1. Consider the following series of measurements of the unknown value x:

$$y_i = a_i x + \varepsilon_i, \quad = 1, \ldots, n,$$

where y_i are measurement results, a_i are known coefficients, and ε_i represent random error of measurement and are independent identically distributed (i.i.d.) with zero mean and variance σ^2:

$$E\varepsilon_i = 0, \quad E\varepsilon_i^2 = \sigma^2, \quad i = 1, \ldots, n$$

(a) What function of $y_1, \ldots, y_n$ (and of $a_1, \ldots, a_n$) would you use as a good estimate for $\widehat{x}$ for x?

Although not obvious, a good estimate for x would be

$$\widehat{x} = \frac{\sum a_i y_i}{\sum a_i^2}.$$

We will show in the next part that this estimate minimizes the sum of the squared distances to the measured data.

(b) Is this estimate optimal in any sense?

For any given estimate x, the sum of the squared differences between observations and estimates of the observations is given by

$$Q(x) = \sum_{i=1}^{n} (y_i - a_i x)^2,$$

for which

$$\begin{aligned} Q(x) &= \left(\sum a_i^2\right) x^2 - \left(2 \sum a_i y_i\right) x + \sum y_i^2 \\ &= \left(\sum a_i^2\right) x^2 - \left(2 \sum a_i y_i\right) x + \frac{\left(\sum a_i y_i\right)^2}{\sum a_i^2} - \frac{\left(\sum a_i y_i\right)^2}{\sum a_i^2} + \sum y_i^2 \\ &\overset{\dagger}{=} \left(\left(\sum a_i^2\right)^{1/2} x - \frac{\sum a_i y_i}{\left(\sum a_i\right)^{1/2}} \right)^2 - \frac{\left(\sum a_i y_i\right)^2}{\sum a_i^2} + \sum y_i^2. \end{aligned}$$

Since only the first term depends on x, and each term is positive, $Q(x)$ is minimized when the first squared term vanishes. This happens precisely when

$$\widehat{x} = \frac{\sum a_i y_i}{\sum a_i^2} =: \frac{U}{V},$$

where U and V are the respective sums.

(c) Is it a biased or an unbiased estimate?

Observe

$$\begin{aligned} \widehat{x} - x &= \left(\sum a_i^2\right)^{-1} \sum a_i (a_i x + \varepsilon_i) - x \\ &= \left(\sum a_i^2\right)^{-1} \left(\sum a_i^2\right) x - x + \left(\sum a_i^2\right)^{-1} \sum a_i \varepsilon_i \\ &\overset{*}{=} \left(\sum a_i^2\right)^{-1} \sum a_i \varepsilon_i. \end{aligned}$$

Taking expectations on both sides and using $E\varepsilon_i = 0$, we see that $\widehat{x}$ is unbiased for x.

(d) What is its variance (expressed through σ^2)?

Since $E\widehat{x} = x$, the variance of $\widehat{x}$ is given by

$$\begin{aligned}
\mathrm{Var}(\widehat{x}) = E(\widehat{x} - x)^2 &\overset{*}{=} E\left(\left(\sum a_i^2\right)^{-1} \sum a_i \varepsilon_i\right)^2 \\
&= E\left(\left(\sum a_i^2\right)^{-2} \sum_{i,j} a_i a_j \varepsilon_j \varepsilon_i\right) \\
&= \left(\sum a_i^2\right)^{-2} \sum_{i,j} a_i a_j E\varepsilon_i \varepsilon_j \\
&= \left(\sum a_i^2\right)^{-2} \sum_{i} a_i^2 \sigma^2 \\
&= \frac{\sigma^2}{V}.
\end{aligned}$$

(e) How would you estimate σ^2 if it is unknown?

