

Signal Recovery with Unknown Sparsity Pattern via Multiple Measurement Vectors

Introduction



Objective:

Recovering sparse signal X from a small set of linear noisy measurements using multiple measurement vectors (MMVs)

Assumption:

- Sparse Clustered Pattern: Non-zero elements of the underlying signal may appear in clusters with an unknown structure on each column of X► Joint-Sparsity: non-zero elements of X appear at the same rows (support
- set of the solution is the same for all columns of X)
- ► Model: $Y = A(s \circ X) + E$ $Y \in \mathbb{R}^{M \times N}$, $A \in \mathbb{R}^{M \times P}$, $\mathbf{s} \in \{0, 1\}^{P \times 1}$, $X \in \mathbb{R}^{P \times N}$, $E \in \mathbb{R}^{M \times N}$, and $(M \ll P)$,
- s accounts for the supports of the solution and o denotes Hadamard product

Proposed algorithm:

C-SBL: sparse Bayesian learning model for sparse signals with unknown clustered pattern

Proposed Statistical Model and Defining Priors

- Measure of clumpiness: $(\Sigma \Delta)_{s} = \sum_{i=2}^{P} |s_{i} s_{i-1}|$, where s is the support learning vector of the solution
 - There exist few transitions for the case where the supports of the solution have a block-sparsity structure
- For example, a constant vector (all ones or all zeros) has a $\Sigma\Delta$ of 0
- More examples:

												ΣΔ=0
												$\Sigma \Delta = 0$
												$\Sigma \Delta = 4$
												ΣΔ=12

Prior on the support-learning component s:

▶ we model the elements of **s** as Bernoulli random variables

$$(s_p; \omega_{0,p}, \omega_{1,p}) \sim \text{Bernoulli}(\frac{\omega_{1,p}}{\omega_{1,p}}), \forall p = 1, 2, ..., P$$

$$\omega_{0,p} + \omega_{1,p}$$

$$\omega_{k,p} = e^{-\alpha(\Sigma\Delta)_{k,p}} \text{Binomial}(\Sigma_{k,p}, P, \gamma_p), \ \forall k = 0, 1$$

► The terms $(\Sigma \Delta)_{k,\rho}$ and $\Sigma_{k,\rho}$ denote the $\Sigma \Delta$ value and the sum over all the elements of **s** for the case where $s_{p} = k$, respectively

$$s_p \sim \text{Bernoulli}(\gamma_p), \ \gamma_p \sim \text{Beta}(\alpha_0, \beta_0), p = 1, \dots, P.$$

- Initial setting: $\alpha_0 = \frac{10}{P}$ and $\beta_0 = 1 \alpha_0$, to encourage sparsity
- Prior on the solution-value matrix X:
- For The columns of the solution-value matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ are assumed to be drawn i.i.d. from the following normal-gamma distribution

 $\mathbf{x}_n \sim \mathcal{N}(\mathbf{0}, \tau^{-1} I_P), \ \tau \sim \operatorname{Gamma}(a_0, b_0), \ n = 1, \ldots, N.$

- \triangleright Due to the lack of prior knowledge on the entries of X, we experimentally se the hyper-parameters to $a_0 = b_0 = 10^{-3}$, endowing X a priori with a fairly high variance
- \triangleright and b₀ denote the shape and rate of the Gamma distribution, respectively
- Prior on the noise component E:
- The entries of E are assumed to be drawn i.i.d. from a Gaussian distribution with an unknown precision ε

$$e_{mn} \sim \mathcal{N}(0, \varepsilon^{-1}), m = 1, \dots, M, n = 1, \dots, N,$$

