Published in final edited form as:

Conf Rec Asilomar Conf Signals Syst Comput. 2015 November ; 2015: 508–512. doi:10.1109/ACSSC.

2015-7421180

On The Block-Sparse Solution of Single Measurement Vectors

Mohammad Shekaramiz, Todd K. Moon, and Jacob H. Gunther

ECE Department and Information Dynamics Laboratory, Utah State University

Abstract

Finding the solution of single measurement vector (SMV) problem with an unknown block-sparsity structure is considered. Here, we propose a sparse Bayesian learning (SBL) algorithm simplified via the approximate message passing (AMP) framework. In order to encourage the block-sparsity structure, we incorporate a parameter called Sigma-Delta as a measure of clumpiness in the supports of the solution. Using the AMP framework reduces the computational load of the proposed SBL algorithm and as a result makes it faster. Furthermore, in terms of the mean-squared error between the true and the reconstructed solution, the algorithm demonstrates an encouraging improvement compared to the other algorithms.

1. Introduction

The SMV problem is a set of linear noisy measurements from a sparse signal $\mathbf{x}$ and is modeled as $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{x} \in \mathbb{R}^N$ is the signal of interest to be reconstructed and $\mathbf{e}$ denotes the noise. In this model, $\mathbf{A} \in \mathbb{R}^{M \times N}$ is a known sensing matrix with $M < N$. In some practical applications the non-zero entries of the sparse signal $\mathbf{x}$ appear in clusters, so they more or less clump together. This feature has been referred to as block-sparsity in the literature [1–6]. One example is magnetoencephalography, which seeks the locations where most brain activities are produced. Such activities exhibit contiguity i.e., they occur in localized regions [3].

For the purpose of promoting the recovery performance of the SMV problem with an unknown block-sparsity framework, Zhang and Rao proposed a sparse Bayesian learning (SBL) algorithm that incorporates intra-block correlation (correlation structure in each block) [2]. In this model, they defined a Gaussian-distributed prior with zero-mean and the covariance that depends on the multiplication of a nonnegative scaling parameter followed by a covariance matrix for each block. Such blocks are assumed to be uncorrelated with each other. In order to simplify the model, reduce the complexity, and suppress the overfitting of the parameters in the model, they further considered the same underlying covariance matrix with different scaling parameters for the blocks as a prior. This matrix is updated via the expectation-maximization (EM) algorithm. In [5], a hierarchical Bayesian approach was proposed to deal with the block-sparse multiple measurement vectors (MMVs) problem. In this case, a prior is incorporated which encourages both contiguity and sparsity in the solution. This prior is based on the parameter referred to as Sigma-Delta (Σ), which is a measure of contiguity over the supports of the solution. The Sigma-Delta parameter is defined as follows

$$(\Sigma\Delta)_s = \sum_{n=2}^N |s_n - s_{n-1}|, \quad (1)$$

where $\mathbf{s}$ is the support learning vector of the solution and has binary values with “1” denoting the active entry of $\mathbf{x}$. The prior on $\mathbf{s}$ depends on the term $e^{-a(\Sigma\Delta)_s}$ where $a > 0$. The larger the weight a is, the more clumpy the solution becomes. Experimental receiver operating curves (ROC) indicate that the performance of the algorithm was satisfactory and encouraging. However, the runtime of the algorithm was not fast. In [6], we proposed the SDsRandOMP algorithm which is essentially a sparse version of RandOMP [7] and incorporates the Sigma-Delta parameter to encourage the block-sparsity. This algorithm is almost greedy-based in which the behavior of Sigma-Delta is modeled by a Gamma distribution with $(\Sigma\Delta)_s \sim \Gamma(1, \theta)$. Although this algorithm is faster than the SBLs, it requires more information i.e., noise variance.

