# Linear Algebra

William F. Moss[*]

## 1 Least Squares Linear Models

Suppose we are given a set of $m$ data points

$$(t_1, y_1), (t_2, y_2), \ldots, (t_m, y_m)$$

which we believe to be samples from some underlying function. Our problem is to approximate the underlying function using all available information. In general, this is a difficult problem because we may know very little about the underlying function and the data may be corrupted with noise. There are two common approaches to approximation of the underlying function: interpolation and curve fitting. In interpolation a function, called an interpolant, is constructed whose graph passes through the data points. In curve fitting a function is constructed which passes "close to" the data points but not necessarily through them. This function is said to "fit" the data. When a significant amount of noise is present in the data, an interpolant may not provide a satisfactory approximation of the underlying function because the interpolant may oscillate rapidly. In this case, a slowly varying function that "fits" the data may provide a better approximation of the underlying function.

A linear model of the form

$$f(t; \mathbf{c}) = \sum_{j=1}^{n} c_j \phi_j(t) \tag{1}$$

is often used to interpolate or fit data. It is a linear combination of "basis" functions, $\phi_1(t), \cdots, \phi_n(t)$, which are selected based on practical considerations, visual inspection of the data, and knowledge of the processes giving rise to the data. The

[*]Department of Mathematical Sciences, Clemson University, Clemson, SC 29634–1907, U.S.A. (`bmoss@math.clemson.edu`). Copyright ©1995,1996 William F. Moss. All rights reserved.

linear model depends on an independent variable $t$ and on a set of $n$ coefficients, $c_1, \ldots, c_n$. The $\mathbf{c}$ in $f(t; \mathbf{c})$ denotes a column vector containing these coefficients. In interpolation, the number of basis functions, $n$, is chosen to be equal to the number of data points, $m$, while for curve fitting $n$ is typically much smaller than $m$. We will assume that $n \leq m$.

As an example, suppose a straight line is to be used to fit $m \geq 2$ data points. We set $n = 2$ and $\phi_1(t) = 1$ and $\phi_2(t) = t$. Then $f$ has the form $f(t; \mathbf{c}) = c_1 + c_2 t$. Generally, this line will be an interpolant only in case $m = 2$. A straight line fit is often used in statistics where this line is called the regression line.

If the linear model is constructed so that it changes much more slowly than an interpolant, then curve fitting can be used as a noise filter. This approach is illustrated in the first problem of this project. Here, samples taken from a straight line are corrupted with noise generated randomly, and then a straight line is fit to the data. The original line is very nearly recovered.

In the mathematical discussion below, we will use the following matrix notation which was introduced in the Orthogonal Methods course outline. Let

$$\mathbf{y} = [y_1, \ldots, y_m]^T, \quad \mathbf{c} = [c_1, \ldots, c_n]^T \quad , A = (a_{ij}) = (\phi_j(t_i)).$$

The $m \times n$ matrix $A$ is often called the design matrix.

The interpolation problem is to find $\mathbf{c}$ so that the interpolation equations

$$f(t_i; \mathbf{c}) = y_i, \quad i = 1, \ldots, m, \tag{2}$$

have a solution. Substituting equation (1) into (2), we find that

$$f(t_i; \mathbf{c}) = \sum_{j=1}^{n} c_j \phi_j(t_i) = (A\mathbf{c})_i \,, \quad i = 1, \ldots, m$$

which is a system of $m$ equations and $n$ unknowns with the matrix form

$$A\mathbf{c} = \mathbf{y}. \tag{3}$$

The curve fitting problem is to find $\mathbf{c}$ so that the graph of $f$ passes "close to" the data points. The most commonly used method for defining "close to" is called the least squares method. In the least squares method, $\mathbf{c}$ is determined by minimizing the sum of the squares of the vertical deviations between the graph of $f$ and the data points. The problem is to find $\mathbf{c}_{ls}$ so that

$$\sum_{i=1}^{m} [f(t_i; \mathbf{c}_{ls}) - y_i]^2 = \min_{\mathbf{c} \in \mathbb{R}^{n \times 1}} \sum_{i=1}^{m} [f(t_i; \mathbf{c}) - y_i]^2 \,. \tag{4}$$

We can convert (4) into matrix form as follows. First, note that

$$f(t_i; \mathbf{c}) = \sum_{j=1}^{n} c_j \phi_j(t_i) = (A\mathbf{c})_i \quad \text{and} \quad \sum_{i=1}^{m} [(A\mathbf{c})_i - y_i]^2 = \|A\mathbf{c} - \mathbf{y}\|_2^2. \tag{5}$$

