Iterative Reconstruction Algorithms for Compressive Sensing based on Relaxed Belief Propagation

Mengke Lian and Henry D. Pfister

Department of Electrical and Computer Engineering
Duke University

International Symposium on Turbo Codes
Hong Kong Polytechnic University
December 6th, 2018

Outline

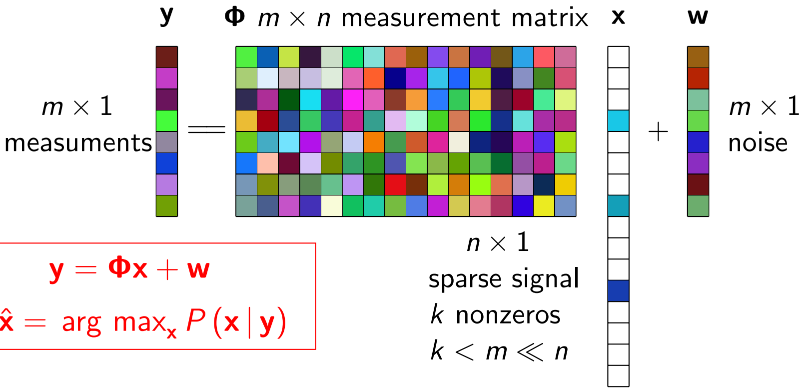
- Background on Compressive Sensing (CS)
- Belief-Propagation (BP) and iterative reconstruction algorithms
- Families of measurement matrices
- RBP-SVD and Numerical experiments
- RBP-SEQ and Numerical experiments

Outline

- Background on Compressive Sensing (CS)
- Belief-Propagation (BP) and iterative reconstruction algorithms
- Families of measurement matrices
- RBP-SVD and Numerical experiments
- RBP-SEQ and Numerical experiments

Compressive Sensing (CS)

- Signal recovery from fewer measurements than Nyquist limit
- Diverse set of signal models and reconstruction goals



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

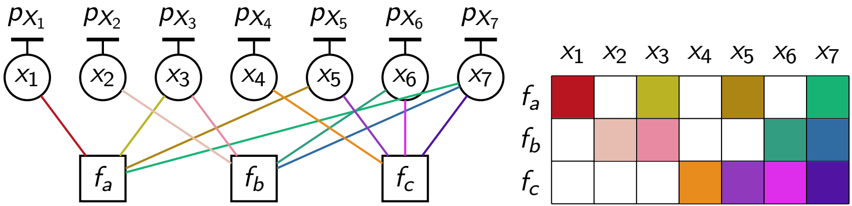
$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} P(\mathbf{x} \mid \mathbf{y})$$

Outline

- Background on Compressive Sensing (CS)
- Belief-Propagation (BP) and iterative reconstruction algorithms
- Families of measurement matrices
- RBP-SVD and RBP-SEQ algorithms
- Numerical experiments

Factor Graphs (FGs)

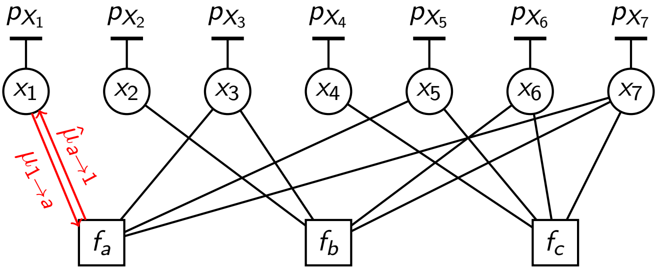
Any measurement matrix $\mathbf{\Phi}$ has a corresponding factor graph FG($\mathbf{\Phi}$).



- Factor Graph
  - Bipartite graph exposing the conditional independence structure for a set of random variables $\mathbf{X} = (X_1, \ldots, X_n) \in \mathcal{X}^n$
  - Joint distribution $\propto$ product of factors defined on subsets
  - BP is a message-passing approx. for marginal inference on FGs
  - Messages in BP are probability density functions over $\mathcal{X}$

Belief Propagation (BP): Messages

Each edge (i, a) in the factor graph is assigned with a pair messages $(\hat{\mu}_{a\to i}, \mu_{i\to a})$, both are probability distribution over $\mathcal{X}$

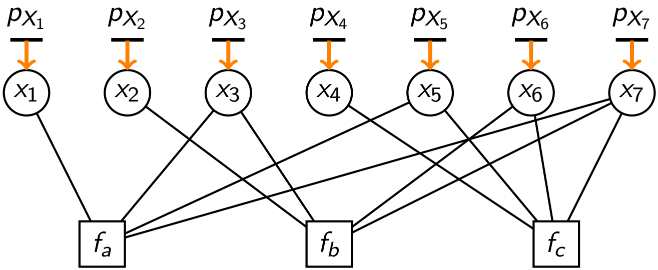


Define $\partial\cdot$ as the "neighborhood" operator, for example

$$\partial a = \{1, 3, 5, 7\} \quad \text{and} \quad \partial 7 = \{a, b, c\}$$

Compressive Sensing Belief Propagation (CSBP)

First studied by Sarvotham, Baron, and Baraniuk in 2005 [BSB10]
Initialization: initialize all $\mu_{i \to a}$ as the prior p_{X_i}



Compressive Sensing Belief Propagation (CSBP)

Algorithm 1 BP Algorithm for Compressive Sensing (CSBP)

Initialize: $\mu_{i\to a} = p_{X_i}, \quad (i,a) \in \mathcal{E}$

for $t = 1$ **to** T **do**

  for $a = 1$ **to** m **do**

    $\hat{\mu}_{a\to i}(x_i) \propto \int_{\mathbf{x}_{\partial a\setminus i}} p_{W_a}\left(y_a - \sum_{j\in\partial a} \Phi_{aj} x_j\right) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j)\, \mathrm{d}\mathbf{x}_{\partial a\setminus i}$

  for $i = 1$ **to** n **do**

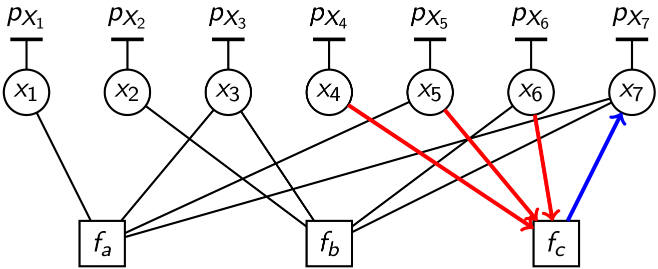
    $\mu_{i\to a}(x_i) \propto p_{X_i}(x_i) \cdot \prod_{b\in\partial i\setminus a} \hat{\mu}_{b\to i}(x_i)$

for $i = 1$ **to** n **do**

  $\mu_i(x_i) \propto p_{X_i}(x_i) \cdot \prod_{b\in\partial i} \hat{\mu}_{b\to i}(x_i), \quad \hat{x}_i = \int \mu_i(x_i)\, \mathrm{d}x_i$ or mode(μ_i)