Recently, research in this area has turned to reducing the computational complexity of their algorithms while preserving the success in the support recovery at a high rate. Such work includes using approximate message passing (AMP) and the expectation-maximization (EM) for the hyperparameters of the SBL algorithms [8–10]. The AMP is essentially a simplified version of the message passing where the number of messages to be propagated is reduced based on Taylor series approximation, averaging over the messages passed to the same node, and the central limit theorem [8,9]. In this case, Al-Shoukairi and Rao incorporated the AMP to their earlier SBL model for both the SMV and MMV problem to reduce the runtime of their algorithms [8]. They avoided imposing a Bernoulli-Gaussian prior on the solution vector and instead simply used a zero-mean Gaussian prior. This simplification causes all the distributions on the factor graph to become Gaussian and results in fewer computations when using message passing.

In this paper, we propose a new SBL algorithm for the block-sparse SMV problem based on the AMP-SBL algorithm proposed in [8]. For this purpose, we use our measure of clumpiness over the supports of the solution (Sigma-Delta) proposed in [5,6] to the AMP-SBL. We refer to our new algorithm as AMP-B-SBL where “-B-” denotes the block-sparsity. To demonstrate the performance of our algorithm, we compare the proposed algorithm with two other algorithms in terms of the normalized mean-squared error between the true and the reconstructed solution. Furthermore, to show the behavior of the algorithm in the signal reconstruction, a random block-sparse case scenario for the SMV has been made. Since it benefits from the AMP, this algorithm turns out to be faster than our previously proposed SBLs [5,6].

2. AMP-B-SBL for Solving SMV

In order to solve for the SMV problem with an unknown block-sparsity structure, we combine our measure of clumpiness of the solution (Sigma-Delta) proposed in [5,6] with the AMP-SBL algorithm introduced in [8]. Here, we define the Sigma-Delta as

$$(\Sigma\Delta)_{(\text{support of } \mathbf{x})} = \sum_{n=2}^N |b(x_n, T) - b(x_{n-1}, T)|, \quad (2)$$

where T is a predetermined threshold. The function $b(.,.)$ in (2) returns a binary value and is defined as follows

$$b(x_n, T) = \begin{cases} 1 & \text{if } |x_n| > T \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

In our previous SBL model [5] we did not have such soft thresholding (3), and regardless of having a very small or large value on x_n , the corresponding support s_n became active ($s_n = 1$). In contrast, here we simply discard such small values by setting the corresponding $s_n = 0$ because such x_p does not have considerable contribution in our measurements. In order not to discard important portions of the signal of interest, we set the threshold T to a small value.

Based on the assumption that the solution vector $\mathbf{x}$ is sparse, we consider an i.i.d. zero-mean Gaussian prior on the components of $\mathbf{x}$. The variance on x_n is defined as α_n which accounts for learning the block-sparsity structure in the solution. These distributions are defined below.

$$\forall n=1, \dots, N, \quad x_n \sim \mathcal{N}(0, \alpha_n) \quad \alpha_n \sim \mathcal{N}\left(e^{\left\{ \frac{(\Sigma\Delta)|_{b(x_{n,:})=0} - (\Sigma\Delta)|_{b(x_{n,:})=1} - 1}{\theta_1} \right\}}, \theta_2\right), \quad (4)$$

where $(\Sigma\Delta)|_{b(x_{n,:})=1}$ is the Sigma-Delta evaluation of the supports of $\mathbf{x}$ in case where s_n is forced to be active, θ_1 is the emphasizing parameter on our measure of clumpiness, and θ_2 is the prior variance on the variance of the variable x_n . The rationale behind assuming this prior on α_n is described in Table 1, in which *cte* denotes a constant value. As an example of Table 1, consider the case where forcing either $s_n = 0$ or $s_n = 1$ 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 as we do this a couple of times. Therefore, α_n needs to be decreased.

According to the defined prior distributions in (4), the joint probability distribution of our model becomes

$$P(\mathbf{y}, \mathbf{x}, \boldsymbol{\alpha}, \theta_1, \theta_2, \sigma^2) \propto P(\mathbf{y}|\mathbf{x}, \sigma^2 I_N) \prod_{n=1}^N \left(P(x_n; 0, \alpha_n) P(\alpha_n; e^{\left\{ \frac{(\Sigma\Delta)|_{b(x_{n,:})=0} - (\Sigma\Delta)|_{b(x_{n,:})=1} - 1}{\theta_1} \right\}}, \theta_2) \right), \quad (5)$$

where the measurement noise is assumed to be $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_N)$.

