LEAST SQUARES and NORMAL EQUATIONS

Background

- Overdetermined Linear systems: consider $Ax = b$ if $A$ is $m \times n$, $x$ is $n \times 1$, $b$ is $m \times 1$ with $m > n$. The linear system is inconsistent if no $x$ satisfies all equations. Note: too many equations, not enough unknowns. Example:

  $x_1 + 2x_2 = 1$
  $2x_1 + 2x_2 = 2$
  $x_1 - x_2 = -1$
  $2x_1 + x_2 = 2$

- The least squares solution: the $x$ that minimizes

\[ ||r||_2 = ||Ax - b||_2 = \left( \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij}x_j - b_i)^2 \right)^{1/2} \]
LS and NORMAL EQUATIONS

• Geometric least squares solution $\bar{x}$:
  $A\bar{x} - b$ should be \textit{orthogonal} to all $Ax$.

• Algebraic least squares solution: consider
  $\|A\bar{x} - b\|_2^2 = \|A(\bar{x} + e) - b\|_2^2$. 
LEAST SQUARES, NORMAL EQUATIONS

The Normal Equations

- The normal equations are $A^T A \bar{x} = A^T b$.
- If $\text{rank}(A) = n$ the normal equations have a unique solution $\bar{x}$.
- Example

$SE$ and $RMSE$: with $r = A\bar{x} - b$

- squared error
  $$SE = ||r||^2_2 = r_1^2 + r_2^2 + \cdots + r_m^2;$$
- root mean squared error
  $$RMSE = \sqrt{\sum_{i=1}^{m} r_i^2 / m} = \sqrt{SE / m}.$$
Data Fitting and Linear Models

- Fitting data to straight line: given data \( \{(t_i, y_i)\}_{i=1}^{m} \), find the line \( y(t) = a + bt \) “closest” to the data points.

“Least Squares” line minimizes sum of squared errors.
Example: $t = [6.8 \ 7.1 \ 7.2 \ 7.4], \ y = [0.8 \ 1.2 \ 0.9 \ 0.9 \ 1.5]$

Matlab

```matlab
 t = [6.8 \ 7.1 \ 7.2 \ 7.4]; \ y = [.8 \ 1.2 \ 0.9 \ 0.9 \ 1.5];
 A = [ones(5,1) \ t\prime]; \ p = (A\prime A)\backslash(A\prime y); 
 tp = [6.8:.01:7.4];
 plot(tp,p(1)+p(2)*tp,t,y,’*’)
```
LEAST SQUARES CONTINUED

• General linear models have \( y(t) = \sum_{j=1}^{n} c_j f_j(t) \),
  for some model functions \( f_i(t) \) (e.g. \( f_j(t) = t^{j-1} \)):
  given data \( \{(t_i, y_i)\}_{i=1}^{m} \), find best \( c_j \)'s.

\[
A = \begin{bmatrix}
f_1(t_1) & f_2(t_1) & f_3(t_1) & \ldots & f_n(t_1) \\
f_1(t_2) & f_2(t_2) & f_3(t_2) & \ldots & f_n(t_2) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
f_1(t_m) & f_2(t_m) & f_3(t_m) & \ldots & f_n(t_m)
\end{bmatrix}
\]

Normal equations solution minimizes \( SE \).
If \( n \) is small and \( m \) is large with small \( SE \), you have
**data compression**.
Examples:
Conditioning for Normal Equations

- Vandermonde Example:
  find best \( y(t) = \sum_{j=1} c_j t^{j-1} \) for data.
  Linear system is \( Vc = y \), with “Vandermonde” \( V \)

\[
\begin{bmatrix}
1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\
1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & t_m & t_m^2 & \cdots & t_m^{n-1}
\end{bmatrix}
\begin{bmatrix}
c_1 \\
\vdots \\
c_n
\end{bmatrix}
= \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_m
\end{bmatrix},
\]

Normal equations solution: solve \( V^T Vc = V^T y \).

Example: \( t_i = 1, 1.1, \ldots, 2 \), with \( n = 8 \).
Matlab

\[
t = [10:20]' / 10; m = 11; n = 8;
V = (t*ones(1,n)).^(ones(m,1)*[0:n-1]);
disp([cond(V) cond(V'*V)]);
\]

\[
2.2303e+08 \quad 4.8086e+16
\]

Fit to data from \( y(t) = 1 + t + t^2 + \cdots + t^7 \).

\[
P = @(t)\text{sum}(t.^[0:n-1]);
for i = 1:m, y(i) = P(t(i)); end
\]

\[
c = (V'*V) \backslash (V'*y'); disp(c');
\]

\[
.98829 \quad 1.058 \quad .87824 \quad 1.1407 \quad .90343 \quad 1.0394 \ldots
\]

\[
.99116 \quad 1.0008
\]

\[
disp(\text{norm}(V*c-y')/\text{sqrt}(m)); \% \text{RMSE}
\]

\[
1.3149e-07
\]

True solution should have all \( c_i = 1 \).

- Theory shows \( K_2(A^T A) = K_2(A)^2 \): normal equations are often illconditioned.