Sparsity-Aware Bayesian Inference and Its Applications

Chandra R. Murthy
Dept. of ECE
Indian Institute of Science

cmurthy@iisc.ac.in

Outline

- Sparse Bayesian learning
- Joint-sparse recovery, guarantees, extns.
- ~~Application to wireless communications~~
- Role of sparsity in linear dynamical systems
- Bayesian inference via deep unfolding

Showing **1-25** of **2,831** results for **sparse bayesian** ×

☐ Conferences (1,737) ☐ Journals (1,034) ☐ Early Access Articles (39) ☐ Magazines (16)

☐ Books (5)

Sun Jun 4, 8:30am-12:00pm

Rethinking Sparsity-Aware Bayesian Learning for Signal Processing and Machine Learning

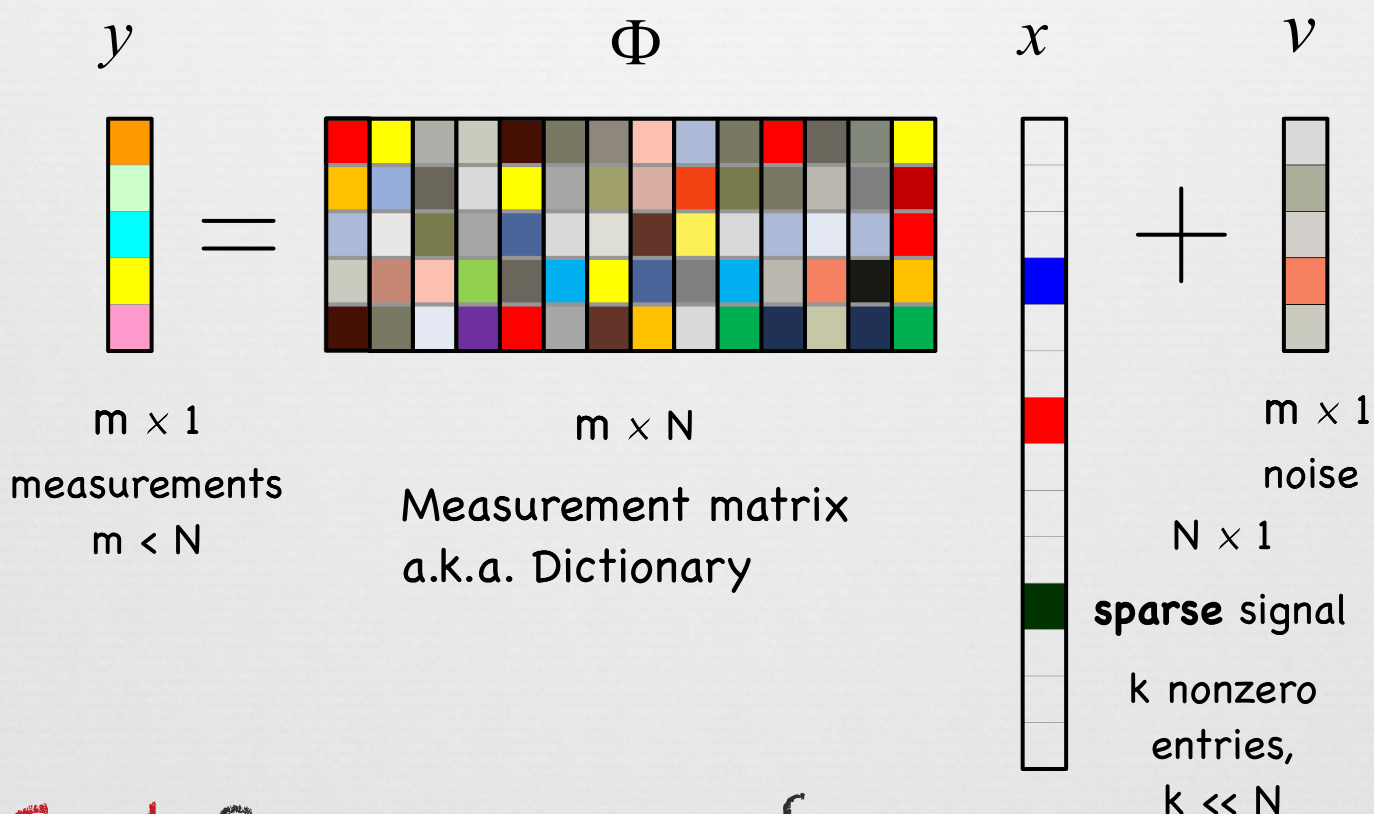
Presenters

Feng Yin (The Chinese University of Hong Kong, Shenzhen), Lei Cheng (Zhejiang University), Sergios Theodoridis (National and Kapodistrian University of Athens)

Part 1: Sparse Bayesian Learning

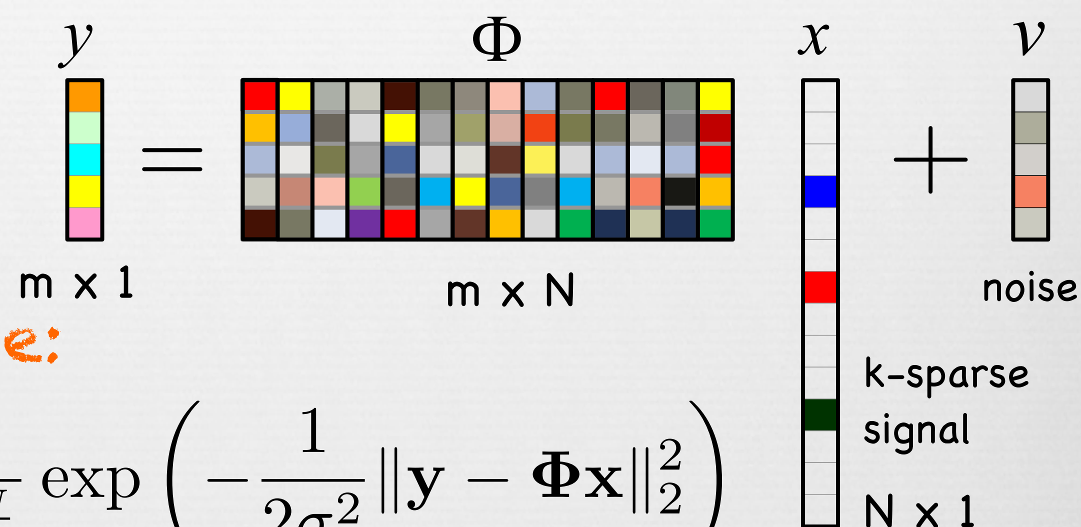
Use a continuum of priors and pick the best one!

