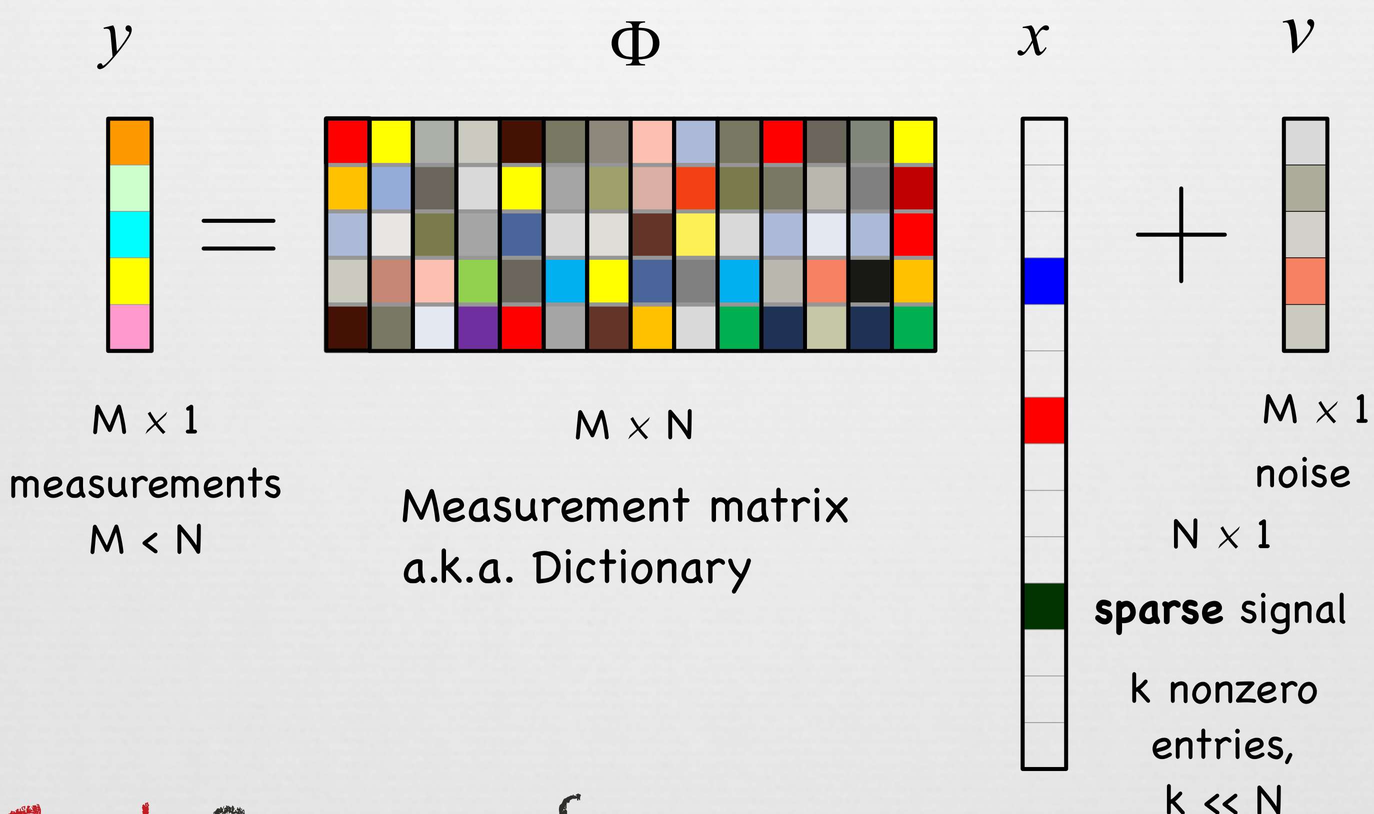
Sparse Bayesian Learning via Approximate Message Passing

Chandra R. Murthy

cmurthy@ece.iisc.ernet.in

Sparse Signal Recovery



- **Goal:** Recover x from y
- M << N: infinitely many solutions

Compressed Sensing

- Deals with two main questions:
  - Design of sensing matrices with recovery guarantees

