More on Least Squares

• Suppose we have the overdetermined linear system
  \[ Ax = b \]
  where \( A \) is \( m \times n \) and \( m > n \).

• We discussed solving instead the Least Squares Problem
  \[ \| Ax - b \| = \min \]

\[ \| Ax - b \|^2 = \min \] (2)

• We saw that we can find \( x \) by solving the normal equations
  \[ A^T Ax = A^T b. \]
• Moreover, we saw that

0. Geometrically we get $Ax$ by projecting $b$ into the column space of $A$ and we get the normal equations by observing that $Ax - b$ is in the orthogonal complement of the column space of $A$.

1. $Ax$ exists and is unique for every $b$.

2. $x$ is unique if the columns of $A$ are linearly independent.

3. $A^T A$ is invertible if and only if the the columns of $A$ are linearly independent.

4. In that case we get

$$x = (A^T A)^{-1} A^T b$$

although we would not normally actually compute the inverse of $(A^T A)$
Example: Suppose you have some data \((x_i, y_i), i = 1, 2, \ldots , n\) and you have reason to believe that the answer is some constant plus a \(2\pi\)-periodic function that looks something like a sine curve. In other words,

\[ y_i \approx f(x_i) = \alpha + \beta \sin x_i + \gamma \cos x_i. \]

In equation form this is

\[ Ax = b \]

where

\[
A = \begin{bmatrix}
1 & \sin x_1 & \cos x_1 \\
1 & \sin x_2 & \cos x_2 \\
\vdots & \vdots & \vdots \\
1 & \sin x_n & \cos x_n
\end{bmatrix}, \quad x = \begin{bmatrix}
\alpha \\
\beta \\
\gamma
\end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_n
\end{bmatrix}
\]

This is an overdetermined linear system. As we discussed in class yesterday, its least squares solution \(x\) is given by the normal equations

\[ A^T Ax = A^T b \]

where, in this case, with all sums running from 1 to \(n\),

\[
A^T A = \begin{bmatrix}
\sum 1 & \sum \sin x_i & \sum \cos x_i \\
\sum \sin x_i & \sum \sin^2 x_i & \sum \sin x_i \cos x_i \\
\sum \cos x_i & \sum \cos x_i \sin x_i & \sum \cos^2 x_i
\end{bmatrix}
\]

and

\[
A^T b = \begin{bmatrix}
\sum y_i \\
\sum y_i \sin x_i \\
\sum y_i \cos x_i
\end{bmatrix}
\]
• Example: Temperatures in Salt Lake City.

• You’d expect (low and high daily) temperatures $T(t)$ in SLC to follow (roughly) some kind of sin curve such as

$$Y(t) = a + b \sin \frac{2\pi t}{12} + c \cos \frac{2\pi t}{12}$$

(3)

where $t$ is the time of the year, measured in months. $a$ is the mean temperature over the whole year. $b$ and $c$ are suitable coefficients.

I found the following data at

http://www.rssweather.com/climate/Utah/Salt%20Lake%20City/

<table>
<thead>
<tr>
<th>Month</th>
<th>Low</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>21.0F</td>
<td>37.0F</td>
</tr>
<tr>
<td>Feb</td>
<td>25.5F</td>
<td>43.0F</td>
</tr>
<tr>
<td>Mar</td>
<td>33.4F</td>
<td>52.8F</td>
</tr>
<tr>
<td>Apr</td>
<td>39.0F</td>
<td>60.9F</td>
</tr>
<tr>
<td>May</td>
<td>46.9F</td>
<td>70.6F</td>
</tr>
<tr>
<td>Jun</td>
<td>55.8F</td>
<td>82.2F</td>
</tr>
<tr>
<td>Jul</td>
<td>63.4F</td>
<td>90.6F</td>
</tr>
<tr>
<td>Aug</td>
<td>62.4F</td>
<td>88.7F</td>
</tr>
<tr>
<td>Sept</td>
<td>52.4F</td>
<td>77.6F</td>
</tr>
<tr>
<td>Oct</td>
<td>41.0F</td>
<td>64.0F</td>
</tr>
<tr>
<td>Nov</td>
<td>30.4F</td>
<td>48.7F</td>
</tr>
<tr>
<td>Dec</td>
<td>22.4F</td>
<td>38.0F</td>
</tr>
</tbody>
</table>

• Suppose you want to find the Least Squares approximation of the form (3) that best approximates the data.
• It would be unreasonable to do this problem by hand. Numerous facilities (e.g., matlab, maple, wolfram alpha, any number of programming languages) exist to get a computer to do the necessary calculations.

