A communication efficient scheme for decentralized estimation of jointly sparse signals

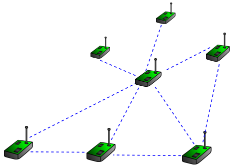
Saurabh Khanna,
Signal Processing for Communication, ECE, IISc

Contents

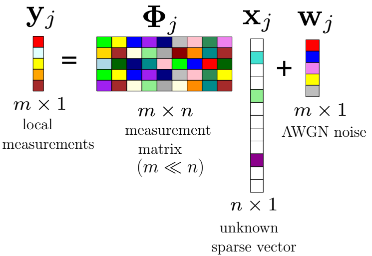
- Decentralized joint sparse signal recovery
  - Problem statement and prior work
- Proposed algorithm
  - Fusion based Sparse Bayesian Learning
- Simulation results

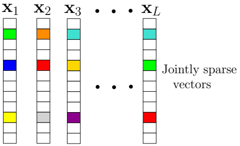
Decentralized joint sparse signal recovery

- Network of L sensor nodes



- Single/Multi hop communication links between nodes
- Measurement model at j^{th} node:





- $\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_L$ are **jointly sparse**

- **Goal:**
  - Decentralized estimation of $\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_L$ at their respective nodes
  - Exploit joint sparsity to reduce no. of local measurements
  - Reduce the amount of internode communication
  - Internode communication restricted to single hop neighborhood

Why decentralized algorithm ?

- Robust to nodal failures, no concept of fusion center
- Energy efficient to implement (think wireless networks)
- Attains centralized solution despite of computations/communications restricted to local neighborhoods

Prior work

Comparison of various decentralized algorithms

Decentralized algorithm	Computational complexity	Performance	Per node, per iteration communication cost
DCOMP	$\star\star\star\star\star$	$\star\star$	$\mathcal{O}(nL)$
DCSP	$\star\star\star\star$	$\star$	$\mathcal{O}(kL)$
DRL-1	$\star$	$\star\star\star$	$\mathcal{O}(nL)$
CB-DSBL	$\star\star$	$\star\star\star\star\star$	$\mathcal{O}(nL)$
FB-DSBL	$\star\star\star$	$\star\star\star\star$	$\mathcal{O}(kL \log n)$

- Algorithms not included in the comparison:
  1. DCS-AMP (involves direct exchange of signal coefficients between nodes)
  2. Turbo BCS (–do–)
  3. Decentralized SA-BMP (restricted to ring topology)

Quick recap of SBL

- SBL stands for **Sparse Bayesian Learning** [Wipf and Rao, 2004]

- **Problem:** Recover unknown sparse vector $\mathbf{x}$ from its noisy, underdetermined, linear measurements $\mathbf{y}$

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{w}$$

  - Impose a sparsity inducing signal prior, $\mathbf{x} \sim \mathcal{N}(0, \mathbf{\Gamma})$
  - $\mathbf{\Gamma} = \mathrm{diag}(\gamma_1, \gamma_2, \ldots \gamma_L)$ model the variance of entries of $\mathbf{x}$
  - If $\mathbf{\Gamma}$ is known, from LMMSE theory, $\hat{\mathbf{x}}_{\text{MAP}} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$

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

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

  - ML estimate $\gamma_{\text{ML}} = \underset{\gamma \in \mathbb{R}_+^n}{\arg\max}\ \log p(\mathbf{y}|\gamma)$ obtained via EM algorithm

$$\text{E step:} \quad Q(\gamma|\gamma^k) = \mathbb{E}_{\mathbf{x}|\mathbf{y},\gamma^k}[\log p(\mathbf{y}, \mathbf{x}|\gamma)]$$

$$\text{M step:} \quad \gamma^{k+1} = \underset{\gamma}{\arg\max}\ Q(\gamma|\gamma^k)$$

Hard and soft support estimates

- At node j, we define:

