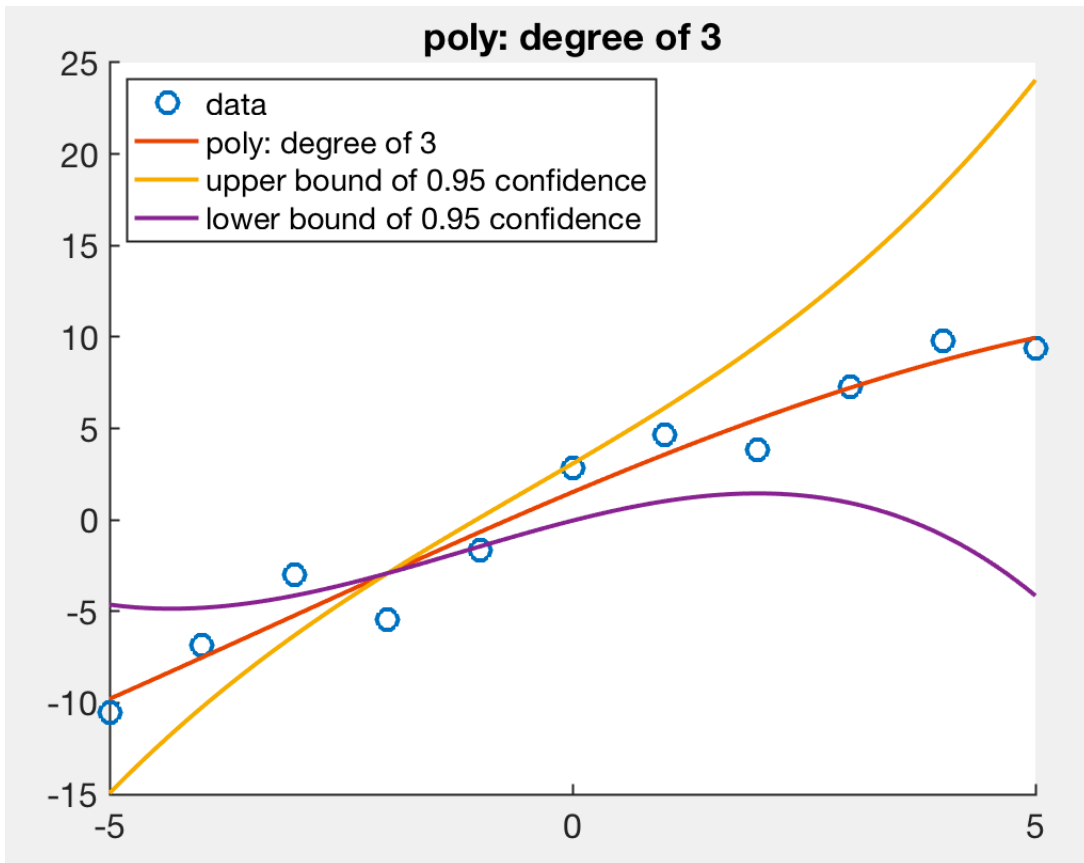