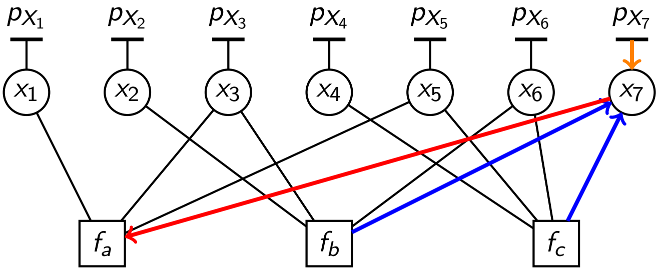
Compressive Sensing Belief Propagation (CSBP)

function-node to variable-node update:



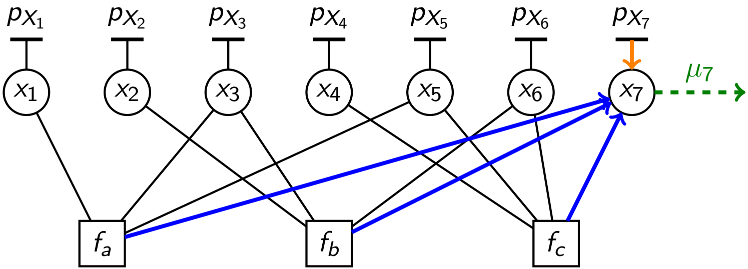
Compressive Sensing Belief Propagation (CSBP)

variable node to function-node update:



Compressive Sensing Belief Propagation (CSBP)

Approximate posterior marginals and estimate $\hat{x}_i$'s

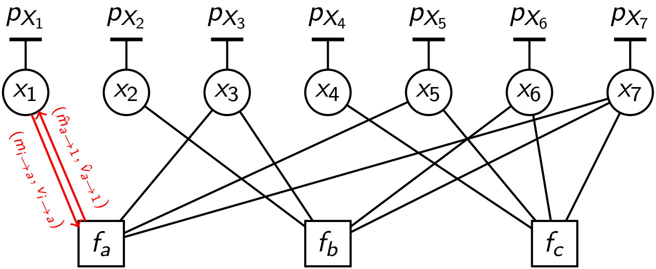


Relaxed Belief Propagation (RBP)

Simplified via Gaussian approximation (GA) by Guo and Wang [GW06]

$$\hat{m}_{a\to i} = \int x \hat{\mu}_{a\to i}(x)\,\mathrm{d}x, \quad \hat{v}_{a\to i} = \int x^2 \hat{\mu}_{a\to i}(x)\,\mathrm{d}x - \hat{m}^2_{a\to i}$$

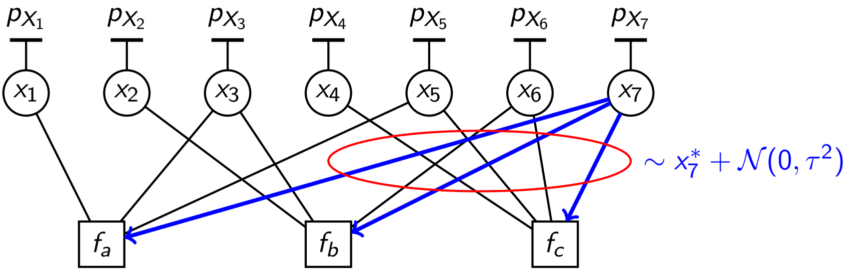
$$m_{i\to a} = \int x \mu_{i\to a}(x)\,\mathrm{d}x, \quad v_{i\to a} = \int x^2 \mu_{i\to a}(x)\,\mathrm{d}x - m^2_{i\to a}$$



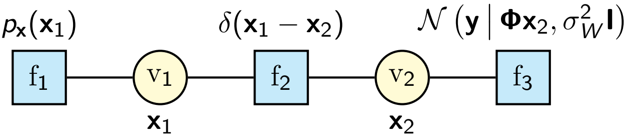
Approximate Message Passing (AMP)

AMP is a simplified RBP for dense measurement matrices [MM10]
Similar to low-complexity CDMA decoder by Kabashima in 2003.

Under mild conditions, all edges of a variable node have very similar output messages due to BP leave-one-out rule. One can approximate by a single message plus a small correction that depends on edge.



Vector AMP (VAMP)



$$p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}) = p_{\mathbf{x}}(\mathbf{x}_1)\,\delta(\mathbf{x}_1 - \mathbf{x}_2)\,\mathcal{N}\left(\mathbf{y} \,\middle|\, \mathbf{\Phi}\mathbf{x}_2, \sigma_W^2 \mathbf{I}\right)$$

- Developed by Schniter, Rangan, and Fletcher [SRF16]
- Based on a cycle-free graph with vector-valued nodes
- Alternating steps of scalar denoising and LMMSE estimation
- Rigorous state evolution for right-rotationally invariant matrices
- Equivalent, in some cases, to orthogonal AMP [ML16]

Algorithm Properties

- CSBP: Computation is accurate but extremely complex; only applies to sparse matrix in practice.
- RBP: Unstable when the measurement matrix is ill-conditioned.
- AMP: Only known to work for iid sub-Gaussian matrices.
- VAMP: Only known to work for right-rotationally invariant matrices.

Goals

- Compare iterative reconstruction algorithm for CS.
- Introduce RBP-SVD to improve stability of RBP for a larger class of measurement matrices.
- Introduce RBP-SEQ to reduce computation complexity when the measurement matrix has stage-wise sparse structure.

Outline

- Background on Compressive Sensing (CS)
- Belief-Propagation (BP) and iterative reconstruction algorithms
- Families of measurement matrices
- RBP-SVD and Numerical experiments
- RBP-SEQ and Numerical experiments

Right-Rotationally Invariant (RRI) Matrix

Definition

A matrix $\mathbf{\Phi}$ *is called right-rotationally invariant if* $\mathbf{\Phi} \overset{\mathrm{d}}{=} \mathbf{\Phi}\mathbf{R}$, *where* $\mathbf{R}$ *is an arbitrary unitary matrix.*

One $m \times n$ RRI matrix with condition number β can be constructed as

$$\mathbf{\Phi} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} = \mathbf{U} \begin{bmatrix} \rho^0 & & & \mathbf{0} \\ & \ddots & & \mathbf{0} \\ & & \rho^{m-1} & \mathbf{0} \end{bmatrix} \mathbf{V}^{\mathrm{T}},$$

where $\beta = \rho^{m-1}$, $\mathbf{U}$ is an arbitrary $m \times m$ unitary matrix, and $\mathbf{V}$ is drawn from Haar measure.