  1. Hard support estimate $\mathbf{b}_j$:
     $\in\{0,1\}^n$, binary vector representing current support estimate

  2. Soft support estimate $\mathbf{g}_j$:

$$\mathbf{g}_j(i) \triangleq \begin{cases} \gamma_j(i) & \text{if } \mathbf{b}_j(i) = 1 \\ 0 & \text{if } \mathbf{b}_j(i) = 0 \end{cases}, \qquad 1 \leq i \leq n$$

- $\mathbf{b}_j$ = support($\mathbf{g}_j$)

Proposed algorithm

- **FB-DSBL**: Fusion based Decentralized Sparse Bayesian Learning

  At node j,

  - **Step-1** Run SBL iteration to update local hyperparameters γ_j
  - **Step-2** Generate hard support estimate $\mathbf{b}_j$ using current estimate of γ_j
  - **Step-3** Generate soft support estimate $\mathbf{g}_j$ and broadcast it to single hop neighbors in $\mathcal{N}_j$
  - **Step-4** Use soft support estimate $\mathbf{g}_{j'}$, $j' \in \mathcal{N}_j$, received from neighboring nodes to update local γ_j
  - Repeat steps 1 to 4, until convergence

FB-DSBL

- **FB-DSBL**: Fusion based Decentralized Sparse Bayesian Learning

  At node j,

  - **Step-1** Run SBL iteration to update local hyperparameters γ_j
  - **Step-2** Generate hard support estimate $\mathbf{b}_j$ using current estimate of γ_j
  - **Step-3** Generate soft support estimate $\mathbf{g}_j$ and broadcast it to single hop neighbors in $\mathcal{N}_j$
  - **Step-4** Use soft support estimate $\mathbf{g}_{j'}$, $j' \in \mathcal{N}_j$, received from neighboring nodes to update local γ_j
  - Repeat steps 1 to 4, until convergence

Generation of hard support estimate $\mathbf{b}_j$

- At j^{th} node, for index i, $(1 \leq i \leq n)$, we define following two hypothesis

$$\mathcal{H}_0 : \mathbf{x}_j(i) = 0$$

$$\mathcal{H}_1 : \mathbf{x}_j(i) \neq 0$$

or equivalently,

$$\mathcal{H}_0 : \gamma_{\boldsymbol{j}}(i) = 0$$

$$\mathcal{H}_1 : \gamma_{\boldsymbol{j}}(i) > 0$$

where $\gamma_{\boldsymbol{j}}$ denotes the local variance parameters

- At node j, for index $i \in [n]$, a log likelihood ratio test (LLRT) is setup as:

Decide $\mathcal{H}_1$ if

$$\log \frac{p(\mathbf{y}_j; \mathcal{H}_1)}{p(\mathbf{y}_j; \mathcal{H}_0)} \geq \theta_{j,i}$$

Generation of hard support estimate $\mathbf{b}_j$

- Per index LLRT:

  Decide $\mathcal{H}_1$ if

$$\log \frac{\mathcal{N}(\mathbf{y}_j; 0, \sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \mathbf{\Gamma}_j^k \mathbf{\Phi}_j)}{\mathcal{N}(\mathbf{y}_j; 0, \sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j)} \geq \theta_{j,i}$$

  where $\mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_j \mathbf{\Phi}_j^T = \sum_{k \neq i} \boldsymbol{\gamma}_{\mathbf{j}}(k) \phi_{j,k} \phi_{j,k}^T$

- After simplification, we get

$$\frac{\left(\mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T \right)^{-1} \mathbf{y}_j \right)^2}{\mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T \right)^{-1} \mathbf{\Phi}_{j,i}} \geq g(\backslash\gamma_j^k(i)) \cdot \left(\frac{1}{\gamma_j^k(i)} + \mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T \right)^{-1} \mathbf{\Phi}_{j,i} \right)$$

  where $g(\backslash\gamma_j^k(i)) = \frac{2\theta_{j,i} + \log \left(1 + \mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T \right)^{-1} \mathbf{\Phi}_{j,i} \right)}{\mathbf{\Phi}_{j,i}^T \left(\sigma_j^2 \mathbf{I}_m + \mathbf{\Phi}_j \tilde{\mathbf{\Gamma}}_{j,i}^k \mathbf{\Phi}_j^T \right)^{-1} \mathbf{\Phi}_{j,i}}$

