Optimal Estimation Methods

(Lecture 5 – Basis Functions and Advanced Least Squares Topics)

Dr. John L. Crassidis

University at Buffalo – State University of New York
Department of Mechanical & Aerospace Engineering
Amherst, NY 14260-4400
johnc@buffalo.edu
http://www.buffalo.edu/~johnc

University at Buffalo The State University of New York

Basis Functions

- Common choice involves powers of t

$$\{1,\, t,\, t^2,\, t^3, \ldots\}$$

- The model is a power series polynomial

$$y(t) = x_1 + x_2 t + x_3 t^2 + \cdots = \sum_{i=1}^{n} x_i t^{i-1}$$

- The H matrix is now given by

$$H = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^{n-1} \end{bmatrix}$$

  - Known as the Vandermonde matrix

Change of Variables

- May be possible to convert nonlinear problem to a linear one using a change of variables

Basis Function	New Form	Change of Variables
$y = x_1 + \frac{x_2}{a} + \frac{x_3}{a^2} + \cdots$	$y = x_1 + x_2 t + x_3 t^2 + \cdots$	$t = \frac{1}{a}, \quad a \neq 0$
$y = Be^{at}$	$z = x_1 + x_2 t$	$z = \ln y, \quad y > 0$ $x_1 = \ln B, \quad B > 0$ $x_2 = a$
$y = x_1 w^{-m} + x_2 w^n$	$z = x_1 + x_2 t$	$z = y\, w^m$ $t = w^{m+n}$
$y = B \exp\left[-\frac{(1-at)^2}{2\sigma^2}\right]$	$z = x_1 + x_2 t + x_3 t^2$	$z = \ln y, \quad y > 0$ $x_1 = \ln B - \frac{\ln e}{2\sigma^2}, \quad B > 0$ $x_2 = \frac{a \ln e}{\sigma^2}$ $x_3 = -\frac{\ln e}{2\sigma^2} a^2$

All $\tilde{y}$ must be > 0

Must be careful because measurement noise all gets converted too.

What started out as Gaussian measurement errors may not be Gaussian anymore!

Use to find good initial guess for NLS

Example

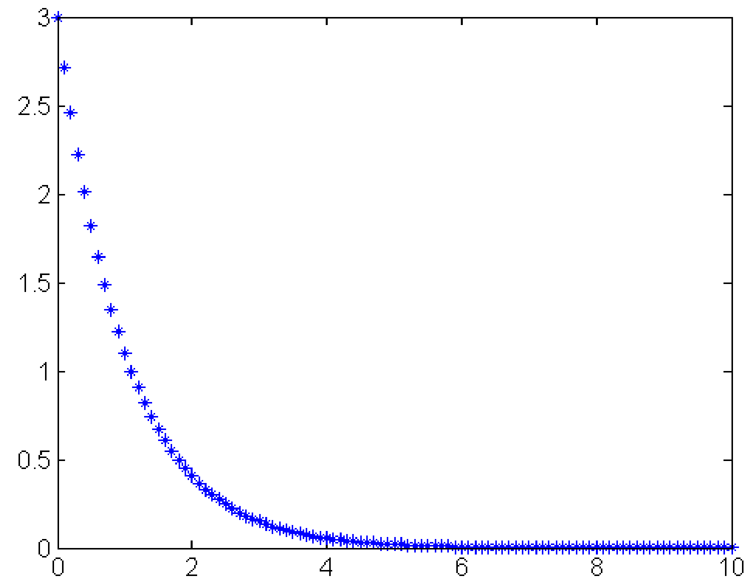
```
% Time
t=[0:0.1:10]';m=length(t);

% Observations
x=[3;-1];
y=x(1)*exp(x(2)*t);

% Change of Variables
z=log(y);
h=[ones(m,1) t];

% Least Squares Solution
x_change=inv(h'*h)*h'*z;

% Change Back
xe=[exp(x_change(1));x_change(2)]
```




```
xe =
  3.0000
 -1.0000
```

Orthogonal Functions (i)

- Definition of orthogonal functions
  - An infinite system of real functions

$$\{\varphi_1(t),\ \varphi_2(t),\ \varphi_3(t), \ldots,\ \varphi_n(t), \ldots\}$$

  is said to be orthogonal on the interval $[\alpha, \beta]$ if

$$\int_\alpha^\beta \varphi_p(t)\varphi_q(t)\,dt = 0 \quad (p \neq q,\ p,\ q = 1,\ 2,\ 3, \ldots)$$

  and

$$\int_\alpha^\beta \varphi_p^2(t)\,dt \equiv c_p \neq 0 \quad (p = 1,\ 2,\ 3, \ldots)$$