 $\varepsilon \sim \operatorname{Gamma}(\theta_0, \theta_1).$

▶ The hyper-parameters are set to $\theta_0 = \theta_1 = 10^{-3}$. This setting may vary unde the required precision or prior knowledge on the noise variance

	r on the emphasizing factor on clumniness of sunnorts α .
	$\alpha \sim \text{Gamma}(a_1, b_1)$
	itial aattings a E and b 1
► In ► Th	That setting: $a_1 = 5$ and $b_1 = 1$ The parameter $\alpha > 0$ specifies the significance of ($\Sigma \Delta$)
► La	arge values of α encourage more contiguity in the support of while small values of α cause s to have many transitions
з,	while small values of a cause s to have many transitions
Joint I	Probability Distribution of the Proposed Model
	Ν/
<i>P</i> (<i>Y</i> , s ,	$(X) \propto P(Y \mathbf{s}, X, \varepsilon) (\prod_{P} P(\mathbf{x}_{n} \boldsymbol{\mu}_{X}, \tau^{-1}I_{P})) P(\tau; \boldsymbol{a}_{0}, \boldsymbol{b}_{0}) P(\varepsilon; \theta_{0}, \theta_{1})$
	n=1
$\prod_{p=1} \prod_{k=0}$	$P(s_p \omega_{0,p},\omega_{1,p})P(\omega_{k,p} \Sigma_{k,p},\gamma_p,\alpha,\mathbf{s}))P(\boldsymbol{\gamma} \mathbf{s},\alpha_0,\beta_0)P(\alpha \mathbf{s},a_1,b)$
Graph	ical Model of the Proposed Bayesian Model
	$a_{0} \qquad \mu_{x} = 0 \qquad a_{1} \qquad \Gamma(\alpha; a_{1}, b_{1}) \qquad b_{1}$ $\Gamma(\tau; a_{0}, b_{0})(\tau^{-1}) \qquad N(x_{n}; \mu_{x}, \tau^{-1}I_{p})(x_{n}) \qquad \alpha$
	b_0 Bernoulli $(Q_p; q_{k,p})$
	$\theta_{0} \qquad \qquad$
	$\frac{1}{(\varepsilon, \theta_0, \theta_1)} \underbrace{\varepsilon}_{\varepsilon} \underbrace{N(e_{mn}; \mu_e, \varepsilon^{-1})}_{e_{mn}} \underbrace{e_{mn}}_{e_{mn}} \underbrace{k = 1, 2}_{e_{mn}}$
	$\theta_1 \bullet \mu_e = 0 \bullet$
	$\theta_1 \bullet \mu_e = 0 \bullet$ $\mu_e = 0 \bullet$ $\mu_e = 0 \bullet$
C-SBL	$\theta_1 \bullet \mu_e = 0 \bullet$ $\mu_e = 0 \bullet$
CDI AL	$\theta_1 \bullet \mu_e = 0 \bullet$ Algorithm
SBL AI	$\theta_{\mu} = 0 \mu_{e} = 0 \mu_{e} = 0$ $P_{0} \bullet h(\gamma_{p}; \alpha_{0}, \beta_{0}) \bullet \alpha_{0}$ $Algorithm$ $gorithm for sparse signal recovery of either SMV or MMVs:$
C-SBL SBL Alg $\Theta^{(i)}_{i=1}$ to or <i>Iter</i> =	$\mu_{e} = 0 \mu_{e} = 0 \mu_{e$
C-SBL SBL $Aig\Theta^{(i)}_{i=1} toor Iter =% SuppFor p –$	$p_{0} \bullet h(\gamma_{p}; \alpha_{0}, \beta_{0}) \bullet \alpha_{0}$ Algorithm gorithm for sparse signal recovery of either SMV or MMVs: $N_{collect} = C-SBL(Y, A, \Theta_{0}, N_{burn-in}, N_{collect})$ $1 \text{ to } N_{burn-in} + N_{collect}$ wort-learning vector component $1 \text{ to } P$
