Choosing the diagonal loading factor for linear signal estimation using cross validation

Jun Tong, Qinghua Guo, Jiangtao Xi, Yanguang Yu[1]
Peter J. Schreier[2]

[1]School of Electrical, Computer & Telecommunication Engineering
The University of Wollongong, Australia
jtong@uow.edu.au

[2]Signal and System Theory Group
Universität Paderborn, Germany
peter.schreier@sst.upb.de

ICASSP 2016

LMMSE estimation

- Consider a system with signal (input) $\mathbf{x}$ and measurement $\mathbf{y}$
- Linear minimum mean-squared error (LMMSE) estimator

$$\widehat{\mathbf{x}} = \mathbf{C}_{yx}^{\dagger}\mathbf{C}_{yy}^{-1}\mathbf{y}$$

minimizes

$$\mathrm{MSE}_{x} \triangleq \mathrm{E}_{x}[||\mathbf{x} - \widehat{\mathbf{x}}||^{2}]$$

- In practice, **sample covariance matrices** (SCMs) computed from length-T training data:

$$\widehat{\mathbf{C}}_{yy} = \frac{1}{T}\mathbf{Y}\mathbf{Y}^{\dagger}, \quad \widehat{\mathbf{C}}_{yx} = \frac{1}{T}\mathbf{Y}\mathbf{X}^{\dagger}$$

- With **low sample support**, LMMSE estimator may perform poorly due to **model mismatch**

LMMSE estimation

- Consider a system with signal (input) $\mathbf{x}$ and measurement $\mathbf{y}$
- Linear minimum mean-squared error (LMMSE) estimator

$$\widehat{\mathbf{x}} = \mathbf{C}_{\mathrm{yx}}^{\dagger}\mathbf{C}_{\mathrm{yy}}^{-1}\mathbf{y}$$

  minimizes

$$\mathrm{MSE}_{\mathrm{x}} \triangleq \mathrm{E}_{\mathrm{x}}[||\mathbf{x} - \widehat{\mathbf{x}}||^2]$$

- In practice, **sample covariance matrices** (SCMs) computed from length-T training data:

$$\widehat{\mathbf{C}}_{\mathrm{yy}} = \frac{1}{T}\mathbf{Y}\mathbf{Y}^{\dagger}, \quad \widehat{\mathbf{C}}_{\mathrm{yx}} = \frac{1}{T}\mathbf{Y}\mathbf{X}^{\dagger}$$

- With **low sample support**, LMMSE estimator may perform poorly due to **model mismatch**

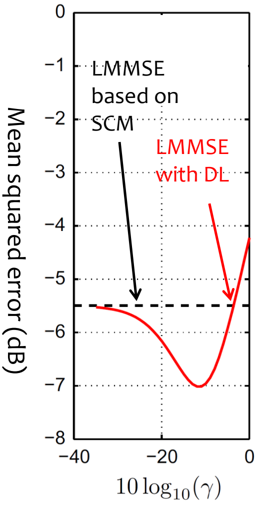
LMMSE estimation with diagonal loading

- Robustness can be improved by diagonal loading (DL) with **diagonal loading factor** (DLF) $\gamma \geq 0$:

$$\widehat{\mathbf{x}} = \widehat{\mathbf{C}}_{\mathrm{yx}}^{\dagger} \left(\widehat{\mathbf{C}}_{\mathrm{yy}} + \gamma \mathbf{I} \right)^{-1} \mathbf{y}$$

- A.k.a. **Tikhonov regularization** or **ridge regression**
- Improves **condition number** of the matrix to be inverted
- Achieves a better **bias-variance trade-off** $\Rightarrow$ lower MSE

How do we choose the DLF?



- DLF γ significantly affects performance
- Typical ad-hoc choice:

$$\gamma = 10\lambda_{\min}$$

- Need methods to automatically tune the DLF

Objective of this work

Given the estimated covariance matrices, automatically choose the optimal γ for

$$\widehat{\mathbf{x}}_{\gamma} = \widehat{\mathbf{C}}_{yx}^{\dagger} \left(\widehat{\mathbf{C}}_{yy} + \gamma \mathbf{I} \right)^{-1} \mathbf{y}$$

such that the MSE of estimating $\mathbf{x}$ is minimized:

$$\gamma^{*} = \arg\min_{\gamma} \mathrm{E}_{x}[||\mathbf{x} - \widehat{\mathbf{x}}_{\gamma}||^{2}]$$