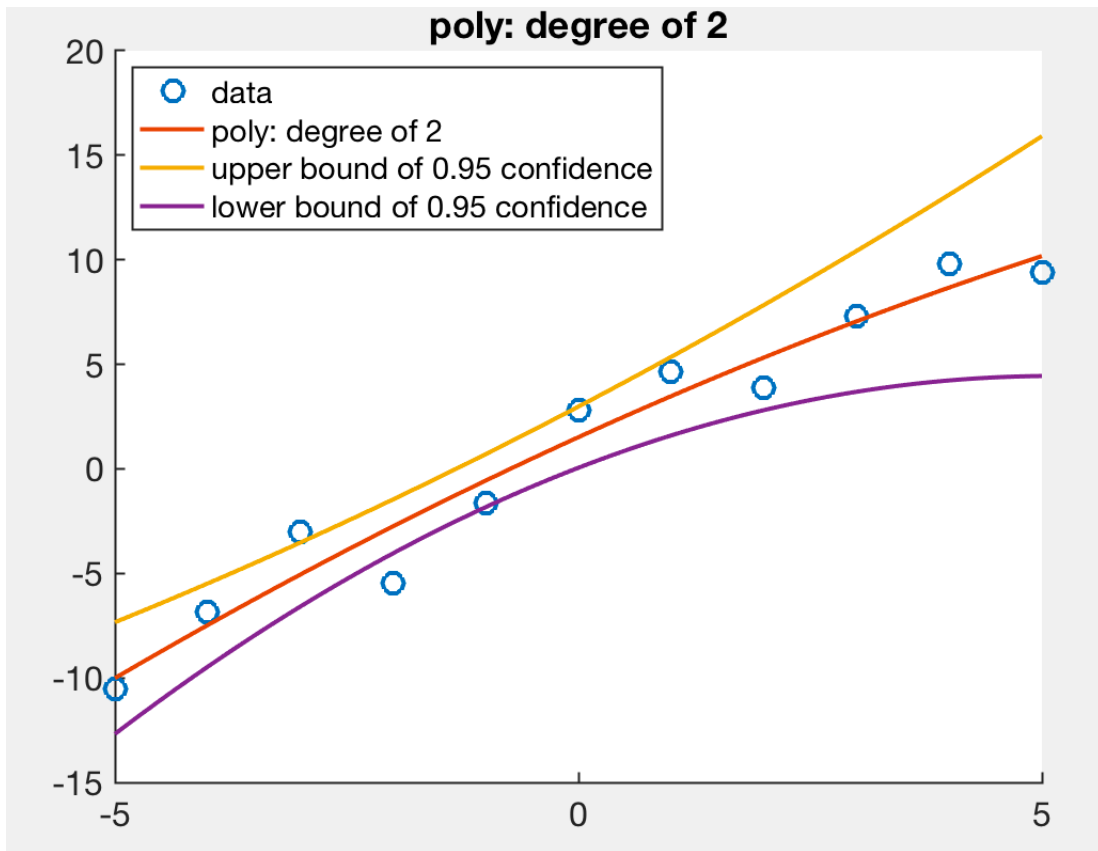
The least square solution is