Generation of hard support estimate $\mathbf{b}_j$

- Per index LLRT:

  Decide $\mathcal{H}_1$ if

$$\frac{\left(\boldsymbol{\Phi}_{j,i}^T\left(\sigma_j^2\mathbf{I}_m+\boldsymbol{\Phi}_j\tilde{\boldsymbol{\Gamma}}_{j,i}^k\boldsymbol{\Phi}_j^T\right)^{-1}\mathbf{y}_j\right)^2}{\boldsymbol{\Phi}_{j,i}^T\left(\sigma_j^2\mathbf{I}_m+\boldsymbol{\Phi}_j\tilde{\boldsymbol{\Gamma}}_{j,i}^k\boldsymbol{\Phi}_j^T\right)^{-1}\boldsymbol{\Phi}_{j,i}} \geq g(\backslash\gamma_j^k(i))\cdot\left(\frac{1}{\gamma_j^k(i)}+\boldsymbol{\Phi}_{j,i}^T\left(\sigma_j^2\mathbf{I}_m+\boldsymbol{\Phi}_j\tilde{\boldsymbol{\Gamma}}_{j,i}^k\boldsymbol{\Phi}_j^T\right)^{-1}\boldsymbol{\Phi}_{j,i}\right)$$

  where $g(\backslash\gamma_j^k(i)) = \frac{2\theta_{j,i}+\log\left(1+\boldsymbol{\Phi}_{j,i}^T\left(\sigma_j^2\mathbf{I}_m+\boldsymbol{\Phi}_j\tilde{\boldsymbol{\Gamma}}_{j,i}^k\boldsymbol{\Phi}_j^T\right)^{-1}\boldsymbol{\Phi}_{j,i}\right)}{\boldsymbol{\Phi}_{j,i}^T\left(\sigma_j^2\mathbf{I}_m+\boldsymbol{\Phi}_j\tilde{\boldsymbol{\Gamma}}_{j,i}^k\boldsymbol{\Phi}_j^T\right)^{-1}\boldsymbol{\Phi}_{j,i}}$.

- Some observations:
  - Under $\mathcal{H}_0$, the test metric is standard chi-squared distributed (DOF = 1)
  - Denominator in LHS is a normalization factor
  - Note that test metric in LHS does not depend on $\gamma_j(i)$
  - $g(\backslash\gamma_j^k(i))$ is independent of $\gamma_j^k(i)$
  - Overall, the LLRT threshold is inversely proportional to $\gamma_j(i)$

Generation of hard support estimate $\mathbf{b}_j$

- Hard support estimate $\mathbf{b}_j$ is generated by performing individual LLRTs for each index $i \in [n]$:

  Decide $\mathcal{H}_1$ if

$$T_{j,i}(\mathbf{y}_j) = \frac{(\phi_{j,i}^T(\sigma_j^2 \mathbf{I}_m + \boldsymbol{\Phi}_j \tilde{\boldsymbol{\Gamma}}_j \boldsymbol{\Phi}_j^T)^{-1} \mathbf{y}_j)^2}{\phi_{j,i}^T(\sigma_j^2 \mathbf{I}_m + \boldsymbol{\Phi}_j \tilde{\boldsymbol{\Gamma}}_j \boldsymbol{\Phi}_j^T)^{-1} \phi_{j,i}} \geq [\mathcal{Q}^{-1}(\alpha)]^2$$

- $\alpha = \mathbb{P}(\mathbf{b}_j(i) = 1 | \mathcal{H}_0)$ for all j, i

FB-DSBL

- **FB-DSBL**: Fusion based Decentralized Sparse Bayesian Learning

  At node j,

  - **Step-1** Run SBL iteration to update local hyperparameters γ_j
  - **Step-2** Generate hard support estimate $\mathbf{b}_j$ using current estimate of γ_j
  - **Step-3** Generate soft support estimate $\mathbf{g}_j$ and broadcast it to single hop neighbors in $\mathcal{N}_j$
  - **Step-4** Use soft support estimate $\mathbf{g}_{j'}$, $j' \in \mathcal{N}_j$, received from neighboring nodes to update local γ_j
  - Repeat steps 1 to 4, until convergence