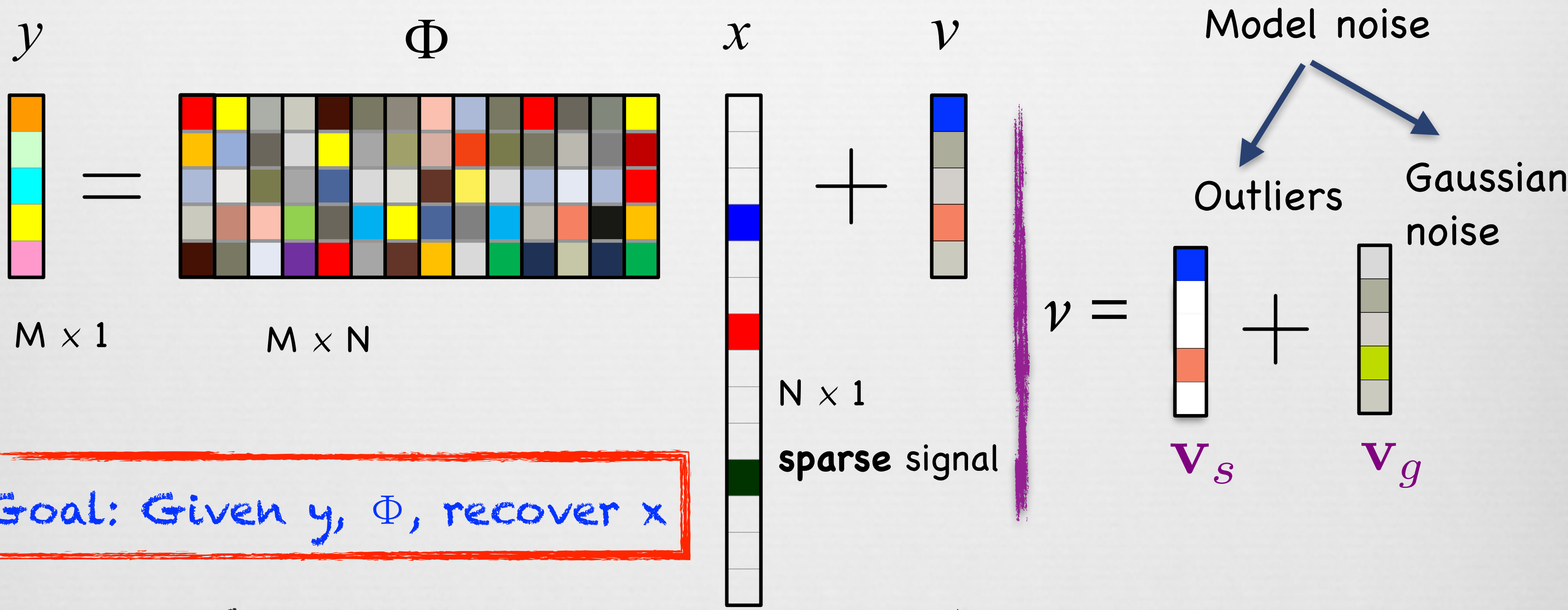
$$\Phi_{M\times N} = \mathbf{A}_{M\times N}\Psi_{N\times N}$$

Sparsifying Basis

  - Computationally efficient recovery

- Our focus: sparse signal recovery from noisy linear underdetermined measurements

Robust Linear Regression: Underdetermined Case



Goal: Given y, Φ, recover x

- Transform into an overcomplete problem:

$$\mathbf{Y} = \mathbf{\Phi x} + \mathbf{\Psi v}_s + \mathbf{v}_g, \quad \text{where } \mathbf{\Psi} = \mathbf{I}$$

$$\text{or} \quad \mathbf{Y} = [\mathbf{\Phi}, \ \mathbf{\Psi}] \begin{bmatrix} \mathbf{x} \\ \mathbf{v}_s \end{bmatrix} + \mathbf{v}_g$$

Sparse recovery algos are now applicable!

Robust Linear Regression: Overdetermined Case

- Measurement model:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{E} + \mathbf{e}$$

$M \times N;$ $M \geq N$ — Outliers; sparse — Noise

- Use SVD: $\mathbf{A} = \mathbf{U}_1 \Sigma \mathbf{V}_1^T;\ \mathbf{U}_2^T \mathbf{A} = \mathbf{0}$

- Processed measurements:

$$\tilde{\mathbf{y}} = \mathbf{U}_2^T \mathbf{y} = \mathbf{U}_2^T \mathbf{E} + \mathbf{U}_2^T \mathbf{e}$$

- Can now directly apply sparse signal recovery algorithms to estimate and remove outliers!

