Sparse Signal Recovery in The Presence of Rank-Deficient Noise

Vinuthna Vinjamuri

SPC Lab
Indian Institute of Science, Bangalore, India.

7 February 2015

Outline

System Model

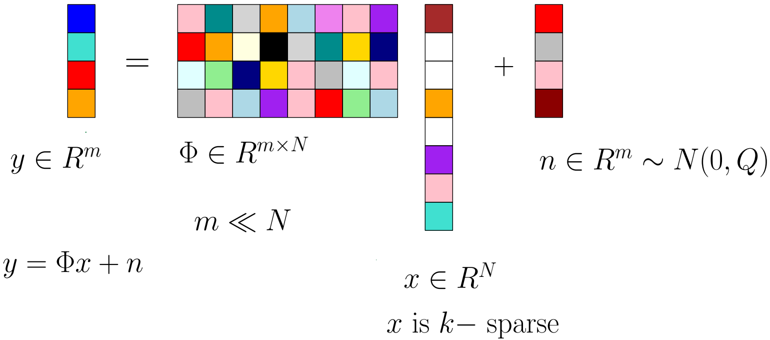


Figure: System Model

Goal: Recover **x** from measurements **y** when, **Q** and $\mathbf{\Phi}$ are known

Solution - EVD

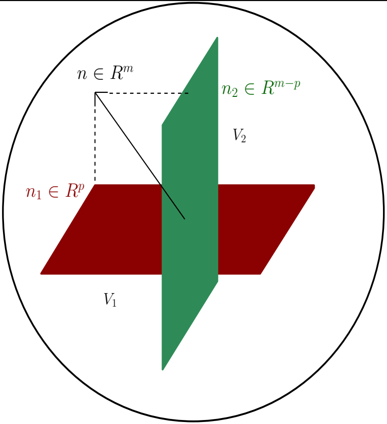


Figure: Projection onto orthogonal subspaces of noise

- EVD of $\mathbf{Q}$ gives $\mathbf{Q} = \mathbf{V}\Lambda\mathbf{V}^\dagger$
- $\mathbf{V}_1 = \mathbf{V}(1:p,:)$
- $\mathbf{V}_2 = \mathbf{V}(p+1:m,:)$
- $\mathbf{V}_1 \perp \mathbf{V}_2$
- $\tilde{\mathbf{y}}_1 = \mathbf{V}_1^\dagger \mathbf{y}$
- $\tilde{\mathbf{y}}_2 = \mathbf{V}_2^\dagger \mathbf{y}$
- $\tilde{\mathbf{y}}_1 = \tilde{\Phi}\mathbf{x} + \mathbf{n}_1$
- $\tilde{\mathbf{y}}_2 = \tilde{\Phi}\mathbf{x}$