Messages exchanged by nodes

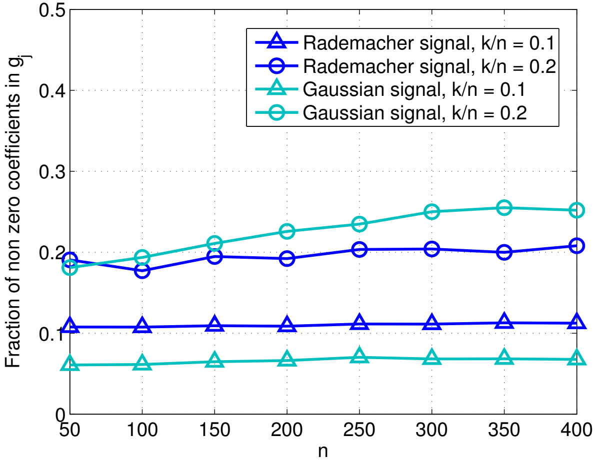
- Structure of message exchanged between the nodes

Fixed length header	No. of non zero entries in $\mathbf{g}_j$	Index and value of 1^{th} non zero entry in $\mathbf{g}_j$	Index and value of 2^{nd} non zero entry in $\mathbf{g}_j$	• • • •	Index and value of last non zero entry in $\mathbf{g}_j$
c_1 bits	$\log n$ bits	$\log n + c_2$ bits	$\log n + c_2$ bits		$\log n + c_2$ bits

- Variable length code used to encode $\mathbf{g}_j$

- $\mathcal{O}(k \log n)$ bits required on average to encode the location and magnitude of non zero entries of soft support estimate $\mathbf{g}_j$

Message size



- SNR = 20 dB, n = 50, m/n = 0.25, L = 10 nodes, trials = 50, $\alpha = 10^{-4}$

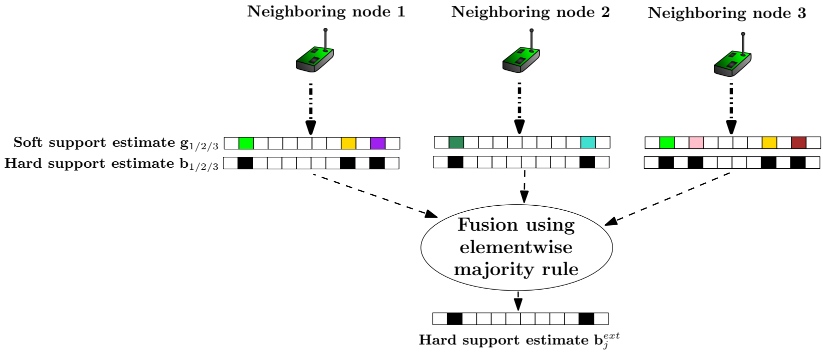
FB-DSBL

- **FB-DSBL**: Fusion based Decentralized Sparse Bayesian Learning

  At node j,

  - **Step-1** Run SBL iteration to update local hyperparameters γ_j
  - **Step-2** Generate hard support estimate $\mathbf{b}_j$ using current estimate of γ_j
  - **Step-3** Generate soft support estimate $\mathbf{g}_j$ and broadcast it to single hop neighbors in $\mathcal{N}_j$
  - **Step-4** Use soft support estimate $\mathbf{g}_{j'}$, $j' \in \mathcal{N}_j$, received from neighboring nodes to update local γ_j
  - Repeat steps 1 to 4, until convergence

Fusion of hard support estimates



- Fuse hard support estimates from neighboring nodes to generate extrinsic hard support $\mathbf{b}_j^{\text{ext}}$

$$\mathbf{b}_j^{\text{ext}}(i) \triangleq \begin{cases} 1 & \text{if } |\mathcal{A}_i^j| \geq \frac{|\mathcal{N}_j|}{2} \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{A}_i^j = \{j' \in \mathcal{N}_j : \mathbf{b}_{j'}(i) = 1\}$.

