Sparse Bayesian Learning for Wavefields from Sensor Array Data

Christoph Mecklenbräuker and Peter Gerstoft

\[ Y = A X + N \]

\[ Y \] is \( n \times L \) measurement matrix

\[ A \] is \( n \times m \) measurement/Dictionary matrix, \( m \gg n \)

\[ X \] is \( m \times L \) desired matrix which is sparse with \( r \) nonzero entries, \( r \ll m \)

\[ N \] is the measurement noise

Multiple and single snapshot compressive beamforming

Peter Gerstoft

CS approach to geophysical data analysis

CS of Earthquakes

Yao, GRL 2011, PNAS 2013

Sequential CS

Mecklenbrauker, TSP 2013

CS beamforming

Xenaki, JASA 2014, 2015

Gerstoft JASA 2015

CS fathometer

Yardim, JASA 2014

CS Sound speed estimation

Bianco, JASA 2016

CS matched field

Gemba, JASA 2016
DOA estimation with arrays

\[ m \in [1, \ldots, M]: \text{sensor} \]
\[ n \in [1, \ldots, N]: \text{look direction} \]
\[ N \gg M \]

\[ y = A_{M \times N} x \]

\[ y = [y_1, \ldots, y_M]^T, \quad x = [x_1, \ldots, x_N]^T \]
\[ A = [a_1, \ldots, a_N] \]
\[ a_n = \frac{1}{\sqrt{M}} [e^{i \frac{2 \pi}{\lambda} r_1 \sin \theta_n}, \ldots, e^{i \frac{2 \pi}{\lambda} r_M \sin \theta_n}]^T \]

CBF: \[ \hat{x} = A^H y \]

CS: \[ \min \left( \|Ax - y\|_2^2 + \mu \|x\|_1 \right) \]

ULA \( M = 8 \), \( \frac{\theta}{\lambda} = \frac{1}{2} \), \([\theta_1, \theta_2] = [0, 5]^\circ\), SNR = 20 dB

High resolution
No sidelobes

MAP

Likelihood (noise complex Gaussian)
\[ p(y | x) \propto \exp \left( -\frac{\|Ax - y\|_2^2}{\sigma^2} \right) \]

Prior (Laplacian)
\[ p(x) \propto \exp \left( -\frac{\|x\|_1}{\nu} \right) \]

Bayes rule
\[ p(x | y) \propto p(y | x)p(x) \propto \exp \left( -\frac{\|Ax - y\|_2^2}{\sigma^2} - \frac{\|x\|_1}{\nu} \right) \]

MAP
\[ \hat{x}_{\text{MAP}} = \arg \min_{\hat{x}} \left[ \|y - Ax\|_2^2 + \mu \|x\|_1 \right] = \hat{x}_{\text{LASSO}}(\mu), \]

\[ \mu \]
CS: LASSO for Multiple Snapshots

Row sparsity constraint

\[ \|X\|_{21} = \sum_{n=1}^{N} x_{n}^{2} \quad \text{with} \quad x_{n}^{2} = \sqrt{\sum_{l=1}^{L} |x_{nl}|^2} \]

CS

\[ \hat{X} = \arg \min_{X \in C^{M \times L}} \|Y - AX\|_2^2 + \mu \|X\|_{21} \]

The complex amplitude of \(X\) is allowed to vary across snapshots, but the sparsity pattern is assumed to be constant across snapshots.

Problem with Degrees of Freedom

• As the number of snapshots (=observations) increases, so does the number of unknown complex source amplitudes.

• PROBLEM: LASSO for multiple snapshots estimates the realizations of the random complex source amplitudes.

• However, we would be satisfied if we just estimated their power

\[ \gamma_m = \mathbb{E}\{|x_{ml}|^2\} \]

• Note that \( \gamma_m \) does not depend on snapshot index \(l\).
The problem revisited

Multiple snapshots: \( l = 1, \ldots, L \)

\[ y_l = Ax_l + n_l. \]

Combining all snapshots (1) becomes

\[ Y = AX + N, \]

Likelihood function (observations conditioned on source amplitudes):

\[
p(Y|X; \sigma^2) = \frac{\exp \left( -\frac{1}{\sigma^2} \| Y - AX \|^2_F \right)}{(\pi \sigma^2)^{NL}}
\]

Sparsity promoted by Gaussian prior?

Sparse Bayesian Learning and the Relevance Vector Machine

Michael E. Tipping
Microsoft Research
St George House, 1 Guildhall Street
Cambridge CB3 3NH, U.K.