Sparse Signal Recovery



- Goal: Recover sparse x from y
- Challenge: m < N, usually m << N

Sparse Bayesian Learning



- **Gaussian noise:**

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{\Phi x}\|_2^2\right)$$

- **Parameterized Gaussian prior:**

$$p(x_i;\gamma_i) = \frac{1}{\sqrt{2\pi\gamma_i}} \exp\left(-\frac{x_i^2}{2\gamma_i}\right), \; \gamma_i \geq 0$$

The EM-SBL Algorithm

1. Initialize $\Gamma = I$
2. Compute

$$\Sigma = \left(\sigma^{-2}\Phi^T\Phi + \left(\Gamma^{(t)}\right)^{-1}\right)^{-1}$$

$$\mu = \sigma^{-2}\Sigma\Phi^T\mathbf{y}$$

Compute posterior distribution

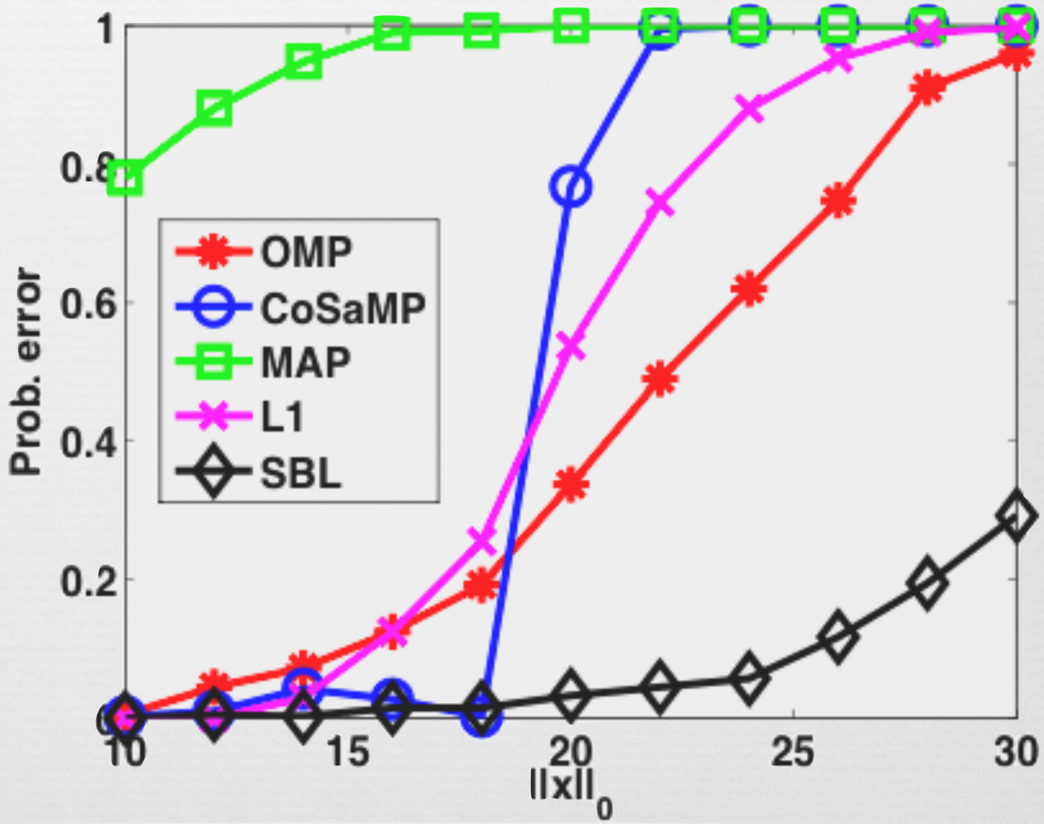
3. Update $\Gamma^{(t+1)} = \mathrm{diag}\left(\mu_i^2 + \Sigma_{ii}\right)$

Update hyperparameters γ_i via type-II ML

4. Repeat steps 2 and 3
5. Output μ after convergence

Empirical Example

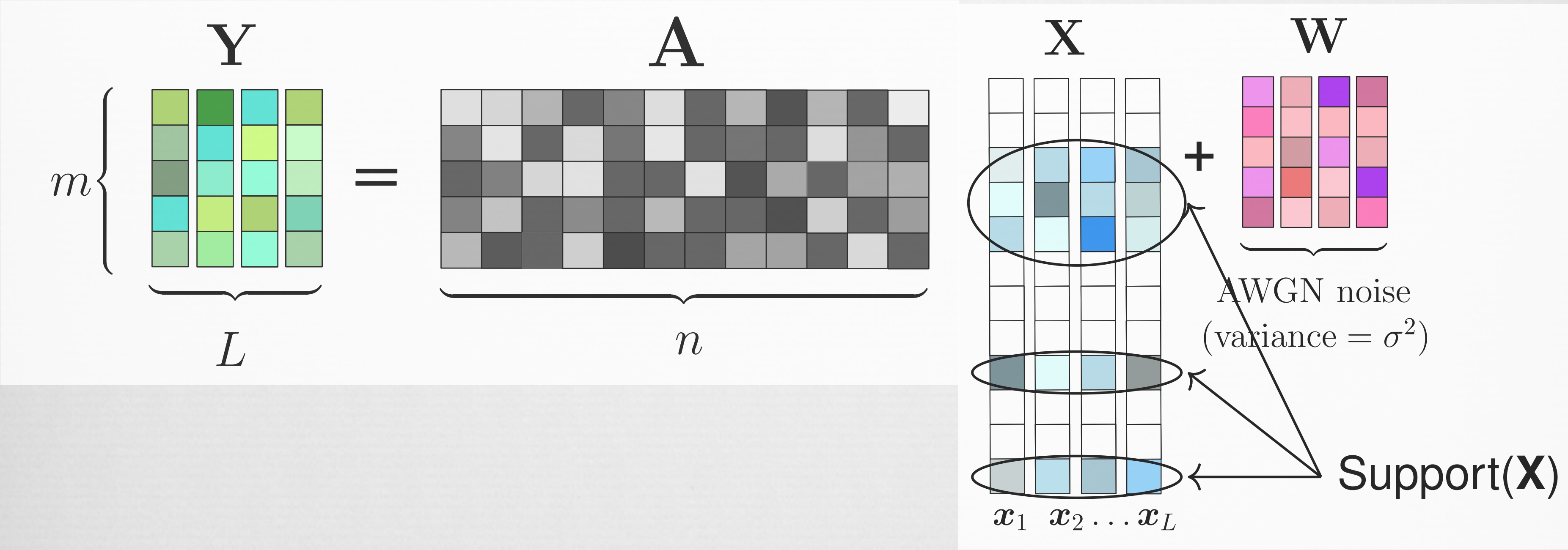
- Generate random 50 x 100 matrix A
- Generate sparse vector x_0
- Compute $y = Ax_0$
- Solve for x_0, average over 1000 trials
- Repeat for different sparsity values



Highly scaled nonzero entries

Joint Sparse Recovery

Observation model

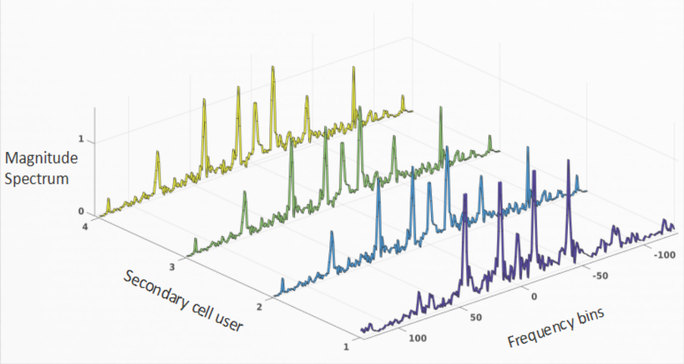


Let k = number of nonzero rows in X.