Updating γ using extrinsic information

- If $\mathbf{b}_j^{\text{ext}}(i) = 1$, update hyperparameter γ as a weighted average:

$$\gamma_j^{\text{new}}(i) = \frac{\gamma_j(i) + \sum_{j' \in \mathcal{N}_j} \mathbf{b}_{j'}(i)\mathbf{g}_{j'}(i)}{1 + \sum_{j' \in \mathcal{N}_j} \mathbf{b}_{j'}(i)}$$

Updating γ using extrinsic information

- If $\mathbf{b}_j^{\text{ext}}(i) = 1$, shrink hyperparameter $\gamma_j(i)$.
- Shrinkage of $\gamma_{\boldsymbol{j}}(i)$:
  - results in a drop in probability of false detection (of zero coefficient at i)
  - also results in a drop in probability of detection (of non zero coefficient at i)
  - must be commensurate with the extrinsic belief in detecting a zero at i^{th} index

Updating γ using extrinsic information

- If $\mathbf{b}_j^{\text{ext}}(i) = 1$, shrink hyperparameter $\gamma_j(i)$.
- Shrinkage of $\gamma_{\boldsymbol{j}}(i)$:
  - results in a drop in probability of false detection (of zero coefficient at i)
  - also results in a drop in probability of detection (of non zero coefficient at i)
  - must be commensurate with the extrinsic belief in detecting a zero at i^{th} index
- Extrinsic belief of finding a zero at i^{th} index $\propto \mathbb{P}(\mathbf{b}_j^{\text{ext}}(i) = 1|\mathcal{H}_0)$

$$\mathbb{P}(\mathbf{b}_j^{\text{ext}}(i) = 1|\mathcal{H}_0) = \sum_{k=\frac{|\mathcal{N}_j|}{2}}^{|\mathcal{N}_j|} \binom{|\mathcal{N}_j|}{k} \alpha^k (1-\alpha)^{|\mathcal{N}_j - k|}$$

Updating γ using extrinsic information

- Proposed solution: Shrink $\gamma(i)$ such that:

$$P_{\text{FA}}(\text{LLRT at index } i) = P_{\text{FA}}(\text{detector}\mathcal{Z})$$

$$\text{where } \mathcal{Z} = AND(\mathbf{b}_j(i), \mathbf{b}_j^{\text{ext}}(i))$$

Updating γ using extrinsic information

- Proposed solution: Shrink $\gamma(i)$ such that:

$$P_{\mathrm{FA}}(\text{LLRT at index } i) = P_{\mathrm{FA}}(\text{detector}\mathcal{Z})$$

$$\text{where } \mathcal{Z} = AND(\mathbf{b}_j(i), \mathbf{b}_j^{\mathrm{ext}}(i))$$

$$\begin{aligned} P_{\mathrm{FA}}(\text{detector}\mathcal{Z}) &= \mathbb{P}(\mathbf{b}_j(i) = 1, \mathbf{b}_j^{\mathrm{ext}}(i) = 1 | \mathcal{H}_0) \\ &= \mathbb{P}(\mathbf{b}_j(i) = 1 | \mathcal{H}_0) \cdot \mathbb{P}(\mathbf{b}_j^{\mathrm{ext}}(i) = 1 | \mathcal{H}_0) \\ &= \alpha \cdot \mathbb{P}(\mathbf{b}_j^{\mathrm{ext}}(i) = 1 | \mathcal{H}_0) \qquad (< \alpha) \end{aligned}$$