• The following matlab code and its outputs show the results of doing a Least Squares Approximation.
% approximate low and high temperatures in SLC by sin and cos
% curve
% see http://www.rssweather.com/climate/Utah/Salt%20Lake%

dlow=[21.3;25.5;33.4;39.0;46.9;55.8;63.4;62.4;52.4;41.0;30.0];
pi = 2.0*acos(0.0);

for i =1:12
  O(i,1) = 1.0;
  S(i,1) = sin(2*i/12*pi);
  C(i,1) = cos(2*i/12*pi);
end

A=[O'*O,O'*S,O'*C;S'*O,S'*S,S'*C;C'*O,C'*S,C'*C]
B=[O'*low;S'*low;C'*low];

X= A\B

for i = 1:12
  app = X(1) + X(2)*S(i,1) + X(3)*C(i,1);
  err = low(i)-app;
  res = [i,low(i), app, err];
  disp(res)
end
\[ A = \begin{bmatrix} 12.0000 & -0.0000 & -0.0000 \\ -0.0000 & 6.0000 & 0.0000 \\ -0.0000 & 0.0000 & 6.0000 \end{bmatrix} \]

\[ X = \begin{bmatrix} 41.1583 \\ -10.9147 \\ -16.9332 \end{bmatrix} \]

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>21.3000</td>
<td>21.0364</td>
<td>0.2636</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0000</td>
<td>25.5000</td>
<td>23.2393</td>
<td>2.2607</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0000</td>
<td>33.4000</td>
<td>30.2436</td>
<td>3.1564</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0000</td>
<td>39.0000</td>
<td>40.1725</td>
<td>-1.1725</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.0000</td>
<td>46.9000</td>
<td>50.3655</td>
<td>-3.4655</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.0000</td>
<td>55.8000</td>
<td>58.0915</td>
<td>-2.2915</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.0000</td>
<td>63.4000</td>
<td>61.2803</td>
<td>2.1197</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.0000</td>
<td>62.4000</td>
<td>59.0774</td>
<td>3.3226</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.0000</td>
<td>52.4000</td>
<td>52.0731</td>
<td>0.3269</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.0000</td>
<td>41.0000</td>
<td>42.1442</td>
<td>-1.1442</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.0000</td>
<td>30.4000</td>
<td>31.9511</td>
<td>-1.5511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.0000</td>
<td>22.4000</td>
<td>24.2252</td>
<td>-1.8252</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
% approximate low and high temperatures in SLC by sin and cos
% curve
% see http://www.rssweather.com/climate/Utah/Salt%20Lake%20City/

high=[37.0;43.4;52.8;60.9;70.6;82.2;90.6;88.7;77.6;64.0;48.0;37.0];
pi = 2.0*acos(0.0);

for i =1:12
  O(i,1) = 1.0;
  S(i,1) = sin(i/6*pi);
  C(i,1) = cos(i/6*pi);
end

A=[O'*O,O'*S,O'*C;S'*O,S'*S,S'*C;C'*O,C'*S,C'*C]
B=[O'*high;S'*high;C'*high];

X= A\B

for i = 1:12
  app = X(1) + X(2)*S(i,1) + X(3)*C(i,1);
  err = high(i)-app;
  res = [i,high(i), app, err];
  disp(res)
end
\[ A = \begin{bmatrix}
12.0000 & -0.0000 & -0.0000 \\
-0.0000 & 6.0000 & 0.0000 \\
-0.0000 & 0.0000 & 6.0000 \\
\end{bmatrix} \]

\[ X = \begin{bmatrix}
62.8750 \\
-13.7609 \\
-21.7808 \\
\end{bmatrix} \]

\[ \begin{bmatrix}
1.0000 & 37.0000 & 37.1318 & -0.1318 \\
2.0000 & 43.4000 & 40.0673 & 3.3327 \\
3.0000 & 52.8000 & 49.1141 & 3.6859 \\
4.0000 & 60.9000 & 61.8481 & -0.9481 \\
5.0000 & 70.6000 & 74.8573 & -4.2573 \\
6.0000 & 82.2000 & 84.6558 & -2.4558 \\
7.0000 & 90.6000 & 88.6182 & 1.9818 \\
8.0000 & 88.7000 & 85.6827 & 3.0173 \\
9.0000 & 77.6000 & 76.6359 & 0.9641 \\
10.0000 & 64.0000 & 63.9019 & 0.0981 \\
11.0000 & 48.7000 & 50.8927 & -2.1927 \\
12.0000 & 38.0000 & 41.0942 & -3.0942 \\
\end{bmatrix} \]
Perhaps more impressive is a graphic display of our data and its approximations:

Figure 1. Temperature in Salt Lake City.
• As we’ve seen, things get more convenient if we have matrices with orthonormal columns. Let’s first look at the case that $A$ itself have orthonormal columns.
• Next let’s see what happens if we have a $QR$ factorization of $A$.

• Suppose

$$A = QR$$

where $R$ is upper triangular and $Q$ has orthonormal columns.