Under this model, all the distributions of the joint, conditional, and posterior densities become Gaussian. Therefore, we only need to pass the mean and variance of the messages rather than computing and propagating the full actual messages. The factor graph of our model for the block-sparse SMV (B-SMV) problem is illustrated in Fig. 1. In the graphical model Fig. 1, the large circle nodes represent the random variables of interest, the shaded squares show the function nodes, the small shaded circles are the observations, and the small unshaded circles denote the hyperparameters [11]. Since this main layer of our algorithm is the same as the one introduced in [8, 10], here we use the same notations as [8]. Such function nodes shown in Fig. 1 are defined as follows

$$\begin{aligned} g_m &:= P(y_m | \mathbf{x}, \boldsymbol{\alpha}), \quad m=1, 2, \dots, M \\ f_n &:= P(x_n; 0, \alpha_n), \quad n=1, 2, \dots, N. \end{aligned} \quad (6)$$

Notice that our proposed algorithm adds an additional layer to [8] by incorporating the prior (4) in order to encourage the block-sparsity. This change in the model does not appear in the factor graph Fig. 1. We now describe our reasoning behind the proposed model for the B-SMV problem. As a prior knowledge, we expect the solution vector $\mathbf{x}$ to be sparse. Therefore in order to encourage the sparsity, we assume a zero-mean Gaussian distribution for $\mathbf{x}$ with the variance α_n , $n = 1, 2, \dots, N$ on its components. The supports of the solution are then specified by the binary function $b(\cdot, T)$ where T is a predetermined threshold. In other words, based on the threshold we discard the small-valued components of $\mathbf{x}$ from being considered as the support of the solution. The smaller α_n is, the higher probability it provides to x_n removed from the supports. Based on our previous results in [5,6], we observed that having more clumpy supports in the solution causes higher value of α and vice versa. Therefore, here we made the mean of our Gaussian prior on α_n be proportional to the term $\exp \{ (\sum_{n=1}^N b(x_n, T)=0) - (\sum_{n=1}^N b(x_n, T)=1) \}$.

Below, we represent the message that propagates from each function node to a variable node and vice versa of Fig. 1. The derived messages in (7)-(9) are the same as those derived in [8,12].

- Message from a function node to a variable node:

$$M_{g_m \rightarrow x_n} \propto N(a_{mn}x_n; z_{mn}, c_{mn}), \quad (7)$$

where

$$z_{mn} = y_m - \sum_{q \neq n} a_{mq} \mu_q \quad \text{and} \quad c_{mn} = \alpha^2 + \sum_{q \neq n} |a_{mq}|^2 \Sigma_q.$$

- Message from a variable node to a function node:

$$M_{x_n \rightarrow g_m} \propto \mathcal{N}(x_n; \sum_{l \neq m} \frac{a_{nl} z_{ln} \alpha_n}{c_n + \alpha_n \sum_{l \neq m} a_{ln}^2}, \frac{c_n \alpha_n}{c_n + \alpha_n \sum_{l \neq m} a_{ln}^2}),$$

where c_{nl} (under the large-system-limit) is approximated by [12]

$$c_{nl} \simeq c_n := \frac{1}{M} \sum_{m=1}^M c_{mn}.$$

Using the fact that we have already normalized the sensing matrix A with respect to its columns, we can further approximate

$$\sum_{l \neq m} a_{ln}^2 \simeq \sum_{m=1}^M a_{mn}^2 = 1.$$

Therefore,

$$M_{x_n \rightarrow g_m} \propto \mathcal{N}(x_n; \sum_{l \neq m} a_{nl} Z_{ln} \frac{\alpha_n}{c_n + \alpha_n}, \frac{c_n \alpha_n}{c_n + \alpha_n}). \quad (8)$$