The Problem

- Noiseless case: Given y and Φ, solve

$$\min \|\mathbf{x}\|_0 \text{ subject to } \mathbf{y} = \mathbf{\Phi x}$$

- Noisy case: solve

$$\min \|\mathbf{x}\|_0 \text{ subject to } \|\mathbf{y} - \mathbf{\Phi x}\|_2 \leq \beta$$

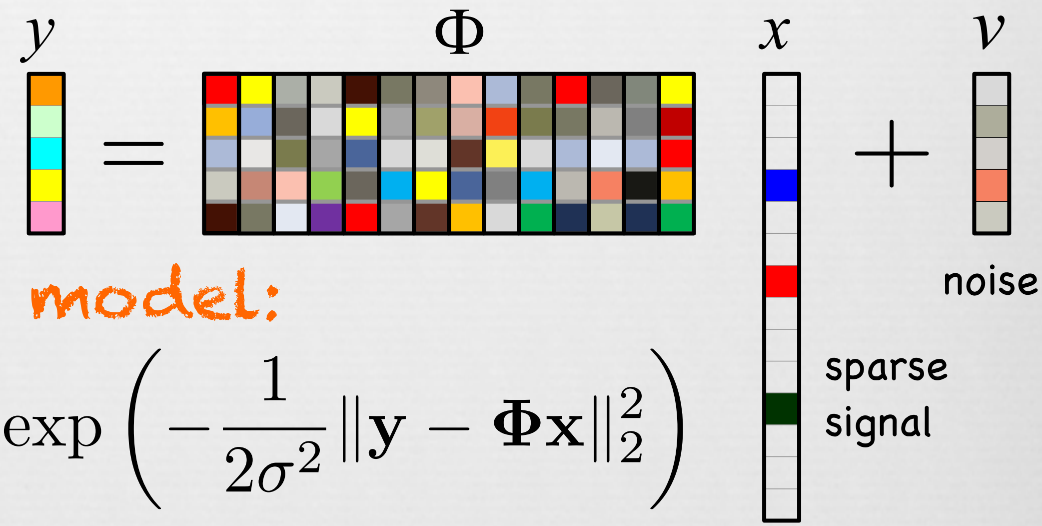
- l_0 norm minimization
  - Combinatorial complexity
  - Not robust to noise

Sparse Bayesian Learning

Use lots of priors and pick the best one!

Sparse Bayesian Learning

- Canonical model



- Gaussian noise model:

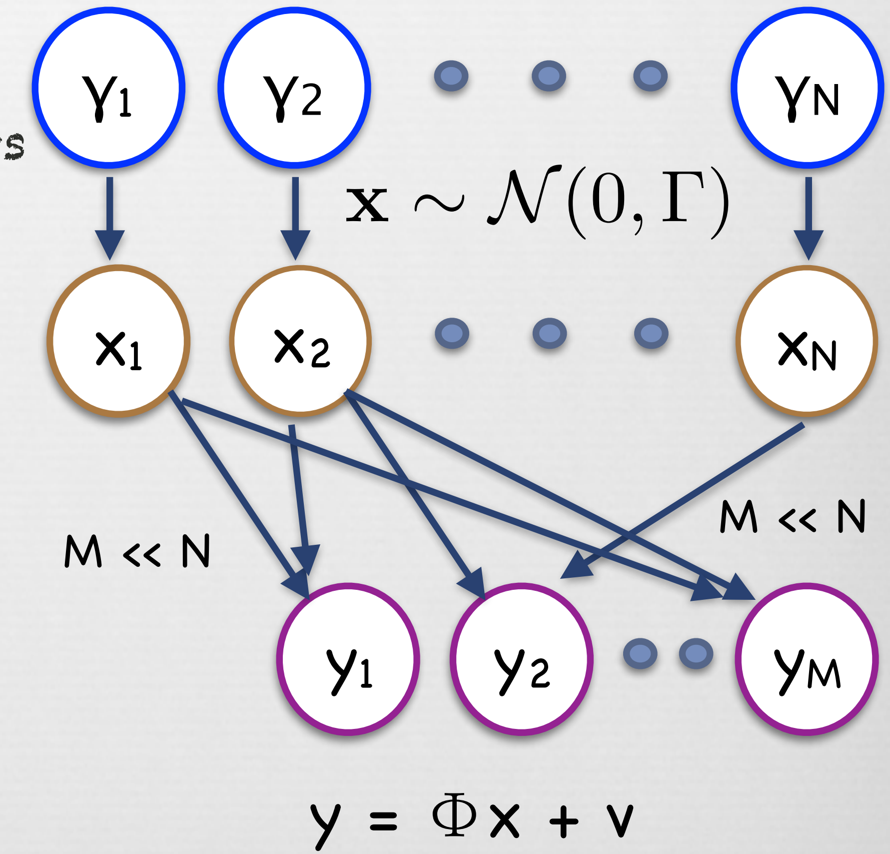
$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{\Phi x}\|_2^2\right)$$