SBLAI SBLAI $\{3^{(i)}\}_{i=1}$ to or <i>Iter</i> = % Supp For $p =$ $\tilde{y}_{mn}^{-p} =$	Algorithm a $\mu_{v} = 0$ $\mu_{v} = 0$ $\mu_{v} = 0$ b $h(y_{p}; \alpha_{0}, \beta_{0}) \bullet \alpha_{0}$ c Algorithm gorithm for sparse signal recovery of either SMV or MMVs: $N_{collect} = \mathbf{C} - \mathbf{SBL}(Y, A, \Theta_{0}, N_{burn-in}, N_{collect})$ $N_{collect} = \mathbf{C} - \mathbf{SBL}(Y, A, \Theta_{0}, N_{burn-in}, N_{collect})$ $1 \text{ to } N_{burn-in} + N_{collect}$ bort-learning vector component 1 to P $= y_{mn} - \sum_{l \neq p}^{P} a_{ml} s_{l} x_{ln}, \forall m = 1 \text{ to } M$
C-SBL SBL A SBL A i i i i i i i i	Algorithm a $\mu_{\nu} = 0$ b $h(r_{p}; \alpha_{0}, \beta_{0}) \bullet \alpha_{0}$ c Algorithm gorithm for sparse signal recovery of either SMV or MMVs: a $N_{collect} = \mathbf{C} - \mathbf{SBL}(Y, A, \Theta_{0}, N_{burn-in}, N_{collect})$ a $N_{collect} = \mathbf{C} - \mathbf{SBL}(Y, A, \Theta_{0}, N_{burn-in}, N_{collect})$ b 1 to $N_{burn-in} + N_{collect}$ b cort-learning vector component c 1 to P e $y_{mn} - \sum_{l\neq p}^{P} a_{ml}s_{l}x_{ln}, \forall m = 1 \text{ to } M$ a $\frac{\gamma_{lp}}{\gamma_{p}} \frac{\Sigma_{1,p}}{P+1-\Sigma_{1,p}} e^{-\alpha((\Sigma\Delta)_{0,p}-(\Sigma\Delta)_{1,p})}$ b $(\mathbf{a}, ^{2}(\Sigma^{N} + x^{2})) = 2\mathbf{a}T(\Sigma^{N} + x, \tilde{\mathbf{a}} = P)$
C-SBL SBL AI SBL AI $3^{(i)}_{i=1}$ to or <i>Iter</i> = % Supp For $p =$ $\tilde{y}_{mn}^{-p} =$ $c = \frac{1}{2}$ $k = \frac{\varepsilon}{2}$ $(s_p -)$	$p_{0} = h(y_{p}; a_{0}, \beta_{0}) \in a_{0}$ Algorithm gorithm for sparse signal recovery of either SMV or MMVs: $N_{collect} = C-SBL(Y, A, \Theta_{0}, N_{burn-in}, N_{collect})$ $N_{collect} = C-SBL(Y, A, \Theta_{0}, N_{burn-in}, N_{collect})$ $N_{collect} = 1 \text{ to } N_{burn-in} + N_{collect}$ wort-learning vector component $1 \text{ to } P$ $= y_{mn} - \sum_{l\neq p}^{P} a_{ml}s_{l}x_{ln}, \forall m = 1 \text{ to } M$ $= \frac{\gamma_{p}}{\gamma_{p}} \frac{\Sigma_{1,p}}{P+1-\Sigma_{1,p}} e^{-\alpha((\Sigma\Delta)_{0,p}-(\Sigma\Delta)_{1,p})}$ $((\mathbf{a}_{p} _{2}^{2}(\sum_{n=1}^{N} x_{pn}^{2})) - 2\mathbf{a}_{p}^{T}(\sum_{n=1}^{N} x_{pn}\tilde{\mathbf{y}}_{n}^{-p}))$ $) \sim \text{Bernoulli}(\frac{1}{1+ce^{k}})$
