NASA Public Access

Author manuscript

Ubiquitous Comput Electron Mob Commun Conf (UEMCON) IEEE Annu. Author manuscript; available in PMC 2017 November 22.

Published in final edited form as:

Ubiquitous Comput Electron Mob Commun Conf (UEMCON) IEEE Annu. 2016 October ; 2016: .

AMP-B-SBL: An algorithm for clustered sparse signals using approximate message passing

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

Electrical and Computer Engineering Department and Information Dynamics Laboratory Utah State University

Abstract

Recently, we proposed an algorithm for the single measurement vector problem where the underlying sparse signal has an unknown clustered pattern. The algorithm is essentially a sparse Bayesian learning (SBL) algorithm simplified via the approximate message passing (AMP) framework. Treating the cluster pattern is controlled via a knob that accounts for the amount of clumpiness in the solution. The parameter corresponding to the knob is learned using expectation-maximization algorithm. In this paper, we provide further study by comparing the performance of our algorithm with other algorithms in terms of support recovery, mean-squared error, and an example in image reconstruction in a compressed sensing fashion.

Keywords

Compressive sensing; Sparse Bayesian learning (SBL); single measurement vector (SMV); clustered pattern; approximate message passing (AMP)

I. Introduction

Compressive sensing (CS) is a promising technique in processing compressible or sparse signals by requiring far few samples than the Nyquist rate [1], [2]. This technique is essentially composed of two main parts. The first part is devoted to the design of the sensing device to be able to measure a compressible signal via very few samples such that all the important information of the underlying signal is captured. The sensing structure is required to satisfy some conditions such as restricted isometry property (RIP) [3]. Under the assumption that all the informational value of the signal has already been compressed into very few samples, the problem then is how to extract all the information and reconstruct the underlying signal again in an efficient manner.

The single measurement vector (SMV) problem is one of the benchmark problems with many applications in CS [4]–[9]. In the SMV problem, a set of linear noisy measurements from a sparse signal $\mathbf{x}$ is modeled as $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{x} \in \mathbb{R}^N$ is the signal that we seek to reconstruct, and $\mathbf{e}$ denotes the measurement noise. In this model, $\mathbf{A} \in \mathbb{R}^{M \times N}$ is a known sensing matrix with $M \ll N$, which satisfies RIP. There exist many algorithms to estimate the sparse solution for the SMVs categorized mainly into either greedy-based algorithms [4], [10] or sparse Bayesian learning (SBL) algorithms [7], [11]. In addition to sparsity, there are many applications in which the sparse signal $\mathbf{x}$ exhibits a clustered pattern. Applications can

be found in gene expression analysis [12] and direction of arrival (DOA) [13]. This feature has been referred to as block-sparsity or sparse clustered pattern in the literature [7], [8], [14], [15]. Regarding the SBL algorithms, two main priors have been considered to encourage the sparsity. The first category involves using Gaussian-inverse-Gamma prior on the entries of the solution. In this model, a zero-mean Gaussian distribution is assumed on each element of the solution to encourage the sparsity. The variances of the Gaussian distributions are assumed to be drawn from inverse-Gamma priors [16]. The reason for using either Gamma or inverse-Gamma hyperpriors is due to the requirement for the non-negative outcomes for variances and also to use the conjugacy property of the distributions for the simplification purposes. The second category is referred to as Bernoulli-Gaussian or spike-and-slab priors. In this case, the solution vector is usually separated into two parts; the support leaning vector and the solution value vector [8], [9], [15]. Although SBL algorithms have been shown to be very promising in estimating sparse signals, they are slow in terms of run-time of the algorithm. Recently, it has been shown that using approximate message passing (AMP) techniques can reduce the complexity of SBL algorithms while maintaining almost the same performance as the regular SBLs [16]–[18]. In [19], we provided an SBL algorithm that uses AMP techniques to solve for the clustered pattern sparse signals, referred to as AMP-B-SBL algorithm. This algorithm is based on AMP-SBL algorithm introduced in [16], into which we incorporated another hyperprior to account for the unknown clustered pattern that may exist in the supports of the solution. We imposed the hyperprior to reflect the amount of clumpiness of the overall supports of the solution. The set of hyperparameters controlling the clustered pattern of the support vector were learned via the expectation-maximization (EM) algorithm.

In this paper, we continue the study of the performance of AMP-B-SBL algorithm. We compare the performance of our algorithm with some of the other existing algorithms. Below, we first provide a short review on the theory behind AMP-B-SBL, and then we demonstrate the simulation results.