- Estimating the posterior on x_n :

$$P(x_n | \mathbf{y}) \propto P(x_n; \alpha_n) \prod_{m=1}^M P(y_m | x_n) \propto M_{f_n \rightarrow x_n} \prod_{m=1}^M M_{g_m \rightarrow x_n}.$$

After simplification, we obtain [8]

$$P(x_n | \mathbf{y}) \propto \mathcal{N}(x_n; \mu_n, \nu_n),$$

where

$$\mu_n = \sum_{m=1}^M a_{mn} z_{mn} \left(\frac{\alpha_n}{c_n + \alpha_n} \right), \quad \text{and} \quad \nu_n = \frac{c_n \alpha_n}{c_n + \alpha_n}. \quad (9)$$

Below, we provide the update rules for the hyperparameters of our algorithm using the expectation-maximization (EM) algorithm. As was mentioned earlier, since we add a layer to the AMP-SBL algorithm to account for the block-sparsity structure, the update rules become different from [8]. We start with describing the update rule for the variance α_n of x_n . Notice that as a prior we consider $\mathbf{x}$ is $x_n \sim \mathcal{N}(0, \alpha_n)$, $\forall n = 1, \dots, N$. Let us define

$$(\Sigma\Delta)_{n,k} := (\Sigma\Delta)|_{b(x_n, \cdot) = k}, \forall k=0, 1. \quad (10)$$

- Update rule for α_n :

$$\alpha_n^{[k+1]} = \arg \min_{\alpha_n} \ln(\alpha_n) + \frac{1}{\alpha_n} \mathbb{E}_{\mathbf{x}|\alpha_n^{[k]}, -} [x_n^2] + \frac{1}{\theta_2} (\alpha_n - e^{\{\frac{(\Sigma\Delta)_{n,0} - (\Sigma\Delta)_{n,1} - 1}{\theta_1}\}})^2.$$

Therefore,

$$\alpha_n^{[k+1]} = \arg \min_{\alpha_n} \ln(\alpha_n) + \frac{\mu_n^2 + \nu_n}{\alpha_n} + \frac{1}{\theta_2} (\alpha_n - e^{\{\frac{(\Sigma\Delta)_{n,0} - (\Sigma\Delta)_{n,1} - 1}{\theta_1}\}})^2. \quad (11)$$

Hence we solve for α_n in

$$\alpha_n^3 - e^{\{\frac{(\Sigma\Delta)_{n,0} - (\Sigma\Delta)_{n,1} - 1}{\theta_1}\}} \alpha_n^2 + \frac{\theta_2}{2} \alpha_n - \frac{\theta_2}{2} (\mu_n^2 + \nu_n) = 0. \quad (12)$$

A solution among all the three possible roots for (12) which minimizes (11) is the update rule for α_n at the next iteration.

- Update rule for σ^2 :

$$\sigma^{2[k+1]} = \arg \max_{\sigma^2} \mathbb{E}_{\mathbf{x}|\mathbf{y}, \theta_1, \theta_2, \alpha, \sigma^{2[k]}} [P(\mathbf{y}, \mathbf{x}, \alpha, \theta_1, \theta_2, \sigma^2)] = \arg \min_{\sigma^2} 2M \ln(\sigma) + \frac{1}{\sigma^2} \mathbb{E}_{\mathbf{x}|\mathbf{y}, -} [\|\mathbf{y} - A\mathbf{x}\|_2^2]$$

Therefore,

$$\sigma^{2[k+1]} = \frac{\|\mathbf{y} - A\boldsymbol{\mu}_{x|-}\|_2^2 + \text{tr}(\Sigma_{x|-} A^T A)}{M} = \frac{\|\mathbf{y} - A\boldsymbol{\mu}_{x|-}\|_2^2 + \sum_{n=1}^N \|\mathbf{a}_n\|_2^2 \Sigma_{x|-}}{M}, \quad (13)$$

where $\boldsymbol{\mu}_{x|-} := [\mu_1, \dots, \mu_N]^T$ and $\Sigma_{x|-} := \text{diag}\{\nu_1, \dots, \nu_N\}$, and $\Sigma_{x|-}$ is an $N \times N$ diagonal matrix.