Solution

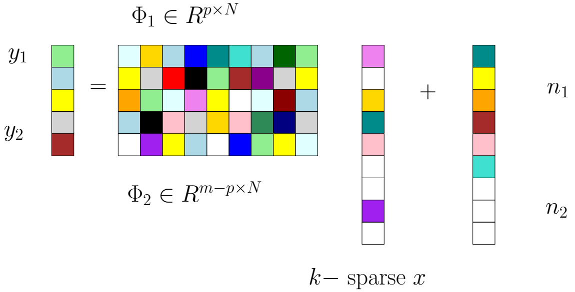


Figure: EVD - CoNo - SBL

Summary-CoNo SBL

Input: $\mathbf{y}$, $\mathbf{\Phi}$ and $\mathbf{Q}$

- Step-1: Initialize $\Gamma \leftarrow \mathbb{I}_N$
- Step-2: Compute $\mathbf{\Sigma}$ and μ using
  $\mathbf{\Sigma} = \Gamma - \Gamma^{\frac{1}{2}}\Theta_1^T\mathbf{B}_{11}\Theta_1\Gamma^{\frac{1}{2}} - \Gamma^{\frac{1}{2}}\Theta_2^{\dagger}\mathbf{U}_1\Theta_2\Gamma^{\frac{1}{2}} +$
  $\Gamma^{\frac{1}{2}}\Theta_1^T\mathbf{\Sigma}_{t1}^{-1}\Theta_1\Theta_2^{\dagger}\mathbf{U}_1\Theta_2\Gamma^{\frac{1}{2}} + \Gamma^{\frac{1}{2}}\Theta_2^{\dagger}\Theta_2\Theta_1^T\mathbf{B}_{11}\Theta_1\Gamma^{\frac{1}{2}}$
  $\boldsymbol{\mu} = \mathbf{\Sigma}\mathbf{\Phi}_1^T\mathbf{D}^{-1}\mathbf{y}_1 + \Gamma^{\frac{1}{2}}\mathbf{U}_2^{\frac{1}{2}}(\Theta_2\mathbf{U}_2^{\frac{1}{2}})^{\dagger}\mathbf{y}_2$
  where $\mathbf{B}_{11} = (\mathbf{\Sigma}_{t1} - \Theta_1\Theta_2^{\dagger}\Theta_2\Theta_1^T)^{-1}\mathbf{\Sigma}_{t1} = (\mathbf{D} + \mathbf{\Phi}_1\Gamma\mathbf{\Phi}_1^T)$,
  $\Theta_1 = \mathbf{\Phi}_1\Gamma^{\frac{1}{2}}$, $\Theta_2 = \mathbf{\Phi}_2\Gamma^{\frac{1}{2}}$, $\mathbf{U}_1 = \mathbf{I}_{m-p} + \Theta_2\Theta_1^T\mathbf{B}_{11}\Theta_1\Theta_2^{\dagger}$ and
  $\mathbf{U}_2 = (\mathbf{I}_N + \Theta_1^T\mathbf{D}^{-1}\Theta_1)^{-1}$
- Step-3: Update Γ using

$$\gamma_i^{t+1} = \Sigma(i,i) + |\mu(i)|^2$$

- Stop:
  $\Gamma^{t+1} - \Gamma^t < 1e-6$ and $t > 600$

Output: μ, Γ

EM-NNL-GAMP

- Recover a non-negative sparse signal $\mathbf{x}$ from noisy linear measurements subject to a linear constraint [1]
- $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, s.t $\mathbf{B}\mathbf{x} = \mathbf{c}$ where $\mathbf{n} \sim \mathcal{N}(0, \chi\mathbb{I})$
- $\mathbf{x}$ is a Bernoulli-Gaussian, $k-$ sparse signal vector in $N-$ dimensional Euclidean space
- Use EM-NNL-GAMP for recovery of $\mathbf{x}$

[1] "An Empirical-Bayes Approach to Recovering Linearly Constrained Non-Negative Sparse Signals", Jeremy Vila and Philip Schniter

EM-NNL-GAMP

- Based on Generalized Approximate Message Passing algorithm (GAMP)
- 2 versions
  - Sum-Product - MMSE estimate
  - Max-Sum - MAP estimate
- Solves

$$\arg\min_{x \geq 0} \frac{1}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||^2 + \lambda||\mathbf{x}||_1 \text{ s.t } \mathbf{B}\mathbf{x} = \mathbf{c}$$

- Tune λ using EM algorithm

EM-NNL-GAMP Cont..

Definitions:

- $\mathbf{z}_i = \mathbf{A}_{i,*}\mathbf{x}$
- $\therefore \mathbf{y}_i = \mathbf{z}_i + \mathbf{n}_i \forall i = 1 \dots m$
- Goal: To find MAP estimate of $\mathbf{x}$ i.e., $\Pr(\mathbf{x}|\mathbf{y};\chi)$
- Baye's Rule: $\Pr(\mathbf{x}|\mathbf{y}) \propto \Pr(\mathbf{x};\gamma)\Pr(\mathbf{y}|\mathbf{x};\chi)$
- $\Pr(\mathbf{x};\gamma)$ as variable node and $\Pr(\mathbf{y}|\mathbf{x};\chi)$ as factor node

EM-NNL-GAMP Cont..

Definitions Cont..