Gallager-Gaussian (GG) Matrix

Definition

A $(w_c, w_r, \sigma_1^2, \sigma_0^2)$ *Gallager-Gaussian matrix* $\mathbf{\Phi}$ *is constructed from a* (w_c, w_r) *binary Gallager 'mask' matrix* $\mathbf{G}$*, then the entries are drawn independently from* $\Phi_{ij} \sim \mathcal{N}(0, \sigma^2_{G_{ij}})$.

An example of $(3, 4)$ Gallager matrix

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Gallager-Gaussian (GG) Matrix

Definition

A $(w_c, w_r, \sigma_1^2, \sigma_0^2)$ Gallager-Gaussian matrix $\mathbf{\Phi}$ *is constructed from a (w_c, w_r) binary Gallager 'mask' matrix* $\mathbf{G}$*, then the entries are drawn independently from $\Phi_{ij} \sim \mathcal{N}(0, \sigma^2_{G_{ij}})$.*

Let $p_{\mathrm{bg}} = \mathcal{N}(0, \sigma_0^2)$, and $p_{\mathrm{fg}} = \mathcal{N}(0, \sigma_1^2)$, where $\sigma_1^2 \gg \sigma_0^2$.

$$\mathbf{\Phi} \sim \begin{bmatrix} p_{\mathrm{bg}} & p_{\mathrm{fg}} & p_{\mathrm{fg}} & p_{\mathrm{bg}} & p_{\mathrm{fg}} & p_{\mathrm{bg}} & p_{\mathrm{fg}} & p_{\mathrm{bg}} \\ p_{\mathrm{fg}} & p_{\mathrm{bg}} & p_{\mathrm{bg}} & p_{\mathrm{fg}} & p_{\mathrm{bg}} & p_{\mathrm{fg}} & p_{\mathrm{bg}} & p_{\mathrm{fg}} \\ p_{\mathrm{bg}} & p_{\mathrm{fg}} & p_{\mathrm{fg}} & p_{\mathrm{bg}} & p_{\mathrm{fg}} & p_{\mathrm{fg}} & p_{\mathrm{bg}} & p_{\mathrm{bg}} \\ p_{\mathrm{fg}} & p_{\mathrm{bg}} & p_{\mathrm{bg}} & p_{\mathrm{fg}} & p_{\mathrm{bg}} & p_{\mathrm{bg}} & p_{\mathrm{fg}} & p_{\mathrm{fg}} \\ p_{\mathrm{fg}} & p_{\mathrm{fg}} & p_{\mathrm{bg}} & p_{\mathrm{bg}} & p_{\mathrm{bg}} & p_{\mathrm{fg}} & p_{\mathrm{fg}} & p_{\mathrm{bg}} \\ p_{\mathrm{bg}} & p_{\mathrm{bg}} & p_{\mathrm{fg}} & p_{\mathrm{fg}} & p_{\mathrm{fg}} & p_{\mathrm{bg}} & p_{\mathrm{bg}} & p_{\mathrm{fg}} \end{bmatrix}$$

RBP-SVD Algorithm

Consider problem

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{w}, \quad \text{with} \quad w_a \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_W^2), \quad \forall\, a \in \{1, \dots, m\}$$

When $\text{rank}(\mathbf{\Phi}) = r < m$, we can use the compact SVD

$$\mathbf{\Phi} = \underbrace{\begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_r \end{bmatrix}}_{m \times r} \underbrace{\begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_r \end{bmatrix}}_{r \times r} \underbrace{\begin{bmatrix} \mathbf{v}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{v}_r^{\mathrm{T}} \end{bmatrix}}_{r \times n} = \mathbf{USV}^T$$

RBP-SVD Algorithm

Consider problem

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{w}, \quad \text{with} \quad w_a \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_W^2), \quad \forall\, a \in \{1, \ldots, m\}$$

Multiplying $\mathbf{S}^{-1}\mathbf{U}^{\mathrm{T}}$ on both sides, we get

$$\underbrace{\mathbf{S}^{-1}\mathbf{U}^{\mathrm{T}}\mathbf{y}}_{\tilde{\mathbf{y}}} = \mathbf{S}^{-1}\mathbf{U}^{\mathrm{T}}\left(\mathbf{U}\mathbf{S}\mathbf{V}^{T}\right)\mathbf{x} + \mathbf{S}^{-1} \underbrace{\mathbf{U}^{\mathrm{T}}\mathbf{w}}_{\sim\mathcal{N}(\mathbf{0},\sigma_W^2 \mathbf{I}_r)},$$

where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma_W^2 \mathbf{I}_m)$ is rotationally invariant

RBP-SVD Algorithm

Consider problem

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{w}, \quad \text{with} \quad w_a \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_W^2), \quad \forall\, a \in \{1, \ldots, m\}$$

Now, effective measurement matrix $\mathbf{V}^{\mathrm{T}}$ is well-conditioned ($\kappa(V^T) = 1$). RBP-SVD is completed by running RBP on the following reformulated CS problem

$$\tilde{\mathbf{y}} = \mathbf{V}^{\mathrm{T}}\mathbf{x} + \tilde{\mathbf{w}}, \quad \text{where} \quad \tilde{\mathbf{w}} \sim \mathcal{N}\left(\mathbf{0}, \sigma_W^2 \mathbf{S}^{-2}\right)$$

Experimental Results

- Signal prior p_X is chosen to be Bernoulli-Gaussian and $\mathrm{Var}(X) = 1$,

$$p_{X_i}(x) \equiv p_X(x) = 0.9 \times \delta_0(x) + 0.1 \times \mathcal{N}\left(x \,|\, 0, 10\right).$$

  SNR ranges geometrically from 5dB to 40dB.