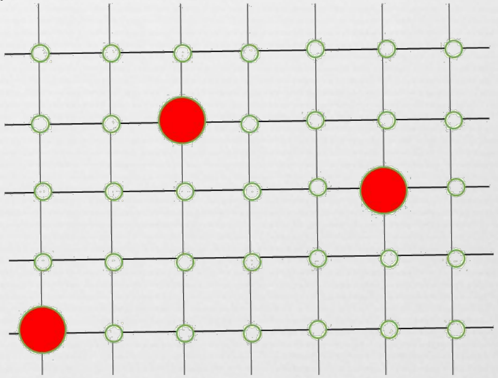
Want to recover X or support(X) from the Multiple Measurement Vectors (MMVs) Y

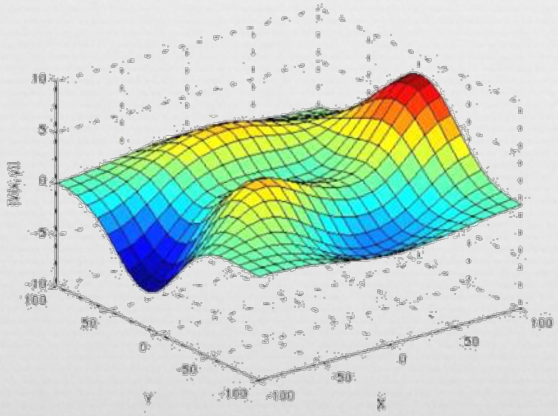
Support Recovery is also Important

Wideband spectrum sensing



Sparse event localization





Subspace filtering by projecting to common signal subspace

MSBL-Sparse Bayesian Learning using MMVs

- Observation model: $\mathbf{Y} = \Phi\mathbf{X} + \mathbf{W}$
- Correlation-aware prior: $\mathbf{x}_j \overset{i.i.d.}{\sim} \mathcal{N}(0, \Gamma),\ \Gamma = \mathrm{diag}(\gamma)$
  - Common Γ enforces same support in columns of X.
  - Gaussian MMVs:

$$\mathbf{y}_j \sim \mathcal{N}(0, \sigma^2\mathbf{I} + \Phi\Gamma\Phi^T)$$

- M-SBL algorithm: $\hat{\gamma} = \underset{\gamma \in \mathbb{R}_+^n}{\mathrm{argmax}} \log p(\mathbf{Y}; \gamma)$
  - Non-convex objective
  - Solved via Expectation Maximization (EM)
  - Estimated support = support($\hat{\gamma}$).

The EM-MSBL Algo

- E Step:

$$\Sigma_j^{k+1} = \Gamma^k - \Gamma^k \Phi_j^T \left(\sigma_j^2 \mathbf{I}_M + \Phi_j \Gamma^k \Phi_j^T\right)^{-1} \Phi_j \Gamma^k$$

$$\mu_j^{k+1} = \sigma_j^{-2} \Sigma_j^{k+1} \Phi_j^T \mathbf{y}_j$$

- M Step:

$$\gamma^{k+1}(i) = \frac{1}{L} \sum_{j=1}^{L} \mu_j^{k+1}(i)^2 + \Sigma_j^{k+1}(i,i)$$

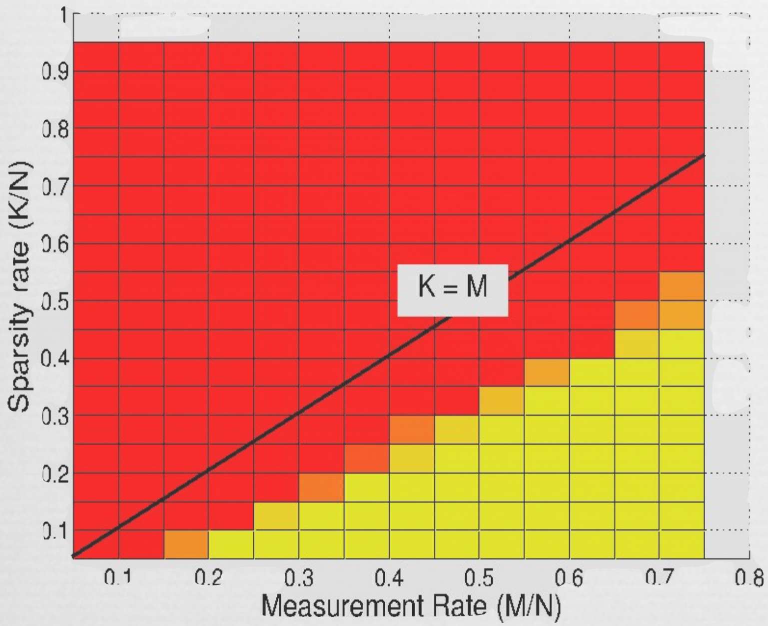
  - Average of the individual estimates of γ_i across measurements

Performance of MSBL

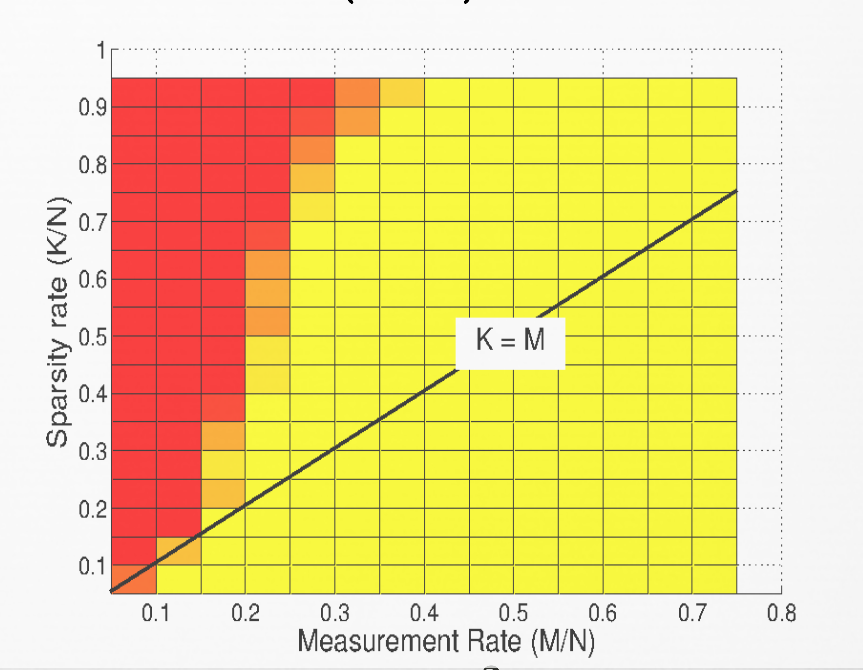
Support recovery phase transition

n=200,
L=400,
SNR=20 dB

Simultaneous Orthogonal Matching Pursuit (SOMP)



