Constrained Optimization

ECE 6279: Spatial Array Processing
Spring 2007
Lecture 16
2/19/07

Prof. Aaron D. Lanterman

School of Electrical & Computer Engineering
Georgia Institute of Technology
AL: 404-385-2548
<lanterma@ece.gatech.edu>

Where We Are in J&D

- Section 7.2 and Appendix C

Maximizing SNR for Known Colored Noise

- In Lecture 13, we showed that for

Data: $\underline{\mathbf{y}} = \mathbf{e}(\vec{k}^0)\underline{s} + \underline{\mathbf{n}}$

$$E\left[|\underline{s}|^2\right] = \sigma_s^2 \quad E\left[\underline{\mathbf{n}}\underline{\mathbf{n}}^H\right] = \mathbf{K}_{\mathbf{n}}$$

we can maximize SNR by choosing

$$z = \mathbf{a}^H \underline{\mathbf{y}} = \mathbf{e}^H(\vec{k}^0)\mathbf{K}_{\mathbf{n}}^{-1}\underline{\mathbf{y}}$$

$$\mathbf{a} = \mathbf{K}_{\mathbf{n}}^{-1}\mathbf{e}(\vec{k}^0)$$

Warning: Notation Change Ahead

- To match the book, we'll change $\mathbf{a}$ to $\mathbf{w}$
- This $\mathbf{w}$ should not be confused with the "shading," "tapering," aka "windowing" weights we had along the diagonal of a matrix called $\mathbf{W}$

Unknown Colored Noise

- What if you don't know $\mathbf{K}_\mathbf{n}$?
- Be pessimistic: assume data is mostly noise, which we want to minimize, so minimize beamformer output

$$E\left[|\underline{z}|^2\right] = E\left[\left|\mathbf{w}^H \underline{\mathbf{y}}\right|^2\right]$$

$$= E[\mathbf{w}^H \underline{\mathbf{y}}\underline{\mathbf{y}}^H \mathbf{w}] = \mathbf{w}^H \mathbf{R}_\mathbf{y} \mathbf{w}$$

Problem Specification

$$\mathbf{w}_{\lozenge} = \arg\min_{\mathbf{w}} \mathbf{w}^H \mathbf{R}_\mathbf{y} \mathbf{w}$$

- Has trivial solution of $\mathbf{w}_{\lozenge} = 0$
- To avoid nonsensical answer, add constraint $\mathbf{C}\mathbf{w} = \mathbf{c}$
- For ex., $\mathbf{e}^H(\vec{k})\mathbf{w} = 1$ assures signal in look direction passes unharmed

Real Vector-Parameter Optimization

$$\nabla_\mathbf{x} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_M} \end{bmatrix} \overset{set}{=} 0$$ to find stationary points

Gradients of Quadratic Forms (Real)

$$\left[\nabla_\mathbf{x}\{\mathbf{x}^T \mathbf{A}\mathbf{x}\}\right]_\ell = \left[\nabla_\mathbf{x}\left\{\sum_{ij} A_{ij} x_i x_j\right\}\right]_\ell$$

$$= \frac{d}{dx_l}\left\{\sum_{ij} A_{ij} x_i x_j\right\} = \sum_i A_{il} x_i + \sum_j A_{lj} x_j$$

$$= \mathbf{A}\mathbf{x} + \mathbf{A}^T\mathbf{x} = 2\mathbf{A}\mathbf{x}$$

If **A** is symmetric

Complex Parameter Optimization (1)

- How can we minimize a real-valued function of a complex variable? $f(\mathrm{Re}\{z\}, \mathrm{Im}\{z\})$
- Could take $\frac{\partial f}{\partial \mathrm{Re}\{z\}}, \frac{\partial f}{\partial \mathrm{Im}\{z\}},$

  but that gets messy

Complex Parameter Optimization (2)

$$\mathrm{Re}\{z\} = \frac{1}{2}[z + z^*], \quad \mathrm{Im}\{z\} = \frac{1}{2j}[z - z^*]$$

$$\begin{bmatrix} 2\mathrm{Re}\{z\} \\ 2j\,\mathrm{Im}\{z\} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}_{\text{invertible}} \begin{bmatrix} z \\ z^* \end{bmatrix}$$

- Can treat z and z* as independent variables

Complex Parameter Optimization (3)

- For scalars, stationary points can be found via

$$\frac{\partial f(z,z^*)}{\partial z} = 0 \quad \underline{\text{or}} \quad \underbrace{\frac{\partial f(z,z^*)}{\partial z^*} = 0}_{\text{usually easier}}$$

- For vectors, <u>must</u> use

$$\nabla_{z^*} f(z,z^*) = 0$$

  to find stationary points

The Lagrangian

treat λ and λ* independently

$$L(\mathbf{w},\boldsymbol{\lambda}) = \mathbf{w}^H \mathbf{R}_y \mathbf{w} + \underbrace{\boldsymbol{\lambda}^H (\mathbf{C}\mathbf{w} - \mathbf{c}) + \boldsymbol{\lambda}^T (\mathbf{C}^* \mathbf{w}^* - \mathbf{c}^*)}_{\text{real-valued}}$$