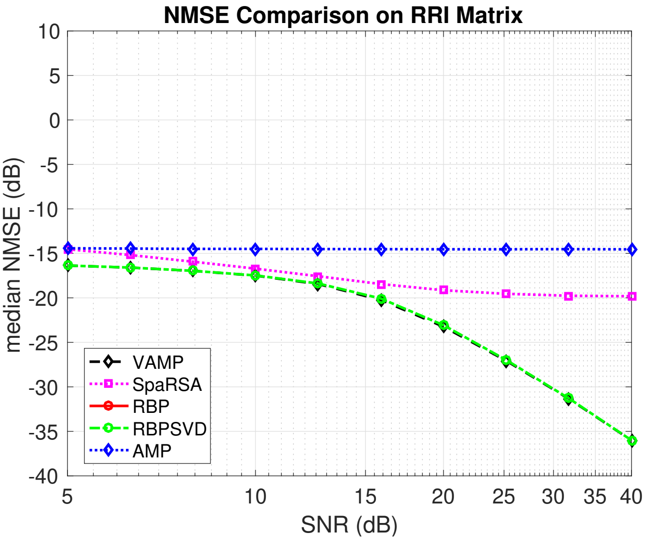
- All measurement matrices have size $m = 512$, $n = 1024$, normalized so that Frobenius norm is n.

- All algorithms have max iteration $T = 30$ and damping $\alpha = 0.95$,

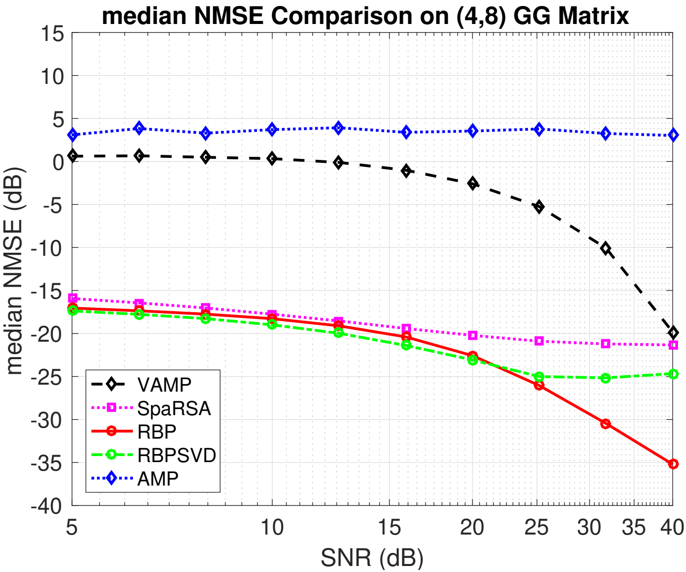
$$\mu^{(t)} = \alpha\, \mu_{\mathrm{new}} + (1 - \alpha)\, \mu^{(t-1)}.$$

- Each point is computed over 500 random matrices and 10 random signals per matrix.

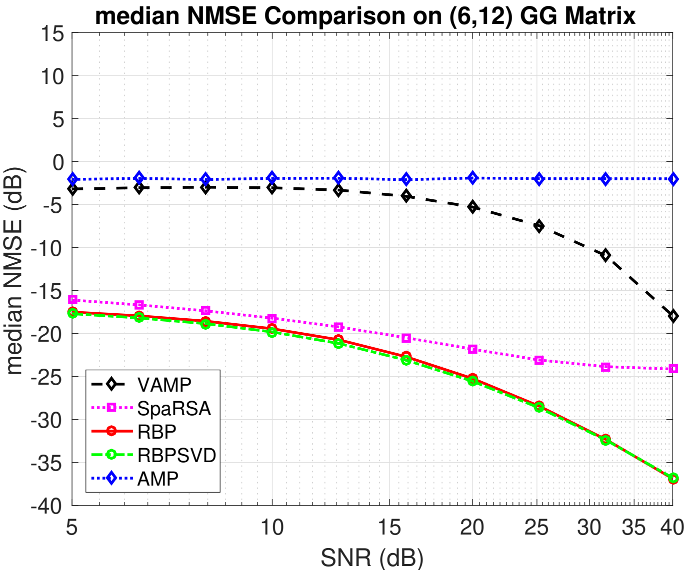
RRI Matrices with $\kappa(\mathbf{\Phi}) = 100$



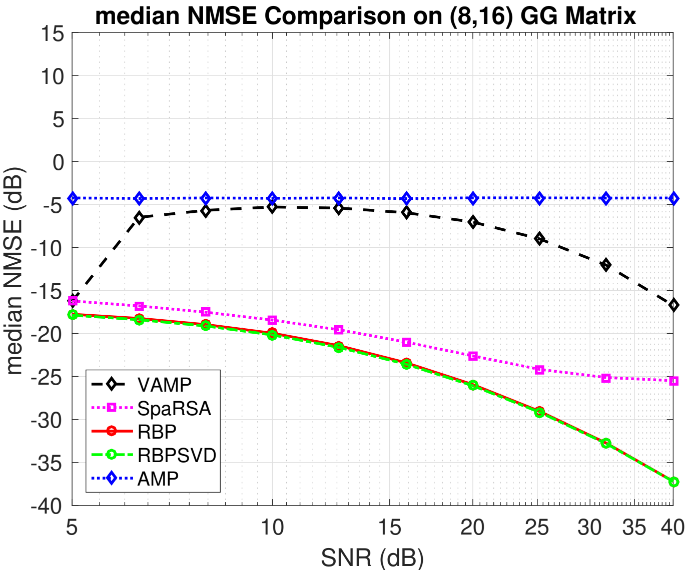
$(w_c = 4, w_r = 8)$ GG Matrices



$(w_c = 6, w_r = 12)$ GG Matrices



$(w_c = 8, w_r = 16)$ GG Matrices



Outline

- Background on Compressive Sensing (CS)
- Belief-Propagation (BP) and iterative reconstruction algorithms
- Families of measurement matrices
- RBP-SVD and Numerical experiments
- RBP-SEQ and Numerical experiments

Sparse-Matrix-Product (SMP) Matrix

Sometimes a dense matrix can be decomposed into sequential product of several sparse matrices, one can utilize the sparsity of the factor matrices to achieve more efficient computation.