C-SBL SBL SBL A SBL A a b i i i i i i i i	$\int_{A_{\mu}=0}^{P_{\mu}=0} \int_{A_{\mu}=0}^{P_{\mu}\bullet h(\gamma_{p};\alpha_{0},\beta_{0})\bullet \alpha_{0}} d\phi$ Algorithm for sparse signal recovery of either SMV or MMVs: $\frac{1}{N_{collect}} = C-SBL(Y, A, \Theta_{0}, N_{burn-in}, N_{collect})$ $\frac{1}{1 \text{ to } N_{burn-in} + N_{collect}}$ wort-learning vector component $\frac{1}{1 \text{ to } P} = y_{mn} - \sum_{l\neq p}^{P} a_{ml}s_{l}x_{ln}, \forall m = 1 \text{ to } M$ $\frac{-\gamma_{p}}{\gamma_{p}} \sum_{P=1-\sum_{i,p}} e^{-\alpha\left((\Sigma\Delta)_{0,p}-(\Sigma\Delta)_{1,p}\right)}$ $\left(\left(\ \mathbf{a}_{p}\ _{2}^{2}\left(\sum_{n=1}^{N} x_{pn}^{2}\right)\right) - 2\mathbf{a}_{p}^{T}\left(\sum_{n=1}^{N} x_{pn}\tilde{\mathbf{y}}_{n}^{-p}\right)\right)$ $) \sim \text{Bernoulli}\left(\frac{1}{1+ce^{k}}\right)$ lution-value matrix component = 1 to P
C-SBL SBL SBL A SBL A a b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i b i i i i i i i i	$p_{\Delta} = p_{\Delta} = p_{\Delta} = 0$ $p_{\Delta} = h(y_{\rho}; \alpha_{0}, \beta_{0}) = \alpha_{0}$ $= Algorithm$ gorithm for sparse signal recovery of either SMV or MMVs: $N_{collect} = C-SBL(Y, A, \Theta_{0}, N_{burn-in}, N_{collect})$ $= 1 \text{ to } N_{burn-in} + N_{collect}$ bort-learning vector component $= 1 \text{ to } P$ $= y_{mn} - \sum_{l\neq p}^{P} a_{ml} s_{l} x_{ln}, \forall m = 1 \text{ to } M$ $= \frac{\gamma_{D}}{\gamma_{D}} \frac{\sum_{l=D}}{P+1-\sum_{l\neq p}} e^{-\alpha \left((\Sigma\Delta)_{0,p} - (\Sigma\Delta)_{1,p}\right)}$ $((\mathbf{a}_{p} _{2}^{2} (\sum_{n=1}^{N} x_{pn}^{2})) - 2\mathbf{a}_{p}^{T} (\sum_{n=1}^{N} x_{pn} \tilde{\mathbf{y}}_{n}^{-p}))$ $) \sim \text{Bernoulli}(\frac{1}{1+ce^{k}})$ Iution-value matrix component $= 1 \text{ to } P$ $= (\tau + \varepsilon s_{l}^{2} \mathbf{a}_{l} _{2}^{2})^{-1}$
C-SBL SBL Algo SBL Algo SBL Algo SBL Algo SBL Algo i = 1 to or <i>Iter</i> = % Supp For $p =$ $\tilde{y}_{mn}^{-p} =$ $c = \frac{1}{2}$ $k = \frac{\varepsilon}{2}$ $(s_p -)$ % So For <i>I</i> $\Sigma_x =$ $\bar{\mu} =$ For	Additional states Additional states additio
C-SBL Alg -SBL Alg $\exists^{(i)}_{i=1 \text{ to}}$ $\sigma \text{ Iter} =$ % Supp For $p =$ $\tilde{y}_{mn}^{-p} =$ $c = \frac{1}{2}$ $k = \frac{\varepsilon}{2}$ $(s_p -)$ % So For I $\Sigma_x =$ $\bar{\mu} =$ For \tilde{y}_{mn}	Algorithm a a b a b b b b b b b b b b
C-SBLAI -SBLAI -SBLAI $3^{(i)}_{i=1 \text{ to}}$ or <i>Iter</i> = % Supp For $p =$ $\tilde{y}_{mn}^{-p} =$ $c = \frac{1}{2}$ $k = \frac{\varepsilon}{2}$ $(s_p -)$ % So For <i>I</i> $\Sigma_x =$ $\bar{\mu} =$ For \tilde{y}_{mn}	Algorithm provide $p_{k,-0} = p_{k,-0} = p_{k,-0}$
C-SBL SBL AI SBL AI SBL AI SBL AI SBL AI SBL AI S S S S S S S S	$ \int_{\mathbb{R}^{n}} \int_{\mathbb{R}^{n}$