- Update rule for θ_2 :

$$\theta_2^{[k+1]} = \arg \min_{\theta_2} \mathbb{E}_{\mathbf{x}|\mathbf{y}, \theta_2^{[k]}, -} [N \ln(\theta_2) + \frac{1}{\theta_2} \sum_{n=1}^N (\alpha_n - e^{\{(\Sigma\Delta)_{n,0} - (\Sigma\Delta)_{n,1} - 1\}})^2].$$

Therefore,

$$\theta_2^{[k+1]} = \frac{\sum_{n=1}^N (\alpha_n - e^{\{(\Sigma\Delta)_{n,0} - (\Sigma\Delta)_{n,1} - 1\}})^2}{N}. \quad (14)$$

Finally, based on the AMP algorithm in [8], our AMP-B-SBL is described as follows. The stopping condition of the algorithm can be based on a predetermined number of iterations or convergence of the solution to a tight bound.

AMP-B-SBL Algorithm for the block-sparse SMV problem:

- Definitions

$$F_n(k_n, \alpha_n, c) = k_n \frac{\alpha_n}{c + \alpha_n}$$

$$G_n(\alpha_n, c) = \frac{c \cdot \alpha_n}{c + \alpha_n}$$

$$F'_n(\alpha_n, c) = \frac{\alpha_n}{c + \alpha_n}$$

- Message updates

For $n = 1, 2, \dots, N$

$$k_n = \sum_{m=1}^M a_{mn}^* z_m + \mu_n$$

$$\mu_n = F_n(k_n, \alpha_n, c)$$

$$\nu_n = G_n(\alpha_n, c)$$

End

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

$$z_m = y_m - \sum_{n=1}^N a_{mn} \mu_n + \frac{z_m}{M} \sum_{n=1}^N F'_n(\alpha_n, c), \forall m = 1, \dots, M$$

- Parameter updates

Updating α :

$\forall n = 1, 2, \dots, N$, solve for α_n in

$$\alpha_n^3 - e^{\{\frac{(\Sigma\Delta)_{n,0} - (\Sigma\Delta)_{n,1} - 1}{\theta_1}\}} \alpha_n^2 + \frac{\theta_2}{2} \alpha_n - \frac{\theta_2(\mu_n^2 + \nu_n)}{2} = 0$$

which is the minimizer of

$$f(\alpha_n) = \ln(\alpha_n) + \frac{\mu_n^2 + \nu_n}{\alpha_n} + \frac{1}{\theta_2} (\alpha_n - e^{\{\frac{(\Sigma\Delta)_{n,0} - (\Sigma\Delta)_{n,1} - 1}{\theta_1}\}})^2$$

Updating the noise variance σ^2 :

$$\sigma^{2[k+1]} = \frac{\|\mathbf{y} - A\boldsymbol{\mu}\|_2^2 + \sum_{n=1}^N \|\mathbf{a}_n\|_2^2 \nu_n}{M}$$

Updating the variance of $\boldsymbol{\alpha}$:

$$\theta_2^{[k+1]} = \frac{1}{N} \sum_{n=1}^N \left(\alpha_n - e^{\left\{ \frac{(\Sigma\Delta)_{n,0} - (\Sigma\Delta)_{n,1-1}}{\theta_1} \right\}^2} \right)$$

In the above algorithm, the hyperparameter θ_1 is a tuning parameter. Since we use Gaussian message passing modeling, it is faster than the conventional SBL modeling. In other words, it provides a trade-off between the exactness of the SBLs and the fast runtime of the greedy-based algorithms.