Continuing from the last line of the derivation of $\widehat{x}$, we have

$$\begin{aligned}
Q(\widehat{x}) &\overset{\dagger}{=} 0 + \sum y_i^2 - \frac{U^2}{V} \\
&= \sum (a_i x + \varepsilon_i)^2 - V\widehat{x}^2 \\
&= \sum (x^2 a_i^2 + 2x a_i \varepsilon_i + \varepsilon_i^2) - V\widehat{x}^2 \\
&= 2x \sum a_i \varepsilon_i + \sum \varepsilon_i^2 - V(\widehat{x}^2 - x^2).
\end{aligned}$$

Taking expectations on both sides and using $\frac{\sigma^2}{V} = E(\widehat{x} - x)^2 = E\widehat{x}^2 - 2xE\widehat{x} + x^2 = E\widehat{x}^2 - x^2$,we obtain

$$\begin{aligned}
EQ(\widehat{x}) &= 0 + n\sigma^2 - V\frac{\sigma^2}{V} \\
&= (n-1)\sigma^2.
\end{aligned}$$

Hence, an unbiased estimate of σ^2 is given by

$$\widehat{\sigma^2} = \frac{Q(\widehat{x})}{n-1} = \frac{1}{n-1}\left(W - \frac{U^2}{V}\right),$$

where $W = \sum y_i^2$.

(f) What would you use as an estimate for $\mathrm{Var}(\widehat{x})$ if σ^2 is unknown?

Based on (d) and (e), an unbiased estimate for $\mathrm{Var}(\widehat{x})$ is

$$\widehat{\mathrm{Var}(\widehat{x})} = \frac{\widehat{\sigma^2}}{V} = \frac{1}{n-1}\left(\frac{W}{V} - \widehat{x}^2\right)$$

(g) Suppose that the variance σ^2 is known. What "canonical information" would be sufficient to extract from the series of observations

$$(y_1, a_1), \ldots, (y_n, a_n), \quad i = 1, \ldots, n$$

in order to compute the estimate $\widehat{x}$, and its variance $\mathrm{Var}(\widehat{x})$?

From the previous derivations, we need only the quantities $U = \sum a_i y_i$ and $V = \sum a_i^2$ to compute

$$\widehat{x} = \frac{U}{V}, \quad \text{and} \quad \mathrm{Var}(\widehat{x}) = \frac{\sigma^2}{V}.$$

(h) Suppose that the variance σ^2 is NOT known. What "canonical information" would be sufficient to extract from the series of observations in order to compute $\widehat{x}$,$\widehat{\sigma^2}$, and $\widehat{\mathrm{Var}(\widehat{x})}$.

From the previous derivations, we need only the quantities $U = \sum a_i y_i, V = \sum a_i^2, W = \sum y_i^2$, and n to compute

$$\widehat{x} = \frac{U}{V}, \quad \text{and} \quad \widehat{\mathrm{Var}(\widehat{x})} = \frac{1}{n-1}\left(\frac{W}{V} - \widehat{x}^2\right).$$

(i) How should we update such "information" when a new observation (y_{n+1}, a_{n+1}) arrives?

Since each element of the canonical information is expressed as a sum, we can easily update via the following schematic:

$$\begin{array}{l} (U, V, W, n) \longrightarrow \\ \qquad\qquad\quad \nearrow \\ (y_{n+1}, a_{n+1}) \end{array} \oplus \longrightarrow (U + y_{n+1}a_{n+1}, V + a_{n+1}^2, W + y_{n+1}^2, n+1)$$

(j) How should we "combine" (merge) two pieces of "canonical information"?

Again, because the canonical information is expressed through sums, information is easily combined via the following schematic:

$$\begin{array}{l} (U, V, W, n) \\ \qquad\qquad\quad \searrow \\ (U', V', W', n') \end{array} \oplus \longrightarrow (U + U', V + V', W + W', n + n')$$

2. Write a program which illustrates simple linear regression (or a more general variant of linear regression) and implements accumulation of canonical information.

(a) For some fixed parameters a and b (or, in a more general case $a_1, \ldots, a_m$) generate a sequence of "observations" (x_i, y_i):

$$y_i = f(x_i) = \varepsilon_i$$

where

$$f(x) = a + bx \quad \text{or} \quad f(x) = a_1 + a_2 x + a_3 x^2 + \cdots + a_m x^{m-1}$$

ε_i are i.i.d. with zero mean and $\mathrm{E}\varepsilon_i^2 = \sigma^2$. Values x_i can be generated randomly with some mean and variance.

