Pattern-Coupled Sparse Bayesian Learning for Recovery of Block-Sparse Signals

Geethu Joseph

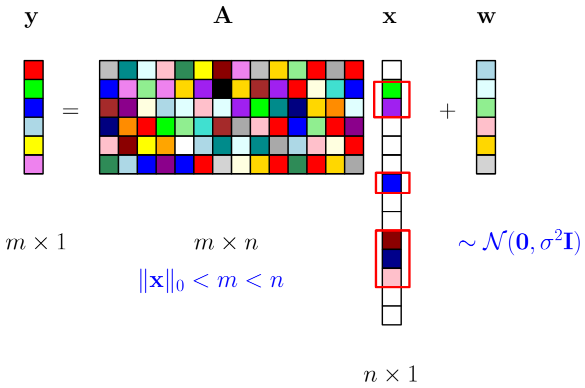
SPC Lab, IISc

July 18, 2015

Paper Reviewed

- Title: Pattern-Coupled Sparse Bayesian Learning for Recovery of Block-Sparse Signals
- Authors: Jun Fang, Yanning Shen, Hongbin Li, and Pu Wang
- Publication date: January 15, 2015
- Journal name: IEEE Transactions on Signal Processing, vol. 63, no. 2, pp. 360-372

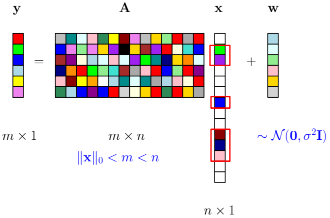
Block-Sparse Signal Recovery Problem



- **Goal**: Recover block-sparse vector$\boldsymbol{x}$ from $\boldsymbol{y}$
- Unknown block-sparsity structure

Sparse Bayesian Learning

- Impose a fictitious sparsity inducing prior on x



$$x \sim \mathcal{N}(0, \Gamma)$$
$$\Gamma = \mathrm{diag}\{\gamma_1, \gamma_2, \ldots, \gamma_n\}$$

E-step: $Q(\boldsymbol{\Gamma}|\boldsymbol{\Gamma}^{(r)}) = E_{\mathbf{x}|\mathbf{y};\boldsymbol{\Gamma}^{(r)}} \log p(\mathbf{y}, \mathbf{x}; \boldsymbol{\Gamma})$

Iterate

M Step: $\boldsymbol{\Gamma}^{(r+1)} = \arg\max Q(\boldsymbol{\Gamma}|\boldsymbol{\Gamma}^{(r)})$

ML estimate of $\boldsymbol{\Gamma}$

$$\hat{\mathbf{x}} = E(\mathbf{x}|\mathbf{y}; \boldsymbol{\Gamma})$$

Pattern-Coupled Hierarchical Model

- Sparsity patterns of neighboring coefficients are statistically dependent
- Parameters:
  - $\boldsymbol{\alpha}$: hyperparameters associated with co-efficients
    - As $\alpha_i \to \infty$, then $x_i \to 0$
  - β: pattern relevance between neighboring coefficients

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{\alpha}) &\sim \prod_{i=1}^{n} p(x_i|\alpha_i, \alpha_{i+1}, \alpha_{i-1}) \\ p(x_i|\alpha_i, \alpha_{i+1}, \alpha_{i-1}) &= \mathcal{N}\left(x_i|0, (\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1})^{-1}\right) \end{aligned}$$

- Assume $\alpha_0 = \alpha_{n+1} = 0$

Pattern-Coupled Hierarchical Model

- Sparsity patterns of neighboring coefficients are statistically dependent

$$
\begin{aligned}
p(\boldsymbol{x}|\boldsymbol{\alpha}) &\sim \prod_{i=1}^{n} p(x_i|\alpha_i, \alpha_{i+1}, \alpha_{i-1}) \\
p(x_i|\alpha_i, \alpha_{i+1}, \alpha_{i-1}) &= \mathcal{N}\left(x_i|0, (\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1})^{-1}\right)
\end{aligned}
$$

- Gamma distribution over hyperparameters

$$
p(\boldsymbol{\alpha}) = \prod_{i=1}^{n} \text{Gamma}(\alpha_i|a, b)
$$

Some Insights

- Model:
  $p(x_i|\alpha_i, \alpha_{i+1}, \alpha_{i-1}) = \mathcal{N}\left(x_i|0, (\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1})^{-1}\right)$
- As $\alpha_i \to \infty$, then $x_i \to 0$
- Sporadic errors are reduced and consecutive errors are much unlikely
  - Nonzero to zero misidentification drives the associated hyperparameter to ∞
  - Zero to nonzero misidentification is reduced as either one of its neighboring hyperparameters goes to ∞
- Flexible to accommodate conventional sparse signals
  - If $x_i \neq 0$ is an isolated nonzero coefficient, $\{\alpha_i, \alpha_{i\pm1}\}$ have finite values and $\{\alpha_{i\pm2}\}$ becomes ∞
- Associating multiple neighbor parameters could lead to excessive coupling