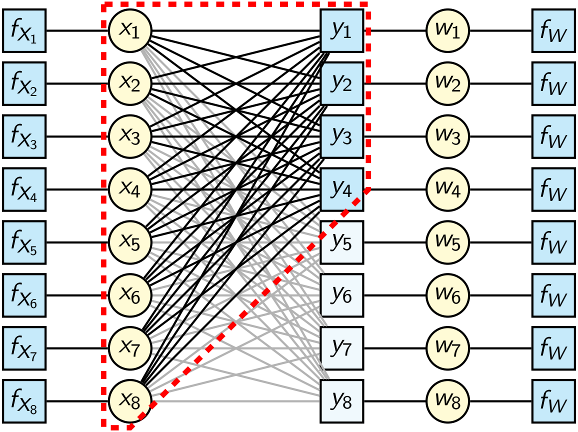
A (K, δ)-SMP matrix $\mathbf{\Phi}$ is constructed by K component matrices $\mathbf{F}^{(1)}, \ldots, \mathbf{F}^{(K)}$, where δ is the fraction of nonzero entries in $\mathbf{F}^{(k)}$ s.t.

$$\mathbf{\Phi} = \mathbf{F}^{(K)}\mathbf{F}^{(K-1)} \cdots \mathbf{F}^{(1)} \quad \textit{and} \quad \mathrm{nnz}(\mathbf{\Phi}) \gg \sum_{k=1}^{K} \mathrm{nnz}(\mathbf{F}^{(k)}),$$

where the nonzero entries in $\mathbf{F}^{(k)}$ are drawn independently from $\mathcal{N}(0, 1)$, and $\mathrm{nnz}(\cdot)$ stands for the number of nonzero entries of a matrix.

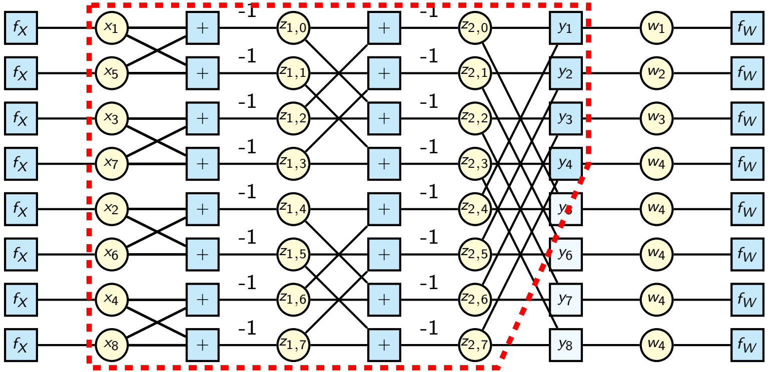
RBP-SEQ Algorithm Inspiration

Suppose the measurement matrix is the first m rows of $\mathbf{F}_n$.



n VN, m FN, nm edges. Complexity $O(n^2)$

Using butterfly diagram to compute FWHT:



$n\log_2(n)$ VN, $n\log_2(n) - n + m$ FN, $3n(\log_2(n) - 1) + 2m$ edges

Complexity $O(n\log(n))$

RBP-SEQ Algorithm

For SMP matrices we have sparse component matrices $\mathbf{F}^{(k)}$ for $k \in \{1, \ldots, K\}$ and

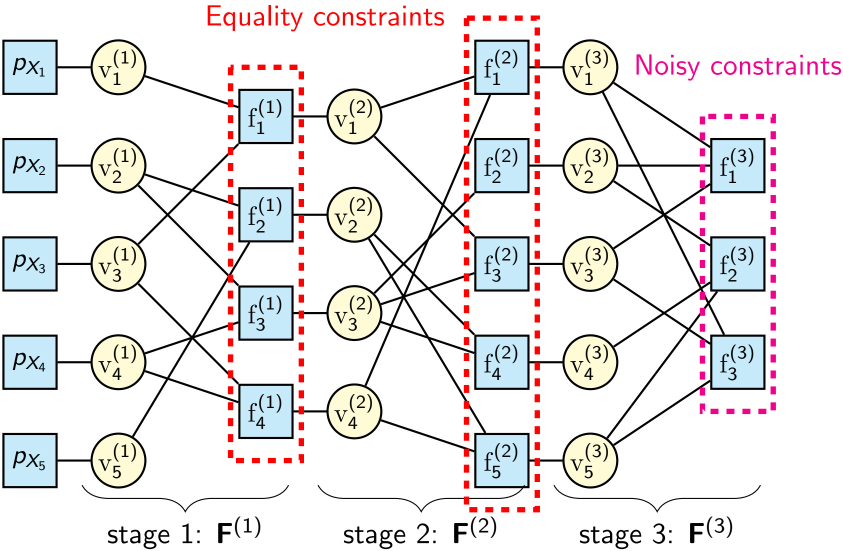
$$\mathbf{\Phi} = \mathbf{F}^{(K)}\mathbf{F}^{(K-1)} \cdots \mathbf{F}^{(1)} \quad \textit{and} \quad \mathrm{nnz}(\mathbf{\Phi}) \gg \sum_{k=1}^{K} \mathrm{nnz}(\mathbf{F}^{(k)}).$$

Similarly, the complexity is $O(\mathrm{nnz}(\mathbf{\Phi}))$ for standard RBP and $O\left(\sum_{k=1}^{K} \mathrm{nnz}(\mathbf{F}^{(k)})\right)$ for RBP-SEQ.
Denote $\tilde{\mathbf{x}}^{(1)} = \mathbf{x}$ and $\tilde{\mathbf{x}}^{(k+1)} = \mathbf{F}^{(k)}\tilde{\mathbf{x}}^{(k)}$, we have K subproblems:

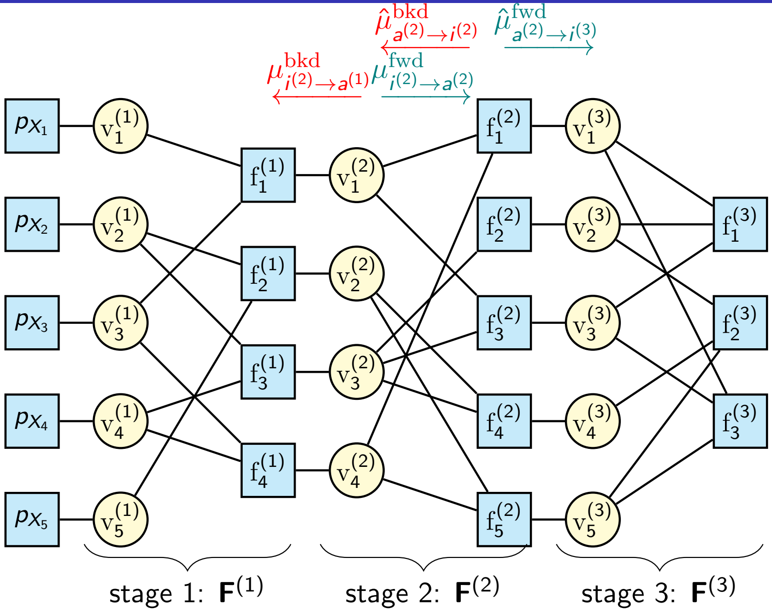
$$\tilde{\mathbf{y}}^{(k)} = \mathbf{0}_{n_{k+1}\times 1} = \begin{bmatrix}\mathbf{F}^{(k)} & -\mathbf{I}\end{bmatrix} \begin{bmatrix}\tilde{\mathbf{x}}^{(k)} \\ \tilde{\mathbf{x}}^{(k+1)}\end{bmatrix}, \qquad \forall\, k \in \{1, \ldots, K-1\}$$

$$\tilde{\mathbf{y}}^{(K)} = \mathbf{y} = \mathbf{F}^{(K)}\tilde{\mathbf{x}}^{(K)} + \mathbf{w}$$