- A more general problem: Optimize the shrinkage factors (α, γ) for the estimate

$$\widehat{\mathbf{x}}_{\alpha,\gamma} = \widehat{\mathbf{C}}_{yx}^{\dagger} \left(\alpha \widehat{\mathbf{C}}_{yy} + \gamma \mathbf{I} \right)^{-1} \mathbf{y}$$

which reduces to DL for $\alpha = 1$

Objective of this work

Given the estimated covariance matrices, automatically choose the optimal γ for

$$\widehat{\mathbf{x}}_{\gamma} = \widehat{\mathbf{C}}_{\mathsf{yx}}^{\dagger} \left(\widehat{\mathbf{C}}_{\mathsf{yy}} + \gamma \mathbf{I} \right)^{-1} \mathbf{y}$$

such that the MSE of estimating $\mathbf{x}$ is minimized:

$$\gamma^* = \arg\min_{\gamma} \mathrm{E}_{\mathsf{x}}[||\mathbf{x} - \widehat{\mathbf{x}}_{\gamma}||^2]$$

- A more general problem: Optimize the shrinkage factors (α, γ) for the estimate

$$\widehat{\mathbf{x}}_{\alpha,\gamma} = \widehat{\mathbf{C}}_{\mathsf{yx}}^{\dagger} \left(\alpha \widehat{\mathbf{C}}_{\mathsf{yy}} + \gamma \mathbf{I} \right)^{-1} \mathbf{y}$$

which reduces to DL for $\alpha = 1$

Related work

Techniques based on **random matrix theory** (RMT) and **large system assumption**

1. Optimize **estimation of the covariance matrix**
   - **Examples:** Ledoit and Wolf (J. Multivariate Analysis 2004), Stoica et al. (TSP 2008), Chen et al. (TSP 2010)
   - Achieves near-optimal covariance matrix estimation
   - But generally suboptimal for **signal estimation**
2. Maximize **SINR**
   - **Examples:** Mestre and Lagunas (TSP 2006), Zhang et al. (TSP 2013)
   - Generally suboptimal for minimizing MSE
3. Minimize **MSE**
   - **Examples:** Wen et al. (SPL 2013), Zhang et al. (TSP 2013)
   - based on SCM
   - do not account for **differently distributed** training and application data

Related work

Techniques based on **random matrix theory** (RMT) and **large system assumption**

1. Optimize **estimation of the covariance matrix**
   - **Examples:** Ledoit and Wolf (J. Multivariate Analysis 2004), Stoica et al. (TSP 2008), Chen et al. (TSP 2010)
   - Achieves near-optimal covariance matrix estimation
   - But generally suboptimal for **signal estimation**
2. Maximize **SINR**
   - **Examples:** Mestre and Lagunas (TSP 2006), Zhang et al. (TSP 2013)
   - Generally suboptimal for minimizing MSE
3. Minimize **MSE**
   - **Examples:** Wen et al. (SPL 2013), Zhang et al. (TSP 2013)
   - based on SCM
   - do not account for **differently distributed** training and application data

Related work

Techniques based on **random matrix theory** (RMT) and **large system assumption**

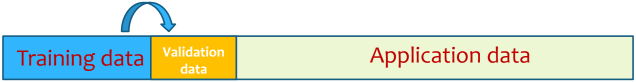
1. Optimize **estimation of the covariance matrix**
   - **Examples:** Ledoit and Wolf (J. Multivariate Analysis 2004), Stoica et al. (TSP 2008), Chen et al. (TSP 2010)
   - Achieves near-optimal covariance matrix estimation
   - But generally suboptimal for **signal estimation**
2. Maximize **SINR**
   - **Examples:** Mestre and Lagunas (TSP 2006), Zhang et al. (TSP 2013)
   - Generally suboptimal for minimizing MSE
3. Minimize **MSE**
   - **Examples:** Wen et al. (SPL 2013), Zhang et al. (TSP 2013)
   - based on SCM
   - do not account for **differently distributed** training and application data

This work

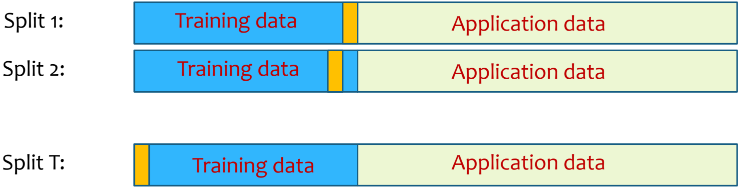
- We propose choosing the DLF based on **cross-validation** (CV)
- We derive **computationally efficient** calculation schemes
- Not based on random matrix theory
- Explicitly target the **minimization of the MSE** for signal estimation
- Allow **different distributions** for training and application data