Sparse Bayesian Learning (M-SBL)



- Recoverable support size k grows as $O(m^2)$!
- Using correlation-structure aware priors is helpful

Part 2: Performance Guarantees

Sufficient conditions for support recovery by M-SBL

Sufficient Conditions for Support Recovery in SBL

- Single measurement vector ($L = 1$)
- Noiseless observations
- Result: SBL correctly recovers the support for all $1 \leq k < \text{spark}(\Phi) - 1$
  - spark: smallest num. of lin. dep. cols
  - Usually, in CS, $\text{spark}(\Phi) = m + 1$
- For l1 recovery, $1 \leq k \leq O(m / \log N)$

[Chen and Huo, 06]

First Support Recovery Guarantee for M-SBL

- Common sensing matrix $\Phi \overset{i.i.d.}{\sim} \mathcal{N}(0, 1/m)$
- M-SBL recovers the true support with vanishing probability of error, provided

$$m = \Theta(k \log N) \text{ and } L = \Omega\left(\frac{N}{k}\log N + N \log k + N \log\log N\right)$$

- Or

$$m = \Theta(\sqrt{k} \log N) \text{ and } L = \Omega\left(\frac{N}{\sqrt{k}}\log N + N\sqrt{k} \log k + N\sqrt{k} \log\log N\right)$$

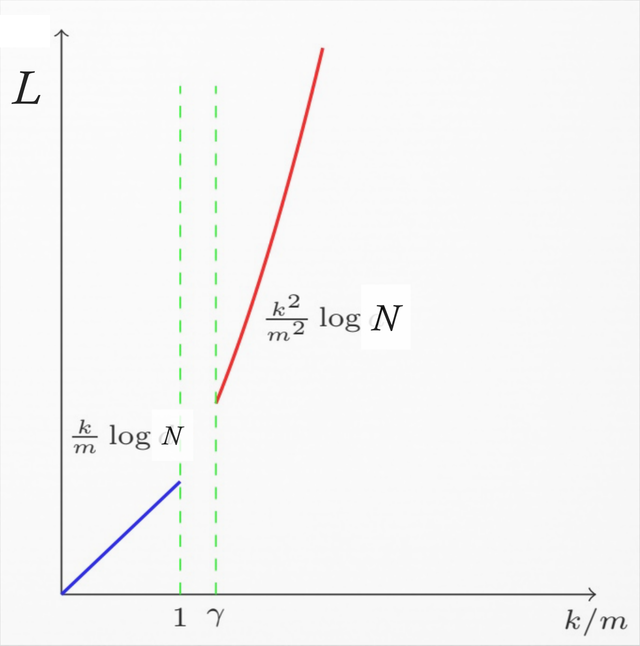
S. Khanna and M., T-IT Nov. 2022

Second Support Recovery Guarantee for M-SBL

- Sensing matrix $\Phi_i \overset{i.i.d.}{\sim} \mathcal{N}(0, 1/m), i = 1, 2, \ldots, L$
- For $(\log k)^2 \leq m < k/2$ and $1 \leq k \leq N-1$, the sample complexity for successful support recovery is

$$L = \Theta\left(\frac{k^2}{m^2} \log k(N-k)\right)$$

- In fact, this bound can be achieved using a very SIMPLE algorithm!



L. Ramesh, M., and H. Tyagi, T-IT Dec. 2021

Simple Algorithm

- Observations $\mathbf{y}_i = \Phi_i \mathbf{x}_i + \mathbf{w}_i$, $i = 1, 2, \ldots, L$
- Compute the diagonal entries of the "pseudo" covariance matrix

$$\frac{1}{L} \sum_{j=1}^{L} \Phi_j^T \mathbf{y}_j \mathbf{y}_j^T \Phi_j$$

- Declare the indices corresp. top k diagonal entries as the support!

Part 3: New Algorithms

Covariance matching is the key!

New Interpretation of M-SBL Cost Function

- M-SBL cost:

$$-\log p(\mathbf{Y};\gamma) = -\sum_{j=1}^{L} \log \mathcal{N}\left(\mathbf{y}_j; 0, \sigma^2 \mathbf{I}_m + \mathbf{\Phi}\Gamma\mathbf{\Phi}^T\right)$$

$$\propto \log|\sigma^2 \mathbf{I}_m + \mathbf{\Phi}\Gamma\mathbf{\Phi}^T| + \mathrm{tr}\left(\left(\sigma^2 \mathbf{I}_m + \mathbf{\Phi}\Gamma\mathbf{\Phi}^T\right)^{-1}\left(\frac{1}{L}\mathbf{Y}\mathbf{Y}^T\right)\right)$$

$$\propto \mathcal{D}^{\mathrm{Bregman}}_{-\log\det}\left(\frac{1}{L}\mathbf{Y}\mathbf{Y}^T, \sigma^2 \mathbf{I}_m + \mathbf{\Phi}\Gamma\mathbf{\Phi}^T\right) + \mathrm{const.\ terms}$$

- Motivates covariance matching based approaches to sparse recovery

Covariance Matching Framework

- Observation Model:

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{X} + \mathbf{W}$$

Correlation-aware Gaussian prior

$$\mathbf{x}_j \sim \mathcal{N}(0, \mathrm{diag}(\gamma))$$

$$\mathbf{y}_j \sim \mathcal{N}(0, \sigma^2\mathbf{I}_m + \mathbf{\Phi}\mathbf{\Gamma}\mathbf{\Phi}^T)$$

- Principle:

Parametrized covariance matrix

$$\hat{\gamma} = \arg\min_{\gamma \in \mathbb{R}_+^n} \mathbf{dist}\left(\frac{1}{L}\mathbf{Y}\mathbf{Y}^T, \sigma^2\mathbf{I}_m + \mathbf{\Phi}\mathbf{\Gamma}\mathbf{\Phi}^T\right)$$