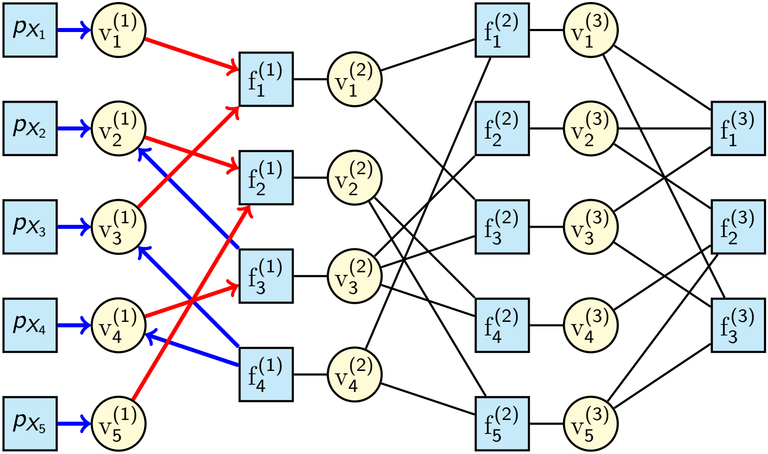
RBP-SEQ Algorithm



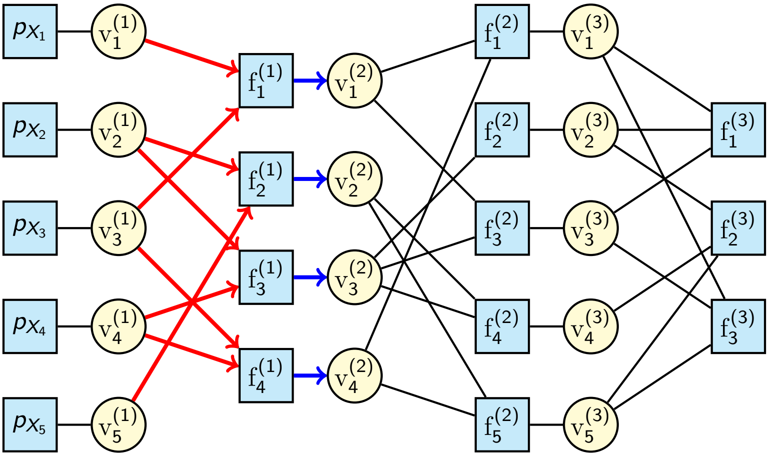
RBP-SEQ Algorithm



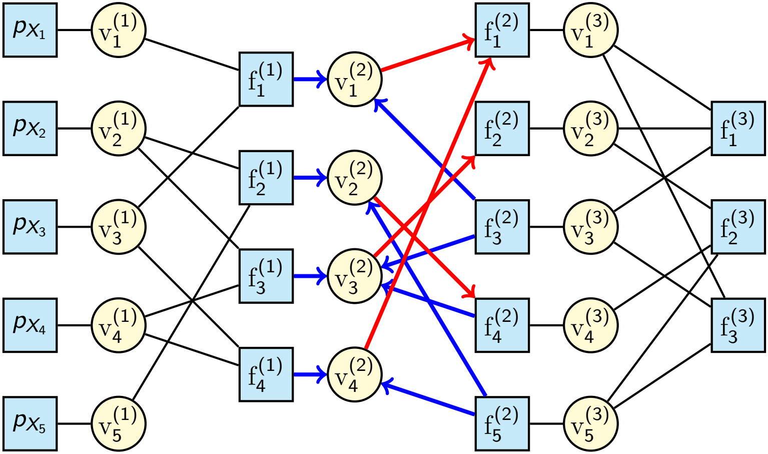
RBP-SEQ Algorithm



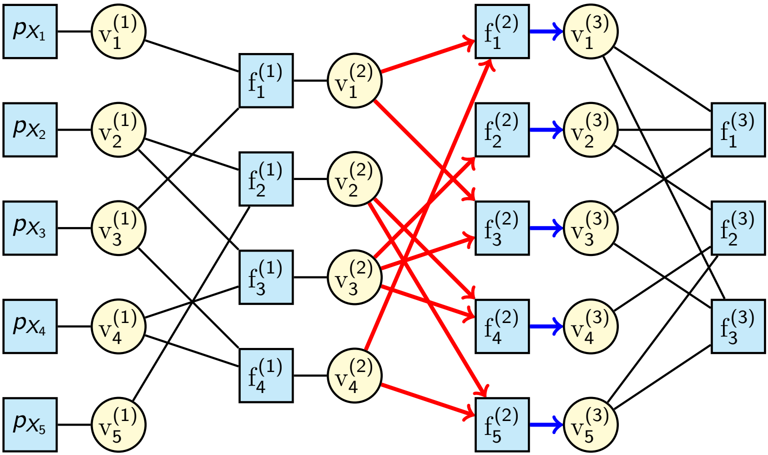
RBP-SEQ Algorithm



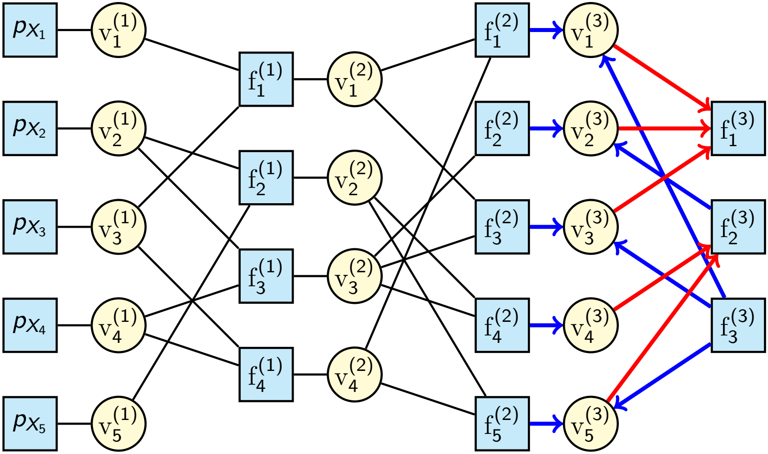
RBP-SEQ Algorithm



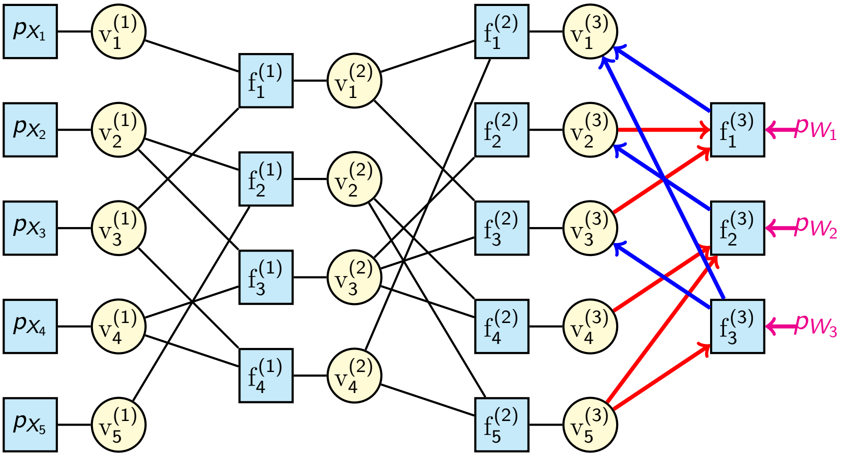
RBP-SEQ Algorithm



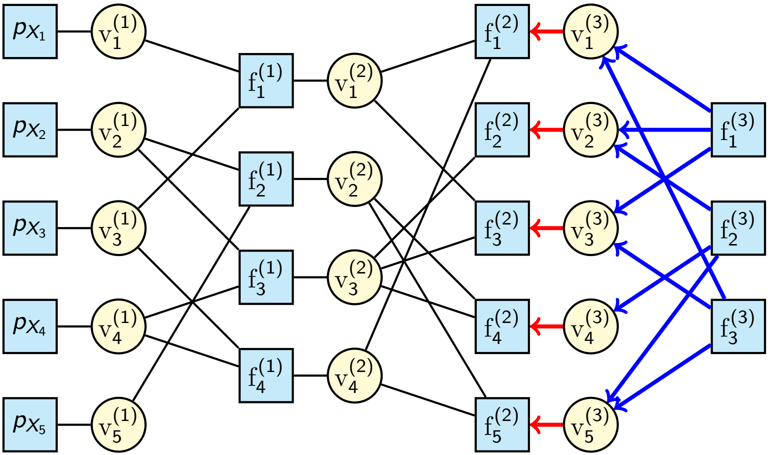
RBP-SEQ Algorithm



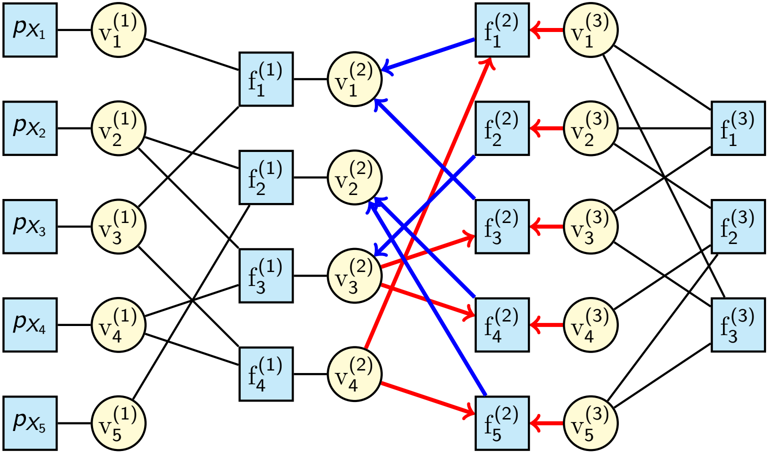
RBP-SEQ Algorithm



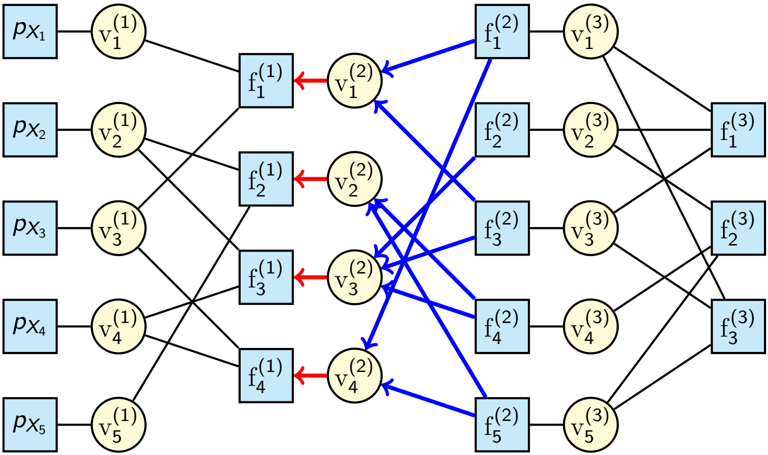
RBP-SEQ Algorithm



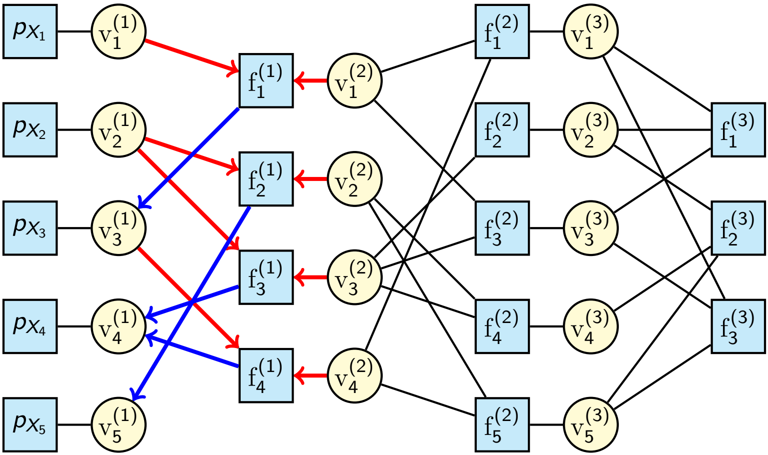
RBP-SEQ Algorithm



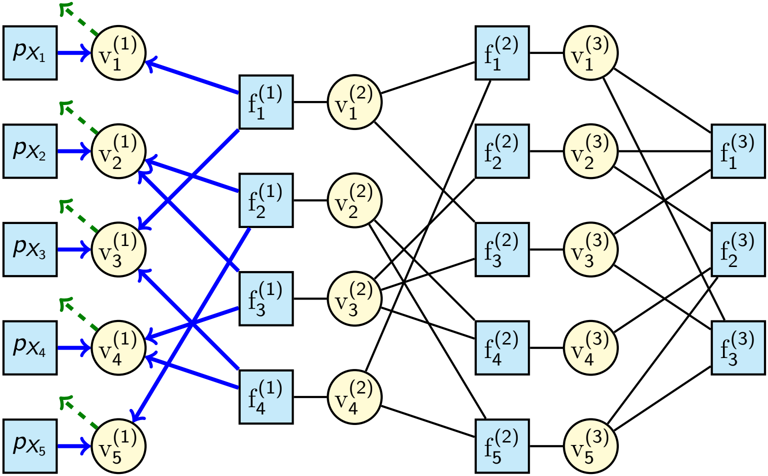
RBP-SEQ Algorithm



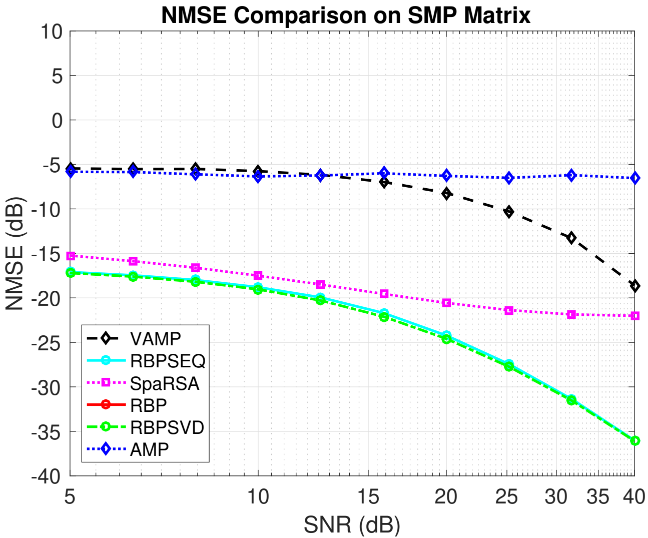
RBP-SEQ Algorithm



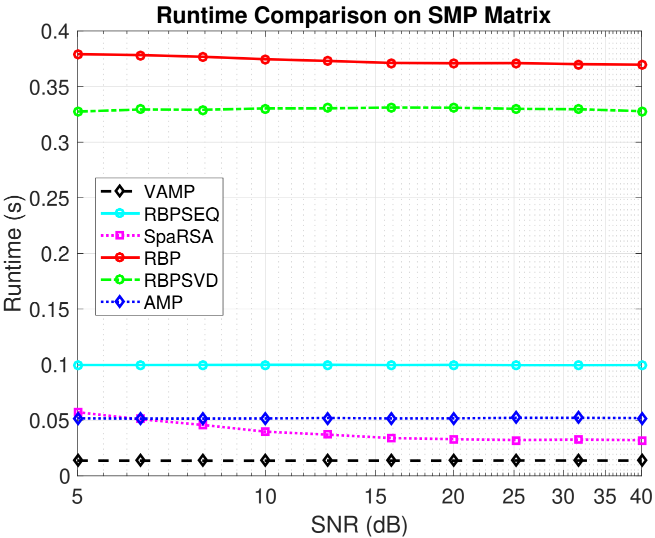
RBP-SEQ Algorithm



$(K = 2, \delta = 0.08)$ SMP Matrices



$(K = 2, \delta = 0.08)$ SMP Matrices



Comparisons and Conclusions

	RRI NMSE	GG NMSE	SMP NMSE	SMP Speed
SpaRSA [WNF09]	middle	middle	middle	fast