Proposed Bayesian Approach

- Assume noise varianceσ^2 is known
- Posterior distribution $p(\boldsymbol{x}|\boldsymbol{\alpha}, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\phi})$

$$
\begin{aligned}
\boldsymbol{\mu} &= \sigma^{-2}\boldsymbol{\phi}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{y} \\
\boldsymbol{\phi} &= \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} + \sigma^2\boldsymbol{D}\right)^{-1} \\
\boldsymbol{D} &= \mathrm{diag}\left\{\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1}\right\}_{i=1}^{n}
\end{aligned}
$$

- MAP estimate of sparse vector: $\hat{\boldsymbol{x}}_{\mathrm{MAP}} = \boldsymbol{\mu}$
- **Problem**: Estimate the set of n hyperparameters $\boldsymbol{\alpha}$
  - Use EM formulation with $\boldsymbol{x}$ as hidden variable

EM Algorithm

- E-step

$$Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)}) = \sum_{i=1}^{n} \Bigg(a \log \alpha_i - b\alpha_i + \frac{1}{2} \log\left(\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1}\right) - \frac{1}{2}\left(\alpha_i + \beta\alpha_{i+1} + \beta\alpha_{i-1}\right)\left(\hat{\mu}_i^2 + \hat{\phi}_{i,i}\right)\Bigg)$$

  - $\boldsymbol{\mu}$ and $\boldsymbol{\phi}$ are mean and covariance of $\boldsymbol{x}$ computed using $\boldsymbol{\alpha}^{(t)}$
- M-step: No closed form expression for $\boldsymbol{\alpha}$
  - Gradient descent methods are computationally intense
  - At the optimal point $\alpha_i^* \in \left[\frac{a}{0.5c_i+b}, \frac{a+1.5}{0.5c_i+b}\right]$
    - $c_i = \left(\hat{\mu}_i^2 + \hat{\phi}_{i,i}\right) + \beta\left(\hat{\mu}_{i+1}^2 + \hat{\phi}_{i+1,i+1}\right) + \beta\left(\hat{\mu}_{i-1}^2 + \hat{\phi}_{i-1,i-1}\right)$
  - Choose sub-optimal solution

$$\hat{\alpha}_i = \frac{a}{0.5c_i + b}$$

  - Update rule gives negative feedback when α_i is large

Algorithm

- Input: $\{\boldsymbol{y}, \boldsymbol{A}, \sigma^2\}$
- Parameters: $\{a, b, \beta, \tau, \epsilon\}$
- At iteration t
  - Update hyperparamters:$\hat{\alpha}_i^{(t)} = \begin{cases} \frac{a}{0.5c_i^{(t)}+b} & \text{if} \hat{\alpha}_i^{(t)} < \tau \\ 10^8 & \text{if} \hat{\alpha}_i^{(t)} \geq \tau \end{cases}$
  - Compute $\hat{\boldsymbol{\mu}}^{(t)}$ and $\hat{\boldsymbol{\phi}}^{(t)}$ using $\boldsymbol{\alpha}^{(t)}$
  - MAP estimate of sparse vector:$\hat{\boldsymbol{x}}^{(t)} = \hat{\boldsymbol{\mu}}^{(t)}$
  - Continue until $\|\hat{\boldsymbol{x}}^{(t)} - \hat{\boldsymbol{x}}^{(t-1)}\|_2 \leq \epsilon$
- Output: $\hat{\boldsymbol{x}}^{(t)}$

Choice of Parameters

- Choice of a is not critical: Stable recovery in a reasonable region $a \in [0.5, 2]$
- As in conventional SBL, b is chosen as small value $\sim 10^{-4}$
- Chocosing $\beta \in (0, 1]$ performs better than $\beta = 0$
  - Safe choice is value closer to 0 imposing mild coupling effect
- Stable recovery over a range of values for $\tau \in [0.5 \times 10^3, 5 \times 10^3]$

Complexity and Convergence

- Number of floating point operations per iterations $\mathcal{O}(m^3)$
  - Same as conventional SBL
- No convergence guarantees
  - Works in practice

Summary

- A new SBL algorithm for handling block sparsity
- Outperforms other existing methods
- Interesting directions to explore:
  1. An algorithm with convergence guarantees
  2. Automatically learn whether or not signal has block structure
  3. Other models for capturing block sparsity