Leave-one-out cross validation (LOOCV)

- Choosing γ is a **model selection problem**
- Assume first that the training and application data are identically distributed
- Reserve some of the training data for model validation **under the signal estimation criterion**:



- LOOCV splits repeatedly, reserving **one symbol** for validation each time:



Direct implementation of LOOCV is expensive

Using SCMs and T samples, LOOCV chooses

$$\gamma^* = \arg\min_{\gamma} \frac{1}{T} \sum_{i=1}^{T} ||\mathbf{x}_i - \mathbf{W}^{\dagger}_{\sim i,\gamma}\mathbf{y}_i||^2$$

with $\mathbf{W}_{\sim i,\gamma} = \left(\mathbf{Y}_{\sim i}\mathbf{Y}^{\dagger}_{\sim i} + \gamma\mathbf{I}\right)^{-1}\mathbf{Y}_{\sim i}\mathbf{X}^{\dagger}_{\sim i}$

- If we test K candidates for γ, this requires KT matrix inversions
- For N-dimensional $\mathbf{y}$, the resulting **complexity** $O(KTN^3)$ can be prohibitive

Direct implementation of LOOCV is expensive

Using SCMs and T samples, LOOCV chooses

$$\gamma^* = \arg\min_{\gamma} \frac{1}{T} \sum_{i=1}^{T} ||\mathbf{x}_i - \mathbf{W}^{\dagger}_{\sim i,\gamma} \mathbf{y}_i||^2$$

with $\mathbf{W}_{\sim i,\gamma} = \left(\mathbf{Y}_{\sim i}\mathbf{Y}^{\dagger}_{\sim i} + \gamma \mathbf{I}\right)^{-1} \mathbf{Y}_{\sim i}\mathbf{X}^{\dagger}_{\sim i}$

- If we test K candidates for γ, this requires KT matrix inversions
- For N-dimensional $\mathbf{y}$, the resulting **complexity** $O(KTN^3)$ can be prohibitive

Computationally efficient implementation

- For SCMs, we apply the **Woodbury matrix identity** to simplify the problem to

$$\gamma^* = \arg\min_{\gamma} \left\| \mathbf{X} - \mathbf{X}\left(\mathbf{B}_\gamma - \mathbf{D}_{B_\gamma}\right)\left(\mathbf{I} - \mathbf{D}_{B_\gamma}\right)^{-1} \right\|^2$$

  where

$$\mathbf{B}_\gamma \triangleq \mathbf{Y}^\dagger \left(\mathbf{Y}\mathbf{Y}^\dagger + \gamma\mathbf{I}\right)^{-1} \mathbf{Y}$$

  and $\mathbf{D}_{B_\gamma}$ is a diagonal matrix with diagonal entries of $\mathbf{B}_\gamma$
- This is a **univariate optimization problem**, which can be solved using standard tools
- Computing the **SVD** of **Y** can further **accelerate** the evaluation of the cost function for different candidates γ

Computationally efficient implementation

- For SCMs, we apply the **Woodbury matrix identity** to simplify the problem to

$$\gamma^* = \arg\min_{\gamma} \left\| \mathbf{X} - \mathbf{X} \left(\mathbf{B}_\gamma - \mathbf{D}_{\mathrm{B}_\gamma} \right) \left(\mathbf{I} - \mathbf{D}_{\mathrm{B}_\gamma} \right)^{-1} \right\|^2$$

  where

$$\mathbf{B}_\gamma \triangleq \mathbf{Y}^\dagger \left(\mathbf{Y}\mathbf{Y}^\dagger + \gamma \mathbf{I} \right)^{-1} \mathbf{Y}$$

  and $\mathbf{D}_{\mathrm{B}_\gamma}$ is a diagonal matrix with diagonal entries of $\mathbf{B}_\gamma$
- This is a **univariate optimization problem**, which can be solved using standard tools
- Computing the **SVD** of **Y** can further **accelerate** the evaluation of the cost function for different candidates γ