We encode a preference for smoother (less complex) functions by making the popular choice of a zero-mean Gaussian prior distribution over \( w \):

\[
p(w|\alpha) = \prod_{i=0}^{N} \mathcal{N}(w_i|0, \alpha_i^{-1}), \quad (5)
\]

with \( \alpha \) a vector of \( N + 1 \) hyperparameters. Importantly, there is an individual hyperparameter associated independently with every weight, moderating the strength of the prior thereon.\(^4\)

To complete the specification of this hierarchical prior, we must define hyperpriors over \( \alpha \), as well as over the final remaining parameter in the model, the noise variance \( \sigma^2 \). These quantities are examples of scale parameters, and suitable priors thereover are Gamma distributions (see, e.g., Berger, 1985):
Sparsity promoted by Gaussian prior?

B. Prior

We assume a multivariate zero-mean circularly symmetric complex normal distribution for the source vector $x_l$ with the hyperparameter $\gamma = (\gamma_1, \ldots, \gamma_M)^T$, with all $\gamma_m \geq 0$.

\[
p(X; \gamma) = \prod_{l=1}^{L} \prod_{m=1}^{M} p_m(x_{ml}; \gamma_m),
\]

\[
p_m(x_{ml}; \gamma_m) = \begin{cases} 
\delta(x_{ml}), & \text{for } \gamma_m = 0; \\
\frac{1}{\pi \gamma_m} e^{-|x_{ml}|^2/\gamma_m}, & \text{for } \gamma_m > 0.
\end{cases}
\]

Proceeding with Bayes rule

Given the array data observations $Y$, the posterior pdf is found using Bayes rule conditioned on $\gamma, \sigma^2$.

\[
p(X|Y; \gamma, \sigma^2) \propto \frac{p(Y|X, \gamma, \sigma^2)p(X|\gamma; \sigma^2)}{p(Y; \gamma, \sigma^2)}
\]

The denominator $p(Y; \gamma, \sigma^2)$ is the evidence term which for a given $\gamma, \sigma^2$ is a normalization factor and is neglected at first,

\[
p(X|Y; \gamma, \sigma^2) \propto p(Y|X; \sigma^2)p(X; \gamma)
\]

\[
\propto e^{-\text{tr}((X-\mu_x)^H \Sigma_x^{-1}(X-\mu_x))}
\]

\[
\propto \frac{1}{(\pi^N \det \Sigma_x)^L}
\]
Evidence

To determine the hyperparameters $\gamma_1, \gamma_2, \ldots, \gamma_M$, and $\sigma^2$ the evidence is maximized.

The evidence is the product of the likelihood and the prior integrated over the complex source signals.

$$p(Y; \gamma, \sigma^2) = \int_{\mathbb{R}^{2ML}} p(Y \mid X; \sigma^2)p(X; \gamma) \, dX,$$

$$= \frac{e^{- \text{tr}(Y^H \Sigma_y^{-1} Y)}}{(\pi^N \det \Sigma_y)^L}$$

$$\log p(Y; \gamma, \sigma^2) \propto - \text{tr} \left( Y^H \Sigma_y^{-1} Y \right) - L \log \det \Sigma_y$$

Maximizing the Evidence: Covariance Fitting

Solving for $\tilde{\Gamma}$ yields

$$\tilde{\Gamma} = A_M^{+} \left( S_y - \bar{\sigma}^2 I_N \right) A_M^{+H}, \quad (25)$$

$$\bar{\sigma}^2 = \frac{1}{N - K} \text{tr} \left( (I_N - A_M A_M^{+}) S_y \right), \quad (26)$$

where $A_M^{+}$ denotes the Moore-Penrose pseudo inverse of $A_M$.

with the sample covariance matrix

$$S_y = \frac{1}{L} YY^H. \quad (23)$$
Maximizing the Evidence: Covariance Fitting

\[ \frac{\partial \log p(Y; \gamma, \sigma^2)}{\partial \gamma_m} = \frac{1}{\gamma_m^2 \mathcal{L}} \left\| \mu_m \right\|_2^2 - a_m^H \Sigma_y^{-1} a_m = 0 \]

\[ \gamma_m^{\text{new}} = \frac{1}{\sqrt{\mathcal{L}}} \left\| \mu_m \right\|_2 / \sqrt{a_m^H \Sigma_y^{-1} a_m}. \quad (\text{RVM-ML1}) \]

