AMTH142

Lecture 13

Least Squares Curve Fitting

Contents

1 Least Squares
   1.1 The General Problem ........................................... 2

2 Linear Least Squares .............................................. 3
   2.1 The Computational Problem .................................... 3
   2.2 Scilab ............................................................. 4
1 Least Squares

In the previous lecture we looked at interpolation problems where we were given some data \((x_i, y_i), i = 1, \ldots, n\) and we wanted to find a function \(y = f(x)\) which interpolated the data so that \(y_i = f(x_i), i = 1, \ldots, n\). This approach is only applicable when the data are relatively free from error, otherwise the interpolating function will exhibit the same kind of fluctuations that are present in the data.

When we are dealing with data containing random errors the most common approach to fitting a function to data is least squares data fitting.

1.1 The General Problem

The general least squares problem can be formulated as follows: given data \((x_i, y_i), i = 1, \ldots, n\) and a function

\[
y = f(x, a_1, a_2, \ldots, a_k)
\]

depending on the parameters \(a_1, \ldots, a_k\), let

\[
r_i = y_i - f(x_i, a_1, \ldots, a_k)
\]

denote the distance between the data point \((x_i, y_i)\) and the graph of \(f(x, a_1, a_2, \ldots, a_k)\).

![Graph](image)

We want to find values for the parameters \(a_1, \ldots, a_k\) which minimizes the sum of squares of the \(r_i\)

\[
\text{Minimize } S^2 = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} (y_i - f(x_i, a_1, \ldots, a_k))^2
\]
Typically the number of parameters, \( k \), is much smaller than the number of data points, \( n \).

Least squares problems are classified as linear or nonlinear, depending on whether or not the function \( f(x, a_1, \ldots, a_k) \) depends linearly or non-linearly on the parameters \( a_1, \ldots, a_k \).

## 2 Linear Least Squares

The most common data fitting problem is fitting a straight line
\[
y = ax + b
\]
to data. This is an example of a linear least squares problem with two parameters, in this case \( a \) and \( b \), to be determined. More generally, fitting a polynomial of degree \( k \)
\[
y = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k
\]
to data is another example of linear least squares. Although \( y \) is a nonlinear function of \( x \) it is a linear function of the parameters \( a_0, \ldots, a_k \).

Linear least squares have a number of similarities to the general interpolation problem discussed in the previous lecture. Like interpolation it can be formulated as a problem in linear algebra.

### 2.1 The Computational Problem

As for the general interpolation problem, suppose we want to fit a function
\[
y = a_1 f_1(x) + a_2 f_2(x) + \cdots + a_k f_k(x)
\]
where the \( f_j(x) \) are given basis functions. The least squares approach leads us to minimizing the sum of squares
\[
S^2 = \sum_{i=1}^{n} [y_i - (a_1 f_1(x_i) + a_2 f_2(x_i) + \cdots + a_k f_k(x_i))]^2
\]
Note that the sum is over the data points, and once the data and basis functions are given, the sum of squares \( S^2 \) is a function of the parameters \( a_1, \ldots, a_k \) only.
The individual terms inside the square brackets
\[ r_i = y_i - (a_1 f_1(x_i) + a_2 f_2(x_i) + \cdots + a_k f_k(x_i)) \]
can be written in vector/matrix form
\[
\begin{bmatrix}
  r_1 \\
  r_2 \\
  \vdots \\
  r_n
\end{bmatrix} = 
\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_n
\end{bmatrix} - 
\begin{bmatrix}
  f_1(x_1) & f_2(x_1) & \cdots & f_k(x_1) \\
  f_1(x_2) & f_2(x_2) & \cdots & f_k(x_2) \\
  \vdots & \vdots & \ddots & \vdots \\
  f_1(x_n) & f_2(x_n) & \cdots & f_k(x_n)
\end{bmatrix} 
\begin{bmatrix}
  a_1 \\
  a_2 \\
  \vdots \\
  a_k
\end{bmatrix}
\]

Our aim is to find \(a_1, \ldots, a_k\) to minimize \(S^2 = \sum r_i^2\). [Note that when we have the same number of parameters \(a_i\) as data points, the problem reduces to an interpolation problem. In this case we could find values for the parameters so that all the \(r_i\) are zero.]

In the case of a straight line, our basis functions are \(f_1(x) = 1\) and \(f_2(x) = x\) and the equations above become
\[
\begin{bmatrix}
  r_1 \\
  r_2 \\
  \vdots \\
  r_n
\end{bmatrix} = 
\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_n
\end{bmatrix} - 
\begin{bmatrix}
  1 & x_1 \\
  1 & x_2 \\
  \vdots & \vdots \\
  1 & x_n
\end{bmatrix} 
\begin{bmatrix}
  a_1 \\
  a_2 \\
  \vdots \\
  a_k
\end{bmatrix}
\]

### 2.2 Scilab

We will look at example of fitting a straight line to data. First we will generate some \(x\) and \(y\) data where
\[ y = -3x + 5 + \text{random perturbation} \]

\[ \text{--->} x = (0:20)' ; \]
\[ \text{--->} y = -3*x + 5 + \text{rand}(x, \ 'normal') ; \]
\[ \text{--->} \text{plot2d}(x, y, -1) \]
Figure 1: The Data

To fit a straight line to this data we first create the matrix in previous section

```matlab
--> aa = [ones(x) x]
    aa =

    1.0000   0.0000
    1.0000   1.0000
    1.0000   2.0000
    1.0000   3.0000
    1.0000   4.0000
```
The least squares problem is solved with the backslash operator, \, the same way as linear equations are solved.

```
--a = aa\y
a  =

!  4.7074061 !
! - 2.989726 !
```

The components of the solution, a, are the coefficients of the basis functions $f_1(x) = 1$ and $f_2(x) = x$. We can now evaluate and plot the straight line

```
-->xx = 0:0.1:20;

-->yy = a(1) + a(2)*xx;

-->plot2d(xx, yy)
```
Here is another example. Again we will generate some data, but this time using trigonometric functions:

```matlab
--> x = (0:20)';

--> y = 4*sin(x) + 3*sin(2*x) + 2*sin(4*x) + 0.5*rand(x, 'normal');

--> plot2d(x, y, -1)
```
In this case our basis functions will be

\[ f_1(x) = \sin(x) \]
\[ f_2(x) = \sin(2x) \]
\[ f_3(x) = \sin(4x) \]

Our computations follow the same pattern as before:

--->aa = [sin(x) sin(2*x) sin(4*x)];

--->a = aa\y
a =

! 3.9324226 !
\[ a(1) \cdot \sin(xx) + a(2) \cdot \sin(2 \cdot xx) + a(3) \cdot \sin(4 \cdot xx) \]

-->plot2d(xx, yy)