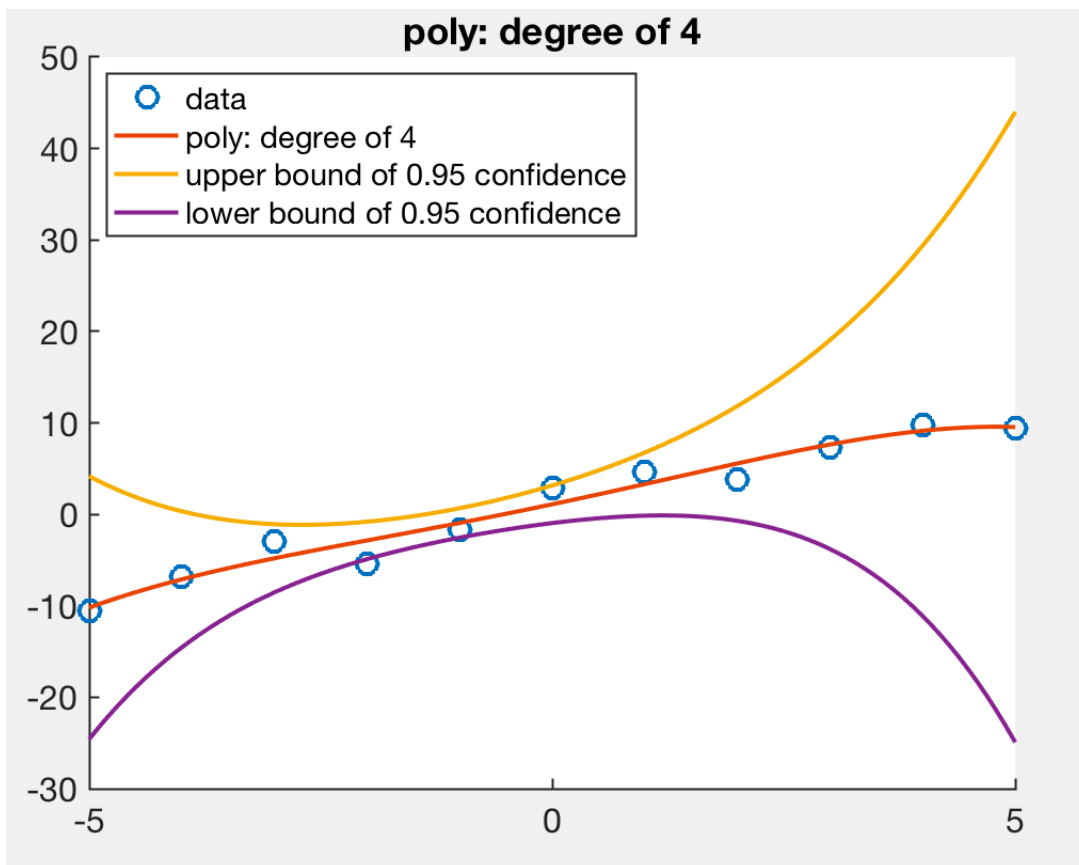
$$\begin{aligned} \mathbf{m}^{est} &= [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} = \begin{bmatrix} \frac{1}{N} & 0 \\ 0 & \frac{1}{\sum_1^N x_i^2} \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{N} & 0 \\ 0 & \frac{1}{\sum_1^N x_i^2} \end{bmatrix} \begin{bmatrix} \sum_1^N d_i \\ \sum_1^N x_i d_i \end{bmatrix} \\ &= \begin{bmatrix} \frac{\sum_1^N d_i}{N} \\ \frac{\sum_1^N x_i d_i}{\sum_1^N x_i^2} \end{bmatrix} \end{aligned}$$