C-SEL SBL AI SBL AI SBL AI SBL AI SBL AI SBL AI S i = 1 to or <i>Iter</i> = % Supp For $p =$ $\tilde{y}_{mn}^{-p} =$ $c = \frac{1}{2}$ $k = \frac{\varepsilon}{2}$ $(s_p -)$ % So For <i>I</i> $\sum_{x} =$ $\bar{\mu} =$ For Y For <i>P</i> For <i>P</i> For <i>P</i> For <i>P</i> For <i>P</i> For <i>P</i> For <i>P</i> For <i>P</i> For <i>P</i> For <i>P</i> S For <i>I</i> $\sum_{x} =$ $\bar{\mu} =$ For For <i>I</i> $\sum_{x} =$ $\bar{\mu} =$ For For <i>I</i> $\sum_{x} =$ For For <i>I</i> $\sum_{x} =$ For For For <i>I</i> S For <i>I</i> S S For <i>I</i> S S For <i>I</i> S S For <i>I</i> S S For <i>I</i> S S For <i>I</i> S S For <i>I</i> S S For I S S S S For I S S S S S S S S	Algorithm a b b b b b b c b b c b b c b b c b c b c b c b c c b b c c c b c c c c c c c c c c
C-SBL SBL Algo $\Theta^{(i)}_{i=1}$ to or <i>Iter</i> = % Supp For $p =$ $\tilde{y}_{mn}^{-p} =$ $c = \frac{1}{2}$ $k = \frac{\varepsilon}{2}$ $(s_p -)$ % So For <i>I</i> $\Sigma_x =$ $\bar{\mu} =$ For $\tilde{y}_{mn}^{\mu} =$ For $\chi_x =$ $\bar{\mu} =$ For $\tilde{y}_{mn}^{\mu} =$ For $(s_p -)$	Add Constraints Add Constraints add Series add Series add Series add Series add Series b add Series add Series
C-SBL SBL Algo \exists (<i>i</i>) $i=1$ to or <i>Iter</i> = % Supp For $p =$ $\tilde{y}_{mn}^{-p} =$ $c = \frac{1}{2}$ $k = \frac{\varepsilon}{2}$ $(s_p -)$ % So For <i>I</i> $\Sigma_x =$ $\tilde{\mu} =$ For \tilde{y}_{mn} $\mu =$ For \tilde{y}_{mn} $\mu =$ For \tilde{y}_{mn} $\mu =$ For $(s_p -)$ End Fo $(\gamma_p -)$ End Fo $(\gamma_p -)$ $(\alpha_{ -}) \sim$	$p_{\mathbf{y}} = p_{\mathbf{y}_{n},\mathbf{q}_{n}} = p_{\mathbf{y}_{n},\mathbf{q}_{n}} = p_{\mathbf{y}_{n},\mathbf{q}_{n},\mathbf{q}_{n},\mathbf{q}_{n}} = \mathbf{A}$ a Algorithm for sparse signal recovery of either SMV or MMVs: $\frac{1}{N_{collect}} = \mathbf{C} \cdot \mathbf{SBL}(Y, A, \Theta_{0}, N_{burn-in}, N_{collect}) + 10 N_{burn-in} + N_{collect}$ Sourchearning vector component $\frac{1}{1 \text{ to } P} = \frac{1}{1 + 2} \sum_{k=0}^{p} a_{ml} s_{k} x_{k}, \forall m = 1 \text{ to } M$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } M$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } M$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } M$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } M$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } M$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } M$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } M$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } M$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } M$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } M$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{i=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