II. AMP-B-SBL: An algorithm for solving SMVs with unknown clustered pattern

In the AMP-B-SBL algorithm for the SMV problem, we assume *i.i.d.* zero-mean Gaussian priors on the components of the solution as follows

$$x_n \sim \mathcal{N}(0, \alpha_n), \quad \forall n=1, \dots, N. \quad (1)$$

As was discussed in section I, having priors of the form (1) promotes sparsity in the solution. We model the measurement noise in the SMV model as $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_M)$. To encourage the clustered pattern in the supports of the solution $\mathbf{x}$, in [19] we defined a measure of clumpiness as follows

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

where the function $b(\cdot, \cdot)$ is defined as

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

The threshold T is set to a small value, meaning that the estimated components of $\mathbf{x}$ which possesses an amplitude less than T are discarded from the support set and are deemed to be zero. Then, we incorporated the measure of clumpiness into our modeling by adding a hyperprior on each variance α_n as follows [19]

$$\alpha_n \sim \mathcal{N}\left(e^{\left\{ \frac{(\sum \Delta)_{b(x_n, \cdot)=0} - (\sum \Delta)_{b(x_n, \cdot)=1}^{-1}}{\theta_1} \right\}}, \theta_2\right), \quad (4)$$

where $(\sum \Delta)_{b(x_n, \cdot)=k}, \forall k = 0, 1, \forall n = 1, \dots, N$ is defined as the measure of clumpiness in the support of $\mathbf{x}$ in case where the support corresponded to x_n is set to k . For example $(\sum \Delta)_{b(x_n, \cdot)=1}$ is the case of forcing the index of x_n to be active. Here, we abbreviate $(\sum \Delta)_{b(x_n, \cdot)=k}$ to $(\sum \Delta)_{n,k}$. The parameter θ_1 in (4) is the emphasis parameter on our measure of clumpiness, and is left as a free parameter in our model. The parameter θ_2 is another hyperparameter and is learned via the EM algorithm. For the rationale behind assuming (4) as a prior on α_n , one can refer to [19]. The joint probability distribution of our model using (1) and (4) becomes

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

By using the AMP technique defined in [16], [19], [20] and after some algebra, the posterior inference on x_n is estimated as follows

$$\begin{aligned}
P(x_n|\mathbf{y}) &\propto P(x_n;\alpha_n) \prod_{m=1}^M P(y_m|x_n) \\
&\propto N(x_n;\mu_n,\nu_n), \text{ where} \quad (6)
\end{aligned}$$

$$\mu_n = \sum_{m=1}^M a_{mn} z_{mn} \left(\frac{\alpha_n}{c_n + \alpha_n} \right), \quad \text{and} \quad \nu_n = \frac{c_n \alpha_n}{c_n + \alpha_n}. \quad (7)$$

In (7), the terms z_{mn} and c_n are defined as

$$z_{mn} := y_m - \sum_{q \neq m} a_{mq} \mu_q \text{ and } c_n = \frac{1}{M} \sum_{m=1}^M c_{mn},$$

where $c_{mn} = \sigma^2 + \sum_q a_{mq}^2 \nu_q$. (More explanation can be found in [19].) Finally, the update rules for the hyper-parameters $\mathbf{a}$, σ , and θ_2 were obtained using the EM algorithm as the solution to the optimization problems below.

- Update rule for α_n :

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

- Update rule for σ^2 :

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

- Update rule for θ_2 :

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

The simplified version of the above update rules can be found in the pseudo code of the AMP-B-SBL algorithm as described below. The stopping condition of our algorithm can be set based on either a predetermined number of iterations or convergence of the solution to a tight bound.

AMP-B-SBL Algorithm [19]:**• Definitions**

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

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

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

• 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

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

which is the minimizer of

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

Updating the noise variance σ^2 :

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

Updating the variance of α :

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

III. Simulation Results

In this section, we demonstrate the performance of AMP-B-SBL algorithm compared to orthogonal matching pursuit (OMP) [21], basis pursuit denoising (BPDN) [5], MFOCUSS [22], and AMP-SBL [16] algorithms for solving the SMV problem. In all the simulations, we used the default settings for MFOCUSS algorithm. For the OMP, the stopping condition

is set to $\sqrt{0.5N\sigma^2}$. In AMP-B-SBL, we set $c^{[10]} = 10$, $\alpha_n = 2 \times 10^{-4}$, $\forall n = 1, \dots, N$, $T = 0.001$, $\theta_1 = 8$, $\theta_2^{[0]} = 0.6$, and the number of iterations is set to 1000. For the AMP-SBL algorithm, we set $c^{[0]} = 10$, $\gamma_n = 2 \times 10^{-4}$, $\forall n = 1, \dots, N$, and the number of iterations is set to 1000. Finally, for BPDN we stopped the algorithm once $\|\mathbf{y} - A\hat{\mathbf{x}}\|$ becomes less than 0.75.