Empirical covariance matrix

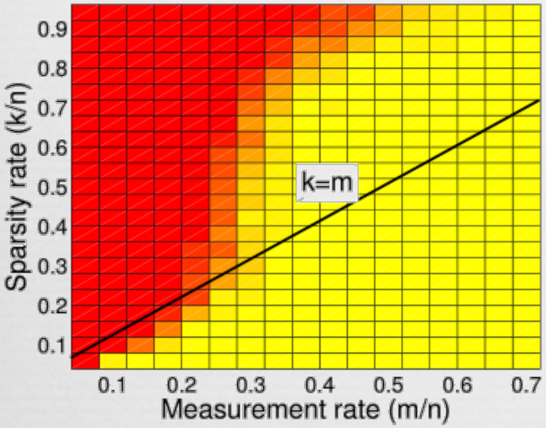
Support estimate = Support($\hat{\gamma}$)

Algorithms

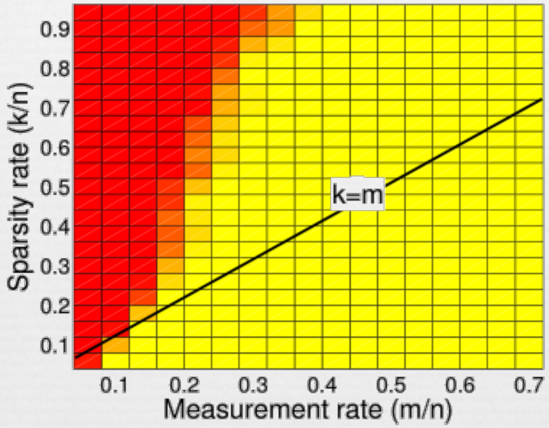
- Approach 1
  - Distance = Frobenius norm
  - Algorithm = CoLASSO [Pal, Vaidyanathan, 15]
- Approach 2
  - Distance = Log-Det Bregman Divergence
  - Algorithm = M-SBL [Wipf & Rao, 07]
- Approach 3
  - Distance = $\alpha-$Rényi Divergence
  - Algorithm = Rényi Divergence based Covariance Matching Pursuit (RD-CMP) [Khanna & M., 17]

Performance

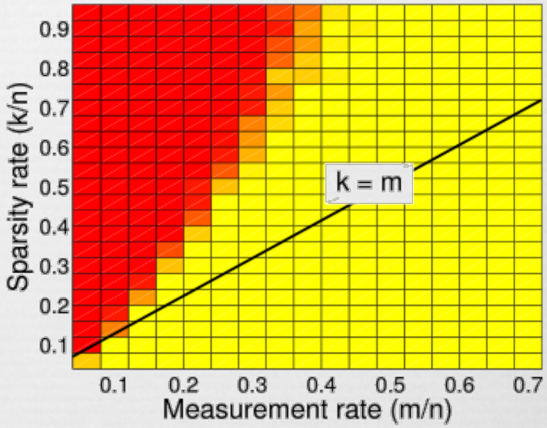
Co-LASSO



MSBL

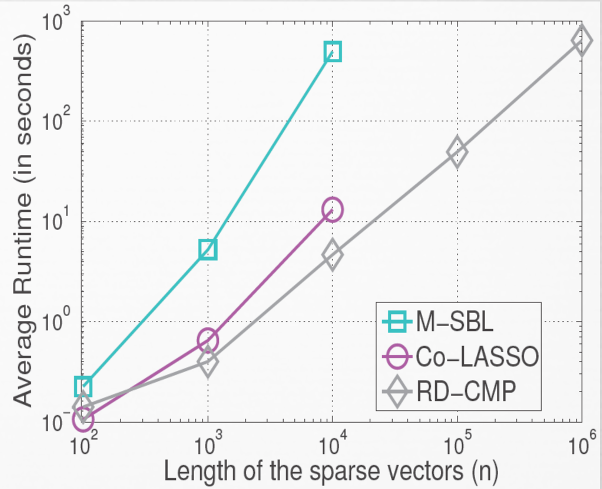


RD-CMP



SNR = 10 dB; n = 200; L = 200

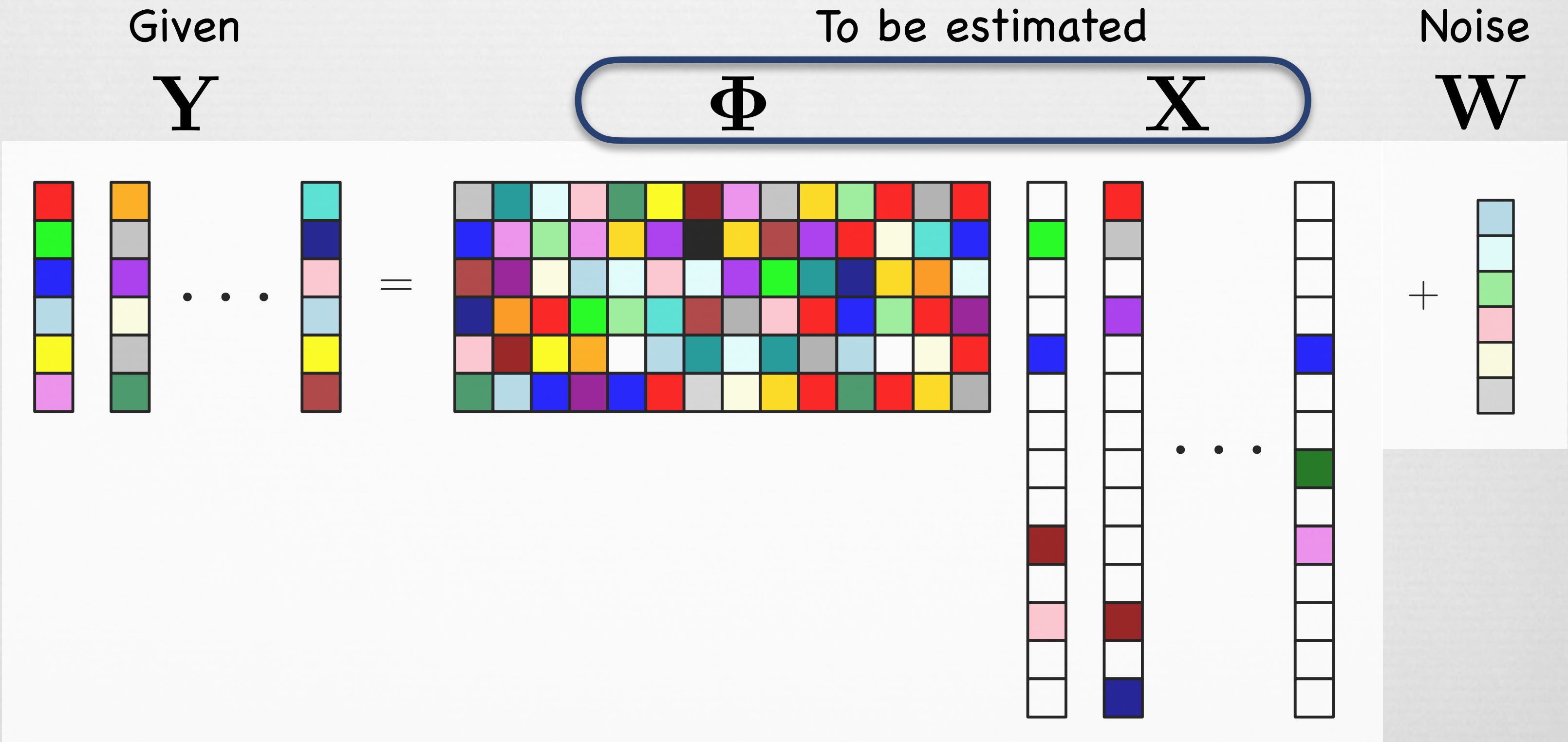
SNR = 10 dB; k = 50 log n
m = 0.75 k, mL = 50 k log n