{p=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{D} \sum_{p=1}^{L} \sum_{p=1}^{p} a_{ml} s_{i} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{L} = p^{T} \sum_{p=1}^{T} \sum_{p=1}^{T} \sum_{p=1}^{T} x_{p} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{L} = p^{T} \sum_{p} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{L} = p^{T} \sum_{p} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{L} = p^{T} \sum_{p} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{L} = p^{T} \sum_{p} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{L} = p^{T} \sum_{p} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{L} = p^{T} \sum_{p} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{L} = p^{T} \sum_{p} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{L} = p^{T} \sum_{p} x_{k}, \forall m = 1 \text{ to } N$ $\frac{\gamma_{L} = p^{T} \sum_{p} x_{k}, \forall m = 1 \text$
C-SBL SBL A SBL A SBL A SBL A SBL A S S S S S S S S	$P_{n_{n}} = P_{n_{n}} e^{-p_{n}} P_{n_{n}} e^{-p_{n}} P_{n_{n}} e^{-p_{n}} $
C-SBL SBL Algo $\Theta^{(i)}_{i=1} = 1$ to or <i>Iter</i> = % Supp For $p =$ $\tilde{y}_{mn}^{-p} =$ $c = \frac{1}{2}$ $k = \frac{\varepsilon}{2}$ $(s_p -)$ % Sol For <i>I</i> $\Sigma_x =$ $\bar{\mu} =$ For $\tilde{y}_{mn}^{\mu} =$ For $\chi_x =$ $\bar{\mu} =$ For $\tilde{y}_{mn}^{\mu} =$ For $(s_p -)$ End Fo $(\gamma_p -)$	$\int_{k}^{\infty} \int_{k}^{\infty} e^{-e^{-e^{-k}}} e^{-\beta \left(y_{i},z_{i},z_{i},\beta_{i}\right) \cdot dz}$ a Algorithm for sparse signal recovery of either SMV or MMVs: $\int_{N_{coldect}} e^{-c} - C-SBL(Y, A, \Theta_{0}, N_{burn-in}, N_{coldect})$ $f to N_{burn-in} + N_{coldect}$ wort-learning vector component $f to P$ $\int_{n}^{\infty} \frac{\sum_{i,p}}{p^{i+1}-\sum_{i,p}} e^{-\alpha \left((\Sigma\Delta)_{i,p}-(\Sigma\Delta)_{i,p}\right)}$ $((\mathbf{a}_{p} _{2}^{2}(\sum_{l=1}^{n} x_{pl}^{2})) - 2\mathbf{a}_{p}^{T}(\sum_{l=1}^{n} x_{pl}\mathbf{y}_{pl}^{T}))$ $) \sim \text{Bernoulli}\left(\frac{1}{1+e^{k}}\right)$ Iution-value matrix component $f to P$ $= (\tau + cs_{i}^{2} \mathbf{a}_{l} _{2}^{2})^{-1}$ $= s_{i}(\sum_{i,q} A_{i})$ $k_{i}(k_{i}) - \sqrt{N}(\mu_{X}, \Sigma_{X})$ $d For For \int_{n}^{\infty} \text{Deta}\left(\alpha_{0} + 2(1 + \sum_{k\neq p}^{p} s_{k}), \beta_{0} + 2(P - \sum_{k\neq p}^{p} s_{k})\right) f Gamma(a_{0} + \frac{MP}{2}, b_{0} + \frac{1}{2} X _{F}^{2}) Gamma(a_{0} + \frac{MP}{2}, b_{0} + \frac{1}{2} X _{F}^{2}) Gamma(a_{1} + \sum_{p-1}^{p} (\Sigma\Delta)_{0,p}, b_{1} + \sum_{p-1}^{p} (\Sigma\Delta)_{1,p}) N_{burn-in}$