From (4) and (5) we see that

$$\sum_{i=1}^{m} [f(t_i; \mathbf{c}_{ls}) - y_i]^2 \le \sum_{i=1}^{m} [f(t_i; \mathbf{c}) - y_i]^2 \quad \text{for all } \mathbf{c} \tag{6}$$

implies that

$$\|A\mathbf{c}_{ls} - \mathbf{y}\|_2^2 \le \|A\mathbf{c} - \mathbf{y}\|_2^2 \quad \text{for all } \mathbf{c}. \tag{7}$$

Taking the square root of both sides of (7), yields the standard matrix form of the linear least squares problem. Find $\mathbf{c}_{ls}$ so that

$$\|A\mathbf{c}_{ls} - \mathbf{y}\|_2 = \min_{\mathbf{c} \in \mathbb{R}^{n\times 1}} \|A\mathbf{c} - \mathbf{y}\|_2. \tag{8}$$

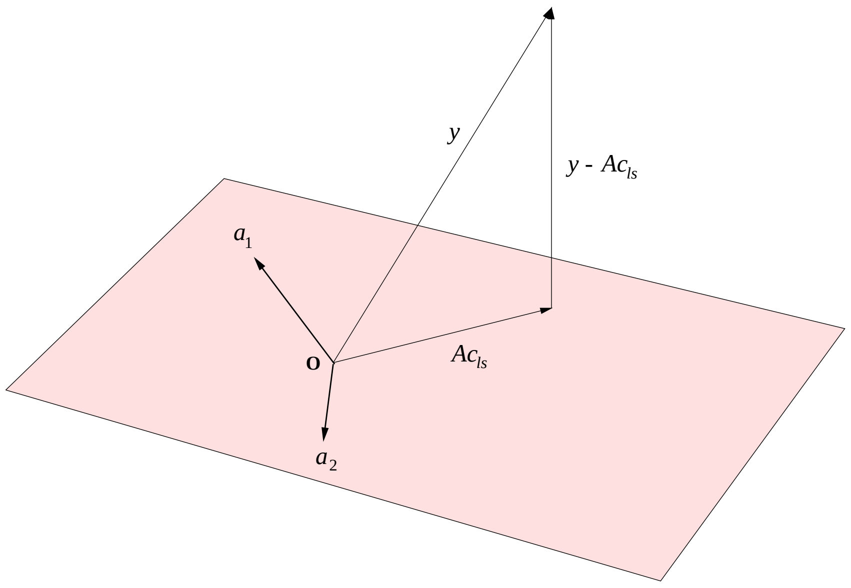

Figure 1: The $m = 3$, $n = 2$ case.

Figure 1 provides geometric insight into the problem of solving (3) and (8) for the case $m = 3$ and $n = 2$. Here $\mathbf{y} \in \mathbb{R}^{3\times 1}$ and $A = [\mathbf{a}_1, \mathbf{a}_2]$ is $3 \times 2$ with two linearly independent columns, $\mathbf{a}_1$ and $\mathbf{a}_2$. Now $Ran(A)$ is the span$\{\mathbf{a}_1, \mathbf{a}_2\}$, that is, the set of all linear combinations of the columns of $A$. It is a two-dimensional subspace of $\mathbb{R}^{3\times 1}$ and is visualized in Figure 1 by the plane which passes through the origin and contains $\mathbf{a}_1, \mathbf{a}_2$. Also, for any $\mathbf{c} \in \mathbb{R}^{2\times 1}$, $A\mathbf{c} = c_1\mathbf{a}_1 + c_2\mathbf{a}_2 \in Ran(A)$. Now if $\mathbf{y}$ is not in the plane $Ran(A)$, as is the case in Figure 1, the linear system (3) and the interpolation problem will not have a solution. On the other hand, the least squares problem always has a solution $\mathbf{c}_{ls}$. The vector $A\mathbf{c}_{ls}$ is the vector in the plane $Ran(A)$, which is closest to $\mathbf{y}$. Also, the vector $\mathbf{y} - A\mathbf{c}_{ls}$ will be orthogonal to the the plane $Ran(A)$. Because the columns of $A$ are linearly independent, or equivalently because $rank(A) = 2$, $\mathbf{c}_{ls}$ is the only solution to the least squares problem. Now imagine a case where $\mathbf{a}_1$ and $\mathbf{a}_2$ lie along the same line. In this case, $Ran(A)$ is a one-dimensional subspace of $\mathbb{R}^{3\times 1}$ and can be visualized by a line passing through the origin. Now there are infinitely many ways to describe a vector in $Ran(A)$ as a linear combination of $\mathbf{a}_1$ and $\mathbf{a}_2$, and so there are infinitely many solutions to the least squares problem. We would prefer to have a unique linear model which fits the data. This will happen if the basis functions used in the linear model are chosen so that the columns of $A$ are linearly independent.