(b) Accumulate canonical information, i.e., at each step when a new observation (x_i, y_i) is produced update canonical information.

(c) Illustrate $\widehat{f(x)}$.

(d) Illustrate $\mathrm{Var}(\widehat{f(x)})$, assuming that σ^2 is known.

(e) Illustrate $\mathrm{Var}\widehat{f(x)})$, assuming that σ^2 is NOT known.

In your repoprt present the source code and a few (around 3) nice graphs showing estimations for "small","intermediate", and "large" numbers of observations.

Solution

The attached `MATLAB` codes implement curve fitting using the methods outlined in Lectures 04a-04b. Data (x_i, y_i) were generated in three stages using built-in `MATLAB` psuedo-random number generation. The x_i are uniformly distributed data in $[0, 1]$, hence $\mathrm{E}x_i = \frac{1}{2}$. The y_i are such that

$$y_i = a + bx_i + \varepsilon_i$$

where ε_i are normally distributed with $\mathrm{E}\varepsilon_i = 0$ and $\mathrm{E}\varepsilon_i^2 = .2$ and $a = 2$ and $b = 3$.

The results of the simulation are presented in the three figures below

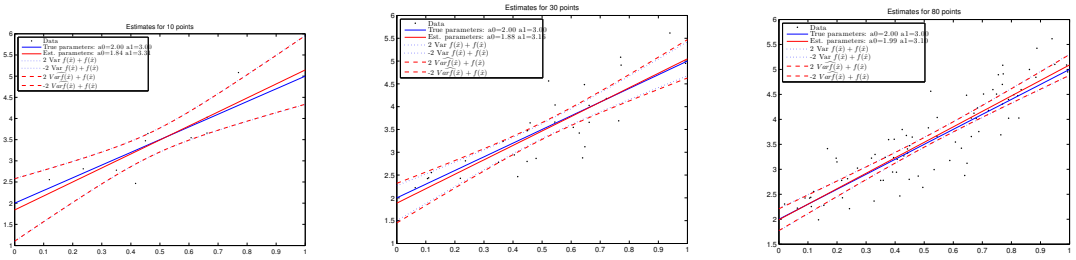

Figure 1: Estimates and quantified uncertainty in them are illustrated for $n = 10, 30, 80$ points. The data points $ax_i + b$ are shown as black points, the line $ax + b$ is shown in blue, and the estimated line $\widehat{a}x + \widehat{b}$ is shown in red. The variance for each estimate $\widehat{x}$ when σ^2 is known is shown as a band with dotted blue lines with width $4\mathrm{Var}(\widehat{f(x)})$. Similarly, the band for when σ^2 is unknown is shown with red dashed lines. Note that estimation for a given x is most precise near $\mathrm{E}x_i = .5$

□

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                               %
% problem2.m                                                                    %
%                                                                               %
% These codes illustrate an implementation of fitting a curve to data using    %
% linear regression by updating the canonical information for linear regression%
% presented in Lectures 04a - 04b of Math 491 - Spring 2015.                    %
%                                                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
F = @(x)( [ones(size(x)), x] ); % Simple linear regression functions f1(x) = 1, f2(x) = x
a = [2 3]'; % Parameters to fit
sigma2 = .2; % Epsilon variance

% Something more complicated is possible
%F = @(x)( [ones(size(x)), x, x.^2, sin(x)] );
%a = [3 1 4 1]';

n1 = 10; % How many points to fit initially
n2 = 20; % How many additional points at to fit at second stage
n3 = 50; % How many additional points at to fit at third stage

x1 = rand(n1,1);% Generate n1 points in [0,1]
y1 = F(x1)*a + sqrt(sigma2)*randn(n1,1);% Generate noisey response
x = x1;
y = y1;

