Chapter 8. Least Squares and the Normal Equation

Load the `LinearAlgebra` and `plots` packages for access to the `GenerateMatrix`, `LinearSolve`, `LeastSquares`, and `display` procedures. with(`LinearAlgebra`) : with(`plots`):

**Least Squares Approximations in \( \mathbb{R}^n \)**

Approximation theory often starts with a linear system \( Ax = b \) that has no solutions. Typically, \( A \) is an \( m \times n \) matrix with \( m > n \) and rank \( n \). So, \( b \) is a vector that is not in the column space of \( A \). When this happens the next best thing is the solution to \( Ax = Pb \), where \( P \) is the matrix that projects \( b \) orthogonally to the column space of \( A \). The solution \( a \) for this equation is referred to as the *least squares solution* to \( Ax = b \) because, since \( Pb \) is the vector in \( S \) that is closest to \( b \), \( \| Aa - b \|_2 < \| Ax - b \|_2 \) for all \( x \neq a \) in \( \mathbb{R}^n \). Thus, using \( a \) as the solution to \( Ax = b \) yields the least squared error.

**Example 1.** We seek the linear function \( f(x) := a_1 + a_2 x \) that best approximates the three data points \( DP := \{[1, 2], [4, 3], [5, 5]\} \), see the plot below. The plot is named \( DP\text{plot} \) so we can show it later.

```
plot(DP, x = 0..6, 0..6, style = point, symbol = solidcircle, symbolsize = 18); DP\text{plot} := %:
```

![Plot of data points](image)

Assuming the graph of \( f \) passes through all three points yields the following list of three linear equations.

\[
eqns := \left[ f(DP[k, 1]) = DP[k, 2] \right] \text{ for } k = 1..3
\]

\[
\begin{align*}
a_1 + a_2 &= 2, \\
a_1 + 4a_2 &= 3, \\
a_1 + 5a_2 &= 5
\end{align*}
\]  

(1)

Apply `GenerateMatrix` to the list named `eqns` to generate the coefficient matrix and constant vector for the system. They are named \( A \) and \( b \) respectively.

\[
A, b := \text{GenerateMatrix}(\eqns, [a_1, a_2])
\]

\[
\begin{bmatrix}
1 & 1 \\
1 & 4 \\
1 & 5
\end{bmatrix},
\begin{bmatrix}
2 \\
3 \\
5
\end{bmatrix}
\]  

(2)

To build the matrix projecting \( b \) to the column space of \( A \), orthogonalize \( A \)'s columns then calculate \( P \).

\[
u, v := \text{op}(\text{GramSchmidt }([\text{Column}(A, 1), \text{Column}(A, 2)])) =
\begin{bmatrix}
1 \\
1 \\
1
\end{bmatrix},
\begin{bmatrix}
\frac{7}{3} \\
\frac{2}{3} \\
\frac{5}{3}
\end{bmatrix}
\]
The Normal Equation

The hardest part of the process described above is orthogonalization to calculate the projection matrix. Fortunately, it can be avoided. We can find the least squares approximation without calculating $P$. Start with $A \mathbf{x} = P \mathbf{b}$ and multiply both sides by the transpose $A^+$ of $A$ to get $A^+ \mathbf{A} \mathbf{x} = A^+ P \mathbf{b}$. This simplifies to the so-called normal equation

$$A^+ A \mathbf{x} = A^+ b,$$

because $A^+ P = (A^+ A)^+ = (P A)^+ = A^+$. Thus Gram-Schmidt orthogonalization is not needed. Indeed, because $A^+ A$ is invertible (exercise) the following formula can be used for the least squares solution to $A \mathbf{x} = \mathbf{b}$

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

As a bonus, the equation above yields a formula for the projection matrix $P$. Multiply that equation by $A$ to obtain $A \mathbf{x} = A (A^+ A)^{-1} A^+ b$. Since $A \mathbf{x} = P \mathbf{b}$ also, $A (A^+ A)^{-1} A^+ b = P \mathbf{b}$. This is true for all $\mathbf{b}$ implying that

$$P = A (A^+ A)^{-1} A^+.$$

We will refer to this as the normal equation formula for the matrix projecting orthogonally to the column space of an $m \times n$ matrix $A$ having rank $n$. Here is how the formula works for the matrix $A$ in Example 1. Note that periods are needed to force matrix multiplication.
Example 2. We now want to approximate the data shown below. It looks periodic, period 14, and it rises, so we use a function of the form \( g(x) := c_1 + c_2 x + c_3 \cos(\frac{2 \pi x}{14}) + c_4 \sin(\frac{2 \pi x}{14}) \). The data points are \( DP := \{ [0, 2], [3, 3], [6, 8], [9, 9], [14, 8] \} \). The plot is named \( DP\text{plot} \) so we can show it later.

\[
\text{plot}(DP, x = 0 .. 15, 0 .. 10, \text{style} = \text{point}, \text{symbol} = \text{solidcircle}, \text{symbolsize} = 18); \quad DP\text{plot} := \%
\]