- Parameterized Gaussian prior:

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

Graphical Model

- Markov chain: γ -> x -> y
- γ: nonnegative hyperparameters
- Potential advantages:
  - Given γ, $p(x|y;\gamma)$ is Gaussian: easy to find point estimates
  - Averaging over x -> fewer local minima in $p(\gamma|y)$
  - γ can be used to tie parameters together: fewer params. to estimate



Hierarchical Bayesian Framework

- First, estimate hyperparameters: $\hat{\gamma} = \arg\max_{\gamma} p(\gamma|\mathbf{y})$
  - γ : deterministic and unknown, or random with hyperprior distbn.
- Then, find posterior distribution p(x|y;$\hat{\gamma}$)

$$p(\mathbf{x}|\mathbf{y};\hat{\gamma}) = \mathcal{N}(\mu_x, \Sigma_x)$$

$$\mu_x = \hat{\Gamma}\Phi^T\left(\Phi\hat{\Gamma}\Phi^T + \lambda\mathbf{I}\right)^{-1}\mathbf{y}$$

$$\Sigma_x = \hat{\Gamma} - \hat{\Gamma}\Phi^T\left(\Phi\Gamma\Phi^T + \lambda\mathbf{I}\right)^{-1}\Phi\hat{\Gamma}$$

- For point estimates: e.g., posterior mean: $\mathbb{E}\left(\mathbf{x}|\mathbf{y};\hat{\gamma}\right)$

Sparse Bayesian Methods

- Estimate γ_i from the data: Type-II ML

$$\mathcal{L}(\Gamma) = \log p(\mathbf{y};\Gamma) = \log \int p(\mathbf{y}|\mathbf{x};\Gamma)p(\mathbf{x};\Gamma)\mathrm{d}\mathbf{x}$$

$$p(\mathbf{y};\Gamma) = \mathcal{N}\left(0, \underbrace{\sigma^2\mathbf{I} + \Phi\Gamma\Phi^T}_{\Sigma_{\mathbf{y}}}\right)$$

- When γ is random: can find MAP estimates
  - Just add $\sum_{i=1}^{N} \log p(\gamma_i)$ term to log likelihood fn
- SBL cost function: $\mathcal{L}(\Gamma) \propto -\log\det(\Sigma_{\mathbf{y}}) - \mathbf{y}^T\Sigma_{\mathbf{y}}^{-1}\mathbf{y}$

Optimization via EM

- Log likelihood of the complete data

$$-\log p(\mathbf{y},\mathbf{x};\gamma) = \frac{\|\mathbf{y}-\mathbf{\Phi}\mathbf{x}\|_2^2}{2\sigma^2} + \frac{1}{2}\left[\sum_{i=1}^{N}\frac{x_i^2}{\gamma_i} + \log\gamma_i\right] - \sum_{i=1}^{N}\log p(\gamma_i)$$

$-\log p(\mathbf{y}|\mathbf{x};\gamma)$ indep. of γ

$-\log p(\mathbf{x};\gamma)$ func. of γ

Facilitates type-II algorithms

- E-Step: compute "Q-function"

$$Q\left(\Gamma|\Gamma^{(t)}\right) = \mathbb{E}_{\mathbf{x}|\mathbf{y};\Gamma^{(t)}}\left[-\log p(\mathbf{y},\mathbf{x};\Gamma)\right]$$

$$\doteq \sum_{i=1}^{N}\frac{\mathbb{E}(x_i^2|\mathbf{y};\Gamma^{(t)})}{\gamma_i} + \log\gamma_i$$

from previous iteration

- Easy to compute: $p(x_i|\mathbf{y};\Gamma^{(t)})$ is Gaussian

The EM Iterations

- **E-step (continued):** $p\left(\mathbf{x}|\mathbf{y};\Gamma^{(t)}\right) = \mathcal{N}(\mu, \Sigma)$

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

- **M-step:** maximize $Q(\Gamma|\Gamma^{(t)})$ given posteriors gathered in the E-step:

$\mathbb{E}(x_i^2|\mathbf{y};\Gamma^{(t)})$

$$\Gamma^{(t+1)} = \arg\max_{\gamma_i \geq 0} Q\left(\Gamma|\Gamma^{(t)}\right) = \boxed{\mathrm{diag}\left(\mu_i^2 + \Sigma_{ii}\right)}$$

- Component-wise updates

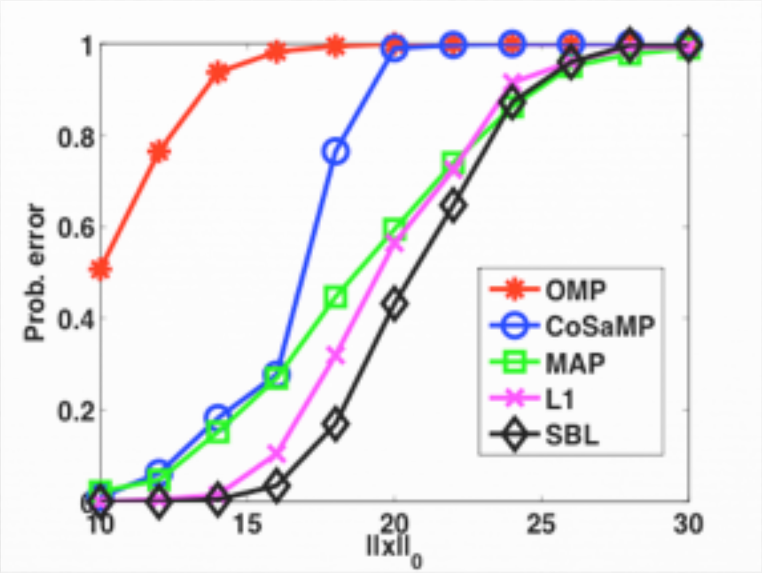
Can recover type-I methods by treating γ as hidden and taking expectation over γ instead of x

The SBL Algorithm

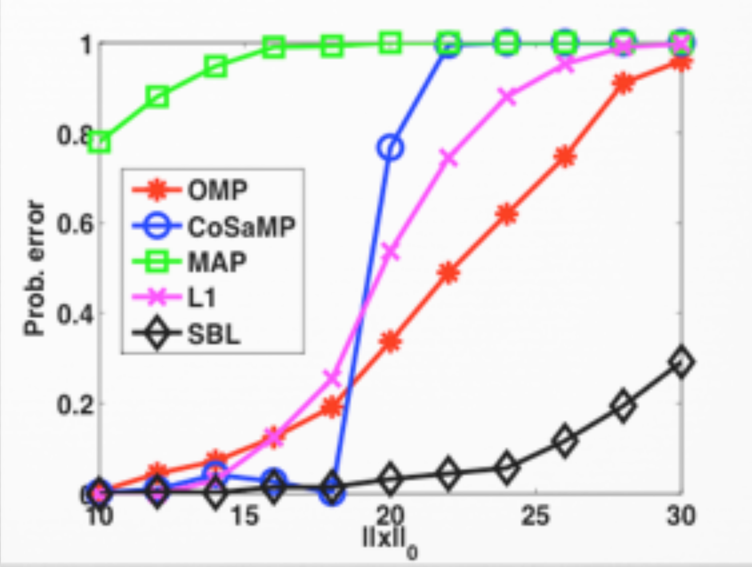
1. Initialize $\Gamma = I$
2. Compute
$$\mu = \sigma^{-2}\left(\sigma^{-2}\Phi^T\Phi + \left(\Gamma^{(t)}\right)^{-1}\right)^{-1}\Phi^T\mathbf{y}$$
$$\Sigma = \left(\sigma^{-2}\Phi^T\Phi + \left(\Gamma^{(t)}\right)^{-1}\right)^{-1}$$
3. Update $\Gamma^{(t+1)} = \mathrm{diag}\left(\mu_i^2 + \Sigma_{ii}\right)$
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



Unit magnitude entries



Highly scaled entries

Approximate Message Passing

- AMP [Donoho, Maleki, Montanari 09]:
  - Uses loopy belief propagation + Gaussian approximations to solve LASSO
  - Key advantage: low complexity
- In SBL:
  - All Gaussian PDFs: approximation is not necessary
  - Only need to track means and variances
  - Can replace computationally expensive E-step with the AMP based iterations

Factor Graph

- In the E-Step, we're after

$$p\left(\mathbf{x}|\mathbf{y};\Gamma^{(t)}\right) \propto p(\mathbf{y}|\mathbf{x})p\left(\mathbf{x};\Gamma^{(t)}\right)$$