- $prox_g(\hat{\mathbf{v}}; \mu^{\mathbf{v}}) = \arg\min_{x \in \mathbb{R}} g(\mathbf{x}) + \frac{1}{2\mu^{\mathbf{v}}} |\mathbf{x} - \hat{\mathbf{v}}|^2$
- $\hat{\mathbf{r}}$ and $\hat{\mathbf{p}}$ are awgn corrupted versions of true coefficient $\mathbf{x}$ and transformed coefficient $\mathbf{z}$ respectively
- $\Pr(\mathbf{z}|\hat{\mathbf{p}}; \mu^{\mathbf{p}}) = \frac{\Pr(\mathbf{y}_m|\mathbf{z})\mathcal{N}(\mathbf{z};\hat{\mathbf{p}},\mu^p)}{\int_x \Pr(\mathbf{y}_m|\mathbf{z})\mathcal{N}(\mathbf{z};\hat{\mathbf{p}},\mu^p)}$
- $\Pr(\mathbf{x}|\hat{\mathbf{r}}; \mu^{\mathbf{r}}) = \frac{\Pr(\mathbf{x})\mathcal{N}(\mathbf{x};\hat{\mathbf{r}},\mu^r)}{\int_x \Pr(\mathbf{x})\mathcal{N}(\mathbf{x};\hat{\mathbf{r}},\mu^r)}$

EM-NNL-GAMP Cont..

Input: $\Pr_{\mathbf{x}_n}(\mathbf{x})$, $\Pr_{\mathbf{y}_m|\mathbf{z}_m}(\mathbf{y}|\mathbf{z})$, $\mathbf{A}_{mn}$, T_{max}, $\epsilon_{gamp} > 0$

- Step-1: Initialize
  - $\forall n : \hat{\mathbf{x}_n}(1) = \int_{\mathbf{x}} \mathbf{x} \Pr_{\mathbf{x}_n}(\mathbf{x})$
  - $\forall n : \mu_n^{\mathbf{x}}(1) = \int_{\mathbf{x}} |\mathbf{x} - \hat{\mathbf{x}}_n(1)|^2 \Pr_{\mathbf{x}_n}(\mathbf{x})$
  - $\forall m : \hat{\mathbf{s}}_m(0) = 0$
- Step-2:
  $\forall m$:
  - $\mu_m^{\mathbf{p}}(t) = \sum_{n=1}^{N} |\mathbf{A}_{mn}|^2 \mu_n^{\mathbf{x}}(t)$
  - $\hat{\mathbf{p}}_m(t) = \sum_{n=1}^{N} \mathbf{A}_{mn}\hat{\mathbf{x}}_n(t) - \mu_m^p(t)\hat{\mathbf{s}}_m(t-1)$
  - $\hat{\mathbf{z}}_m(t) = prox_{-\log \Pr_{\mathbf{y}_m|\mathbf{z}_m}}(\hat{\mathbf{p}}_m(t); \mu_m^p(t)$
  - $\mu_m^{\mathbf{z}}(t) = \mu_m^p(t) prox'_{-\log \Pr_{\mathbf{y}_m|\mathbf{z}_m}}(\hat{\mathbf{p}}_m(t); \mu_m^p(t)$
  - $\mu_m^{\mathbf{s}}(t) = (1 - \mu_m^{\mathbf{z}}(t)/\mu_m^{\mathbf{p}}(t))/\mu_m^{\mathbf{p}}(t)$
  - $\hat{\mathbf{s}_m}(t) = (\hat{\mathbf{z}}_m(t) - \hat{\mathbf{p}}_m(t))/\mu_m^{\mathbf{p}}(t)$

EM-NNL-GAMP Cont..

- Step-3:$\forall n$:
  - $\mu^r_n(t) = (\sum_{m=1}^{M} |\mathbf{A}_{mn}|^2 \mu^{\mathbf{s}}_m(t))^{-1}$
  - $\hat{\mathbf{r}}_n(t) = \hat{\mathbf{x}}_n(t) + \mu^r_n(t) \sum_{m=1}^{M} \mathbf{A}^{\star}_{mn} v\hat{e}cs_m(t)$
- Step-4:Update
  $\hat{\mathbf{x}_n}(t+1) = prox_{-\log \mathrm{Pr}_{\mathbf{x}_n}}(\hat{\mathbf{r}_n}(t); \mu^{\mathbf{r}}_n(t))$
  $\mu^{\mathbf{x}}_n(t+1) = \mu^{\mathbf{r}}_n(t) prox'_{-\log \mathrm{Pr}_{\mathbf{x}_n}}(\hat{\mathbf{r}}_n(t); \mu^{\mathbf{r}}_n(t))$
- Stop: $\sum_{n=1}^{N} |\hat{\mathbf{x}}_n(t+1) - \hat{\mathbf{x}}_n(t)|^2 < \epsilon_{gamp} \sum_{n=1}^{N} |\hat{\mathbf{x}}_n(t)|^2$