Different distributions of training and application data

- **Training** and **application data** may have **different distributions** (e.g., orthogonal training)
- In this case, we exploit **spatial correlation** between entries of $\mathbf{y}$
  - E.g., in the MIMO channel model $\mathbf{y} = \mathbf{Hx} + \mathbf{z}$, this correlation is introduced by $\mathbf{H}$
- **Assumption:** Estimates of covariance matrices $(\widehat{\mathbf{C}}_{\mathrm{yx}}, \widehat{\mathbf{C}}_{\mathrm{yy}})$ available
- Perform **spatial LOOCV** on the **application data**. That is, choose γ to **minimize the MSE** of predicting $y_d^{(n)}$ from $\mathbf{y}_d^{(\sim n)}$:

$$\gamma^* = \arg\min_{\gamma} \frac{1}{ND} \sum_{d=1}^{D} \sum_{n=1}^{N} \left| y_d^{(n)} - \widehat{y}_{d,\gamma}^{(n)} \right|^2$$

with length-D application data of dimension N

- **Training** and **application data** may have **different distributions** (e.g., orthogonal training)
- In this case, we exploit **spatial correlation** between entries of $\mathbf{y}$
  - E.g., in the MIMO channel model $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}$, this correlation is introduced by $\mathbf{H}$
- **Assumption:** Estimates of covariance matrices $(\widehat{\mathbf{C}}_{\mathrm{yx}}, \widehat{\mathbf{C}}_{\mathrm{yy}})$ available
- Perform **spatial LOOCV** on the **application data**. That is, choose γ to **minimize the MSE** of predicting $y_d^{(n)}$ from $\mathbf{y}_d^{(\sim n)}$:

$$\gamma^* = \arg\min_{\gamma} \frac{1}{ND} \sum_{d=1}^{D} \sum_{n=1}^{N} \left| y_d^{(n)} - \widehat{y}_{d,\gamma}^{(n)} \right|^2$$

with length-D application data of dimension N

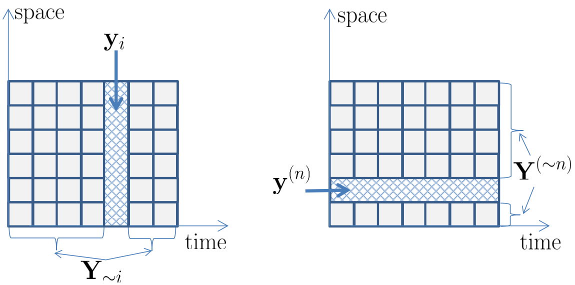
Different distributions of training and application data

- **Training** and **application data** may have **different distributions** (e.g., orthogonal training)
- In this case, we exploit **spatial correlation** between entries of $\mathbf{y}$
  - E.g., in the MIMO channel model $\mathbf{y} = \mathbf{Hx} + \mathbf{z}$, this correlation is introduced by $\mathbf{H}$
- **Assumption:** Estimates of covariance matrices ($\widehat{\mathbf{C}}_{\text{yx}}, \widehat{\mathbf{C}}_{\text{yy}}$) available
- Perform **spatial LOOCV** on the **application data**. That is, choose γ to **minimize the MSE** of predicting $y_d^{(n)}$ from $\mathbf{y}_d^{(\sim n)}$:

$$\gamma^* = \arg\min_{\gamma} \frac{1}{ND} \sum_{d=1}^{D} \sum_{n=1}^{N} \left| y_d^{(n)} - \widehat{y}_{d,\gamma}^{(n)} \right|^2$$

with length-D application data of dimension N

- Perform **spatial LOOCV** on the **application data**:



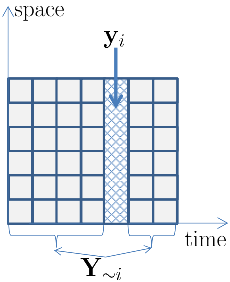
Split the **training** data w.r.t. **time**

Split the **application** data w.r.t. **space**

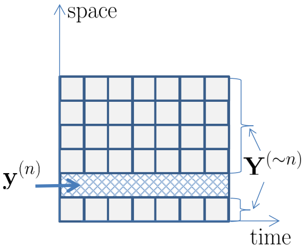
- As before, **computationally efficient** implementations are derived using the Woodbury matrix identity

LOOCV on application data

- Perform **spatial LOOCV** on the **application data**:



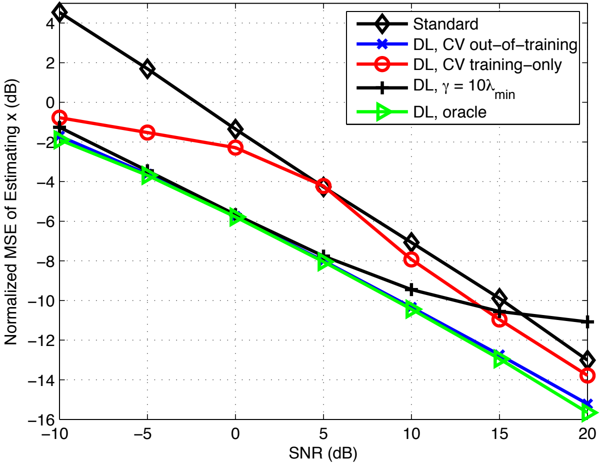
Split the **training** data w.r.t. **time**