Updating γ using extrinsic information

- Proposed solution: Shrink $\gamma(i)$ such that:

$$P_{\text{FA}}(\text{LLRT at index } i) = P_{\text{FA}}(\text{detector}\mathcal{Z})$$

$$\text{where } \mathcal{Z} = AND(\mathbf{b}_j(i), \mathbf{b}_j^{\text{ext}}(i))$$

$$\begin{aligned}
P_{\text{FA}}(\text{detector}\mathcal{Z}) &= \mathbb{P}(\mathbf{b}_j(i) = 1, \mathbf{b}_j^{\text{ext}}(i) = 1|\mathcal{H}_0) \\
&= \mathbb{P}(\mathbf{b}_j(i) = 1|\mathcal{H}_0) \cdot \mathbb{P}(\mathbf{b}_j^{\text{ext}}(i) = 1|\mathcal{H}_0) \\
&= \alpha \cdot \mathbb{P}(\mathbf{b}_j^{\text{ext}}(i) = 1|\mathcal{H}_0) \qquad (< \alpha)
\end{aligned}$$

- Backpropagating the new $P_{\text{FA}}(\text{LLRT})$ to obtain corresponding new threshold $\theta_{j,i}^{\text{new}}$

$$\theta_{j,i}^{\text{new}} = \left(\mathcal{Q}^{-1}\left(\frac{P_{\text{FA}}(\text{detector}\mathcal{Z})}{2}\right)\right)^2$$

Updating γ using extrinsic information

- Old and new LLRT thresholds for node j and i^{th} index,

$$\theta_{j,i}^{\text{old}} = \left(\mathcal{Q}^{-1}\left(0.5\alpha\right)\right)^2$$

$$\theta_{j,i}^{\text{new}} = \left(\mathcal{Q}^{-1}\left(0.5P_{\text{FA}}(\text{detector}\mathcal{Z})\right)\right)^2$$

Updating γ using extrinsic information

- Old and new LLRT thresholds for node j and i^{th} index,

$$\theta_{j,i}^{\text{old}} = \left(\mathcal{Q}^{-1}\left(0.5\alpha\right)\right)^2$$

$$\theta_{j,i}^{\text{new}} = \left(\mathcal{Q}^{-1}\left(0.5 P_{\text{FA}}(\text{detector}\,\mathcal{Z})\right)\right)^2$$

- Then, we can write

$$\eta \triangleq \left(\frac{\mathcal{Q}^{-1}(0.5 P_{\text{FA}}(\text{detector}\,\mathcal{Z}))}{\mathcal{Q}^{-1}(0.5\alpha)}\right)^2 = \frac{g(\backslash\boldsymbol{\gamma}_j^{k,new}(i))\cdot\left(\frac{1}{\boldsymbol{\gamma}_j^{k,new}(i)}+\boldsymbol{\Phi}_{j,i}^T\left(\sigma_j^2\mathbf{I}_m+\boldsymbol{\Phi}_j\tilde{\boldsymbol{\Gamma}}_{j,i}^k\boldsymbol{\Phi}_j^T\right)^{-1}\boldsymbol{\Phi}_{j,i}\right)}{g(\backslash\boldsymbol{\gamma}_j^{k}(i))\cdot\left(\frac{1}{\boldsymbol{\gamma}_j^{k}(i)}+\boldsymbol{\Phi}_{j,i}^T\left(\sigma_j^2\mathbf{I}_m+\boldsymbol{\Phi}_j\tilde{\boldsymbol{\Gamma}}_{j,i}^k\boldsymbol{\Phi}_j^T\right)^{-1}\boldsymbol{\Phi}_{j,i}\right)}$$

Updating γ using extrinsic information

- Old and new LLRT thresholds for node j and i^{th} index,

$$\theta_{j,i}^{\text{old}} = \left(\mathcal{Q}^{-1}\left(0.5\alpha\right)\right)^2$$
$$\theta_{j,i}^{\text{new}} = \left(\mathcal{Q}^{-1}\left(0.5P_{\text{FA}}(\text{detector}\mathcal{Z})\right)\right)^2$$

