ECE 7251: Signal Detection and Estimation

Spring 2002
Prof. Aaron Lanterman
Georgia Institute of Technology

Lecture 10, 1/28/02 (slides revised 2/8/02):
Estimation Under Additive Gaussian Noise
(a.k.a. Least Squares Solutions)

A Boring Definition Slide

- Define the gradient vector

$$\nabla_q g(q) \equiv \begin{bmatrix} \frac{\partial g_1(q)}{\partial q_1} & \frac{\partial g_2(q)}{\partial q_1} & \cdots & \frac{\partial g_M(q)}{\partial q_1} \\ \frac{\partial g_1(q)}{\partial q_2} & \frac{\partial g_2(q)}{\partial q_2} & & \frac{\partial g_M(q)}{\partial q_2} \\ \vdots & & \ddots & \vdots \\ \frac{\partial g_1(q)}{\partial q_M} & \frac{\partial g_2(q)}{\partial q_M} & \cdots & \frac{\partial g_M(q)}{\partial q_M} \end{bmatrix}$$

Warning: Different authors use different definitions!

Transformed CR Bounds

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

$$F_a(q) = [\nabla_q \{g(q)\}]^{-1} F_q(q) [\nabla_q^T \{g(q)\}]^{-1}$$
$$\mathrm{cov}_q(\hat{a}) \geq \nabla_q^T \{g(q)\} F_q^{-1}(q) \nabla_q \{g(q)\}$$

- If *g* has an inverse, we can also write:

$$F_a(q) = \nabla_a \{g^{-1}(a)\} F_q(q) \nabla_a^T \{g^{-1}(a)\}$$
$$\mathrm{cov}_q(\hat{a}) \geq [\nabla_a^T \{g^{-1}(a)\}]^{-1} F_q^{-1}(q) [\nabla_a \{g^{-1}(a)\}]^{-1}$$

- Usually substitute $q = g^{-1}(a)$ to get

$$F_a(q = g^{-1}(a)) = F_a(a)$$
$$\mathrm{cov}_{q=g^{-1}(a)}(\hat{a}) = \mathrm{cov}_a(\hat{a})$$

A Quick Word of Warning

Recall ML estimation is preserved under nonlinear transformations

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

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

Signal in Additive Gaussian Noise

- Signal in Gaussian noise; signal parameterized by *q*, perhaps in a nonlinear fashion

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

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

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

Least Squares Solutions

- ML solution must satisfy:

$$\nabla_q \{l(q)\} = \nabla_q \{s(q)\} K^{-1} [s(q) - \frac{1}{n} \sum_{i=1}^{n} y_i] = 0$$

- 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 Guassian!
- If *s(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:

$$F(q) = E[\nabla_q \{l(q)\} \nabla_q^T \{l(q)\}]$$
$$= E[\nabla_q \{s(q)\} K^{-1} [s(q) - y][s(q) - y]^T K^{-1} \nabla_q^T \{s(q)\}]$$
$$= \nabla_q \{s(q)\} K^{-1} \underbrace{E[[s(q) - y][s(q) - y]^T]}_{=K} K^{-1} \nabla_q^T \{s(q)\}]$$
$$= \nabla_q \{s(q)\} K^{-1} \nabla_q^T \{s(q)\}$$

- Aside: for complex (Goodman) Gaussian data:

$$F(q) = 2\mathrm{Re}\{\nabla_q \{s(q)\} K^{-1} \nabla_q^H \{s(q)\}\}$$

Linear Least Squares

- Suppose *s(q)=Hq*, i.e. *y= Hq+w*

$$\nabla_q \{s(\hat{q})\} K^{-1} [s(\hat{q}) - y_i]^T = 0$$
$$\nabla_q \{H\hat{q}\} K^{-1} [H\hat{q} - y] = 0$$
$$H^T K^{-1} [H\hat{q} - y] = 0$$
$$H^T K^{-1} H\hat{q} - H^T K^{-1} y = 0$$
$$H^T K^{-1} H\hat{q} = H^T K^{-1} y$$
$$\hat{q}_{LS} = (H^T K^{-1} H)^{-1} H^T K^{-1} y$$

Linear Least Squares Con't

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

- If data is Gaussian, then

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

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

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

Ex: Medical Imaging

- Emission tomography; projection data is Poisson

$$y_i \sim Poisson(Hq), \quad E[y_i] = Hq, \quad \mathrm{var}[y_i] = E[y_i]$$

- Approximate Poisson distribution with Gaussian

$$y_i \sim N(Hq, \underbrace{Hq}_{\text{Roughly estimate with } y_i})$$
$$y_i \sim N(Hq, y_i)$$

- 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!

$$q \sim N(0, K_q)$$
$$Y = Hq + W, \quad W \sim N(0, K_w)$$

- Recall formulas for linear MMSE:

$$\hat{E}[q \mid Y] = m_q + K_{qY} K_Y^{-1} (Y - m_Y)$$
$$E[ee^T] = K_q - K_{qY} K_Y^{-1} K_{Yq}$$

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

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

$$K_{Yq} = HK_q$$

$$K_Y = E[(Hq + w)(Hq + w)^T]$$
$$= E[Hqq^T H^T] + E[ww^T] = HK_q H^T + K_W$$

The Linear MMSE Estimator

- Linear MMSE estimator:

$$\hat{E}[q \mid Y] = K_{qY} K_Y^{-1} Y$$
$$= K_q H^T (H K_q H^T + K_W)^{-1} Y$$
$$= (K_q^{-1} + H^T K_W^{-1} H)^{-1} H^T K_W^{-1} Y$$

(by Matrix Inversion Lemma)

- Expected error matrix:

$$E[ee^T] = K_q - K_{qY} K_Y^{-1} K_{Yq}$$
$$= K_q - K_q H^T (H K_q H^T + K_W)^{-1} H K_q$$
$$= (K_q^{-1} + H^T K_W^{-1} H)^{-1}$$

(by M.I.L.)

Why Use Matrix Inversion Lemma?

- Interesting interpretation of:

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

- If K_q is "large," prior is uninformative:

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

Analogous to non-Bayesian LS solution on previous slides!

- Often computationally useful if Θ is of smaller dimension than Y:

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