AMTH142                                      Practical 7

Scilab — Least Squares

This practical is an extension of Lecture 13 on least squares.

Contents

1  Linear Least Squares                      2
   1.1  Fitting Polynomials to Data          2
   1.2  Evaluating Polynomials              2
   1.3  Example                             3
   1.4  Polynomial Interpolation            3
1 Linear Least Squares

1.1 Fitting Polynomials to Data

We will begin by writing a function to fit a polynomial to data. We will assume that the data \( x \) and \( y \) are given by column vectors, and we wish to fit a polynomial of degree \( n \) to the data. The main part of the calculation is creating the matrix

\[
\begin{bmatrix}
1 & x_1 & x_1^2 & \cdots & x_1^n \\
1 & x_2 & x_2^2 & \cdots & x_2^n \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_m & x_m^2 & \cdots & x_m^n
\end{bmatrix}
\]

function \( p = \text{polyfit}(x, y, n) \)

\[
m = \text{length}(x);
\]

if (length(y) \( \neq m) \)

   error("Data have different sizes")
end

aa = zeros(m, n+1)
aa(:,1) = ones(m,1)
for \( k = 2:n+1 \)
   aa(:,k) = x.^((k-1))
end

\( p = \text{aa}\backslash y \)
endfunction

1.2 Evaluating Polynomials

We also need one to evaluate the polynomial. The most efficient way to evaluate polynomials is by Horner’s method. If

\[
p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n
\]

then, Horner’s method evaluates it using

\[
p(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + x(\ldots (a_{n-1} + x a_n) \ldots)))))
\]

function \( y = \text{polyval}(p, x) \)

\( n = \text{length}(p) \)
\begin{verbatim}
    y = zeros(x)
    for k = n:-1:1
      y = y.*x + p(k)
    end
endfunction
\end{verbatim}

1.3 Example
We will repeat the straight line example from Lecture 13:
\begin{verbatim}
-->
x = (0:20)';

-->
y = -3*x + 5 + rand(x, 'normal');

-->
p = polyfit(x, y, 1)

\end{verbatim}
\begin{verbatim}
p =
! 5.5144187 !
! - 3.060465 !

-->
xx = 0:0.1:20;

-->
yy = polyval(p, xx);

-->
plot2d(x, y, -1)

-->
plot2d(xx, yy)
\end{verbatim}

1.4 Polynomial Interpolation
If the number of coefficients in the polynomial is equal to the number of data points, then the least squares problem becomes an interpolation problem. We can redo the example of polynomial interpolation from Lecture 12.
\begin{verbatim}
-->
x = [1 2 4 5 7]';

-->
y = [2 4 1 7 3]';

-->
p = polyfit(x, y, 4)
\end{verbatim}
p =
!  - 22.666667 !
!  43.683333 !
!  - 23.508333 !
!  4.8166667 !
!  - 0.325 !

--> xx = 1:0.01:7;

--> yy = polyval(p, xx);

--> plot2d(x, y, -1)

--> plot2d(xx, yy)