  - Many orthogonal functions exist
    - Sines and Cosines, Bessel Functions, Hermite Polynomials, Legendre Polynomials, Spherical Harmonics, Chebyshev Polynomials, etc.
    - We'll focus on sines and cosines

Orthogonal Functions (ii)

- Assume T is the period under consideration over any interval centered at $t = T/2$
- The orthogonality condition on the individual integrals of the terms $\sin(2\pi pt/T)$ and $\cos(2\pi pt/T)$are trivial to prove on the interval $[0, T]$ for $p = 1,2,3,\ldots$
- Look at the following integral

$$\int_0^T \sin(c\,t)\sin(d\,t)\,dt = \frac{1}{2}\int_0^T [\cos(c\,t - d\,t) - \cos(c\,t + d\,t)]\,dt$$

$$= \left[\frac{\sin(c\,t - d\,t)}{2(c-d)} - \frac{\sin(c\,t + d\,t)}{2(c+d)}\right]\Bigg|_0^T$$

- If $c = 2\pi p/T$ and $d = 2\pi q/T$ then it is easy to see that this integral is identically zero for any $p \neq q$

Fourier Series (i)

- The Fourier series of a function is a harmonic expansion of sines and cosines

$$y(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + \sum_{n=1}^{\infty} b_n \sin(n\omega t)$$

with $\omega = 2\pi/T$ and n is an integer

- To compute a coefficient, such as a_1, multiply both sides by $\cos(\omega t)$ and integrate from 0 to T

$$\int_0^T y(t)\cos(\omega t)\,dt = a_0 \underbrace{\int_0^T \cos(\omega t)\,dt}_{0} + a_1 \int_0^T [\cos(\omega t)]^2\,dt + \underbrace{\cdots}_{0} + b_1 \underbrace{\int_0^T \cos(\omega t)\sin(\omega t)\,dt}_{0} + \underbrace{\cdots}_{0}$$

- All are orthogonal functions except for two of them!

- Solving for a_1 gives

$$a_1 = \frac{\int_0^T y(t)\cos(\omega t)\,dt}{\int_0^T [\cos(\omega t)]^2\,dt}$$

- Compute the integral in the denominator

$$\int_0^T [\cos(\omega t)]^2\,dt = \left[\frac{t}{2} + \frac{\sin(2\omega t)}{4\omega}\right]\Bigg|_0^T = \frac{T}{2}$$

- Determine a_0 by integrating the original series

$$\int_0^T y(t)\,dt = \cancelto{= T a_0}{\int_0^T a_0\,dt} + \cancelto{0}{\int_0^T \sum_{n=1}^{\infty} a_n \cos(n\omega t)\,dt} + \cancelto{0}{\int_0^T \sum_{n=1}^{\infty} b_n \sin(n\omega t)\,dt}$$

$$a_0 = \frac{1}{T}\int_0^T y(t)\,dt$$

Fourier Series (iii)

- Keep going along the same approach for the other coefficients to yield the Fourier coefficients

$$a_0 = \frac{1}{T}\int_0^T y(t)\,dt$$

$$a_n = \frac{2}{T}\int_0^T y(t)\cos(n\omega t)\,dt$$

$$b_n = \frac{2}{T}\int_0^T y(t)\sin(n\omega t)\,dt$$

- Provides the classic result

Fourier Series (iv)

- Let's derive the Fourier coefficients using least squares
  - Consider minimizing the function

$$J = \frac{1}{2}\int_0^T [y(t) - \hat{\mathbf{x}}^T \mathbf{h}(t)]^T [y(t) - \hat{\mathbf{x}}^T \mathbf{h}(t)]\, dt$$

where

$$\begin{aligned}\mathbf{h}(t) &\equiv \begin{bmatrix} h_1(t) & h_2(t) & h_3(t) & \cdots \end{bmatrix}^T \\ &= \begin{bmatrix} 1 & \cos(\omega t) & \sin(\omega t) & \cdots & \cos(n\omega t) & \sin(n\omega t) \end{bmatrix}^T\end{aligned}$$

  - Performing the multiplications in the loss function gives

$$J = \frac{1}{2}\int_0^T [y(t)]^2\, dt - \left[\int_0^T y(t)\, \mathbf{h}^T(t)\, dt\right] \hat{\mathbf{x}} + \frac{1}{2}\hat{\mathbf{x}}^T \left[\int_0^T \mathbf{h}(t)\, \mathbf{h}^T(t)\, dt\right] \hat{\mathbf{x}}$$