RD-CMP is a fast covariance matching based MMV solver! [Khanna & M., 17]

Dictionary Learning

- Matrix factorization problem:



SBL framework for DL

- Type-II ML: solve $\max_{\boldsymbol{\Lambda}=\{\boldsymbol{\Phi},\boldsymbol{\Gamma}\}} \log p(\mathbf{Y};\boldsymbol{\Lambda})$
- EM procedure:
  - E-step: update statistics of X, as before
  - M-step: separable in variables $\boldsymbol{\Phi}, \boldsymbol{\Gamma}$
    - Closed-form update for $\boldsymbol{\Gamma}$
    - Non-convex in $\boldsymbol{\Phi}$
    - Alternating minimization (AM): update one column of $\boldsymbol{\Phi}$ at a time

Image Denoising Example

(a) Original image
(b) Corrupted image, PSNR = 20 dB
(c) DL-SBL, PSNR = 28.96 dB, run time = 105.7 s
(d) SimCO, PSNR = 28.64 dB, run time = 58.7 s
(e) DL-MM, PSNR = 28.54 dB, run time = 98.7 s
(f) KSVD, PSNR = 28.34 dB, run time = 76.7 s
(g) SGK, PSNR = 27.44 dB, run time = 82.5 s
(h) PAU, PSNR = 27.44 dB, run time = 84.5 s
(i) MOD, PSNR = 27.42 dB, run time = 79.2 s

- 512 x 512 image "Barbara"
- Goal: remove AWGN
- Learn dictionary using 1000 8 x 8 blocks, randomly chosen
- N = 256
- Learn dictionary
- Reconstruct image using OMP

[G. Joseph and M., TSP 2020]

DL-SBL Guarantees

- Cost function converges, iterates converge to stationary points
- Global minimum of the DL-SBL cost function occurs at the desired soln., sparse local minima
- FIRST convergence guarantee for DL algorithms!

[Joseph & M., TSP 2020]

Part 4: From Compressed Sensing to Control Theory

Linear dynamical systems

Applications

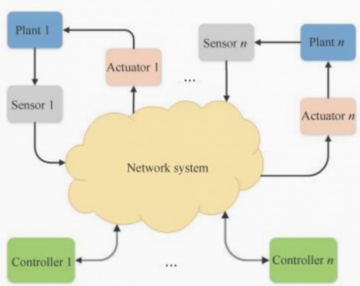
Sparse initial state

Diffusion processes

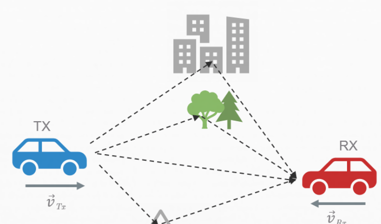
Epidemic spreading

Fake news spreading

Sparse control



Networked control system



Wireless channel

Network opinion manipulation

Sparsity and Linear Dynamical Systems

- System Model: $$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k$$ $$\mathbf{y}_k = \mathbf{C}_{(k)}\mathbf{x}_k + \mathbf{w}_k$$
- Goal: observe, control, stabilize linear dynamical systems under sparsity constraints
  - Some examples:
    - With known inputs: recover sparse initial state from observations
    - With unknown sparse inputs: recover state and inputs from observations
    - Design sparse inputs to reach a desired state

Sparse Initial State: Observability

- Recovery problem: $$\begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{K-1} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{(0)} \\ \mathbf{C}_{(1)}\mathbf{A} \\ \vdots \\ \mathbf{C}_{(K-1)}\mathbf{A}^{K-1} \end{bmatrix} \mathbf{x}_0$$
- Recoverability depends on RIC of the "effective" measurement matrix
- Sufficient number of measurements:
  - Independent, iid $\mathbf{C}_{(k)}$: $Km \sim s\log(N/s)$
  - Single $\mathbf{C}_{(k)}$ with iid entries: $Km \sim s\log^2 s \log^2 N$
  - Matrix $\mathbf{A}$ "well conditioned"

[Joseph & M., SPL 2018, TSP 2019]

Sparse Controllability

- Problem: find sparse $\mathbf{u}_k$ s.t.

$$\mathbf{x}_{\text{final}} - \mathbf{A}^K \mathbf{B} \mathbf{x}_{\text{init}} = [(\mathbf{A}^{K-1}\mathbf{B})\ (\mathbf{A}^{K-2}\mathbf{B}) \ldots (\mathbf{B})] \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_K \end{bmatrix}$$

- Necessary and sufficient conditions for s-sparse controllability:
  - For all $\lambda \in \mathbb{C}$, Rank$\{[\mathbf{A} - \lambda \mathbf{I} \quad \mathbf{B}]\}$ = N
  - $s \geq N -$ Rank$(\mathbf{A})$
- No more than N sparse inputs needed to steer the system to a desired state

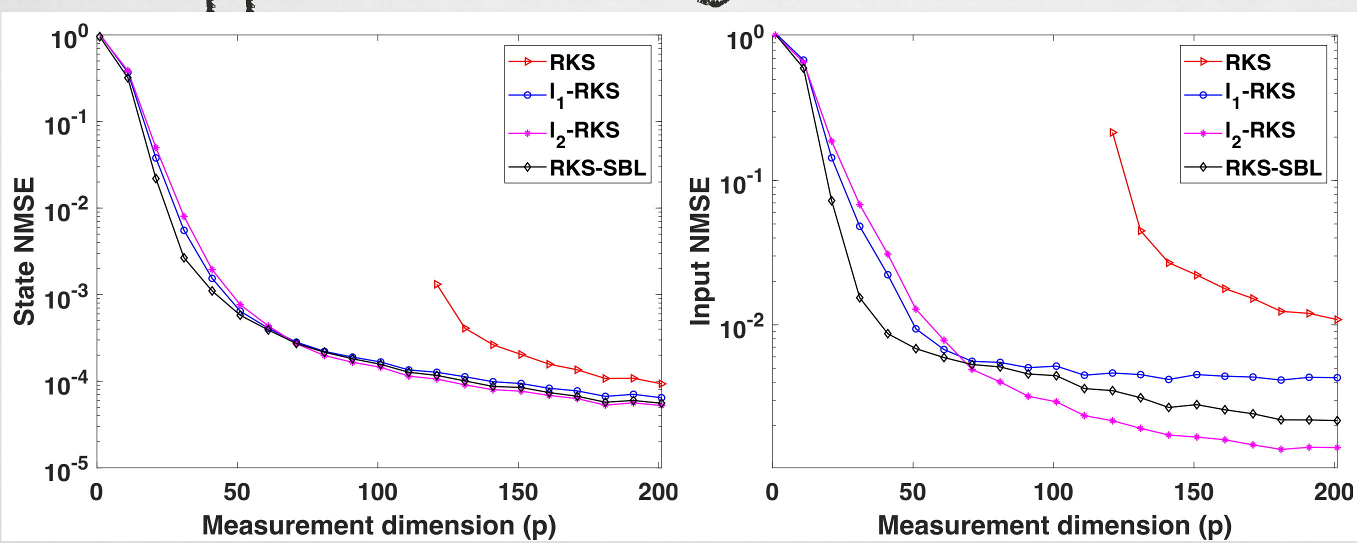
[Joseph & M., TAC 2021]