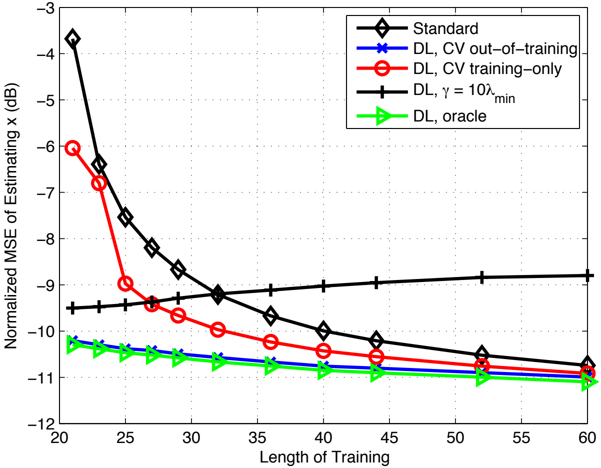
Split the **application** data w.r.t. **space**

- As before, **computationally efficient** implementations are derived using the Woodbury matrix identity

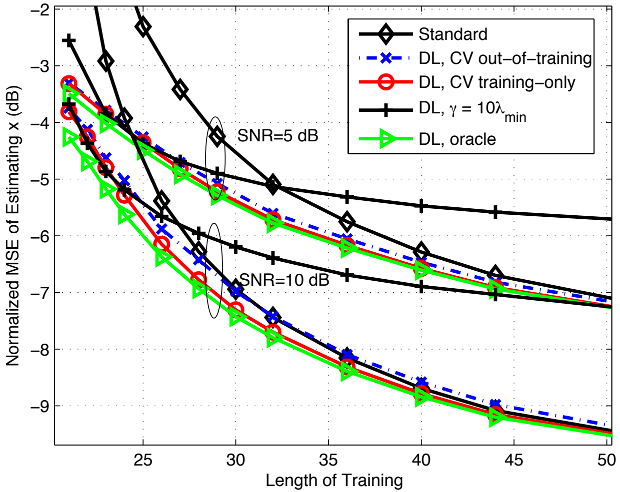
20×20 MIMO channel model: $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}$
Training: orthogonal (DFT), $T = 24$
Application data: Gaussian, $D = 24$



Training: orthogonal (DFT), varying T
Application data: Gaussian, $D = 24$
SNR: 10 dB



Application and training data identically distributed



Conclusions

- LOOCV can be **efficiently** used to choose the **DLF minimizing MSE**
- Can handle both **identically distributed** and **differently distributed** training and application data, by splitting w.r.t. **time** and **space**, respectively
- These ideas can be generalized to the shrinkage estimator

$$\widehat{\mathbf{x}}_{\alpha,\gamma} = \widehat{\mathbf{C}}_{yx}^{\dagger} \left(\alpha \widehat{\mathbf{C}}_{yy} + \gamma \mathbf{I}\right)^{-1} \mathbf{y}$$

- See our forthcoming journal paper:
  J. Tong, P. J. Schreier, Q. Guo, S. Tong, J. Xi, and Y. Yu, "Shrinkage of covariance matrices for linear signal estimation using cross-validation," to appear in *IEEE Trans. Signal Processing*

Conclusions

- LOOCV can be **efficiently** used to choose the **DLF minimizing MSE**
- Can handle both **identically distributed** and **differently distributed** training and application data, by splitting w.r.t. **time** and **space**, respectively
- These ideas can be generalized to the shrinkage estimator

$$\widehat{\mathbf{x}}_{\alpha,\gamma} = \widehat{\mathbf{C}}_{\mathrm{yx}}^{\dagger} \left(\alpha \widehat{\mathbf{C}}_{\mathrm{yy}} + \gamma \mathbf{I}\right)^{-1} \mathbf{y}$$

- See our forthcoming journal paper:
  J. Tong, P. J. Schreier, Q. Guo, S. Tong, J. Xi, and Y. Yu, "Shrinkage of covariance matrices for linear signal estimation using cross-validation," to appear in *IEEE Trans. Signal Processing*