Output:$\forall m, n$: $\hat{\mathbf{z}}_m(t)$, $\mu^{\mathbf{z}}_m(t)$, $\hat{\mathbf{r}}_n(t)$, $\mu^{\mathbf{r}}_n(t)$, $\hat{\mathbf{x}}_n(t+1)$, $\mu^{\mathbf{x}}_n(t+1)$

Numerical Results

- $N = 100$
- $k = 10$
- $SNR = \begin{pmatrix} 10 & 15 & 20 & 25 & 30 & 35 & 40 \end{pmatrix}$
- $m = 60$
- $rank = 1 : 2 : m$, Typical $rank = m/2$
- Number of realizations $= 600$

EM-NNL-GAMP *vs.* CoNo-SBL - SNR

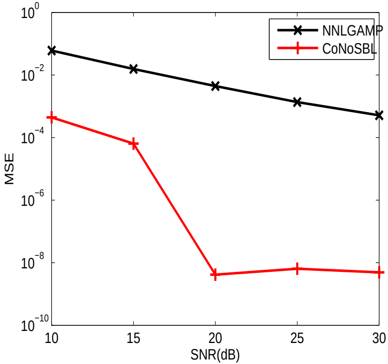


Figure: MSE vs. SNR

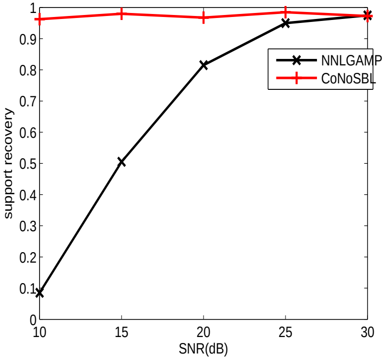


Figure: Support Recovery vs. SNR

EM-NNL-GAMP *vs.* CoNo-SBL - Rank

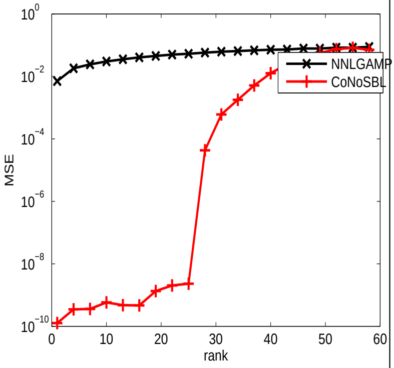


Figure: MSE vs. Rank

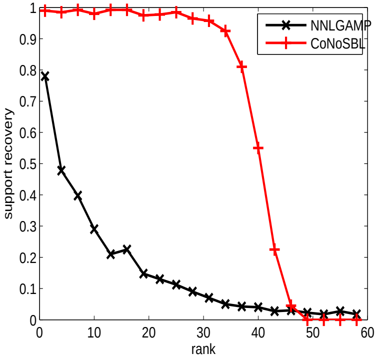


Figure: Support Recovery vs. Rank

MMV Setup

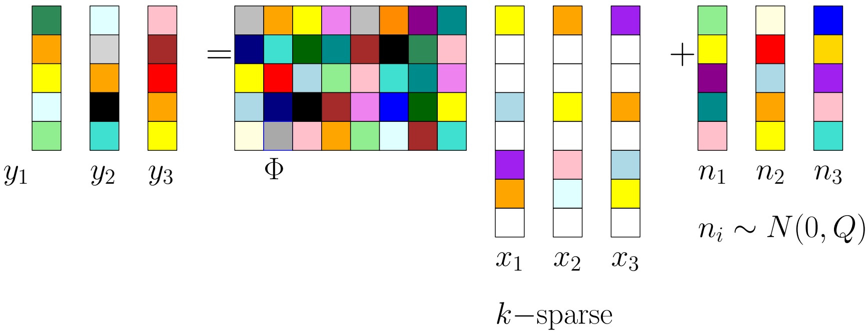


Figure: Multiple Measurement Vector Setup

- Definitions: Consider $L-$ Measurement vectors
- $\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_L \end{pmatrix}$, $\mathbf{Y} = \begin{pmatrix} \mathbf{y}_1 & \dots & \mathbf{y}_L \end{pmatrix}$ and $\mathbf{N} = \begin{pmatrix} \mathbf{n}_1 & \dots & \mathbf{n}_L \end{pmatrix}$
- Assume $\mathbf{x}_i \sim \mathcal{N}(0, \Gamma\mathbb{I}) \forall i = 1 \dots L$, where $\Gamma = diag(\gamma_i)$
- when $\mathbf{Q} = \sigma^2\mathbb{I}$, the update equations for γ, the posterior parameters on $\mathbf{X}$, $\boldsymbol{\Sigma}$ and $\mathcal{M}$ are given by
  - $\boldsymbol{\Sigma} = \Gamma - \Gamma\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Phi}\Gamma$, where $\boldsymbol{\Sigma}_t = (\sigma^2\mathbb{I} + \boldsymbol{\Phi}\Gamma\boldsymbol{\Phi}^T)$