Design of Sparse Control Inputs

- Time-varying support:
  - Piecewise OMP
  - Piecewise inverse scale-space algo
- Fixed support: Reformulate as a block-sparse recovery problem. Many options!
  - Block OMP
  - Group LASSO
  - Block SBL, ...

Joint Recovery of State and Sparse Inputs

- Problem: recover $\{\mathbf{x}_k, \mathbf{u}_k : \|\mathbf{u}_k\|_0 \ll n\}_{k=1}^{K}$ from $\{\mathbf{y}_k\}_{k=1}^{K}$, with
$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k$$
$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{D}_k\mathbf{u}_k + \mathbf{v}_k$$
- Approaches: Regularizer-based; Bayesian



(a) NMSE in state estimation (b) NMSE in input estimation

RKS: Robust Kalman smoothing (classical approach)

Open Issues

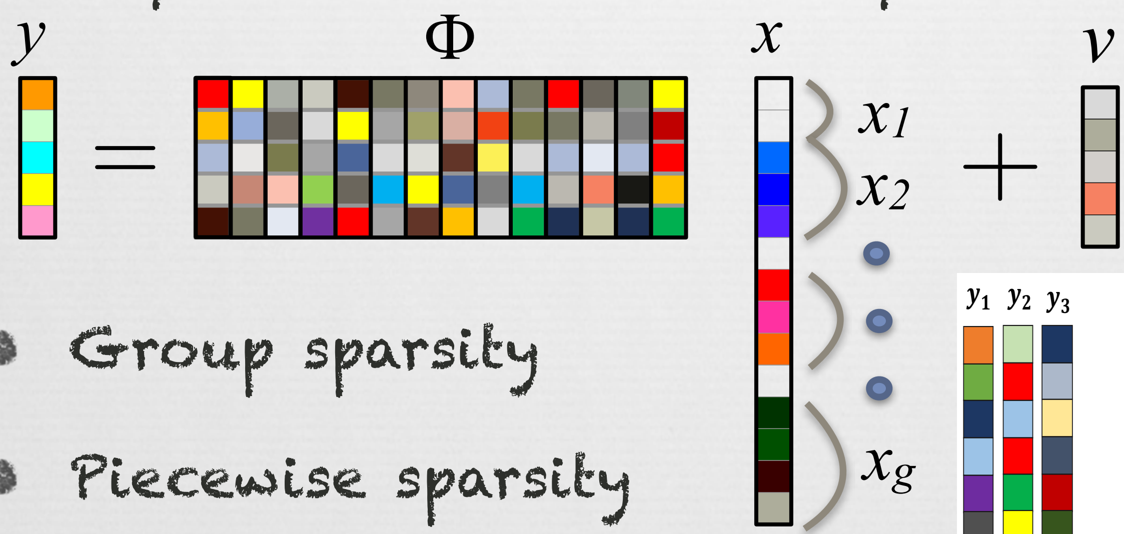
- Handling energy + sparsity constraints in the control of LDS
- Better algorithms for
  - state recovery under sparsity constraints
  - designing sparse inputs
  - system identification, e.g., using active learning
- Theoretical guarantees
- NEW APPLICATIONS!

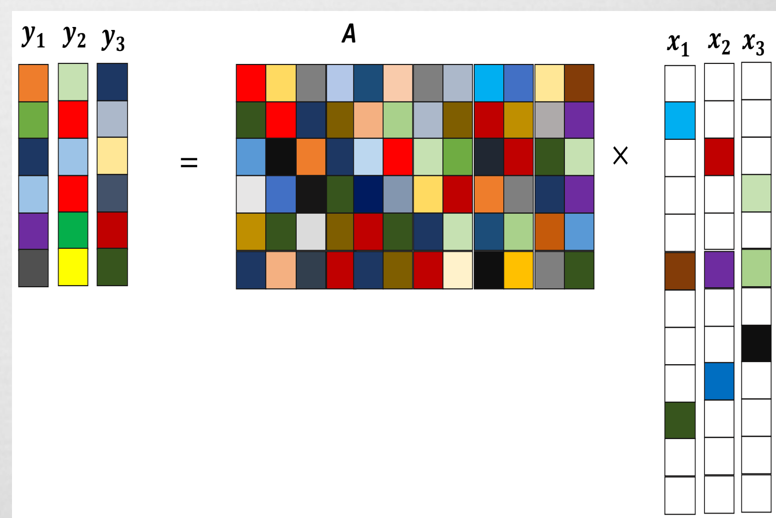
Part 5: Deep Unfolding

Learn any underlying structure, without hand-crafting priors, cost functions, or developing new algorithms!

