
Special Cases of Maximum-Likelihood Estimation

**ECE 6279: Spatial Array Processing
Spring 2011
Lecture 28**

Prof. Aaron D. Lanterman

**School of Electrical & Computer Engineering
Georgia Institute of Technology**

AL: 404-385-2548

<lanterma@ece.gatech.edu>

Copyright 2011 Aaron Lanterman



Our Deterministic Gaussian Model

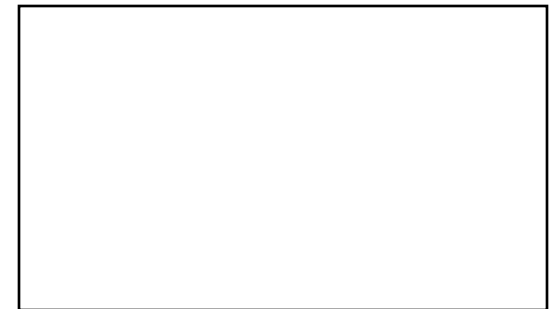
- N_s sources in additive noise

$$\underline{\mathbf{y}}(l) = \sum_{n=1}^{N_s} \mathbf{e}(\boldsymbol{\theta}_n) s_n(l) + \underline{\mathbf{n}}(l)$$

$$\underline{\mathbf{n}} \sim CN(0, \mathbf{K}_n)$$

$$\mathbf{s}(l) = \begin{bmatrix} s_1(l) \\ \vdots \\ s_{N_s}(l) \end{bmatrix}$$

Modeled as deterministic parameters we need to estimate



The Single-Source Case

- **Recall ML estimate of signal for a fixed Θ**

$$\hat{\mathbf{s}} = \left[\mathbf{D}^H(\Theta) \mathbf{K}_n^{-1} \mathbf{D}(\Theta) \right]^{-1} \mathbf{D}(\Theta)^H \mathbf{K}_n^{-1} \mathbf{y}$$

- **Suppose we only have one source**

$$\hat{s} = \left[\mathbf{e}^H(\theta) \mathbf{K}_n^{-1} \mathbf{e}(\theta) \right]^{-1} \mathbf{e}(\theta)^H \mathbf{K}_n^{-1} \mathbf{y}$$

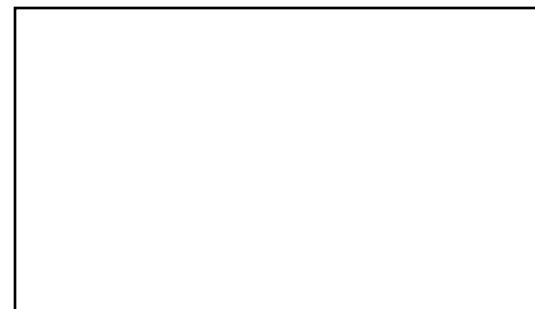


A Blast from the Past

$$\hat{s} = \frac{\mathbf{e}^H(\boldsymbol{\theta})\mathbf{K}_n^{-1}}{\underbrace{\mathbf{e}^H(\boldsymbol{\theta})\mathbf{K}_n^{-1}\mathbf{e}(\boldsymbol{\theta})}_{\equiv \mathbf{a}^H}} \mathbf{y} = \mathbf{a}^H \mathbf{y}$$

- Recall $\mathbf{a}^H \propto \mathbf{e}^H(\boldsymbol{\theta})\mathbf{K}_n^{-1}$ are the weights that maximize SNR

–Showed this
in Lecture 13

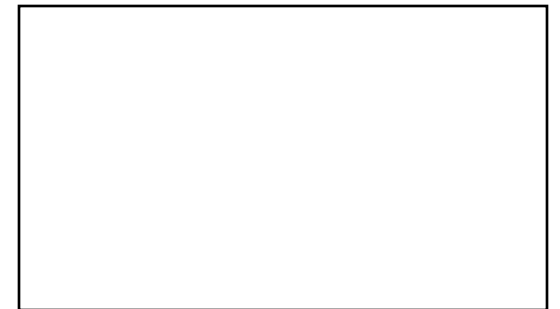


Another Blast from the Past

- If $\mathbf{K}_n = \sigma_n^2 \mathbf{I}$ then

$$\hat{s} = \frac{\mathbf{e}^H(\boldsymbol{\theta}) \mathbf{K}_n^{-1}}{\mathbf{e}^H(\boldsymbol{\theta}) \mathbf{K}_n^{-1} \mathbf{e}(\boldsymbol{\theta})} \mathbf{y} = \frac{\mathbf{e}^H(\boldsymbol{\theta})}{\mathbf{e}^H(\boldsymbol{\theta}) \mathbf{e}(\boldsymbol{\theta})} \mathbf{y}$$
$$= \frac{\mathbf{e}^H(\boldsymbol{\theta})}{M} \mathbf{y}$$

- Old delay-and-sum conventional beamformer



Maximum-Likelihood Procedure

- ML procedure said to maximize

$$\text{tr} \left\{ \mathbf{D}(\mathbf{D}^H \mathbf{D})^{-1} \mathbf{D}^H \hat{\mathbf{R}}_y \right\} \text{ over } \Theta$$

- For one source

$$\begin{aligned} \text{tr} \left\{ \mathbf{e}(\mathbf{e}^H \mathbf{e})^{-1} \mathbf{e}^H \hat{\mathbf{R}}_y \right\} &= \frac{1}{M} \text{tr} \left\{ \mathbf{e} \mathbf{e}^H \hat{\mathbf{R}}_y \right\} \\ &= \frac{1}{M} \text{tr} \left\{ \mathbf{e}^H \hat{\mathbf{R}}_y \mathbf{e} \right\} = \underbrace{\mathbf{e}^H (\boldsymbol{\theta}) \hat{\mathbf{R}}_y \mathbf{e}(\boldsymbol{\theta})}_{\text{Power out of our old conventional}} / M \end{aligned}$$

delay-and-sum beamformer!



Take-Home Messages (1)

- If you have a **single source** in **Gaussian noise**, maximizing the SNR turns out to be optimal (in the ML estimation sense) under the deterministic Gaussian model
- If you have non-Gaussian noise or multiple sources, maximizing SNR is **not** the optimal ML procedure
 - But usually still pretty useful!



Take-Home Message (2)

- If you have a **single source** in white Gaussian noise, the conventional delay-and-sum beamformer is optimal in the ML sense under the deterministic Gaussian model
- If you have non-Gaussian noise or multiple sources the conventional beamformer is not the optimal ML procedure
 - But usually still pretty useful!



Capon's Argument (1)

- **ML signal estimate for one source is**

$$\hat{\mathbf{s}} = \frac{\mathbf{e}^H(\boldsymbol{\theta})\mathbf{K}_n^{-1}}{\mathbf{e}^H(\boldsymbol{\theta})\mathbf{K}_n^{-1}\mathbf{e}(\boldsymbol{\theta})} \mathbf{y}$$

- **In Capon's application (see p. 358-359 of J&D), he didn't have a good estimate of \mathbf{K}_n**
- **Capon handwaved and said to try replacing \mathbf{K}_n with $\hat{\mathbf{R}}_y$**



Capon's Argument (2)

- **Totally ad-hoc from an ML standpoint; $\hat{\mathbf{R}}_y$ is a terrible estimate of \mathbf{K}_n**
- **But if we go ahead and do it,**

we get

$$\hat{s} = \frac{\mathbf{e}^H(\boldsymbol{\theta})\hat{\mathbf{R}}_y^{-1}}{\mathbf{e}^H(\boldsymbol{\theta})\hat{\mathbf{R}}_y^{-1}\mathbf{e}(\boldsymbol{\theta})} \mathbf{y}$$

which is the MVDR beamformer!