The $m > 3$ case works the same ways as the $m = 3$ case. For $\mathbf{c}$, $A\mathbf{c} \in Ran(A)$, a subspace of $\mathbb{R}^{m\times 1}$. In the least squares problem, we find the vector $\mathbf{c}_{ls}$ so that $A\mathbf{c}_{ls} \in Ran(A)$ is as close to the vector $\mathbf{y}$ as possible. Because we have assumed that $n \le m$, $rank(A) \le n$. If the basis functions have been well chosen, the columns of $A$ will be linearly independent and $rank(A) = n$. In this case, (8) is said to be a full rank least squares problem and has a unique solution. If $rank(A) < n$, (8) is said to be rank deficient and has infinitely many solutions.

In Orthogonal Methods course outline, we have see that the singular value decomposition (SVD) of $A$ can be used to solve the least squares problem. We note that in the $m = n$ case, it can also be used to solve (3), although Gauss elimination with partial pivoting is more frequently used. In the Orthogonal Methods course outline a formula is given for $\mathbf{c}_{ls}$. In the rank deficient case, this formula gives the least squares solution which has the smallest 2-norm. There are two other methods that are commonly used to solve the full rank least squares problem: QR factorization with column pivoting followed by the solution of a triangular system, and solution of the normal equations. However, the SVD approach is regarded as the most computationally reliable method.

As explained in the instructions for this project, you are to create an M-file with filename P1loginid.m containing MATLAB commands and statements to solve the following problems. Note that you can cut MATLAB code from this PDF document and paste it into the MATLAB editor.

- **Problem 1** In this problem you will fit noisy data using a line. We generate the data by taking 20 uniformly spaced samples from the line $y = 1 + 2t$ for $t \in [0, 1]$ and adding a small amount of random noise to each sample. Add the following lines to P1loginid.m.

```
% Project 2. Last Name First Name. GroupXX

echo on

% Problem 1
% create m > 2 noisy data points lying close to a line
m = 20;
t = linspace(0,1,m)’;
epsilon = .2; % Maximum noise amplitude
y = 1 + 2*t + epsilon*rand(m,1);
% Use the model c(1) + c(2)*t
% Construct the design matrix A.
A = [ones(m,1),t];
% Use the SVD to solve the least squares problem.
c = pinv(A)*y
if rank(A) == 2
   disp(’The design matrix is full rank’)
else
   disp(’The design matrix is rank deficient.’)
end
% Since A is full rank, we have found the unique
% solution to the full rank least squares problem.
% When the design matrix is rank deficient, the
% least squares problem has infinitely many
% solutions. The above computation would produce
% the solution that has the smallest 2-norm.
%
% Plot
tt = linspace(0,1,100)’;
yy = c(1) + c(2)*tt;
plot(t,y,’o’,tt,yy)
```

At the MATLAB prompt, type P2loginid to run your M-file. Note that a semicolon at the end of a line suppresses screen output for that line. You should see your plot in a window labeled Figure No. 1. Next, add three more lines for Problem 1. Use the MATLAB commands "xlabel" (`help xlabel`), "ylabel" (`help ylabel`), and "title" (`help title`) to label the $x$ and $y$ axes and to add explanatory text at the top of the plot. Run your M-file again. Notice that the solution to the least squares problem is a coefficient vector which is close to [1, 2].

- **Problem 2** The water level in the North Sea is mainly determined by the so-called $M_2$-tide, whose period is about 12 hours. A reasonable but simple model for predicting the level at any time $t$ is a function of the form

$$h(t) = c_1\phi_1(t) + c_2\phi_2(t) + c_3\phi_3(t), \qquad t \text{ in hours.}$$

where

$$\phi_1(t) = 1, \quad \phi_2(t) = \sin\left(\frac{2\pi t}{12}\right) \quad \phi_3(t) = \cos\left(\frac{2\pi t}{12}\right).$$

Here we are using basis functions which are periodic with period 12 hours. Does this seem reasonable? We wish to estimate the values of the unknown coefficients $c_1$, $c_2$, and $c_3$ from a set of measured values of water level $y_1, \ldots, y_m$ made at corresponding times $t_1, \ldots, t_m$ by solving a linear least squares problem as discussed above. From calculus we know that determining $c_1$, $c_2$, and $c_3$ corresponds to estimating the mean water level (bias), the amplitude of variation about the mean, and the phase of the oscillatory water level.