% Function for returning canonical data
get_canonical_data = @(x,y)( struct('T',F(x)'*F(x), ... % Form T matrix
                                    'b',F(x)'*y,    ... % Form B'y
                                    'V',y'*y,       ... % Form sum of squares for y
                                    'n',size(x,1)));    % The number of data points
info1 = get_canonical_data(x,y);
info = info1;

% Function for transforming canonical information into estimates
get_estimates = @(info) ( struct('ahat',info.T \ info.b, ...
                                 'cov_ahat',sigma2*inv(info.T), ...
                                 'hat_cov_ahat', (info.V - info.b'*inv(info.T)*info.b)/(info
                                     .n - size(info.b,1))*inv(info.T) ,...
                                 'hat_sigma2', (info.V - info.b'*inv(info.T)*info.b)/(info.n
                                     - size(info.b,1))));
estimates = get_estimates(info);
figure(1)
plot_estimates(info,estimates,x,y,F,a,sigma2); % See plot_estimates.m
title(sprintf('Estimates for %d points',n1))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Generate n2 new data points
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
input(sprintf('Strike Enter to add %d points',n2))
x2 = rand(n2,1);% Generate n1 points in [0,1]
y2 = F(x2)*a + sqrt(sigma2)*randn(n2,1);% Generate noisey response
x = [x1;x2];
y = [y1;y2];

info2 = get_canonical_data(x2,y2);

% Function for combining canonical information.
combine_information = @(info1,info2)(struct('T',info1.T+info2.T,...
                                            'b',info1.b+info2.b,     ...
                                            'V',info1.V+info2.V,         ...
                                            'n',info1.n+info2.n));

info = combine_information(info1,info2);
estimates = get_estimates(info);

figure(2)
```

```matlab
plot_estimates(info,estimates,x,y,F,a,sigma2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Generate n3 new data points
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
input(sprintf('Strike Enter to add %d points',n3))
x3 = rand(n3,1);% Generate n1 points in [0,1]
y3 = F(x3)*a + sqrt(sigma2)*randn(n3,1);% Generate noisey response
x = [x;x3]; % Although it is not necessary to keep all of the data,
y = [y;y3]; % we do so to illustrate in the following plots

% Having defined all of the functions, the algorithm for updating information is concisely presented below
info3 = get_canonical_data(x3,y3);
info = combine_information(info,info3);
estimates = get_estimates(info);
figure(3)
plot_estimates(info,estimates,x,y,F,a,sigma2);
```

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                            %
% plot_estimates.m                                                           %
%                                                                            %
% This function plots curve fitting estimates from canonical information and %
% data generated in the file problem2.m                                      %
%                                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [fhat,var_fhat,f] = plot_estimates(info,estimates,x,y,F,a,sigma2)
  xx = [0:.01:1]'; % Make a "fine" grid to estimate on
  fhat = F(xx)*estimates.ahat; % Estimate at each point
  var_fhat = sigma2*dot( F(xx)*inv(info.T), F(xx),2 ); % Sigma known
  hat_var_fhat = estimates.hat_sigma2*dot( F(xx)*inv(info.T), F(xx),2 ); % Sigma unknown
  f = plot(x,y,'k.',...
           xx,F(xx)*a,'b-', ...
           xx,fhat,'r-',...
           xx,fhat+2*sqrt(var_fhat),'b:',...
           xx,fhat-2*sqrt(var_fhat),'b:',...
           xx,fhat+2*sqrt(hat_var_fhat),'r--',...
           xx,fhat-2*sqrt(hat_var_fhat),'r--');
  legend({'Data',...
         sprintf('True parameters: a0=%.2f a1=%.2f',a),...
         sprintf('Est. parameters: a0=%.2f a1=%.2f',estimates.ahat),...
         '2 Var $f(\hat{x}) + f(\hat{x})$',...
         '-2 Var $f(\hat{x}) + f(\hat{x})$',...
         '2 $\widehat{Var f(\hat{x})} + f(\hat{x})$',...
         '-2 $\widehat{Var f(\hat{x})} + f(\hat{x})$'},...
         'Location','Northwest',...
         'Interpreter','Latex');
  title(sprintf('Estimates for %d points',info.n))
```