A Great Confusion

- **MVDR (a.k.a. Capon's method) is sometimes erroneously referred to as "maximum likelihood"**
 - That's nonsense! MVDR does not arise from any known statistical ML procedure
- **Our earlier derivation of MVDR from a constrained optimization procedure is due to Lacoss (1971)**



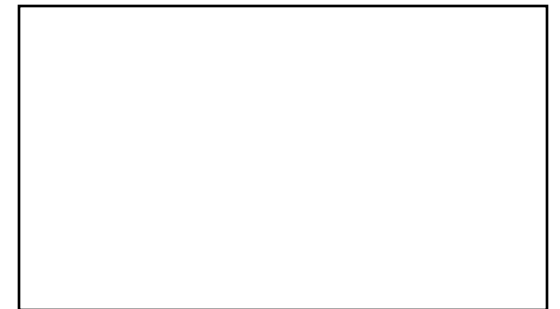
Recollections and Observations

- Remember MVDR was a solution to

$$\mathbf{w}^H \mathbf{R}_y \mathbf{w} \text{ s.t. } \mathbf{w}^H \mathbf{e}(\theta) = 1$$

- Yes, this is an optimization problem; but it's not a maximum-likelihood optimization problem

- Doesn't require a probability model for the data



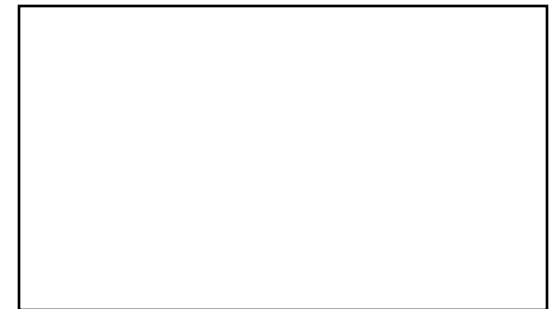
Our Stochastic Gaussian Model

- **N sources in additive noise**

$$\underline{\mathbf{y}}(l) = \sum_{n=1}^N \mathbf{e}(\boldsymbol{\theta}_n) \underline{s}_n(l) + \underline{\mathbf{n}}(l)$$

$$\underline{\mathbf{n}} \sim CN(0, \mathbf{K}_n)$$

$$\underline{\mathbf{s}} = \begin{bmatrix} \underline{s}_1 \\ \vdots \\ \underline{s}_N \end{bmatrix} \sim CN(0, \mathbf{K}_s)$$



Maximum-Likelihood Estimation

- **Goal: maximize**

$$-L \ln \det \mathbf{K}_y(\xi) - L \operatorname{tr} \left[\hat{\mathbf{K}}_y \mathbf{K}_y^{-1}(\xi) \right]$$

where

$$\mathbf{K}_y(\xi) = \mathbf{D}(\Theta) \mathbf{K}_s \mathbf{D}^H(\Theta) + \mathbf{K}_n$$

over ξ , which represents all the parameters (angles, signal powers, correlation coefficients)



Asymptotic ML Est. for Stochastic Model

- **If you have**
 - “Lots of snapshots,”
 - Uncorrelated sources,
 - and I.I.D. receiver noise
- **Then VT-IV, pp. 985-991 shows that you can get rid of the power parameters $\sigma_1^2, \dots, \sigma_N^2$ and find ML estimates of angles by maximizing...**



Asymptotic ML Procedure

- **New goal: maximize**

$$-\ln \det \left[\mathbf{P}_D \hat{\mathbf{K}}_y \mathbf{P}_D + \sigma_n^2 \mathbf{P}_D^\perp \right]$$
$$-\text{tr} \left[\mathbf{P}_D^\perp \hat{\mathbf{K}}_y \right] / \sigma_n^2$$

over Θ

where $\mathbf{P}_D = \mathbf{D}(\mathbf{D}^H \mathbf{D})^{-1} \mathbf{D}^H$

$$\mathbf{P}_D^\perp = \mathbf{I} - \mathbf{P}_D$$