Suppose we have made the measurements:

| $t_j$ | 0 | 2 | 4 | 6 | 8 | 10 | hours |
|---|---|---|---|---|---|---|---|
| $y_j$ | 1.0 | 1.6 | 1.4 | 0.6 | 0.2 | 0.8 | meters |

Add the following lines to your M-file to solve this linear least squares problem having $m = 6$ data points and $n = 3$ basis functions.

```
% Problem 2

m = 6;
t = [0;2;4;6;8;10]; % hours
```

```
y = [1;1.6;1.4;0.6;0.2;0.8]; % meters
% design matrix
A = [ones(m,1),sin(pi*t/6),cos(pi*t/6)];
c = pinv(A)*y % least squares solution
% Plot the data and the model
% plot vectors
tt = linspace(0,10,100)’;
yy = [ones(100,1),sin(pi*tt/6),cos(pi*tt/6)]*c;
figure % create a new plot window
plot(t,y,’o’,tt,yy)
```

At the MATLAB prompt, type P2loginid to run your M-file. You should see your plot for problem 2 in a window labeled Figure No. 2. Next, add three more lines for Problem 2. Use the MATLAB commands "xlabel" (`help xlabel`), "ylabel" (`help ylabel`), and "title" (`help title`) to label the $x$ and $y$ axes and to add explanatory text at the top of the plot. Run your M-file again. Notice that the model passes "close to" the data points but does not interpolate them.

- **Problem 3** The interpolation equations (2) can be satisfied if and only if the linear system (3) has a solution. If $n < m$, there will be more equations than unknowns and this overdetermined system will not generally have a solution. This is clear in Figure 1 where $m = 3$ and $n = 2$. Show that this is the case for the example in Problem 2 by applying the rank test from the Linear Systems and Matrices Outline. You must compare the rank of $A$ to the rank of $[A, \mathbf{y}]$.

- **Problem 4** Add lines to your M-file to find a straight line fit to the following data and plot a graph of the straight line model along with the data on the same set of axes. Label your graph. Note that here we have $(x, y)$ data instead of $(t, y)$ data.

| Density of ore x | 2.8 | 2.9 | 3.0 | 3.1 | 3.2 | 3.2 | 3.2 | 3.2 | 3.3 | 3.4 |
|---|---|---|---|---|---|---|---|---|---|---|
| Iron content y(%) | 30 | 26 | 33 | 31 | 33 | 35 | 33 | 37 | 36 | 33 |

- **Problem 5** Balancing a Chemical Reaction. [1] Balance the chemical reaction

  $$x_1 Cu_2 S + x_2 H^+ + x_3 NO_3^- \longrightarrow x_4 Cu^{2+} + x_5 NO + x_6 S_8 + x_7 H_2 O$$

  by calculating the smallest possible, positive integer coefficients $x_1, \ldots, x_7$. The principles are that the number of atoms of each element must be conserved as well as the total electric charge. This gives six equations for the seven coefficients. Put these equations in the form of a homogeneous system $A\mathbf{x} = \mathbf{0}$. The null space $N(A)$ has dimension one. Find a basis vector for this null space. Find the solution in the form $c\mathbf{v}$ where $\mathbf{v}$ is your basis vector and $c$ is a scalar. Remember that the coefficients must be positive integers and they must be as small as possible. Hint: it is possible to solve this problem using the MATLAB functions augmovie, or ref and slash, or null.

- **Problem 6** The following problem arises later in the course when we model heat conduction in a rod and a vibrating string. Find a scalar $\lambda$ (eigenvalue) and a nontrivial function $X(x)$ (eigenfunction) so that

  $$X''(x) + \lambda X(x) = 0 \tag{9}$$

  $$X(0) = 0 \tag{10}$$

  $$X(1) = 0 \tag{11}$$

  This problem is sometimes referred to as an eigenvalue problem for an ordinary differential equation. It is also called a Sturm-Liouville problem. It can be shown that there are infinitely many positive eigenvalues. We want to approximate the smallest one, which the Sturm-Liouville theory says is $\pi^2$. We will reduce this problem to a matrix eigenvalue problem using a centered finite difference approximation for $X''$.

  First, we discretize the interval $[0, 1]$ with mesh points $x_i = ih, \ i = 0, \ldots, n+1$, and $h = 1/(n+1)$. Next, we use the approximation

  $$X''(x_i) \approx \frac{X(x_{i+1}) - 2X(x_i) + X(x_{i-1})}{h^2} \quad \text{for } i = 1, \ldots, n, \tag{12}$$

[1] This problem appears in *FirstLeaves: A Tutorial Introduction*, B. W. Char et. al., Springer 1992, page 197 (ISBN 0-387-97621-3).

to derive a matrix eigenvalue problem by evaluating (9) at $x_1, \ldots, x_n$, and using (10), (11), and (12). This problem has the form

$$\frac{1}{h^2} AZ = \lambda Z \tag{13}$$

where $Z = [X(x_1), \ldots, X(x_n)]^T$ and $A$ has the number 2 on the main diagonal, the number -1 on the first subdiagonal and first superdiagonal, and is zero elsewhere.

Add the following lines to your M-file to solve this problem. Experiment with the value of $n$.

```
% Problem 6

n = 100;
h = 1/(n+1); e = ones(n,1)/hˆ2;
A = spdiags([-e 2*e -e],-1:1,n,n);
z = eig(A); format long
y = sort(z)/piˆ2;
y(1)
```