Fourier Series (v)

- The necessary conditions gives

$$\underset{n\times 1}{\hat{\mathbf{x}}} = \Bigg[\int_0^T \mathbf{h}(t)\,\mathbf{h}^T(t)\,dt\Bigg]^{-1}_{n\times n} \Bigg[\int_0^T y(t)\,\mathbf{h}(t)\,dt\Bigg]$$

- Since $\mathbf{h}(t)$ represents a set of orthogonal functions on the interval $[0, T]$, then the integral of $\mathbf{h}(t)\,\mathbf{h}^T(t)$ is a diagonal matrix with elements given by $\int_0^T [h_i(t)]^2\,dt$, which leads to

$$\hat{x}_i = \frac{\int_0^T y(t)h_i(t)\,dt}{\int_0^T [h_i(t)]^2\,dt}, \quad i = 1, 2, \ldots, n$$

  - Note that $\mathbf{h}(t)\,\mathbf{h}^T(t)$ is not diagonal itself but its integral over the interval is diagonal
- This is identical to the solution for the Fourier coefficients done previously
- Therefore, the Fourier coefficients are just “least square” estimates using the particular orthogonal basis functions

Matrix Decompositions (i)

- The core component of any least squares algorithm is $(H^T H)^{-1}$ ← m×n inverse
  - As an alternative to direct computation of this inverse, it is common to decompose H in some way which simplifies the calculations and/or is more robust with respect to near singularity conditions
- The QR decomposition factors a full rank matrix H as the product of an orthogonal matrix Q and an upper-triangular matrix R

$$H = QR$$

where Q is an $m \times n$ matrix with $Q^T Q = I$, and R is an upper triangular $n \times n$ matrix with all elements $R_{ij} = 0$ for $i > j$

  - The term $H^T H$ in the normal equations is easier to invert since

$$H^T H = R^T Q^T Q R = R^T R$$

Matrix Decompositions (ii)

- The normal equations simply become

$$R^T R\hat{\mathbf{x}} = R^T Q^T \tilde{\mathbf{y}}$$

or

$$\boxed{R\hat{\mathbf{x}} = Q^T \tilde{\mathbf{y}}}$$

- The solution can easily be accomplished since R is upper triangular
- The real cost is in the $2mn^2$ operations in the *modified Gram-Schmidt algorithm*, which are required to compute Q and R
- Notice it is not necessary to square H (i.e., form $H^T H$)
  - The QR algorithm operates directly on H
- If H is poorly conditioned, it is easy to verify that $H^T H$ is much more poorly conditioned than H itself
  - This gives the QR approach a much more numerically conditioned solution to the least squares problem

Matrix Decompositions (iii)

- Another decomposition of H is the SVD with $H = U\,S\,V^T$

  where U is an $m \times n$ matrix with orthonormal columns, S is an $n \times n$ diagonal matrix, and V is an $n \times n$ orthogonal matrix

  - Note that $U^T U = I$, but $U\,U^T$ is not in general

  skinny form

- Normal equations become

$$(H^T H)\hat{\mathbf{x}} = H^T \tilde{\mathbf{y}}$$

$$(VSU^T USV^T)\hat{\mathbf{x}} = VSU^T \tilde{\mathbf{y}}$$

$$(VSSV^T)\hat{\mathbf{x}} = VSU^T \tilde{\mathbf{y}}$$

$$(SV^T)\hat{\mathbf{x}} = U^T \tilde{\mathbf{y}}$$

- Solution is then given by

$$\boxed{\hat{\mathbf{x}} = VS^{-1}U^T \tilde{\mathbf{y}}}$$

  - Inverse of a diagonal matrix is only required
  - But the SVD is very computational expensive

Kronecker Factorization (i)

- The Kronecker product is defined

$$H = A \otimes B \equiv \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1\beta}B \\ a_{21}B & a_{22}B & \cdots & a_{2\beta}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{\alpha 1}B & a_{\alpha 2}B & \cdots & a_{\alpha\beta}B \end{bmatrix}$$

  where H is an $M \times N$ matrix, A is an $\alpha \times \beta$ matrix, and B is $\gamma \times \delta$ matrix; Kronecker product is only valid when $M = \alpha\gamma$ and $N = \beta\delta$

- Some useful identities

$$(A \otimes B)^T = A^T \otimes B^T \qquad (1)$$

$$(A \otimes B)(C \otimes D) = (A\,C) \otimes (B\,D) \qquad (2)$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}, \quad \text{if } A \text{ and } B \text{ are invertible} \qquad (3)$$