Mohammad Shekaramiz, Todd K. Moon, and Jacob H. Gunther Information Dynamics Laboratory, ECE Dept., Utah State University

Behavior of α with respect to $(\Sigma \Delta)$

For each element s_p we have:

$(\Sigma\Delta) _{0,\rho}$	α
cte	\downarrow
cte	\downarrow
\downarrow	\downarrow
\uparrow	\uparrow
cte	\uparrow
	$(\Sigma \Delta) _{0,p}$ cte \downarrow \uparrow cte

- For example consider the case where forcing either $s_p = 0$ or $s_p = 1$ does not make any change in the evaluation of $(\Sigma \Delta)$
- In this case, though it promotes the clumpiness in the solution, it discourages the solution to be sparse
- Therefore, α needs to be decreased

Simulation Results

- Our MMV model is a set of linear equations where
- The supports of the true solution are binary and drawn from a Bernoulli distribution in such a way to have a random clustered-sparsity structure
- The number of columns in X and Y is set to N = 2
- ► The entries of $\bar{\mathbf{x}}_n$ is drawn i.i.d. from $\mathcal{N}(\mathbf{0}, \sigma_x^2 I_P)$, where $\sigma_x^2 = \mathbf{1}$
- The true solution is constructed from $X = \mathbf{s} \circ \overline{X}$
- The sensing matrix $A \in \mathbb{R}^{M \times P}$, with $a_{mn} \sim \mathcal{N}(0, \frac{1}{\sqrt{M}})$, where M varies and P = 100
- ▶ The entries of the noise vector are drawn i.i.d. from $\mathcal{N}(\mathbf{0}, \sigma_n^2)$, in such a way to have SNR = 25 dB
- The measurement matrix Y is computed from Y = AX + E
- In all the simulations, the sparsity level is set to $K_{sp} = 25$
- In the simulations for C-SBL we set $N_{\text{burn-in}} = 500$ and $N_{\text{collect}} = 500$
- We consider two case scenarios
- ▶ case 1: the columns of true solution X are uncorrelated i.e., $\rho = 0$
- case 2: the columns of true solution X have correlation factor of = 0.85
- In order to investigate the performance, we generate 200 random cases using the above settings and then averaging over all the obtained results
- In the figures, λ is the sampling rate and is defined as $\lambda := M/P$

Simulation Results (Performance of C-SBL)







Simulation Results (Comparison with other algorithms)







A Case Scenario (Synthetic data)

In this example $\lambda = 0.7$, N = 2, and SNR = 25dB





Conclusions

A new algorithm for the recovery of sparse signals with unknown clustered pattern is proposed (C-SBL algorithm)

The proposed algorithm can be used for either single- or multiple-measurement vectors in the compressive sensing (CS) applications

► Based on the simulation results, C-SBL outperforms the famous M-SBL [2], T-SBL [3], and MFOCUSS [4] algorithms

Main References

. M. Shekaramiz, T. K. Moon, and J. H. Gunther, *Hierarchical Bayesian approach* for jointly-sparse solution of multiple measurement vectors, 48th Asilomar Conf. of Signals, Systems, and Computers, pp. 1962–1966, Nov. 2014.

2. D.P. Wipf and B. D. Rao, An empirical Bayesian strategy for solving the simultaneous sparse approximation problem, IEEE Trans. on Signal Processing, vol. 55, no. 7, pp. 3704–3716, 2007.

3. Z. Zhang and B. D. Rao, Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning, IEEE Journal of Special Topics in Signal Processing, vol. 5, no. 5, pp. 912–926, 2011.

. S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, Sparse solutions to *linear inverse problems with multiple measurement vectors*, IEEE Transactions on Signal Processing, vol. 53, no. 7, pp. 2477–2488, 2005.