Exploiting Jaffer’s necessary condition

\[ \gamma_m^{\text{new}} = \frac{1}{\sqrt{\mathcal{L}}} \left\| \mu_m \right\|_2 / \sqrt{a_m^H \Sigma_y^{-1} a_m}. \quad (\text{RVM-ML1}) \]

\[ A_M^H \left( S_y - \bar{\Sigma}_y \right) A_M = 0. \]

\[ \gamma_m^{\text{new}} = \frac{1}{\sqrt{\mathcal{L}}} \left\| \mu_m \right\|_2 / \sqrt{a_m^H S_y^{-1} a_m}. \quad (\text{RVM-ML}) \]

Sparse Bayesian Learning Algorithm

0: Given: $\mathbf{A} \in \mathbb{C}^{N \times M}$, $\mathbf{Y} \in \mathbb{C}^{N \times L}$, $s \in \mathbb{N}$
Given: $\sigma_0^2 > 0, \gamma_0 > 0, \epsilon_{\text{max}} > 0, j_{\text{max}} \in \mathbb{N}$

1: sample covariance matrix $\mathbf{S}_y = \frac{1}{L} \mathbf{Y} \mathbf{Y}^H$
2: initialize $j = 0$, $\sigma^2 = \sigma_0^2$, $\Gamma = \gamma_0 \mathbf{I}_s$
3: while ($\epsilon > \epsilon_{\text{max}}$) and ($j < j_{\text{max}}$)
4: $j = j + 1$
5: $\gamma_{\text{old}} = \gamma$, $\Gamma = \text{diag}(\gamma_1, \ldots, \gamma_M)$
6: $\Sigma_y = \sigma^2 \mathbf{I}_N + \mathbf{A} \Gamma \mathbf{A}^H$  \hspace{1cm} (16)
7: $\mu_m = \gamma_m a_m^H \Sigma_y^{-1} \mathbf{Y}$  \hspace{1cm} (13)
8: $\gamma_m^{\text{new}} = \frac{1}{\sqrt{L}} \| \mu_m \|_2 / \sqrt{a_m^H \Sigma_y^{-1} a_m}$  \hspace{1cm} (RVM-ML1)

9: $\mathcal{M} = \{ m \in \mathbb{N} | \gamma_m > 0 \} = \{ m_1, \ldots, m_s \}$  \hspace{1cm} (6)
10: $\mathbf{A}_\mathcal{M} = (a_{m_1}, \ldots, a_{m_K})$
11: $\bar{\sigma}^2 = \frac{1}{N-K} \text{tr} \left( (\mathbf{I}_N - \mathbf{A}_\mathcal{M} \mathbf{A}_\mathcal{M}^+) \mathbf{S}_y \right)$  \hspace{1cm} (26)
12: $\epsilon = \| \gamma - \gamma_{\text{old}} \|_2 / \| \gamma_{\text{old}} \|_2$  \hspace{1cm} (32)
13: end
14: Output: $\mathcal{M}$

Sparse Bayesian Learning Algorithm

0: Given: $\mathbf{A} \in \mathbb{C}^{N \times M}$, $\mathbf{Y} \in \mathbb{C}^{N \times L}$, $s \in \mathbb{N}$
Given: $\sigma_0^2 > 0, \gamma_0 > 0, \epsilon_{\text{max}} > 0, j_{\text{max}} \in \mathbb{N}$

1: sample covariance matrix $\mathbf{S}_y = \frac{1}{L} \mathbf{Y} \mathbf{Y}^H$
2: initialize $j = 0$, $\sigma^2 = \sigma_0^2$, $\Gamma = \gamma_0 \mathbf{I}_s$
3: while ($\epsilon > \epsilon_{\text{max}}$) and ($j < j_{\text{max}}$)
4: $j = j + 1$
5: $\gamma_{\text{old}} = \gamma$, $\Gamma = \text{diag}(\gamma_1, \ldots, \gamma_M)$
6: $\Sigma_y = \sigma^2 \mathbf{I}_N + \mathbf{A} \Gamma \mathbf{A}^H$  \hspace{1cm} (16)
7: $\mu_m = \gamma_m a_m^H \Sigma_y^{-1} \mathbf{Y}$  \hspace{1cm} (13)
8: $\gamma_m^{\text{new}} = \frac{1}{\sqrt{L}} \| \mu_m \|_2 / \sqrt{a_m^H \mathbf{S}_y^{-1} a_m}$  \hspace{1cm} (RVM-ML)