  - $\mathcal{M} = \Gamma\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_t^{-1}\mathbf{Y}$
  - Update: $\boldsymbol{\gamma}_i = \frac{1}{L}\mu_i^2 + \boldsymbol{\Sigma}_{ii}$

- When $\mathbf{Q}$ is Rank-deficient, project all the measurement vectors onto orthogonal subspaces like CoNo-SBL
- $\begin{pmatrix} \mathbf{y}_{11} & \dots & \mathbf{y}_{L1} \end{pmatrix} = \mathbf{\Phi}_1 \begin{pmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_L \end{pmatrix} + \begin{pmatrix} \mathbf{n}_{11} & \dots & \mathbf{n}_{L1} \end{pmatrix}$
  $\begin{pmatrix} \mathbf{y}_{12} & \dots & \mathbf{y}_{L2} \end{pmatrix} = \mathbf{\Phi}_2 \begin{pmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_L \end{pmatrix}$
- Goal: Recover $\mathbf{X}$ using this set-up

M-CoNo-SBL

- Let $\tilde{\mathbf{y}}_i = \begin{pmatrix} \mathbf{y}_{i1} \\ \mathbf{y}_{i2} \end{pmatrix}$, $\tilde{\mathbf{n}}_i = \begin{pmatrix} \mathbf{n}_{i1} \\ \mathbf{n}_{i2} \end{pmatrix} \forall i = 1 \dots L$
- $\therefore \begin{pmatrix} \tilde{\mathbf{y}}_1 & \dots & \tilde{\mathbf{y}}_L \end{pmatrix} = \tilde{\mathbf{\Phi}} \mathbf{X} + \begin{pmatrix} \tilde{\mathbf{n}}_1 & \dots & \tilde{\mathbf{n}}_L \end{pmatrix}$, $\mathbb{E}(\tilde{\mathbf{n}}_i \tilde{\mathbf{n}}_i{}^H) = \mathbf{\Lambda}$, where $\mathbf{\Lambda} = \begin{pmatrix} \mathbf{D} & 0 \\ 0 & \sigma_2^2 \mathbb{I} \end{pmatrix}$, $\sigma_2^2 \to 0$
- The update equation for $\mathbf{\Sigma}$ remains the same. However, the update equation for $\mathcal{M} = \Gamma^{\frac{1}{2}} \mathbf{U}_3^T \Theta_1^T \mathbf{B}_{11} \tilde{\mathbf{Y}}_1 + \Gamma^{\frac{1}{2}} \mathbf{U}_4 \Theta_2^{\dagger} \mathbf{U}_1 \tilde{\mathbf{Y}}_2$