  - The last one is particularly useful for least squares applications

Kronecker Factorization (ii)

- Suppose that $H = A \otimes B$ is true, then the least squares estimate is

$$\hat{\mathbf{x}} = (H^T H)^{-1} H^T \tilde{\mathbf{y}}$$

$$= \left[(A^T \otimes B^T)(A \otimes B)\right]^{-1} (A^T \otimes B^T)\tilde{\mathbf{y}} \longrightarrow \text{used (1)}$$

$$= \left[(A^T A) \otimes (B^T B)\right]^{-1} (A^T \otimes B^T)\tilde{\mathbf{y}} \longrightarrow \text{used (2)}$$

$$= \left[(A^T A)^{-1} \otimes (B^T B)^{-1}\right] (A^T \otimes B^T)\tilde{\mathbf{y}} \longrightarrow \text{used (3)}$$

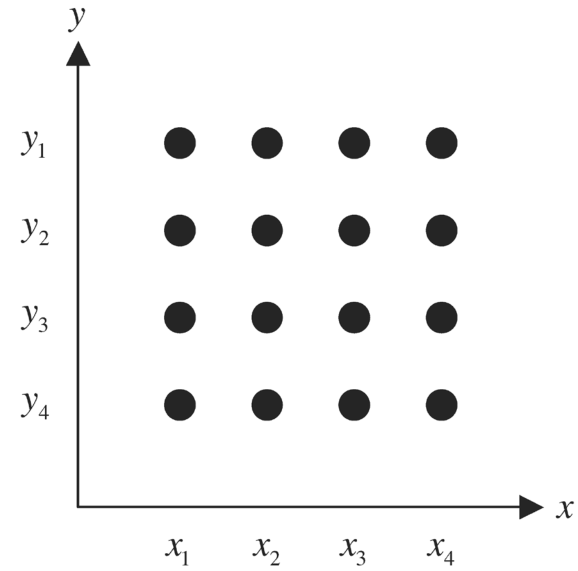
$$= \left\{\left[(A^T A)^{-1} A^T\right] \otimes \left[(B^T B)^{-1} B^T\right]\right\} \tilde{\mathbf{y}} \longrightarrow \text{used (2)}$$

- In essence the Kronecker product approach takes the square root of the matrix dimensions in regard to the computational difficulty
  - Provides a computationally efficient and numerically robust algorithm
- Under what conditions can a matrix be factored as a Kronecker product of smaller matrices?
  - Many exist, but we'll focus on one that's very useful for many applications

Kronecker Factorization (iii)

- Consider fitting a two-variable polynomial to data on an x-y grid

$$z = f(x,\, y) = \sum_{p=0}^{M} \sum_{q=0}^{N} c_{pq} x^p y^q$$



- The measurements are now defined by

$$\tilde{z}_{ij} = f(x_i,\, y_j) + v_{ij}$$

for $i = 1,\, 2,\, \ldots,\, n_x$ and $j = 1,\, 2,\, \ldots,\, n_y$

- Consider the special case of $M = 2$, $N = 1$, $n_x = 4$, and $n_y = 3$
- The two-variable polynomial then becomes

$$z = c_{00} + c_{01} y + c_{10} x + c_{11} x\, y + c_{20} x^2 + c_{21} x^2 y$$

Kronecker Factorization (iv)

- The least squares measurement model is now given by

$$\begin{bmatrix} \tilde{z}_{11} \\ \tilde{z}_{12} \\ \tilde{z}_{13} \\ \vdots \\ \tilde{z}_{41} \\ \tilde{z}_{42} \\ \tilde{z}_{43} \end{bmatrix} = \begin{bmatrix} 1 & y_1 & x_1 & x_1 y_1 & x_1^2 & x_1^2 y_1 \\ 1 & y_2 & x_1 & x_1 y_2 & x_1^2 & x_1^2 y_2 \\ 1 & y_3 & x_1 & x_1 y_3 & x_1^2 & x_1^2 y_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y_1 & x_4 & x_4 y_1 & x_4^2 & x_4^2 y_1 \\ 1 & y_2 & x_4 & x_4 y_2 & x_4^2 & x_4^2 y_2 \\ 1 & y_3 & x_4 & x_4 y_3 & x_4^2 & x_4^2 y_3 \end{bmatrix} \begin{bmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \\ c_{20} \\ c_{21} \end{bmatrix} + \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ \vdots \\ v_{41} \\ v_{42} \\ v_{43} \end{bmatrix}$$

$$\equiv H\mathbf{c} + \mathbf{v}$$

where H, $\mathbf{c}$, and $\mathbf{v}$ have dimensions of 12×6, 6×1, and 12×1, respectively