In the first set of experiments, we generate 200 independent trials and then average over the obtained results. In this case, we evaluate the performance of the algorithms via detection and false alarm rate in support estimation, and also the normalized mean-squared error (NMSE) between the true and the estimation solutions. The NMSE is defined as

$\text{NMSE (dB)} := 20 \log_{10} \frac{\|\mathbf{x} - \hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2}$, where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the true and the estimated solutions, respectively. In each trial, the true solution vector is $\mathbf{x} \in \mathbf{R}^{100}$ with 25 non-zero elements (supports). The supports of the true solution in each trial are randomly drawn so that the support set of the solution exhibits a clustered pattern. The non-zero values of the solution are drawn *i.i.d.* from a zero-mean Gaussian distribution with variance of 1. Each entry of the sensing matrix A is an outcome of a Gaussian distribution $\mathcal{N}(0, 1)$. After constructing the sensing matrix A , we normalize A with respect to its columns. We vary the number of measurements M to show the performance. The elements of measurement noise are drawn *i.i.d.* from $\mathcal{N}(0, \sigma^2)$ such that $\text{SNR} = 25$ dB for all trials. The SNR is defined as $\text{SNR} = 20 \log_{10}(\sigma_x / \sigma)$.

In Fig. 1, we illustrate the empirical results of detection rate *vs.* the ratio $\lambda = M/N$. In the simulations, we discarded those estimated supports which their corresponding amplitudes become less than 0.01.

In Fig. 1 we see that AMP-B-SBL shows the highest detection rate for $\lambda \in [0.05, 0.2]$. For $0.25 \leq \lambda \leq 0.55$ AMP-B-SBL, AMP-SBL, BPDN, and MFOCUSS demonstrate almost the same detection rate. Finally, for $\lambda \geq 0.55$ AMP-B-SBL, AMP-SBL, OMP, and MFOCUSS show approximately the same performance.

In Fig. 2, the false alarm rate (the rate of deciding on wrong supports in the solution) *vs.* λ is illustrated.

In Fig. 2 we observe that our algorithm has a higher false alarm rate compared to the other algorithms for $\lambda = [0, 0.2]$. Comparing the detection rate with the false alarm rate of AMP-B-SBL for such range of λ shows the trade off that exists between the detection rate and false alarm rate. For $0.2 < \lambda \leq 0.3$ all the algorithms show the same performance. However, our algorithm illustrates the *lowest* false alarm rate for $\lambda > 0.3$. Fig. 3 essentially demonstrates the overall performance of the algorithms as a combination of support detection, false alarm rate, and the number of measurements.

From Fig. 3, we observe that the best performance belongs to the AMP-B-SBL algorithm. This shows that incorporating our measure of contiguity in the supports into the SBL algorithm boosts the recovery performance for the clustered pattern sparse signals. Finally, Fig. 4 shows the performance comparison in terms of normalized mean-squared error between the true and estimated solutions.

According to Fig. 4, AMP-B-SBL outperforms the other algorithms in terms of estimating the true solution.

We further evaluate the performance of the algorithms via the following example. In this case we use the image of 112×200 pixels, where we consider the black pixels as the “interesting” locations. The image is shown in Fig. 5.

In our simulations for this image, we assign the value of 1 to the pixels with black color and 0 to the white ones. We then, reshape the matrix corresponding to the image and construct the vector $\mathbf{x} \in \mathbf{R}^{11200 \times 1}$. The number of measurements are set to 5040, i.e., $\lambda = 0.45$. We construct the sensing matrix in the same way we described earlier. Then, the measurements are obtained from $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$ with SNR=25 dB. We feed all the algorithms with the measurement vector $\mathbf{y}$ and the sensing matrix $\mathbf{A}$. The reconstructed images are illustrated in Fig. 6. In all the algorithms, we applied the threshold of 0.01, meaning that we discarded those estimated supports that corresponded to non-zero elements with the absolute value of less than 0.01. Below, we also include the reconstruction based on CLUSSMCMC algorithm [8]. In order to compare the reconstructed images illustrated in Fig. 6, we compare the obtained results based on the NMSE and peak-SNR (PSNR) as shown in Tab. I. According to Tab. I, we observe that CLUSSMCMC provides the best performance in terms of NMSE and PSNR. However, it is much slower compared to other algorithms. In contrast, AMP-B-SBL algorithm provides high performance for NMSE, PSNR, and the speed of the algorithm. By comparing the results of AMP-SBL with AMP-B-SBL, we also see that using the measure of clumpiness into the SBL algorithm definitely improves the reconstruction performance.

IV. Conclusion

Here, we provided further study on our recently proposed AMP-B-SBL algorithm. AMP-B-SBL is a sparse Bayesian learning approach simplified via approximate message passing techniques that is devised for the sparse signals those that may exhibit some sort of clustering pattern in their supports. Based on the simulation results, we showed that the algorithm provides high performance in terms of both the support recovery and the estimating the underlying true solution for signals having unknown clustered patterns.