Summary of M-CoNo-SBL

Input: $\mathbf{Y}$, $\boldsymbol{\Phi}$ and $\mathbf{Q}$

- Step-1: Initialize $\Gamma \leftarrow \mathbb{I}$
- Step-2:
  - $\boldsymbol{\Sigma} = \Gamma - \Gamma^{\frac{1}{2}}\Theta_1^T\mathbf{B}_{11}\Theta_1\Gamma^{\frac{1}{2}} - \Gamma^{\frac{1}{2}}\Theta_2^{\dagger}\mathbf{U}_1\Theta_2\Gamma^{\frac{1}{2}} + \Gamma^{\frac{1}{2}}\Theta_1^T\boldsymbol{\Sigma}_{t1}^{-1}\Theta_1\Theta_2^{\dagger}\mathbf{U}_1\Theta_2\Gamma^{\frac{1}{2}} + \Gamma^{\frac{1}{2}}\Theta_2^{\dagger}\Theta_2\Theta_1^T\mathbf{B}_{11}\Theta_1\Gamma^{\frac{1}{2}}$
  - $\mathcal{M} = \Gamma^{\frac{1}{2}}\mathbf{U}_3^T\Theta_1^T\mathbf{B}_{11}\tilde{\mathbf{Y}}_1 + \Gamma^{\frac{1}{2}}\mathbf{U}_4\Theta_2^{\dagger}\mathbf{U}_1\tilde{\mathbf{Y}}_2$

  where, $\mathbf{B}_{11} = (\boldsymbol{\Sigma}_{t1} - \Theta_1\Theta_2^{\dagger}\Theta_2\Theta_1^T)^{-1}$, $\boldsymbol{\Sigma}_{t1} = (\mathbf{D} + \boldsymbol{\Phi}_1\Gamma\boldsymbol{\Phi}_1^T)$, $\Theta_1 = \boldsymbol{\Phi}_1\Gamma^{\frac{1}{2}}$, $\Theta_2 = \boldsymbol{\Phi}_2\Gamma^{\frac{1}{2}}$, and $\mathbf{U}_1 = \mathbf{I}_{m-p} + \Theta_2\Theta_1^T\mathbf{B}_{11}\Theta_1\Theta_2^{\dagger}$, $\mathbf{U}_3 = \left(\mathbf{I}_N - \Theta_2^{\dagger}\Theta_2\right)$, $\mathbf{U}_4 = (\mathbf{I}_N - \Theta_1^T\boldsymbol{\Sigma}_{t1}^{-1}\Theta_1)$, $\tilde{\mathbf{Y}}_1 = \mathbf{V}_1^H\mathbf{Y}$ and $\tilde{\mathbf{Y}}_2 = \mathbf{V}_2^H\mathbf{Y}$
- Step-3: Update $\gamma_i = \frac{1}{L}\mu_i^2 + \boldsymbol{\Sigma}_{ii}$
- Stop: $\Gamma^{t+1} - \Gamma^t < 1e-6$ and $t \geq 500$

Output: μ, Γ

Numerical Results

- M-SBL vs. SBL
  - $N = 100$
  - $k = 10$
  - $SNR = 10\text{dB}$
  - $L = 1 : 40$
- M-CoNo-SBL vs. CoNo-SBL
  - $N = 100$
  - $k = 10$
  - $SNR = 10\text{dB}$
  - $L = 1 : 40$