9: $\mathcal{M} = \{ m \in \mathbb{N} | \gamma_m > 0 \} = \{ m_1, \ldots, m_s \}$  \hspace{1cm} (6)
10: $\mathbf{A}_\mathcal{M} = (a_{m_1}, \ldots, a_{m_K})$
11: $\bar{\sigma}^2 = \frac{1}{N-K} \text{tr} \left( (\mathbf{I}_N - \mathbf{A}_\mathcal{M} \mathbf{A}_\mathcal{M}^+) \mathbf{S}_y \right)$  \hspace{1cm} (26)
12: $\epsilon = \| \gamma - \gamma_{\text{old}} \|_2 / \| \gamma_{\text{old}} \|_2$  \hspace{1cm} (32)
13: end
14: Output: $\mathcal{M}$
### Sparse Bayesian Learning Algorithm

1. Given: $A \in \mathbb{C}^{N \times M}$, $Y \in \mathbb{C}^{N \times L}$, $s \in \mathbb{N}$

2. Given: $\sigma_0^2 > 0, \gamma_0 > 0, \epsilon_{\text{max}} > 0, j_{\text{max}} \in \mathbb{N}$

3. sample covariance matrix $S_y = \frac{1}{L} YY^H$

4. initialize $j = 0, \sigma^2 = \sigma_0^2, \Gamma = \gamma_0 I_s$

5. while ($\epsilon > \epsilon_{\text{max}}$) and ($j < j_{\text{max}}$)

6. $j = j + 1$

7. $\gamma_{\text{old}} = \gamma, \Gamma = \text{diag}(\gamma_1, \ldots, \gamma_M)$

8. $\Sigma_y = \sigma^2 I_N + A \Gamma A^H$

9. $\mu_m = \gamma_m a_m^H \Sigma_y^{-1} Y$

10. $\gamma_{\text{new}} = \frac{1}{L} \| \mu_m \|_2^2 + (\Sigma_{\hat{w}})_{mm}$ (SBL-EM)

11. $\mathcal{M} = \{ m \in \mathbb{N} | \gamma_m > 0 \} = \{ m_1, \ldots, m_s \}$

12. $A_{\mathcal{M}} = (a_{m_1}, \ldots, a_{m_K})$

13. $\hat{\sigma}^2 = \frac{1}{N - K} \text{tr} \left( (I_N - A_{\mathcal{M}} A_{\mathcal{M}}^+) S_y \right)$

14. $\epsilon = \| \gamma - \gamma_{\text{old}} \|_2 / \| \gamma_{\text{old}} \|_2$

---


### Example Scenario

In the simulation, we consider an array with $N = 20$ antenna elements and intersensor spacing $d = \lambda/2$.

The DOAs for plane wave arrivals are assumed to be on a fine angular grid $[-90:0.5:90] \circ$ and $L = 50$ snapshots are observed.

The CS solution is found using LASSO extended to multiple measurement vectors (multiple snapshots)

3 sources at DOAs $[-3, 2, 75]$ degrees with magnitudes $[4, 13, 10]$. 

Example Scenario

Source 1
DOA = -3°
Magnitude = 4

Source 2
DOA = +2°
Magnitude = 13

Source 3
DOA = +75°
Magnitude = 10

N = 20 elements

Example Results

a) $P_{[dB]}$

b) CS SNR=0, RMSE: 1.2

b) CBF SNR=0, RMSE: 16%

b) SBL SNR=0, RMSE: 0.64

DOA [°]

Array SNR (dB)

Bin count

RMV
SBL-EM
CS
Exhaust
CBF
Music
Example RMSE Performance

\[
\text{SNR} = 10 \log_{10} \frac{E \{ \|Ax_i\|^2 \}}{E \{ \|n_i\|^2 \}} \text{ (dB)},
\]

\[
\text{RMSE} = \sqrt{E \left[ \frac{1}{K} \sum_{k=1}^{K} (\hat{\theta}_k - \theta_{\text{true}})^2 \right]}
\]

Example CPU Time

\[
\text{CPU time (s)}
\]

\[
\text{Snapshots}
\]
Conclusions

- Sparse Bayesian Learning for complex valued array data using evidence maximization.

- In examples it is ~ 50% faster than the SBL Expectation Maximization (SBL-EM) approach.

- For multiple measurement vectors (snapshots) with stationary sources the benefit of RVM-ML is pronounced.
  - For each DOA it uses the hyperparameter $\gamma_m$ as a proxy, with computational effort independent of no. of snapshots.
  - Increasing no. of snapshots improves the RMSE.
  - The RMSE performance of RVM and exhaustive search are equal in this example.

References

References