3. Simulation Results

In this section, we demonstrate the performance of our algorithm compared to two versions the SBL algorithm borrowed from [13] and the approximate message passing SBL (AMP-SBL) [8]. For simulation purposes, the elements of vector $\mathbf{x}_{np}$ are drawn i.i.d. from Gaussian distribution with zero-mean and variance $\sigma_x^2=1$, where $\mathbf{x}_{np}$ denotes a non-sparse solution vector. The supports of the solution are binary and randomly drawn from a Bernoulli distribution in such a way to have a random block-sparsity structure. The true block-sparse solution is then constructed from $\mathbf{x} = \mathbf{s} \circ \mathbf{x}_{np}$, where “ $\circ$ ” denotes the Hadamard product. The sensing matrix A is 100×200 and is randomly drawn from $\mathcal{N}(0, 1)$ and then it is normalized with respect to its columns. The elements of the noise component are drawn from $e_{mn} \sim \mathcal{N}(0, \sigma_n^2)$. We vary σ_n^2 to illustrate the performance over different SNRs. The measurement vector $\mathbf{y}$ is 100×1 and is computed from $\mathbf{y} = A\mathbf{x} + \mathbf{e}$. We evaluate the performance of the proposed algorithm using the normalized mean-squared error defined as

$$\text{NMSE} := \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 / \|\mathbf{x}\|_2^2,$$

where $\hat{\mathbf{x}}$ and $\mathbf{x}$ denote the estimated solution and the true block-sparse solution, respectively. In Fig. 2, we generate 100 different SMV problems with the aforementioned features and then average over the obtained results. All of the compared algorithms in this figure are set to 1000 iterations. In this figure, we set $\sigma_x^2=1$ and change the noise variance $\sigma_n^2=1$ for evaluating the performance at different SNRs. In the legend of Fig. 2, “AMP-B-SBL” and “AMP-SBL” denote our algorithm and the algorithm proposed in [8], respectively. Also, “Trad.-SBL” and “Fast-SBL” are the traditional SBL and the EM-based fast version of the traditional SBL, respectively [13]. The term λ is the emphasis parameter on the sparsity over the error term, and r_{sp} is defined as the sparsity level divided by the length of the original signal i.e., $r_{sp} = K_{sp}/N$. We chose K_{sp} in such a way that on the average $r_{sp} \approx 0.2$. It can be seen from Fig. 2 that the normalized mean-squared error for both Trad.-SBL and Fast-SBL are the same for the same SNR values and almost remain constant as the SNR varies. Our algorithm shows less NMSE compared to those algorithms and also compared to the AMP-

SBL. The reason that our algorithm works better than the AMP-SBL is due to the fact that the AMP-B-SBL accounts for the unknown block-sparsity in the solution.

We now apply AMP-B-SBL to a specific example. In this case, we randomly generated a block-sparse signal and the corresponding set of measurements as described earlier. In our case, the true supports of the solution (non-zeros of $\mathbf{s}$) are $\{28 : 30, 89 : 95, 106, 117, 118, 128 : 131, 135, 146 : 151, 156 : 162, 172, 190, 191, 200\}$ with $\{28 : 30\}$ denoting the 28th through 30th entries of $\mathbf{s}$. For the initialization of the parameters, we set $\theta_1 = 100$, $c^{[0]} = 5$, $\theta_2^{[0]} = 0.002$, and the threshold $T = 0.001$. Fig. 3 and Fig. 4 illustrate the obtained α_n , $\forall n = 1, 2, \dots, N$ after 1000 iterations for the SNR=10 [dB] and 25 [dB], respectively.

In Fig. 5 and Fig. 6 we demonstrate the comparison between the true and the estimated solution of AMP-B-SBL algorithm for our case scenario. It can be seen from Fig. 5 that at the lower SNR, our algorithm could almost successfully find the active entries of the solution. There are also some false supports found by the algorithm and those caused the estimated amplitudes of the solution differ from the true ones. According to Fig. 6, the performance of the algorithm for SNR=25[dB] is satisfactory in terms of both the support detection and the non-zero amplitudes of the true solution. Based on Fig. 6, although the AMP-B-SBL also found some other supports which are not in the true solution, those false supports clumped together and have very small amplitudes. Such clumpiness is due to the fact that the AMP-B-SBL encourages the Sigma-Delta measure of the solution to be small. Finally, in Fig. 7 and Fig. 8 we show the support recovery of our example as it is being updated for the SNRs of 10 [dB] and 25 [dB], respectively.