MMV vs. SMV - Full Rank **Q**

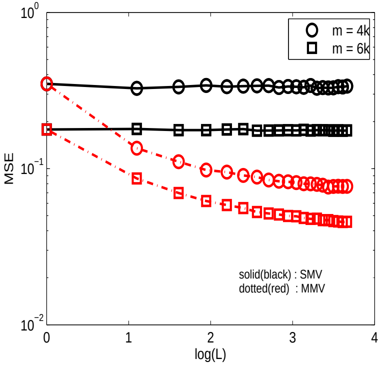


Figure: MSE vs. Rank

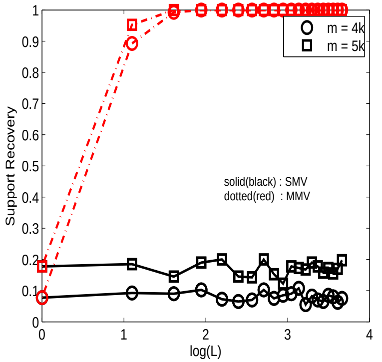


Figure: Support Recovery vs. Rank

MMV vs. SMV - Rank-Deficient Q

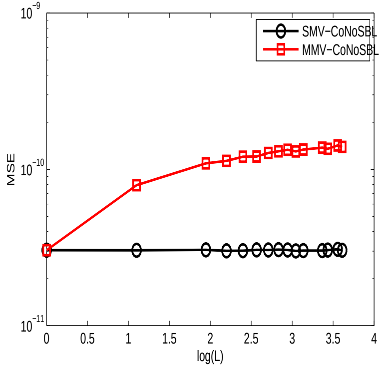


Figure: MSE vs. L

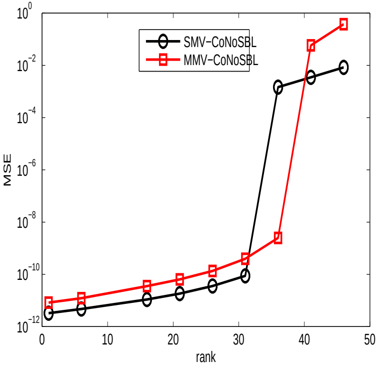


Figure: MSE vs. Rank

Complexities of CoNo-SBL and M-CoNo-SBL

Parameter	SBL	CoNo-SBL	M-SBL	M-CoNo-SBL
$\boldsymbol{\Sigma}$	$\mathcal{O}(N^3)$	$\mathcal{O}((m-p)N(m+N-p))$	$\mathcal{O}(N^3)$	$\mathcal{O}((m-p)N(m+N-p))$
μ	$\mathcal{O}(N^2)$	$\mathcal{O}(mN^2)$	$\mathcal{O}(NmL)$	$\mathcal{O}(N^2(m-p))$
Γ	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$
Eg	10^6	10^6	10^6	10^6

Table: Per Iteration Complexity Comparison of SBL algorithms

Unknown Noise Variance

- Consider SMV set-up and the simplest case when $\mathbf{D} = \sigma^2 \mathbb{I}$
- Joint-estimation of σ^2 and γ using EM algorithm
- Let $\psi = [\ \Gamma^T, \quad \sigma^2\ ]$
- $$\hat{\psi}_{ML} = \underset{\gamma \in \mathbb{R}_+^{N \times 1}, \sigma^2}{\arg\max}\ p(\mathbf{y}; \gamma, \sigma^2)$$

$$\Rightarrow \sigma^{2^{t+1}} = \frac{||\mathbf{y}_1 - \mathbf{\Phi}\mu'||^2 + \sigma^{2^t} \sum_{i=[N]}(1 - {\gamma_i^t}^{-1} \Sigma'_{w_{ii})}}{m}$$

and $\gamma_i = |\mu_i|^2 + \mathbf{\Sigma}_{w_{ii}}$

Summary - UA-CoNo-SBL

Input: $\mathbf{y}$, $\boldsymbol{\Phi}$

- Step-1: Initialize $\Gamma \leftarrow \mathbb{I}$, $\sigma^2 \leftarrow 1$
- Step-2: Compute $\boldsymbol{\Sigma}$ and Γ using
  - $\boldsymbol{\Sigma} = \Gamma - \Gamma^{\frac{1}{2}}\Theta_1^T\mathbf{B}_{11}\Theta_1\Gamma^{\frac{1}{2}} - \Gamma^{\frac{1}{2}}\Theta_2^{\dagger}\mathbf{U}_1\Theta_2\Gamma^{\frac{1}{2}} + \Gamma^{\frac{1}{2}}\Theta_1^T\boldsymbol{\Sigma}_{t1}^{-1}\Theta_1\Theta_2^{\dagger}\mathbf{U}_1\Theta_2\Gamma^{\frac{1}{2}} + \Gamma^{\frac{1}{2}}\Theta_2^{\dagger}\Theta_2\Theta_1^T\mathbf{B}_{11}\Theta_1\Gamma^{\frac{1}{2}}$
  - $\boldsymbol{\mu} = \boldsymbol{\Sigma}\boldsymbol{\Phi}_1^T\mathbf{D}^{-1}\mathbf{y}_1 + \Gamma^{\frac{1}{2}}\mathbf{U}_2^{\frac{1}{2}}(\Theta_2\mathbf{U}_2^{\frac{1}{2}})^{\dagger}\mathbf{y}_2$