AMP	not converge	not converge	not converge	fast
VAMP	low	high	high	fast
RBP	diverge	low	diverge	slow
RBP-SVD	low	low	low	slow
RBP-SEQ	N/A	N/A	low	middle

RBP and AMP do not converge when the matrix is ill-conditioned.

Comparisons and Conclusions

	RRI NMSE	GG NMSE	SMP NMSE	SMP Speed
SpaRSA [WNF09]	middle	middle	middle	fast
AMP	not converge	not converge	not converge	fast
VAMP	low	high	high	fast
RBP	diverge	low	diverge	slow
RBP-SVD	low	low	low	slow
RBP-SEQ	N/A	N/A	low	middle

RBP-SVD and standard RBP both work decently in the metric median NMSE, but RBP has much larger mean NMSE, indicating RBP-SVD is more stable.

Comparisons and Conclusions

	RRI NMSE	GG NMSE	SMP NMSE	SMP Speed
SpaRSA [WNF09]	middle	middle	middle	fast
AMP	not converge	not converge	not converge	fast
VAMP	low	high	high	fast
RBP	diverge	low	diverge	slow
RBP-SVD	low	low	low	slow
RBP-SEQ	N/A	N/A	low	middle

RBP-SEQ takes advantage of the stage-wise sparse structure, thereby reducing the computation complexity. For SMP matrices, RBP diverges, RBP-SEQ outperforms VAMP and behaves similar to RBP-SVD but three times faster.

References

[BSB10] D. Baron, S. Sarvotham, R. G. Baraniuk.
Bayesian compressive sensing via belief propagation.
IEEE Transactions on Signal Processing, 58(1):269–280, 2010.

[GW06] Dongning Guo, Chih-Chun Wang.
Asymptotic Mean-Square Optimality of Belief Propagation for Sparse Linear Systems.
IEEE Information Theory Workshop, strony 194–198. IEEE, 2006.

[MM10] Arian Maleki, Andrea Montanari.
Analysis of approximate message passing algorithm.
Conference on Information Sciences and Systems, 2010.

[SRF16] Philip Schniter, Sundeep Rangan, Alyson K Fletcher.
Vector approximate message passing for the generalized linear model.
arXiv preprint arXiv:1612.01186, 2016.

[WNF09] Stephen J. Wright, Robert D. Nowak, Mário A T Figueiredo.
Sparse reconstruction by separable approximation.
IEEE Transactions on Signal Processing, 57(7):2479–2493, 2009.

Thanks.