- Caveat: for Lagrange trick to work, columns of C must be linearly independent

Gradients of Quadratic Forms (Complex)

$$\left[\nabla_{\mathbf{w}^*}\{\mathbf{w}^H\mathbf{R}\mathbf{w}\}\right]_\ell = \left[\nabla_{\mathbf{w}^*}\left\{\sum_{ij} R_{ij} w_i^* w_j\right\}\right]_\ell$$

$$= \frac{\partial}{\partial w_l^*}\left\{\sum_{ij} R_{ij} w_i^* w_j\right\} = \sum_j R_{lj} w_j$$

$$\nabla_{\mathbf{w}^*}\{\mathbf{w}^H\mathbf{R}\mathbf{w}\} = \mathbf{R}\mathbf{w}$$

Gradiants of Inner Products (Complex)

$$\left[\nabla_{\mathbf{w}^*}\{\mathbf{w}^H\mathbf{v}\}\right]_\ell = \left[\nabla_{\mathbf{w}^*}\left\{\sum_{i} w_i^* v_i\right\}\right]_\ell = v_l$$

$$\nabla_{\mathbf{w}^*}\{\mathbf{w}^H\mathbf{v}\} = \mathbf{v}$$

Dissecting the Lagrangian

$$L(\mathbf{w},\boldsymbol{\lambda}) = \mathbf{w}^H\mathbf{R}_\mathbf{y}\mathbf{w} + \boldsymbol{\lambda}^H(\mathbf{C}\mathbf{w}-\mathbf{c}) + \boldsymbol{\lambda}^T(\mathbf{C}^*\mathbf{w}^* - \mathbf{c}^*)$$

$$\boldsymbol{\lambda}^T\mathbf{C}^*\mathbf{w}^* = (\boldsymbol{\lambda}^T\mathbf{C}^*\mathbf{w}^*)^T = \mathbf{w}^H\mathbf{C}^H\boldsymbol{\lambda}$$

$$\nabla_{\mathbf{w}^*} L(\mathbf{w},\boldsymbol{\lambda}) = \nabla_{\mathbf{w}^*}\{\mathbf{w}^H\mathbf{R}_\mathbf{y}\mathbf{w} + \mathbf{w}^H\mathbf{C}^H\boldsymbol{\lambda}\}$$

$$= \mathbf{R}_\mathbf{y}\mathbf{w} + \mathbf{C}^H\boldsymbol{\lambda}$$

Eliminating the Lagrange Multiplier

$$\mathbf{R}_\mathbf{y}\mathbf{w}_\Diamond + \mathbf{C}^H\boldsymbol{\lambda}_\Diamond = 0$$

$$\mathbf{w}_\Diamond = -\mathbf{R}_\mathbf{y}^{-1}\mathbf{C}^H\boldsymbol{\lambda}_\Diamond$$

want to get rid of Lagrange multiplier

$$= -\mathbf{R}_\mathbf{y}^{-1}\mathbf{C}^H[-(\mathbf{C}\mathbf{R}_\mathbf{y}^{-1}\mathbf{C}^H)^{-1}\mathbf{c}]$$

$$= \mathbf{R}_\mathbf{y}^{-1}\mathbf{C}^H(\mathbf{C}\mathbf{R}_\mathbf{y}^{-1}\mathbf{C}^H)^{-1}\mathbf{c}$$

Use constraint:

$$\mathbf{C}\mathbf{w}_\Diamond = \mathbf{c} \rightarrow -\mathbf{C}\mathbf{R}_\mathbf{y}^{-1}\mathbf{C}^H\boldsymbol{\lambda}_\Diamond = \mathbf{c}$$

$$\boldsymbol{\lambda}_\Diamond = -(\mathbf{C}\mathbf{R}_\mathbf{y}^{-1}\mathbf{C}^H)^{-1}\mathbf{c}$$

Power of Beamformer Output

$$\mathbf{w} = \mathbf{R}_{\mathbf{y}}^{-1}\mathbf{C}^{H}(\mathbf{C}\mathbf{R}_{\mathbf{y}}^{-1}\mathbf{C}^{H})^{-1}\mathbf{c}$$

Power of beamformer output: $\mathbf{w}^{H}\mathbf{R}_{\mathbf{y}}\mathbf{w} =$

$$\mathbf{c}^{H}(\mathbf{C}\mathbf{R}_{\mathbf{y}}^{-1}\mathbf{C}^{H})^{-H}\mathbf{C}\mathbf{R}_{\mathbf{y}}^{-H}\mathbf{R}_{\mathbf{y}}\mathbf{R}_{\mathbf{y}}^{-1}\mathbf{C}^{H}(\mathbf{C}\mathbf{R}_{\mathbf{y}}^{-1}\mathbf{C}^{H})^{-1}\mathbf{c}$$

(Using $\mathbf{A}^{H} = \mathbf{A} \rightarrow \mathbf{A}^{-H} \equiv (\mathbf{A}^{-1})^{H} = \mathbf{A}^{-1}$**)**

$$= \mathbf{c}^{H}(\mathbf{C}\mathbf{R}_{\mathbf{y}}^{-1}\mathbf{C}^{H})^{-1}\mathbf{C}\mathbf{R}_{\mathbf{y}}^{-1}\mathbf{C}^{H}(\mathbf{C}\mathbf{R}_{\mathbf{y}}^{-1}\mathbf{C}^{H})^{-1}\mathbf{c}$$

$$= \mathbf{c}^{H}(\mathbf{C}\mathbf{R}_{\mathbf{y}}^{-1}\mathbf{C}^{H})^{-1}\mathbf{c}$$