Kronecker Factorization (v)

- The matrix H has a Kronecker factorization given by

$$H = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ 1 & x_4 & x_4^2 \end{bmatrix} \otimes \begin{bmatrix} 1 & y_1 \\ 1 & y_2 \\ 1 & y_3 \end{bmatrix} \equiv H_x \otimes H_y$$

where H_x and H_y have dimensions of 4×3 and 3×2, respectively

  - The two-variable Vandermonde matrix can be produced by the Kronecker product of the corresponding one-variable Vandermonde matrices

- Least squares solution simplifies to

$$\boxed{\hat{\mathbf{c}} = (H^T H)^{-1} H^T \tilde{\mathbf{z}} = \left\{ \left[(H_x^T H_x)^{-1} H_x^T \right] \otimes \left[(H_y^T H_y)^{-1} H_y^T \right] \right\} \tilde{\mathbf{z}}}$$

  - Hence, only inverses of 3×3 and 2×2 matrices need to be computed, instead of an inverse of a 6×6 matrix in the standard least squares solution
  - Obviously the Kronecker factorization is useful when it can be applied

Kronecker Factorization (vi)

- The n-dimensional case has gridded data modeled by

$$\begin{aligned} z &= f(x_1, x_2, \ldots, x_n) \\ &= \sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \cdots \sum_{i_n=1}^{N_n} c_{i_1 i_2 \cdots i_n} \phi_{i_1}(x_1) \phi_{i_2}(x_2) \cdots \phi_{i_n}(x_n) \end{aligned}$$

where $\phi_{ij}(x_j)$ are basis functions

- The measurements now follow

$$\tilde{z}_{j_1 j_2 \cdots j_n} \quad \text{at} \quad (x_{1_{j_1}}, x_{2_{j_2}}, \ldots, x_{n_{j_n}})$$

for $j_1 = 1, 2, \ldots, M_1$ through $j_n = 1, 2, \ldots, M_n$

- Vectors used in the least squares algorithm

$$\tilde{\mathbf{z}} = \begin{bmatrix} \tilde{z}_{11\cdots 11} & \cdots & \tilde{z}_{11\cdots 1M_n} & \cdots & \tilde{z}_{M_1 M_2 \cdots M_{n-1} 1} & \cdots & \tilde{z}_{M_1 M_2 \cdots M_{n-1} M_n} \end{bmatrix}^T$$

$$\mathbf{c} = \begin{bmatrix} c_{11\cdots 11} & \cdots & c_{11\cdots 1N_1} & \cdots & c_{N_1 N_2 \cdots N_{n-1} 1} & \cdots & c_{N_1 N_2 \cdots N_{n-1} N_n} \end{bmatrix}^T$$

Kronecker Factorization (vii)

- The matrix H is given by

$$H = H_1 \otimes H_2 \otimes \cdots \otimes H_N$$

  with

$$H_i = \begin{bmatrix} \Phi_1(x_{i_1}) & \Phi_2(x_{i_1}) & \cdots & \Phi_{N_i}(x_{i_1}) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_1(x_{i_{M_i}}) & \Phi_2(x_{i_{M_i}}) & \cdots & \Phi_{N_i}(x_{i_{M_i}}) \end{bmatrix}, \quad i = 1, 2, \ldots, N$$

  where Φ's are sub-matrices composed of the basis functions $\phi_{i_1}(x_1)$ through $\phi_{i_n}(x_n)$

- Least squares estimate

$$\boxed{\hat{\mathbf{c}} = \left\{[(H_1^T H_1)^{-1} H_1^T] \otimes \cdots \otimes [(H_N^T H_N)^{-1} H_N^T]\right\} \tilde{\mathbf{z}}}$$

  - Therefore, the least squares solution is given by a Kronecker product of sub-matrices with much smaller dimension than the original problem

Example (i)

- Consider 21×21 grid over intervals $-2 \le x \le 2$ and $-2 \le y \le 2$

$$\begin{bmatrix} 1 & x & x^2 & x^3 & x^4 & x^5 \end{bmatrix}$$

$$\begin{bmatrix} 1 & y & y^2 & y^3 & y^4 & y^5 \end{bmatrix}$$

  - The 21×6 matrices H_x and H_y are given by

$$H_x = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{21} & x_{21}^2 & x_{21}^3 & x_{21}^4 & x_{21}^5 \end{bmatrix}$$

$$H_y = \begin{bmatrix} 1 & y_1 & y_1^2 & y_1^3 & y_1^4 & y_1^5 \\ 1 & y_2 & y_2^2 & y_2^3 & y_2^4 & y_2^5 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y_{21} & y_{21}^2 & y_{21}^3 & y_{21}^4 & y_{21}^5 \end{bmatrix}$$

Example (ii)

- The 441×36 matrix H is just the Kronecker product of H_x and H_y, so that $H = H_x \otimes H_y$
- Simulation
  - All the coefficients for both polynomials are set to 1
  - No noise is added to the measurements
- Compute the norm of the difference between the estimates and the true values
- Compare the following solutions
  - Standard Least Squares
  - Kronecker Factorization
  - QR Decomposition
  - SVD decomposition
- The standard least squares approach requires the inverse of a 36×36 matrix, while the Kronecker factorization requires two inverses of 6×6 matrices

Example (iii)

```
% Gridded Points
for y = -2:0.2:2;
   rowy=[1 y y^2 y^3 y^4 y^5];
   if y == -2,
      hy=rowy;
   else
      hy=[hy;rowy];
   end
end
for x = -2:0.2:2;
   rowx=[1 x x^2 x^3 x^4 x^5];
   if x == -2,
      hx=rowx;
   else
      hx=[hx;rowx];
   end
end

% H Matrix and True Values
h=kron(hx,hy);xtrue=ones(36,1);ztrue=h*xtrue;
```

Example (iv)

```
% Standard Least Squares
xhat1=inv(h'*h)*h'*ztrue;
norm_ls=norm(xhat1-xtrue)

% Kronecker Solution
xhat2=kron(inv(hx'*hx)*hx',inv(hy'*hy)*hy')*ztrue;
norm_kron=norm(xhat2-xtrue)

% QR Solution
[q,r]=qr(h,0);
xhat3=inv(r)*q'*ztrue;
norm_qr=norm(xhat3-xtrue)

% SVD Solution
[u,s,v]=svd(h,0);
xhat4=v*inv(s)*u'*ztrue;
norm_svd=norm(xhat4-xtrue)
```

worst

```
norm_ls =
 4.6203e-09
```

Best

```
norm_kron =
 4.4496e-13

norm_qr =
 6.7732e-13

norm_svd =
 1.2556e-12
```

normally best

Projections in LS (i)

- The term “normal” in Normal Equations implies that there is a geometrical interpretation to least squares
  - In fact, we will show that the least squares estimate provides the orthogonal projection, hence normal, of measurement vector onto a subspace which is spanned by columns of the matrix H $\underset{m\times1}{y} = \underset{m\times1}{h}\,\underset{1\times1}{x}$

$J = \frac{1}{2}(\tilde{y}^2 - 2y\,h^T\hat{x} + (h^T\hat{x})(\hat{x}h))$

$\nabla_{\hat{x}} J = -y\,h^T + h^T h\,\hat{x}$ $-c$

  - Consider the case of estimating a scalar

$(y_{m\times1})^T(h_{m\times1})\,x_{1\times1}$

$\underbrace{(h^T\hat{x})(\hat{x}h)}_{\text{scalar}} = \hat{x}^2 h^T h$

$$J = \frac{1}{2}(\tilde{\mathbf{y}} - \hat{x}\mathbf{h})^T(\underset{m\times1}{\tilde{\mathbf{y}}} - \hat{x}\mathbf{h})$$

where $\mathbf{h}$ is the basis function vector

  - The necessary conditions yield the following simple solution

$$\hat{x} = \frac{\mathbf{h}^T\tilde{\mathbf{y}}}{\mathbf{h}^T\mathbf{h}}$$

$h^T h$ can only be zero if all h are 0.

  - The residual error is given by

$$\mathbf{e} = (\tilde{\mathbf{y}} - \hat{x}\mathbf{h})$$

Projections in LS (ii)

- Left multiply the residual error by $\mathbf{h}^T$ and substitute the estimate

$$\begin{aligned}\mathbf{h}^T\mathbf{e} &= \mathbf{h}^T(\tilde{\mathbf{y}} - \hat{x}\mathbf{h}) \\ &= \mathbf{h}^T(\tilde{\mathbf{y}} - \frac{\mathbf{h}^T\tilde{\mathbf{y}}}{\mathbf{h}^T\mathbf{h}}\mathbf{h}) \\ &= \mathbf{h}^T\tilde{\mathbf{y}} - \frac{\mathbf{h}^T\tilde{\mathbf{y}}}{\mathbf{h}^T\mathbf{h}}\mathbf{h}^T\mathbf{h} \\ &= 0\end{aligned}$$

h^Te - orthogonal

- This shows that the angle between $\mathbf{h}$ and $\mathbf{e}$ is 90 degrees, so that the line connecting $\tilde{\mathbf{y}}$ to $\hat{x}\mathbf{h}$ must be *perpendicular* to $\mathbf{h}$
- This can easily expanded to the multi-dimensional case where the measurement vector is *projected* onto a subspace rather than just onto a line
  - The vector $\mathbf{p} \equiv H\hat{\mathbf{x}}$ must be the projection of $\tilde{\mathbf{y}}$ onto the column space of H, and the residual error $\mathbf{e}$ must be perpendicular to that space

Projections in LS (iii)

- Extension of scalar case

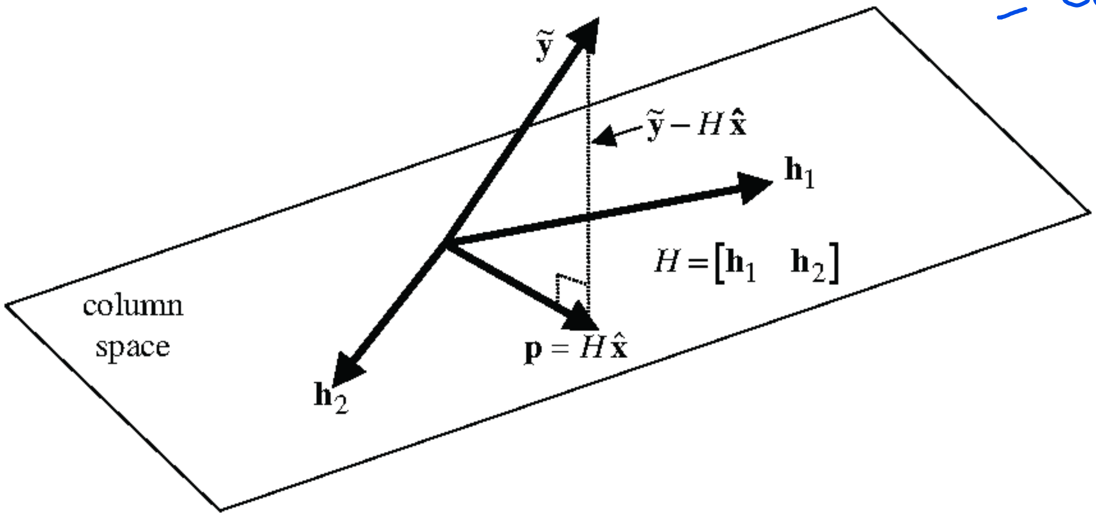


Diagram shows the projection onto the column space of a 3×2 matrix

- The residual error must be $\perp$ to every column ($\mathbf{h}_i$) of H, so that

$$\mathbf{h}_1^T(\tilde{\mathbf{y}} - H\hat{x}) = 0$$
$$\mathbf{h}_2^T(\tilde{\mathbf{y}} - H\hat{x}) = 0$$
$$\vdots$$
$$\mathbf{h}_n^T(\tilde{\mathbf{y}} - H\hat{x}) = 0$$

Gives the normal equations again!

or

$$H^T(\tilde{\mathbf{y}} - H\hat{x}) = \vec{0}$$

zero vector

Projections in LS (iv)

- The projection of the measurement vector onto the column space is

$$\mathbf{p} = H(H^T H)^{-1} H^T \tilde{\mathbf{y}}$$

- Geometrically, this means that the closest point to the measurement vector on the column space of H is $\mathbf{p}$
- The projection matrix is given by

$$\mathcal{P} = H(H^T H)^{-1} H^T$$

- The *projection matrix* follows the *idempotence* property

$$\mathcal{P}\tilde{\mathbf{y}} = [\mathcal{P}\,\mathcal{P}\,\ldots\,\mathcal{P}]\tilde{\mathbf{y}}$$

  - Once a vector has been obtained as the projection onto a subspace using $\mathcal{P}$, it can never be modified by any further application of $\mathcal{P}$
- The prediction error $\mathbf{e}_{\text{min}}$ once the solution has been found is

$$\mathbf{e}_{\text{min}} = (I - \mathcal{P})\tilde{\mathbf{y}}$$

where the matrix $(I - \mathcal{P})$ is the *orthogonal complement* of $\mathcal{P}$

  - It is easy to show that $(I - \mathcal{P})$ must also be a projection matrix, since it projects the measurement vector onto the orthogonal complement

Ball Constraint Problem (i)