## 4.4

The dataset shown in Figure 4.4 is in the file, `linedata01.txt`. Write MatLab scripts to least-squares fit polynomials of degree 2, 3, and 4 to the data. Make plots that show the observed and predicted data. Display the value of each coefficient and its 95% confidence limits. Comment on your results.

Answer:





The raw output is:

The coefficients of polynomial of degree 2 are

1.5138

2.0190

-0.0573

The 0.95 confidence upper bound coefficients of polynomial of degree 2 are

2.9772

2.3253

0.0524

The 0.95 confidence lower bound coefficients of polynomial of degree 2 are

0.0505

1.7126

-0.1669

The coefficients of polynomial of degree 3 are

1.5138  
2.1296  
-0.0573  
-0.0062

The 0.95 confidence upper bound coefficients of polynomial of degree 3 are

3.0691  
2.9687  
0.0593  
0.0372

The 0.95 confidence lower bound coefficients of polynomial of degree 3 are

-0.0415  
1.2906  
-0.1738  
-0.0497

The coefficients of polynomial of degree 4 are

1.0996  
2.1296  
0.0866  
-0.0062  
-0.0058

The 0.95 confidence upper bound coefficients of polynomial of degree 4 are

3.1566  
3.0051  
0.5421  
0.0391  
0.0118

The 0.95 confidence lower bound coefficients of polynomial of degree 4 are

-0.9575

1.2541

-0.3689

-0.0515

-0.0233

Script:

```
clearvars;
D = load('linedata01.txt');

x = D(:,1);
d = D(:,2);
N = length(x);

for i = 2 : 4
    G = ones(N,i+1);
    for n = 1 : i
        G(:,n+1) = x.^n;
    end

    figure(i-1);

    set(gca, 'FontSize',16, 'LineWidth',1);
    hold on;

    plot(x,d, "o", 'LineWidth',2, "MarkerSize",10);
    hold on;

    mest = (G'*G)\(G'*d);
```

```

nline = 100;
xline=linspace(min(x),max(x),nline);
Gline= ones(nline,i+1);
for n = 1 : i
    Gline(:,n+1) = xline.^n;
end

plot(xline,Gline*mest,"-", 'LineWidth',2, "MarkerSize",10);

fprintf("The coefficients of polynomial of degree %g
are\n",i);
disp(mest);

e = d - G*mest;
E = e' * e;
sigma2d= E / (length(d)- (i+1));
Cm = sigma2d * inv(G'*G);

sigma2m = diag(Cm);

mu = mest + 2 * sqrt(sigma2m);
ml = mest - 2 * sqrt(sigma2m);

fprintf("The 0.95 confidence upper bound coefficients of
polynomial of degree %g are\n",i);
disp(mu);

fprintf("The 0.95 confidence lower bound coefficients of
polynomial of degree %g are\n",i);
disp(ml);

plot(xline,Gline*mu,"-", 'LineWidth',2);
plot(xline,Gline*ml,"-", 'LineWidth',2);

```

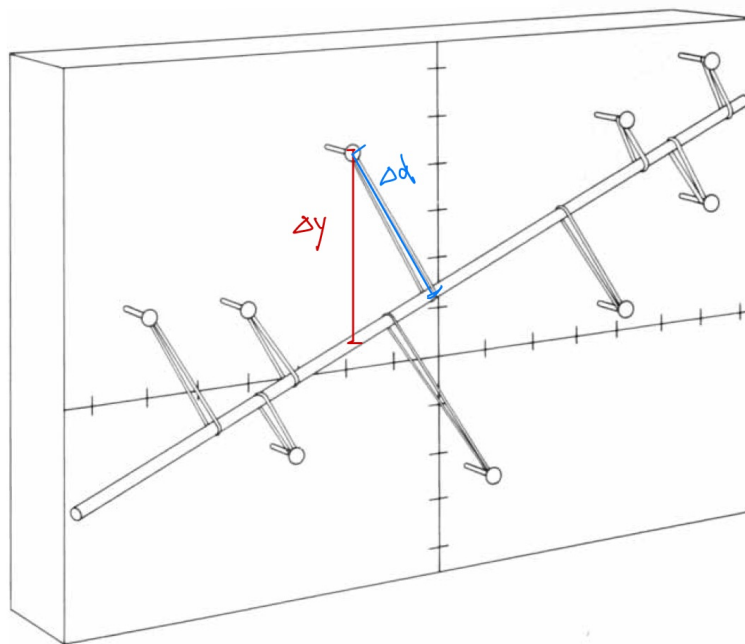
```

    legend({"data", sprintf("poly: degree of %g",i), "upper
bound of 0.95 confidence", "lower bound of 0.95 confidence"
}, 'Location', 'northwest')
    title( sprintf("poly: degree of %g",i));
end

```

## Extra

Give two reasons why the analog gadget illustrated in "Files/Homework/Best-fit\_line\_gadget.pdf" will not generally give the same answer as a least-squares fit. (Hint: if the rubber bands are perfectly elastic, the restoring force is proportional to their length).



*A gadget for finding the line that best fits a series of data points*

Reason 1).

Assume the plot has horizontal x & vertical y axis.

The least-square fit cares about  $\Delta y$ , which is a vertical difference of  $y_{pred}(x_0)$  and  $y(x_0)$ , while the the this line gadget cares about the distance between the data point and the predicted line,  $\Delta d$ , which is perpendicular to the predicted to the predicted line.

Reason 2).

The least square fit is minimizing the sum of  $\Delta y^2$ , while this line gadget is balancing the whole system of stick and rubber bands to get the net force = 0.

Also, the slope of line gadget is also strongly depend on the size (length/scale) of this stick, since the focus, or the 'center', of this stick will cause very different result from the lever principle. The lever principle will provide this gadget a 'built-in' weighted effect associated how far the data point is away from the stick's center. And least-square fit does not weight the data in that way.