- Then, we can write

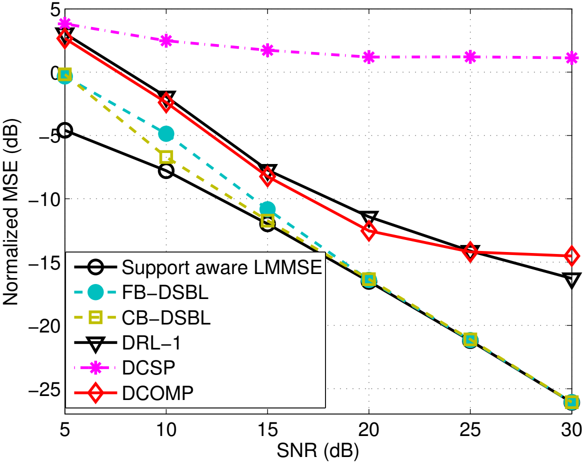
$$\eta \triangleq \left(\frac{\mathcal{Q}^{-1}(0.5P_{\text{FA}}(\text{detector}\mathcal{Z}))}{\mathcal{Q}^{-1}(0.5\alpha)}\right)^2 = \frac{g(\backslash\gamma_j^{k,new}(i))\cdot\left(\frac{1}{\gamma_j^{k,new}(i)}+\mathbf{\Phi}_{j,i}^T\left(\sigma_j^2\mathbf{I}_m+\mathbf{\Phi}_j\tilde{\mathbf{\Gamma}}_{j,i}^k\mathbf{\Phi}_j^T\right)^{-1}\mathbf{\Phi}_{j,i}\right)}{g(\backslash\gamma_j^k(i))\cdot\left(\frac{1}{\gamma_j^k(i)}+\mathbf{\Phi}_{j,i}^T\left(\sigma_j^2\mathbf{I}_m+\mathbf{\Phi}_j\tilde{\mathbf{\Gamma}}_{j,i}^k\mathbf{\Phi}_j^T\right)^{-1}\mathbf{\Phi}_{j,i}\right)}$$

to get the update rule

$$\boxed{\gamma_{\boldsymbol{j}}^{\text{new}}(i) = \frac{\gamma_{\boldsymbol{j}}^k(i)}{\eta + (\eta-1)\gamma_{\boldsymbol{j}}^k(i)(\phi_{j,i}^T(\sigma_j^2\mathbf{I}_m + \mathbf{\Phi}_j\tilde{\mathbf{\Gamma}}^k\mathbf{\Phi}_j^T)^{-1}\phi_{j,i})}}$$

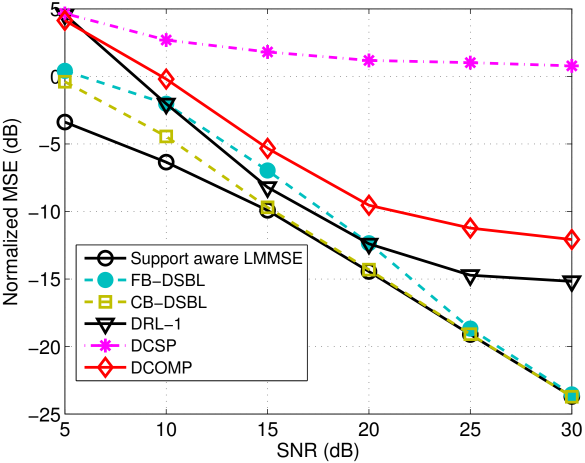
$$\text{where } \mathbf{\Phi}_j\tilde{\mathbf{\Gamma}}_j\mathbf{\Phi}_j^T = \sum_{k\neq i}\gamma_{\boldsymbol{j}}(k)\phi_{j,k}\phi_{j,k}^T$$

MSE performance (Rademacher source)



