Simple Least Squares Fits for Communication Times

William Gropp
www.cs.illinois.edu/~wgropp

Fitting Data to a Linear Equation

- It is common to measure communication times as a function of message length, getting a table of times like this:

N	Time
8192	1.71E-05
16384	1.92E-05
32768	2.37E-05
65536	3.47E-05
131072	5.38E-05
262144	9.47E-05

Fitting Data to a Linear Equation

- We'd like to determine s and r so that the formula
  $T(n) = s + rn$
  "fits" this data. By fit, we mean that the difference between s+rn and T(n) is as small as possible (in some particular sense).

Fitting Data to a Linear Equation

- One common measure is to consider the vector of values n and the corresponding values T(n); these are columns in the table. Then consider
  norm(T-(s+rn))
  The value of s and r that we seek minimize this norm.

Fitting Data to a Linear Equation

- This is the *Linear Least Squares* problem
- The general version is:
  - Solve A x = b, where A is an n x m matrix, b is an n x 1 vector, and x is an m x 1 vector of the coefficients.
- This (except in special cases) doesn't have a solution, so "solve" means to find x such that norm(Ax-b) is minimized.

Creating the Matrix A and Vector b

- For our case, the coefficents are (s,r). We want each row of the matrix to represent one equation T(n) = s + rn. Thus, the equations are

$$s + r * n_1 = T_1$$
$$s + r * n_2 = T_2$$
$$s + r * n_3 = T_3$$

….

- Thus, the matrix A and Vector b are

$$\begin{pmatrix} 1 & n_1 \\ 1 & n_2 \\ 1 & n_3 \\ \ldots & \end{pmatrix} \text{ and } \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ \ldots \end{pmatrix}$$

1867

PARALLEL@ILLINOIS

Solving for The Parameters

- We now need to "solve" $A x = b$. There are several ways to do this, some (much) better than others. For these problems, a good approach is to use a matrix computation program, such as matlab or octave. These implement robust and accurate algorithms for the linear least squares problem. In both Matlab and Octave, the operation
  A \ b
  will compute the least squares solution to $Ax = b$

Example Using Octave

```
>> nrndv=[8192
16384
32768
65536
131072
262144];
>> arndv=[ones(6,1),nrndv];
>> trndv = [ 1.40E-05
1.61E-05
2.08E-05
3.20E-05
5.13E-05
9.19E-05];
>> coefrndv = arndv \ trndv
coefrndv =   1.1221e-05
  3.0764e-10
```