  where, $\mathbf{B}_{11} = (\boldsymbol{\Sigma}_{t1} - \Theta_1\Theta_2^{\dagger}\Theta_2\Theta_1^T)^{-1}$, $\boldsymbol{\Sigma}_{t1} = (\mathbf{D} + \boldsymbol{\Phi}_1\Gamma\boldsymbol{\Phi}_1^T)$, $\Theta_1 = \boldsymbol{\Phi}_1\Gamma^{\frac{1}{2}}$, $\Theta_2 = \boldsymbol{\Phi}_2\Gamma^{\frac{1}{2}}$, $\mathbf{U}_1 = \mathbf{I}_{m-p} + \Theta_2\Theta_1^T\mathbf{B}_{11}\Theta_1\Theta_2^{\dagger}$ and $\mathbf{U}_2 = (\mathbf{I}_N + \Theta_1^T\mathbf{D}^{-1}\Theta_1)^{-1}$
- Step-3: Update
  - $\gamma_i = |\mu_i|^2 + \boldsymbol{\Sigma}_{ii}$
  - $\hat{\sigma}^{2^{t+1}} = \frac{||\mathbf{y}_1 - \boldsymbol{\Phi}\mu'||^2 + \hat{\sigma}^{2^t}\sum_{i=[N]}(1 - \gamma_i^{t^{-1}}\Sigma'_{w_{ii}})}{m}$
- Stop: $\Gamma^{t+1} - \Gamma^t < 1e-6$, $\hat{\sigma^2}^{t+1} - \hat{\sigma^2}^{t} < 1e-4$ and $t > 1000$

Output: μ, Γ, $\hat{\sigma^2}$

Numerical Results

- $N = 100$
- $K = 10$
- $m = 50$
- $p = m/2$
- $SNR = 10\text{dB}$

Support Recovery vs. Rank - UA-CoNo-SBL

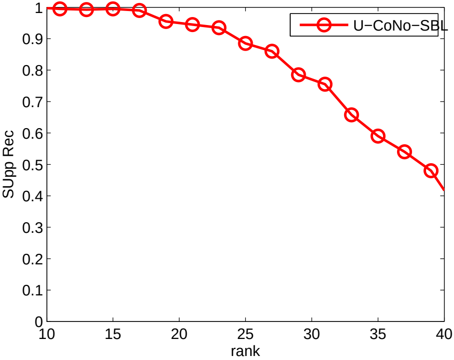
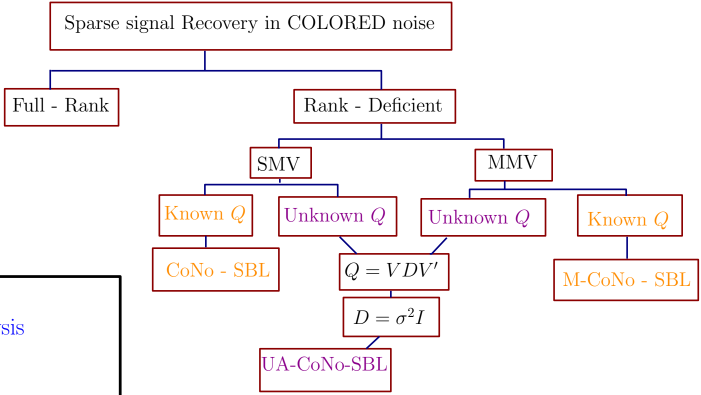


Figure: Support Recovery of the proposed U-CoNo-SBL technique a function of rank p



Analysis

CRB's

Time Complexity

Figure: Summary of work

Future Work

- Overcome Identifiability problem in estimating σ^2 in SBL
- Estimate $\mathbf{Q}$ using MMV-model
- Analyse Time Variant Nature in this set up
- Extend to Robust-PCA and Matrix Completion problems

Thank You