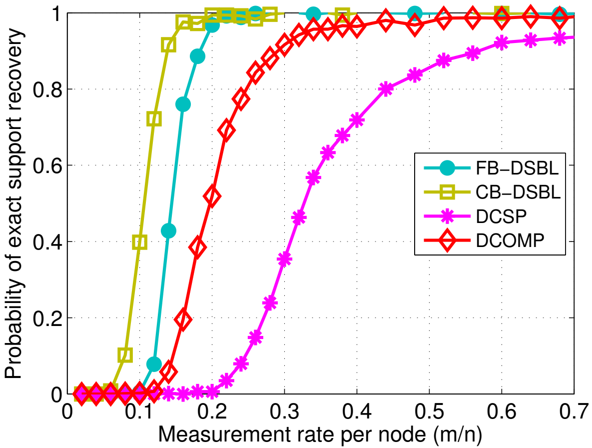
- n = 50, m = 15, 10% sparsity, L = 10 nodes, trials = 200, $\alpha = 10^{-4}$

MSE performance (Gaussian source)



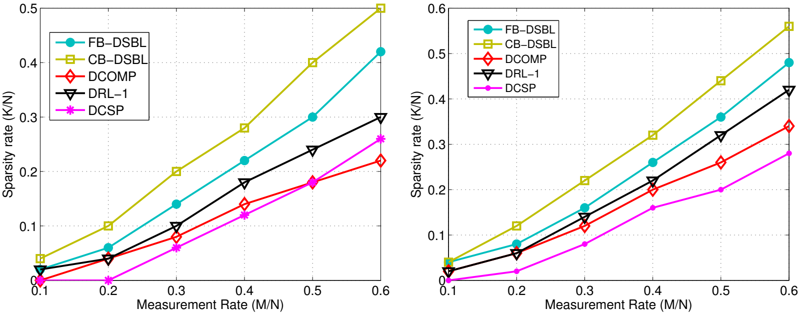
- n = 50, m = 15, 10% sparsity, L = 10 nodes, trials = 200, $\alpha = 10^{-4}$

Support recovery



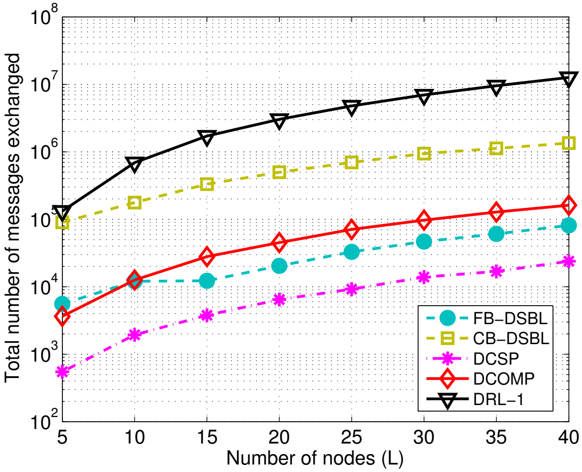
- n = 50, 10% sparsity, L = 10 nodes, SNR = 15 dB, trials = 400, $\alpha = 10^{-4}$

Phase transition characteristics



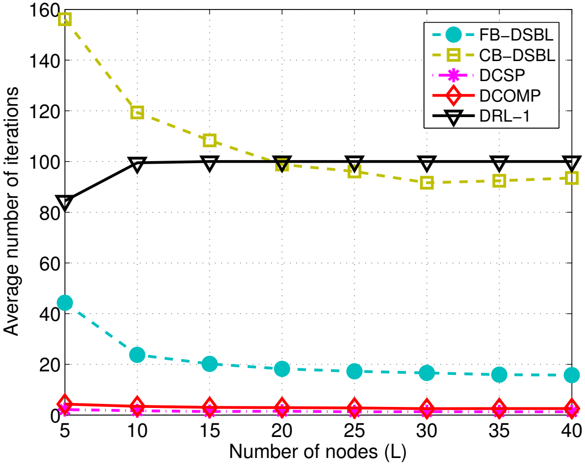
- n = 50, 10% sparsity, L = 10 nodes, SNR = 15 dB, trials = 400, $\alpha = 10^{-4}$

Communication cost



- n = 50, k = 5, m = 10, SNR = 20 dB, trials = 100, $\alpha = 10^{-4}$

Number of iterations



- n = 50, k = 5, m = 10, SNR = 20 dB, trials = 100, $\alpha = 10^{-4}$