

## ECE 7251: Signal Detection and Estimation

Spring 2002  
Prof. Aaron Lanterman  
Georgia Institute of Technology

Lecture 10, 1/28/02:  
Estimation Under Additive Gaussian Noise  
(a.k.a. Least Squares Solutions)

### A Boring Definition Slide

- Define the gradient vector

$$\nabla_{\mathbf{q}} g(\mathbf{q}) \equiv \begin{bmatrix} \frac{\partial g_1(\mathbf{q})}{\partial \mathbf{q}_1} & \frac{\partial g_2(\mathbf{q})}{\partial \mathbf{q}_1} & \dots & \frac{\partial g_M(\mathbf{q})}{\partial \mathbf{q}_1} \\ \frac{\partial g_1(\mathbf{q})}{\partial \mathbf{q}_2} & \frac{\partial g_2(\mathbf{q})}{\partial \mathbf{q}_2} & & \frac{\partial g_M(\mathbf{q})}{\partial \mathbf{q}_2} \\ \vdots & & \ddots & \vdots \\ \frac{\partial g_1(\mathbf{q})}{\partial \mathbf{q}_M} & \frac{\partial g_2(\mathbf{q})}{\partial \mathbf{q}_M} & \dots & \frac{\partial g_M(\mathbf{q})}{\partial \mathbf{q}_M} \end{bmatrix}$$

Warning: Different authors use different definitions!

### Transformed CR Bounds

- Suppose  $\mathbf{a} = g(\mathbf{q})$ , where  $g$  is continuous
- Nice result from p. 230 of Scharf:

$$F_{\mathbf{a}}(\mathbf{q}) = [\nabla_{\mathbf{q}} \{g(\mathbf{q})\}]^{-1} F_{\mathbf{q}}(\mathbf{q}) [\nabla_{\mathbf{q}}^T \{g(\mathbf{q})\}]^{-1}$$

$$\text{cov}_{\mathbf{q}}(\hat{\mathbf{a}}) \geq \nabla_{\mathbf{q}}^T \{g(\mathbf{q})\} F_{\mathbf{q}}^{-1}(\mathbf{q}) \nabla_{\mathbf{q}} \{g(\mathbf{q})\}$$

- If  $g$  has an inverse, we can also write:
 
$$F_{\mathbf{a}}(\mathbf{q}) = \nabla_{\mathbf{a}} \{g^{-1}(\mathbf{a})\} F_{\mathbf{q}}(\mathbf{q}) \nabla_{\mathbf{a}}^T \{g^{-1}(\mathbf{a})\}$$

$$\text{cov}_{\mathbf{q}}(\hat{\mathbf{a}}) \geq [\nabla_{\mathbf{a}}^T \{g^{-1}(\mathbf{a})\}]^{-1} F_{\mathbf{q}}^{-1}(\mathbf{q}) [\nabla_{\mathbf{a}} \{g^{-1}(\mathbf{a})\}]^{-1}$$
- Usually substitute  $\mathbf{q} = g^{-1}(\mathbf{a})$  to get
 
$$F_{\mathbf{a}}(\mathbf{q} = g^{-1}(\mathbf{a})) = F_{\mathbf{a}}(\mathbf{a})$$

$$\text{cov}_{\mathbf{q} = g^{-1}(\mathbf{a})}(\hat{\mathbf{a}}) = \text{cov}_{\mathbf{a}}(\hat{\mathbf{a}})$$

### A Quick Word of Warning

Recall ML estimation is preserved under nonlinear transformations

$$\hat{\mathbf{a}}_{ML} = g(\hat{\mathbf{q}}_{ML})$$

But, other properties are usually not, i.e., if  $\hat{\mathbf{q}}_{ML}$  is efficient, that doesn't necessarily mean  $\hat{\mathbf{a}}_{ML}$  is efficient! (or unbiased, or whatever...)

### Signal in Additive Gaussian Noise

- Signal in Gaussian noise; signal parameterized by  $\mathbf{q}$ , perhaps in a nonlinear fashion

$$y = s(\mathbf{q}) + w, \quad w \sim \mathcal{N}(0, K)$$

$$l(\mathbf{q}) = - \sum_{i=1}^n [s(\mathbf{q}) - y_i] K^{-1} [s(\mathbf{q}) - y_i]^T / 2$$

$$\nabla_{\mathbf{q}} \{l(\mathbf{q})\} = - \sum_{i=1}^n ([s(\mathbf{q}) - y_i] K^{-1} \nabla_{\mathbf{q}}^T \{s(\mathbf{q})\} + \nabla_{\mathbf{q}} \{s(\mathbf{q})\} K^{-1} [s(\mathbf{q}) - y_i]^T) / 2$$

$$= - \sum_{i=1}^n \nabla_{\mathbf{q}} \{s(\mathbf{q})\} K^{-1} [s(\mathbf{q}) - y_i]^T$$

### Least Squares Solutions

- ML solution must satisfy:
 
$$\nabla_{\mathbf{q}} \{l(\mathbf{q})\} = \nabla_{\mathbf{q}} \{s(\mathbf{q})\} K^{-1} [s(\mathbf{q}) - \frac{1}{n} \sum_{i=1}^n y_i]^T = 0$$
- Equivalent to least squares, as it is often called in non-probabilistic contexts (dates back to Gauss studying planetary motions in 1795)
  - If  $K \neq I$ , called weighted least squares
  - No optimality properties if data is not Gaussian!
- If  $s(\mathbf{q})$  is nonlinear, usually need some iterative technique to find a solution

### CRB for the Additive Gaussian Model

- Under the Gaussian model, for real data:

$$\begin{aligned}
 F(\mathbf{q}) &= E[\nabla_{\mathbf{q}} \{l(\mathbf{q})\} \nabla_{\mathbf{q}}^T \{l(\mathbf{q})\}] \\
 &= E[\nabla_{\mathbf{q}} \{s(\mathbf{q})\} K^{-1} [s(\mathbf{q}) - y]^T [s(\mathbf{q}) - y] K^{-1} \nabla_{\mathbf{q}} \{s^T(\mathbf{q})\}] \\
 &= \nabla_{\mathbf{q}} \{s(\mathbf{q})\} K^{-1} \underbrace{E[[s(\mathbf{q}) - y]^T [s(\mathbf{q}) - y]]}_{=K} K^{-1} \nabla_{\mathbf{q}} \{s^T(\mathbf{q})\}] \\
 &= \nabla_{\mathbf{q}} \{s(\mathbf{q})\} K^{-1} \nabla_{\mathbf{q}}^T \{s(\mathbf{q})\}
 \end{aligned}$$

- Aside: for complex (Goodman) Gaussian data:

$$F(\mathbf{q}) = 2\text{Re}\{\nabla_{\mathbf{q}} \{s(\mathbf{q})\} K^{-1} \nabla_{\mathbf{q}}^H \{s(\mathbf{q})\}\}$$

### Linear Least Squares

- Suppose  $s(\mathbf{q}) = H\mathbf{q}$ , i.e.  $y = H\mathbf{q} + w$

$$\begin{aligned}
 \nabla_{\mathbf{q}} \{s(\hat{\mathbf{q}})\} K^{-1} [s(\hat{\mathbf{q}}) - y_i]^T &= 0 \\
 \nabla_{\mathbf{q}} \{H\hat{\mathbf{q}}\} K^{-1} [H\hat{\mathbf{q}} - y] &= 0 \\
 H^T K^{-1} [H\hat{\mathbf{q}} - y] &= 0 \\
 H^T K^{-1} H\hat{\mathbf{q}} - H^T K^{-1} y &= 0 \\
 H^T K^{-1} H\hat{\mathbf{q}} &= H^T K^{-1} y \\
 \hat{\mathbf{q}}_{LS} &= (H^T K^{-1} H)^{-1} H^T K^{-1} y
 \end{aligned}$$

### Linear Least Squares Con't

$$\hat{\mathbf{q}}_{LS} = (H^T K^{-1} H)^{-1} H^T K^{-1} y$$

- If data is Gaussian, then

$$\hat{\mathbf{q}}_{LS} = \hat{\mathbf{q}}_{ML} \sim \mathcal{N}(\mathbf{q}, H^T K^{-1} H)$$

- If data not Gaussian, performance is often hard to analyze; usually must do simulations
- Note if  $K=I$ , then

$$\hat{\mathbf{q}}_{LS} = \underbrace{(H^T H)^{-1} H^T}_{\text{Pseudoinverse}} y$$

### Ex: Medical Imaging

- Emission tomography; projection data is Poisson  $y_i \sim \text{Poisson}(H\mathbf{q})$ ,  $E[y_i] = s(\mathbf{q})$ ,  $\text{var}[y_i] = E[y_i]$

- Approximate Poisson distribution with Gaussian

$$\begin{aligned}
 y_i &\sim N(H\mathbf{q}, \underbrace{H\mathbf{q}}_{\text{Roughly estimate with } y_i}) \\
 y_i &\sim N(H\mathbf{q}, y_i)
 \end{aligned}$$

- Now we have a closed-form (although suboptimal) solution
- Original ML Poisson problem needs something like the expectation-maximization algorithm

### Back to Bayesianland

- Sometimes  $H$ , and even its pseudoinverse, are not well-conditioned; then least-squares crashes and burns
- Solution: convert to Bayesianism and use a prior!

$$\mathbf{q} \sim N(\mathbf{0}, K_q)$$

$$Y = H\mathbf{q} + W, \quad W \sim N(\mathbf{0}, K_w)$$

- Recall formulas for linear MMSE:

$$\hat{E}[\mathbf{q} | Y] = \mathbf{m}_q + K_{qY} K_Y^{-1} (Y - \mathbf{m}_Y)$$

$$E[ee^T] = K_q - K_{qY} K_Y^{-1} K_{Yq}$$

### Stuff We Need Which Wouldn't Easily Fit On the Other Slides

$$K_{qY} = E[\mathbf{q} Y] = E[\mathbf{q} (\mathbf{q}^T H^T + w^T)] = K_q H^T$$

$$K_{Yq} = H K_q$$

$$\begin{aligned}
 K_Y &= E[(H\mathbf{q} + w)(H\mathbf{q} + w)^T] \\
 &= E[H\mathbf{q}\mathbf{q}^T H^T] + E[ww^T] = H K_q H^T + K_w
 \end{aligned}$$

### The Linear MMSE Estimator

- Linear MMSE estimator:

$$\begin{aligned}\hat{E}[q | Y] &= K_{qY} K_Y^{-1} Y \\ &= K_q H^T (H K_q H^T + K_w)^{-1} Y \\ &= (K_q^{-1} + H^T K_w^{-1} H)^{-1} H^T K_w^{-1} Y \\ &\quad \text{(by Matrix Inversion Lemma)}\end{aligned}$$

- Expected error matrix:

$$\begin{aligned}E[ee^T] &= K_q - K_{qY} K_Y^{-1} K_{Yq} \\ &= K_q - K_q H^T (H K_q H^T + K_w)^{-1} H K_q \\ &= (K_q^{-1} + H^T K_w^{-1} H)^{-1} \quad \text{(by M.I.L.)}\end{aligned}$$

### Why Use Matrix Inversion Lemma?

- Interesting interpretation of:

$$\hat{E}[q | Y] = (K_q^{-1} + H^T K_w^{-1} H)^{-1} H^T K_w^{-1} Y$$

- If  $K_q$  is “large,” prior is uninformative:

$$\hat{E}[q | Y] \approx (H^T K_w^{-1} H)^{-1} H^T K_w^{-1} Y$$

Analogous to non-Bayesian LS solution on previous slides!

- Often computationally useful if  $\Theta$  is of smaller dimension than  $Y$ :

$$E[ee^T] = (K_q^{-1} + H^T K_w^{-1} H)^{-1}$$