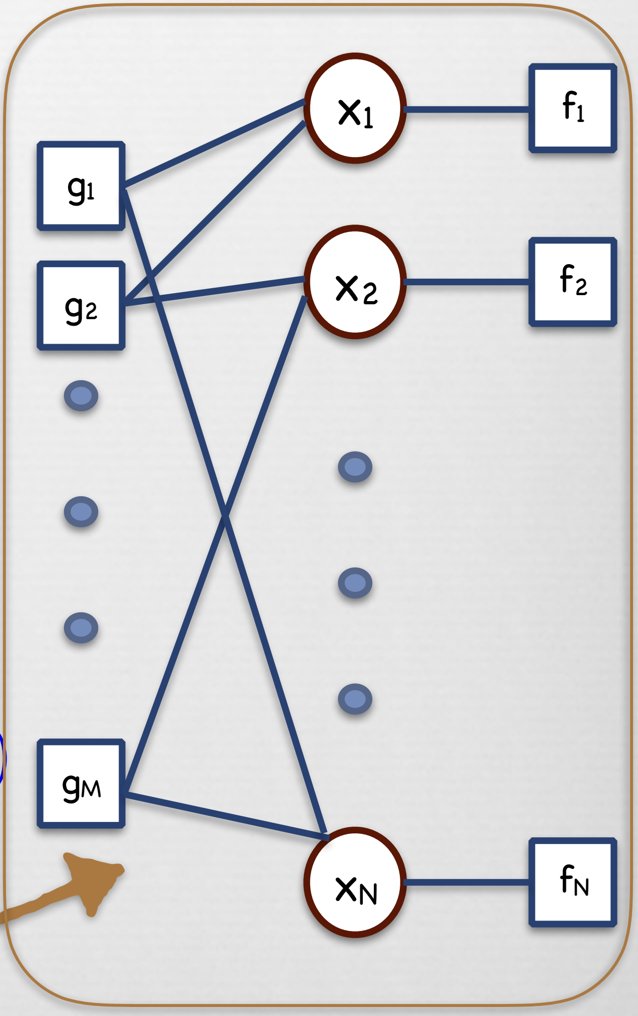
$$\propto \prod_{m=1}^{M} p(y_m|\mathbf{x}) \prod_{n=1}^{N} p(x_n;\gamma_n^{(t)})$$

- And we define

$$g_m(\mathbf{x}) \triangleq p(y_m|\mathbf{x}) = \mathcal{N}(y_m;\Phi_m^H\mathbf{x},\sigma^2)$$

$$f_n(x_n) \triangleq p(x_n;\gamma_n) = \mathcal{N}(x_n;0,\gamma_n)$$

- To get the factor graph



AMP-SBL

- General form of updates:

$$\hat{\mathbf{x}}^{t+1} = \eta_t\left(\Phi^H \mathbf{z}^t + \hat{\mathbf{x}}^t\right)$$

$$\mathbf{z}^t = \mathbf{y} - \Phi\hat{\mathbf{x}}^t + \boxed{\frac{1}{\delta}\mathbf{z}^{t-1}\langle {\eta'}_{t-1}\left(\Phi^H\mathbf{z}^{t-1} + \hat{\mathbf{x}}^{t-1}\right)\rangle}$$

Message passing term

- η_t: soft-thresholding function – linear for SBL

- O(M+N) msg updates: low computational cost!

Definitions:

$$F_n(K_n, c) = K_n\left(\frac{\gamma_n}{c+\gamma_n}\right)$$

$$G_n(K_n, c) = \frac{c\gamma_n}{c+\gamma_n}$$

$$F_n'(K_n, c) = \frac{\gamma_n}{c+\gamma_n}$$

Message Updates:

$$K_n = \sum_{m=1}^{M}\Phi_{mn}^* z_m + \mu_n$$

$$\mu_n = F_n(K_n, c)$$

$$v_n = G_n(K_n, c)$$

$$c = \sigma^2 + \frac{1}{M}\sum_{n=1}^{N} v_n$$

$$z_m = y_m - \sum_{n=1}^{N}\Phi_{mn}\mu_n + \frac{z_m}{M}\sum_{n=1}^{N}F_n'(\mu_n, c)$$

Parameter Update/M-Step:

$$\gamma_n = v_n + \mu_n^2$$

Empirical Example

- N = 200, M = 100, K = 20, Gaussian measurement matrix