References

1. Donoho DL. Compressed sensing. IEEE Trans Info Th. 2006; 52(4):1289–1306.
2. Candes EJ, Wakin MB. An introduction to compressive sampling. IEEE Sig Proc Mag. 2008; 25(2): 21–30.
3. Candes EJ, Romberg JK, Tao T. Stable signal recovery from incomplete and inaccurate measurements. Comm on Pure and Applied Math. 2006; 59(8):1207–1223.
4. Mishali M, Eldar YC. Reduce and boost: Recovering arbitrary sets of jointly sparse vectors. IEEE Trans Sig Proc. 2008; 56(10):4692–4702.
5. Chen, S., Donoho, D. Basis pursuit. 28th Asilomar Conf. of Signals, Systems, and Computers; Oct. 1994; p. 41-44.
6. Gill PR, Wang A, Molnar A. The in-crowd algorithm for fast basis pursuit denoising. IEEE Trans Sig Proc. 2011; 59(10):4595–4605.

7. Zhang Z, Rao BD. 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.
8. Yu L, Sun H, Barbot JP, Zheng G. Bayesian compressive sensing for cluster structured sparse signals. *Signal Process*. 2012; 92(1):259–269.
9. Anderson MR, Winther O, Hansen LK. Bayesian inference for structured spike and slab priors. *Adv Neural Inf Process*. 2014:1745–1753.
10. Elad, M. *Sparse and Redundant Representation: From Theory to Applications in Signal and Image Processing*. Springer; 2010.
11. Fang J, Shen Y, Li F, Li H, Chen Z. Support knowledge aided sparse Bayesian learning for compressed sensing. *IEEE Int Conf on Acoust, Speech and Sig Proc (ICASSP)*. 2015:3786–3790.
12. Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K. Sparsity and smoothness via the fused lasso. *J of the Royal Statistical Society Series B*. 2005; 67(1):91–108.
13. Luo JA, Zhang XP, Wang Z. Direction-of-arrival estimation using sparse variable projection optimization. *Proc Int Symp on Circs and Syst (ISCAS)*. May.2012 :3106–3109.
14. Kwon HE, Rao BD. 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.
15. 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; p. 1962-1966.
16. Al-Shoukairi, M., Rao, BD. Sparse Bayesian learning using approximate message passing. 48th Asilomar Conf. on Sig., Syst. and Compt; Nov. 2014; p. 1957-1961.
17. Donoho, DL., Maleki, A., Montanari, A. Message passing algorithms for compressed sensing: I. motivations and construction. *Proc. Inform. Theory Workshop*; Cairo, Egypt. Jan. 2010; p. 1-5.
18. Ziniel J, Schniter P. Dynamic compressive sensing of time-varying signals via approximate message passing. *IEEE Trans Sig Proc*. 2013; 61(21):5270–5284.
19. Shekaramiz, M., Moon, T., Gunther, J. On the block-sparse solution of single measurement vectors. 49th Asilomar Conf. on Sig., Syst. and Compt; Nov. 2015; p. 508-512.
20. Schniter, P. Turbo reconstruction of structured sparse signals. 44th Conf. Inf. Sci. Syst; Princeton, NJ. Mar. 2010; p. 1-6.
21. Pati, YC., Rezaifar, R., Krishnaprasad, PS. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. 27th Asilomar Conf. on Sig., Syst. and Compt; Nov. 1993; p. 40-44.
22. Cotter SF, Rao BD, Engan K, Delgado KK. Sparse solutions of linear inverse problem with multiple measurement vectors. *IEEE Trans Sig proc*. 2005; 53(7):2477–2488.

Fig. 1.
Comparison in terms of detection rate.

Fig. 2.
Comparison of false alarm rate in support recovery.

Fig. 3.

Comparison based on the difference between the experimental detection and false alarm rate.

Fig. 4.
Comparison in terms of normalized mean-squared error between the true and estimated solution.

Fig. 5.
True Image.

Fig. 6. Results of reconstructed images for OMP, BPDN, MFOCUSS, CLUSSMCMC, AMP-SBL, and AMP-B-SBL.

TABLE I

NMSE and PSNR comparison in image reconstruction.

Algorithm	NMSE (dB)	PSNR (dB)	speed
OMP	-1.7124	10.9425	Very Fast
BPDN	-8.7905	17.0205	Very Fast
MFOCUSS	-10.6522	18.8822	Slow
CLUSSMCMC	-27.42	35.6504	Slow
AMP-SBL	-13.4642	22.6943	Fast
AMP-B-SBL	-24.5600	32.7901	Fast