Assuming the graph of \( g \) passes through all five points yields the following list of five linear equations.

\[
\text{eqns} := \left[ g(DP[k, 1]) = DP[k, 2] \right] \mid k = 1..5
\]

\[
c_1 + c_3 = 2, \quad c_1 + 3 c_2 + c_3 \cos\left(\frac{3}{7} \pi\right) + c_4 \sin\left(\frac{3}{7} \pi\right) = 3, \quad c_1 + 6 c_2 - c_3 \cos\left(\frac{1}{7} \pi\right) + c_4 \sin\left(\frac{1}{7} \pi\right) = 8, \quad c_1 + 9 c_2 - c_3 \cos\left(\frac{2}{7} \pi\right) - c_4 \sin\left(\frac{2}{7} \pi\right) = 9, \quad c_1 + 14 c_2 + c_3 = 8
\]

Apply \text{GenerateMatrix} to the list named \( \text{eqns} \) to generate the coefficient matrix and constant vector for the system. They are named \( C \) and \( b \) respectively. The \text{evalf} procedure is applied to get decimals which are easier to work with in subsequent calculations.

\[
C, b := \text{evalf}\left( \text{GenerateMatrix}\left( \text{eqns}, [c_k \mid k = 1..4] \right) \right)
\]

\[
\begin{bmatrix}
1. & 0. & 1. & 0. & 2. \\
1. & 3. & 0.2225209335 & 0.9749279123 & 3. \\
1. & 6. & -0.9009688678 & 0.4338837393 & 8. \\
1. & 9. & -0.6234898018 & -0.7818314825 & 9. \\
1. & 14. & 1. & 0. & 8. \\
\end{bmatrix}
\]

Now use \text{LinearSolve} to solve the normal equation \( C^+ C x = C^+ b \) and obtain the vector solution to \( C x = P b \). Name the solution \( c \).

\[
c := \text{LinearSolve}(C^+ C, C^+ b) = \begin{bmatrix}
3.57139663320840 \\
0.437818495583968 \\
-1.83016546473049 \\
-0.940389514957001 \\
\end{bmatrix}
\]

The approximating function is
Let \( g(x) = 3.57 + 0.438x - 1.83 \cos(0.449x) - 0.940 \sin(0.449x) \).

Below is a display of the data points and the approximation followed by the associated root squared error.

\[
display(DPplot, plot(g(x), x = 0..15)) =
\]

The root squared error (RSE): \( \|c - b\|_2 = 0.929116954181172017 \)

### Chapter 8 Procedures

<table>
<thead>
<tr>
<th>Math Entry (Example)</th>
<th>Typical Application</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LinearSolve</strong></td>
<td>Find the vector solution to ( Ax = b ) where ( A := \begin{bmatrix} 1 &amp; 2 &amp; 3 \ 3 &amp; 2 &amp; 1 \end{bmatrix} ) and ( b := (1, 1) ): ( x := \text{LinearSolve}(A, b, \text{free} = t) = \begin{bmatrix} t_3 \ 1 + 2 - 2 t_3 \ t_3 \end{bmatrix} )</td>
<td>Maple denotes the free variable as ( t_3 ). Check the general solution. ( Ax = \begin{bmatrix} 1 \ 1 \end{bmatrix} ) Here is the solution when ( t_3 = 2 ). ( \text{subs}(t_3 = 2, x) = \begin{bmatrix} 2 \ -7/2 \ 2 \end{bmatrix} )</td>
</tr>
<tr>
<td><strong>GenerateMatrix and GenerateEquations</strong></td>
<td>Given matrix ( A ) and vector ( b ), ( A, b := % \begin{bmatrix} 2 &amp; -3 \ 1 &amp; 1 \end{bmatrix}, \begin{bmatrix} 1 \ 2 \end{bmatrix} )</td>
<td>Convert to matrix form the general linear system defined below. ( \sum_{j=1}^{3} a_{i,j} x_j = b_i ) : ( i, j = 1, 2, 3 ): ( \text{GenerateMatrix}(\text{linsys}, {x_1, x_2, x_3}) ) [ \begin{bmatrix} a_{1,1} &amp; a_{1,2} &amp; a_{1,3} \ a_{2,1} &amp; a_{2,2} &amp; a_{2,3} \ a_{3,1} &amp; a_{3,2} &amp; a_{3,3} \end{bmatrix} \begin{bmatrix} x_1 \ x_2 \ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \ b_2 \ b_3 \end{bmatrix} ]</td>
</tr>
<tr>
<td><strong>The LeastSquares Procedure</strong></td>
<td>Get the least squares solution for the matrix and vector in output (2). ( A, b := (2) )</td>
<td>( \text{LeastSquares}(A, b) = \begin{bmatrix} 15 \ 13 \ 17 \ 26 \end{bmatrix} )</td>
</tr>
</tbody>
</table>

**Maple Lab Manual**

Chapter 8: Least Squares

Chapter 8 Procedures