4. Conclusion

We proposed a new algorithm for solving the block-sparse SMV problem. By assuming Gaussian distributions for all the variables, we were able to use the factor graph which only needs to propagate the mean and variance of the full messages. We further reduced the computational load of the problem using the approximate message passing. We showed that dependence of the solution variances on Sigma-Delta encourages the solution to have a block-sparsity structure and also reduces the normalized mean-squared error between the true and the estimated sparse signal.

Acknowledgments

This work is supported by the Grant NASA NNX13 AD 39 A.

References

1. Mishali M, Eldar YC. Reduce and boost: Recovering arbitrary sets of jointly sparse vectors. *IEEE Trans Sig Proc.* Oct.2008 56:4692–4702.
2. Zhang Z, Rao B. Recovery of block sparse signals using the framework of block sparse Bayesian learning. *IEEE Int Conf on Acoust, Speech and Sig Proc (ICASSP)*. 2012:3345–3348.
3. Kwon HE, Rao B. On the benefits of the block-sparsity structure in sparse signal recovery. *IEEE Int Conf on Acoust, Speech and Sig Proc (ICASSP)*. 2012:3685–3688.
4. Zhang Z, Rao BD. Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation. *IEEE Trans Sig Proc.* Apr.2013 61:2009–2015.

5. Shekaramiz M, Moon T, Gunther J. Hierarchical Bayesian approach for jointly-sparse solution of multiple-measurement vectors. 48th Asilomar Conf on Sig, Syst and Compt. Nov.2014 :1962–1966.
6. Shekaramiz M, Moon T, Gunther J. On the block-sparsity of multiple-measurement vectors. Presented at the IEEE Sig Proc and SP Edu Workshop, Utah, Aug. 2015
7. Elad M, Yavneh I. A plurality of sparse representations is better than the sparsest one alone. IEEE Trans Info Theory. Oct.2009 55:4701–4714.
8. Al-Shoukairi M, Rao B. Sparse Bayesian learning using approxiamte message passing. 48th Asilomar Conf on Sig, Syst and Compt. Nov.2014 :1957–1961.
9. Donoho DL, Maleki A, Montanari A. Message passing algorithms for compressed sensing: I. motivations and construction. Proc Inform Theory Workshop, (Cairo, Egypt). Jan.2010 :1–5.
10. Ziniel J, Schniter P. Dynamic compressive sensing of time-varying signals via approximate message passing. IEEE Trans Sig Proc. Nov.2013 61:5270–5284.
11. Bishop, CM. Pattern Recognition And Machine Learning. Springer; 2009.
12. Schniter P. Turbo reconstruction of structured sparse signals. Conf Inf Sci Syst, Princeton, NJ. Mar. 2010 :1–5.
13. Zhang, Z. [Accessed Sep. 21, 2015] SBL algorithms matlab code. <https://sites.google.com/site/researchbyzhang/software>

Figure 1.
Factor graph for B-SMV.

Figure 2.
Performance when varying the SNR.

Figure 3.
Updated α for SNR=10[dB] when using AMP-B-SBL.

Figure 4.
Updated α for SNR=25[dB] when using AMP-B-SBL.

Figure 5.
Comparison of $\mathbf{x}$ with $\hat{\mathbf{x}}$ for SNR=10[dB].

Figure 6.
Comparison of $\mathbf{x}$ with $\hat{\mathbf{x}}$ for SNR=25[dB].

Figure 7.
Support recovery using AMP-B-SBL for SNR=10[dB].

Figure 8.
Support recovery using AMP-B-SBL for SNR=25[dB].

Table 1
Behavior of α_n with respect to Sigma-Delta

$(\Sigma) _{b(x_n)=1}$	$(\Sigma) _{b(x_n)=0}$	α_n
<i>cte</i>	<i>cte</i>	↓
↑	<i>cte</i>	↓
<i>cte</i>	↓	↓
<i>cte</i>	↑	↑
↓	<i>cte</i>	↑

NASA Author Manuscript

NASA Author Manuscript

NASA Author Manuscript