- Consider the minimization of

$$J = \frac{1}{2}(\tilde{\mathbf{y}} - H\hat{\mathbf{x}})^T(\tilde{\mathbf{y}} - H\hat{\mathbf{x}})$$

subject to a spherical (ball) constraint

$$\sqrt{\hat{\mathbf{x}}^T\hat{\mathbf{x}}} \leq \gamma$$

- Solution is given using an SVD approach (derivation not shown here)

$$H = USV^T$$

$$\left[\mathbf{v}_1, \ \ldots \ , \ \mathbf{v}_n\right] = V$$

$$\mathbf{z} = U^T\tilde{\mathbf{y}}$$

$$r = \text{rank}(H) \rightarrow \text{usually } n$$

$$S = \text{diag}\begin{bmatrix} s_1 & \cdots & s_n \end{bmatrix}$$

$$\mathbf{z} = \begin{bmatrix} z_1 & z_2 & \cdots & z_n \end{bmatrix}^T$$

Ball Constraint Problem (ii)

- If the following inequality is true

$$\sum_{i=1}^{r} \left(\frac{z_i}{s_i} \right)^2 > \gamma^2$$

then find λ^* such that

$$\sum_{i=1}^{r} \left(\frac{s_i z_i}{s_i^2 + \lambda^*} \right)^2 = \gamma^2$$

  - It can be shown that there exists a unique positive solution for λ^* which can be found using Newton's root solving method

- Optimal estimate given by

$$\hat{\mathbf{x}} = \sum_{i=1}^{r} \left(\frac{s_i z_i}{s_i^2 + \lambda^*} \right) \mathbf{v}_i$$

- If the inequality is not satisfied then the estimate is given by

$$\hat{\mathbf{x}} = \sum_{i=1}^{r} \left(\frac{z_i}{s_i} \right) \mathbf{v}_i$$

Example (i)

- Consider the following model

$$y = x_1 + x_2 t + x_3 t^2$$

  with true values of 3, 2, 1
  - Given a set of 101 measurements we are asked to find an estimate such that $\gamma^2 = 14$
    - Standard deviation of measurement noise is set to 1
  - After forming the H matrix, we determine that the rank of H is $r = 3$
  - Singular values are given by

$$S = \text{diag}\begin{bmatrix} 456.3604 & 15.5895 & 3.1619 \end{bmatrix}$$

  - For this case the inequality is satisfied
  - The optimal value for λ^* is determined using Newton's root solving with a starting value of 0, and converges to a value of $\lambda^* = 0.245$

Example (ii)

- Optimal estimate given by

$$\hat{\mathbf{x}} = \begin{bmatrix} 3.0209 \\ 1.9655 \\ 1.0054 \end{bmatrix}$$

  - Note that the norm is 14 so the inequality is satisfied exactly
- Standard least squares estimate

$$\hat{\mathbf{x}}_{\text{ls}} = \begin{bmatrix} 3.0686 \\ 1.9445 \\ 1.0067 \end{bmatrix}$$

  - The norm is 14.2109 so the inequality is not satisfied
- It is important to note that the solution can vary from run-to-run because of the noise on the measurements
  - Sometimes a solution is given such that the norm is less than 14
  - Still satisfies the constraint though

Example (iii)

```
% Measurements and Other Parameters
t=[0:0.1:10]';
y=3+2*t+t.*t;
ym=y+randn(length(t),1);
h=[ones(length(t),1) t t.*t];
[u,s,v]=svd(h,0);
z=u'*ym;
r=rank(h);

le=0;
for i=1:length(z)
 le=le+(z(i)/s(i,i))^2;
end

% Value for gamma
gam=sqrt(1^2+2^2+3^2);
```

Example (iv)

```
% Main SVD Algorithm
xe=zeros(3,1);
lam=0;fc=0;fdc=0;fstop=100;lami=0;
if le > gam*gam
 while(norm(fstop)>1e-8)
  lami=lami+1;
  for i=1:length(z);
   fc=fc+(s(i,i)*z(i)/(s(i,i)^2+lam))^2;
   fdc=fdc-2*(s(i,i)*z(i))^2*(s(i,i)^2+lam)^(-3);
  end

  fc=fc-gam*gam;
  fstop=fc/fdc;
  lam=lam-fc/fdc;

  if (lami>20000) break; end
 end
```

Example (v)

```
 for i=1:length(z);
  xe=xe+s(i,i)*z(i)/(s(i,i)^2+lam)*v(:,i);
 end

else

 for i=1:length(z)
  xe=xe+(z(i)/s(i,i))*v(:,i);
 end

end
```