Applied Numerical Analysis | (7th Edition)

Problem

Use the data of APP with several approaches to extrapolate backward to estimate the expected expenditure for 1980. How do these values compare to 492 billion, the amount actually spent?

APP

The cost of government welfare programs adds significantly to our taxes. The table below gives data for several years:

Year	**Expenditures in billions of dollars**
1985	731
1986	782
1987	833
1988	886
1989	956
1990	1049
1991	1159
1992	1267
1993	1367
1994	1436
1995	1505

Use the data between 1991 to 1994 to estimate what the value would be in 1995 and compare to the value in the table. Do this

a. From a cubic interpolating polynomial.

b. From the least-squares line.

c. From the least-squares quadratic.

d. From a cubic spline.

From each of these, project to find what one would anticipate the expenditures for the year 2000 might be; then find what the actual expenditures were for comparison.

Step-by-step solution

Step 1 of 11

a) When using numerical approximation techniques, it is useful to have programs written to aid in numerical calculations to avoid calculations by hand. Using Matlab, create an

M-file named "lagrange.m" which will contain Matlab code that implements the algorithm to interpolate Lagrange polynomials and return the coefficients of the polynomial in Lagrange form

Inside this file, write the code to implement the method. One example of a correctly executable routine based on the algorithm description in the text is as follows:

```
function rtn = lagrange ( X, F, x0 )
%given a vector list of x-values and a vector
%list of y-values,the lagrangian interpolation of
%x0 is calculated.
n = length ( X );
coeffmatrix = zeros(1,n);
f=0;
%Lagragian Interpolation loop
for i = 1:n
P=1; coeff=1;
for j = 1:n
if (j ~= i)
P=P*(x0-X(j))/(X(i)-X(j));
coeff=coeff/(X(i)-X(j));
end
end
f = f + P * F(i);
coeffmatrix(i)=coeff*F(i);
end
%outputs
f
```

Comment

Step 2 of 11

Letting 0 correspond to 1985, 1 correspond to 1986, and so forth, use the following Matlab commands to extrapolate the value at -5 corresponding to 1980.

INPUT:

```
>> format shortG
>> x=[0,1,2,3,4,5,6,7,8,9,10];
>> f=[731,782,833,886,956,1049,1159,1267,1367,1436,1505];
>> lagrange(x,f,5)
```

OUTPUT:

f =

640226

Comment

Step 3 of 11

b) For almost all least-squares problems in this text, the data can be fit to a **linear combination of basis functions** $\{1, x_1, x_2, \ldots, x_n\}$ (these are function vectors such as *x*, *y*, z^3, $\ln x$, e^x, and so forth) of the form

$$y = \beta_1 + \beta_2 x_1 + \beta_3 x_2 + \cdots + \beta_{n+1} x_n$$

which when fitting the equation to *m* data points leads to the system of equations (expressed compactly in matrix notation)

$$\sum_{j=1}^{n+1} X_{ij}\beta_j = y_i, (i = 1, 2, \ldots, m),$$

$$\mathbf{y} = \mathbf{X}\beta$$

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1,(n+1)} \\ X_{21} & X_{22} & \cdots & X_{2,(n+1)} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{m,(n+1)} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n+1} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

which is known as the **general linear least-squares model.**

Comment

Step 4 of 11

The get the least squares solution to the system, the deviation function

$$\begin{aligned} S(\beta) &= \|\mathbf{y} - \mathbf{X}\beta\|^2 \\ &= (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) \\ &= \mathbf{y}^T\mathbf{y} - \beta^T\mathbf{X}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{X}\beta \\ &= \mathbf{y}^T\mathbf{y} - 2\beta^T\mathbf{X}^T\mathbf{y} + \beta^T\mathbf{X}^T\mathbf{X}\beta. \end{aligned}$$

must be minimized, so differentiating with respect to β and setting $S(\beta) = 0$ yields the normal equation and solving for β yields

$$\beta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

This form leads to a computationally efficient solution to the problems and numerical calculations can be directly entered into the Matlab command window.

Comment

Step 5 of 11

To get the solution using a least-squares line, use the following sequence of Matlab commands

INPUT:

>> X=[ones(size(x')) x'];

>> B=(X'*X)\(X'*f')

OUTPUT:

B =

676.77

82.3

So the model is

$$f(x) = 82.3x + 676.77$$

Comment

Step 6 of 11

To extrapolate at $x = -5$, use the Matlab command

Comment

Step 7 of 11

INPUT:

>> 676.77+82.3*(-5)

OUTPUT:

ans =

265.57

Comment

Step 8 of 11

c) To get the solution using a least-squares quadratic, use the following sequence of Matlab commands

Comment

Step 9 of 11

INPUT:

>> X=[ones(size(x')) x' x'.^2];

>> B=(X'*X)\(X'*f')

OUTPUT:

B =

715.32

56.601

2.5699

So the model is

$$f(x) = 2.5699x^2 + 56.601x + 715.32$$

Comment

Step 10 of 11

To extrapolate at $x = -5$, use the Matlab command

Comment

Step 11 of 11

INPUT:

>> 715.32+56.601*(-5)+2.5699*(-5)^2

OUTPUT:

ans =

496.56

Looking at the Matlab outputs, the least-square quadratic is the closest approximation to the true value 492.

Comment

Was this solution helpful? 0 0