Other Sparse Structures

- Any additional structure, when present, is important to model & exploit

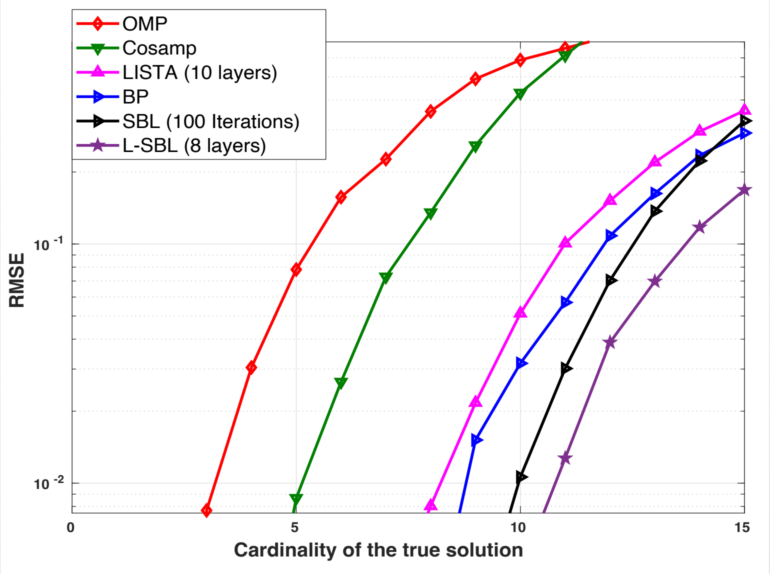




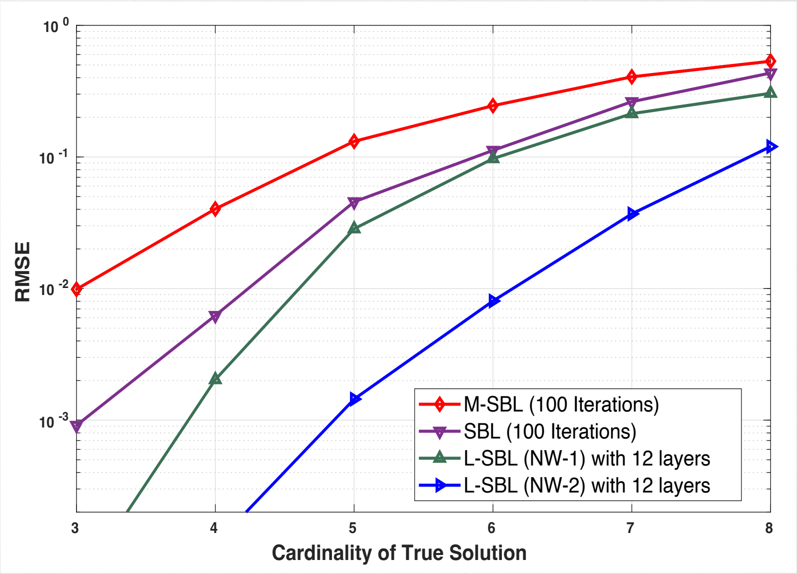
- Group sparsity
- Piecewise sparsity
- Inclusion-exclusion
- Varying sparsity pattern

Unfolded SBL

- Can unfold the SBL iterations
  - E-step: computes the posterior; custom layer
  - M-step: updates hyperparams; dense network



Sparse recovery performance



Time-varying support (arbitrary pattern)

[R. J. Peter and M., ArXiv 2019]

Summary

- Sparsity-aware Bayesian inference:
  - Superior guarantees translating to excellent performance
  - Ultra-fast algorithms and simple updates
  - Versatile framework

- Many opportunities to innovate!

- Reference: G. Joseph, S. Khanna, C. R. Murthy, R. Prasad, S. S. Thoota, "Sparsity-aware Bayesian inference and its applications," Handbook of Statistics, Elsevier, 2022.

Acknowledgements


Geethu Joseph


Saurabh Khanna


Ranjitha Prasad


Vinuthna Vinjamuri

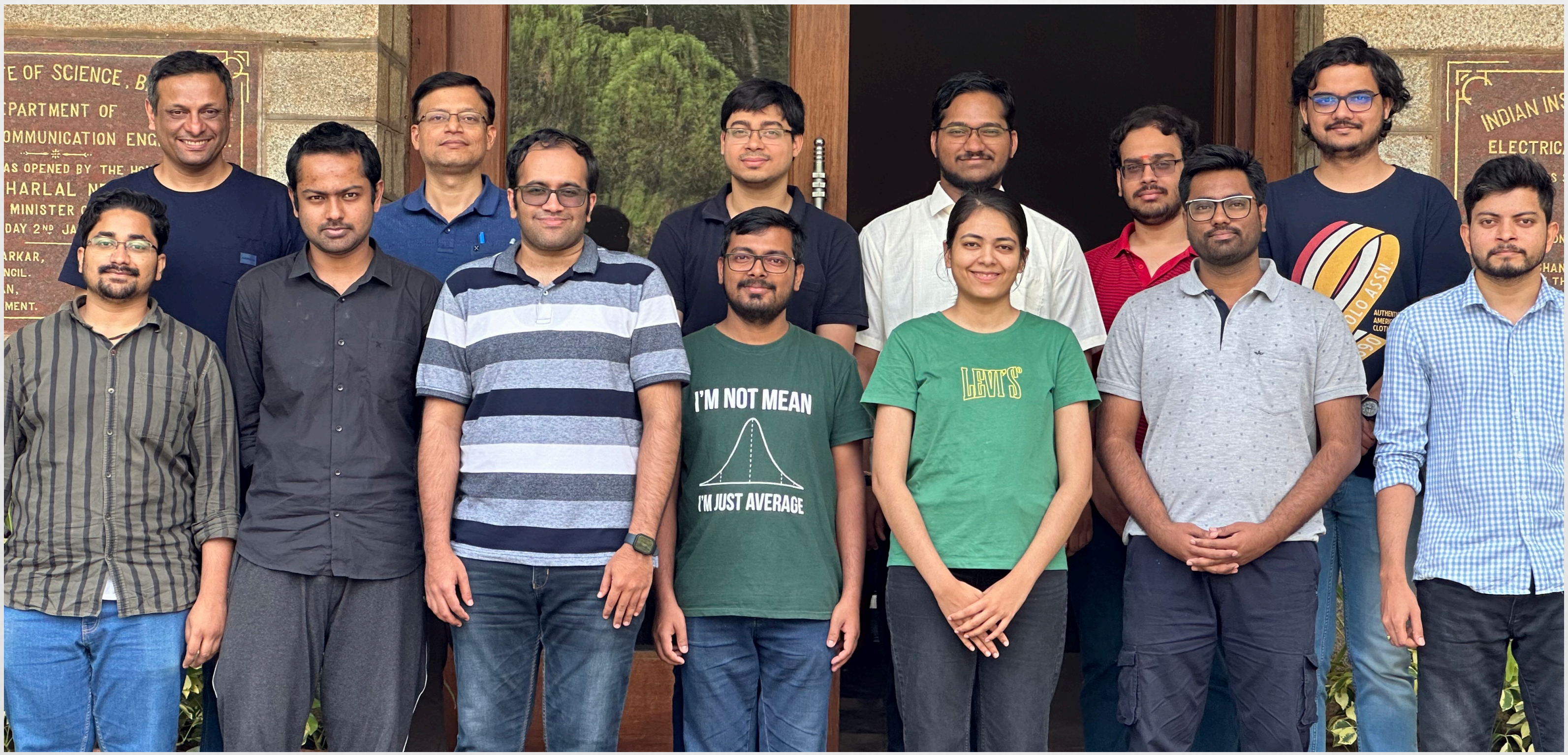

Lekshmi Ramesh


Arunkumar K. P.


Dheeraj Prasanna


Sai Thoota

Thank you!



IISc

Contact: cmurthy@iisc.ac.in

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान