Sparse Bayesian Learning Using Approximate Message Passing

Maher Al-Shoukairi and Bhaskar Rao
Dept. of ECE University of California, San Diego, La Jolla, CA. Email: malshouk@ucsd.edu, brao@ucsd.edu

***Abstract*—We use the approximate message passing framework (AMP) [1] to address the problem of recovering a sparse vector from undersampled noisy measurements. We propose an algorithm based on Sparse Bayesian learning (SBL) [2]. Unlike the original EM based SBL that requires matrix inversions, the proposed algorithm has linear complexity, which makes it well suited for large scale problems. Compared to other message passing techniques, the algorithm requires fewer approximations, due to the conditional Gaussian prior assumption on the original vector. Numerical results show that the proposed algorithm has comparable and in many cases better performance than existing algorithms despite significant reduction in complexity.**

I. Introduction

In the basic single measurement vector (SMV) sparse signal recovery problem, we wish to reconstruct the original sparse signal $\boldsymbol{x} \in \mathbb{R}^N$ from $M \leq N$ noisy linear measurements $\boldsymbol{y} \in \mathbb{R}^M$:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{N \times M}$ is a known measurement matrix and $\boldsymbol{e} \in \mathbb{R}^M$ is the additive noise modeled by $\boldsymbol{e} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. If $\boldsymbol{x}$ is sufficiently sparse and $\boldsymbol{A}$ is well designed, accurate recovery of $\boldsymbol{x}$ is possible.

While this SMV model has a wide range of applications, such as magnetic resonance imaging (MRI), direction of arrival (DOA) estimation and electroencephalography (EEG)/ magnetoencephalography (MEG). The model can be extended to address the case when multiple measurement vectors are available. The multiple measurement vector (MMV) model can be used for a number of applications, such as DOA estimation and EEG/MEG source localization. The MMV model is described by:

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{E}, \tag{2}$$

where $\boldsymbol{Y} \triangleq [\boldsymbol{Y}_{\boldsymbol{.1}}, ..., \boldsymbol{Y}_{\boldsymbol{.T}}] \in \mathbb{R}^{N \times T}$ is the measurement matrix containing T measurement vectors, $\boldsymbol{X} \triangleq [\boldsymbol{X}_{\boldsymbol{.1}}, ..., \boldsymbol{X}_{\boldsymbol{.T}}] \in \mathbb{R}^{M \times T}$ is the unknown source matrix. And $\boldsymbol{E}$ is the unknown noise matrix. In this model we assume a common sparsity profile, which means that the support, which is the location of non zero elements, in $\boldsymbol{X}$ is identical across all columns. In practice it was also found that the different measurement vectors experience temporal correlation. It was found in [3] that exploiting this temporal correlation will yield superior performance when compared to solving the MMV problem while assuming independent vectors.

Compared to other techniques such as greedy algorithms, mixed norm optimization and reweighted algorithms, using Bayesian techniques for sparse signal recovery generally achieves the best performance. Among these Bayesian algorithms is the sparse Bayesian learning algorithm (SBL) which was first introduced by Tipping [4] and then used for the first time for sparse signal recovery by Wipf and Rao [2]. SBL has some appealing properties since its global minimum was proven to be at the sparsest solution [2], unlike l_1 based algorithms. And it was also proven to have less number of local minima when compared to classical recovery algorithms such as the FOCUSS algorithm [5]. However, the SBLs complexity level makes it unsuitable for the use on large scale problems such as imaging for example. In this work, we develop an algorithm based on the same assumptions used in SBL, but use the belief propagation method proposed in [6] to recover the sparse signal. Compared to the original EM based SBL which requires matrix inversions, the proposed approach results in linear complexity, which makes it suitable for large scale problems.

The approximate message passing algorithm was first introduced by Donoho, Maleki and Montenari in [1]. The algorithm uses loopy belief propagation to solve the l_1 minimization problem. It uses the central limit theorem to approximate all messages on the factor graph by Gaussian messages, which requires keeping track of the mean and variance of these messages instead of calculating the actual message at each step. The technique also uses Taylor series approximation to reduce the number of messages even further. The authors also provided a framework to apply the algorithm to Bayesian techniques. The use of a Bayesian framework with belief propagation was proposed previously in [7] [8]. However, because of the Bernoulli-Gaussian or Gaussian mixture priors imposed on the original vector $\boldsymbol{x}$, several approximations had to be made in order to make all the messages on the factor graph Gaussian. On the other hand, the SBL conditional Gaussian prior employed in our approach, results in Gaussian messages all over the graph, without the need for approximations. In addition to that, the Gaussian prior results in a much simpler model, and easier message scheduling especially when we consider the case of multiple measurements with time correlation.

Our numerical results show that the proposed algorithm has comparable performance to other algorithms and in many cases performs slightly better, while reducing the complexity significantly and simplifying the model.

II. SMV AMP-SBL

As mentioned before, in this work we will follow the assumptions of the original SBL algorithm. SBL assumes the Gaussian liklihood model:

$$P(\boldsymbol{y} \mid \boldsymbol{x}; \sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp(-\frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2) \tag{3}$$

Using a hierarchical Bayesian framework, it was shown in [9] that a number of sparsity promoting priors can be represented using a Gaussian scale mixture. A conditional Gaussian prior is imposed on $P(\boldsymbol{x}; \boldsymbol{\gamma})$, and the desired prior on $P(\boldsymbol{x})$ can be obtained by changing the prior on $P(\boldsymbol{\gamma})$. In the SBL case, a non-informative prior is assumed on $\boldsymbol{\gamma}$. The conditional prior on $\boldsymbol{x}$ becomes a parametrized Gaussian (4), where the parameter vector $(\boldsymbol{\gamma})$ can be learned from the data.

$$P(\boldsymbol{x}; \boldsymbol{\gamma}) = \prod_{n=1}^{N} (2\pi\gamma_n)^{-\frac{1}{2}} \exp(-\frac{x_n^2}{2\gamma_n}) \tag{4}$$

This prior was shown in [4] to be a sparsity promoting prior on $\boldsymbol{x}$. And since it is Gaussian when conditioned on $\boldsymbol{\gamma}$, it is appealing to use with the AMP framework.

Based on the previous assumptions and using (5), we can construct the factor graph in figure 1.

$$P(\boldsymbol{x} \mid \boldsymbol{y}) = P(\boldsymbol{y} \mid \boldsymbol{x}) P(\boldsymbol{x}; \boldsymbol{\gamma}) \tag{5}$$

Where $g_m = P(y_m \mid \boldsymbol{x})$ and $f_n = P(x_n; \gamma_n)$.

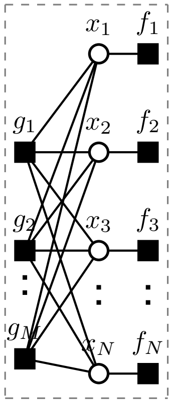

Fig. 1: AMP SBL Factor Graph

Since all the functions on the factor graph are Gaussian pdfs, it will be shown that all the messages passed by the sum product algorithm are also Gaussian, and we only need to keep track of the mean and variance of these messages.

$$V_{f_n \to x_n}(x_n) \propto f_n(x_n; \gamma_n) \tag{6}$$

$$V_{g_m \to x_n}(x_n) \propto \int_{x_{-n}} g_m(x_n) \prod_{q \neq n} V_{x_q \to g_m}(x_n) \tag{7}$$

Where $\int_{x_{-n}}$ denotes the integration over all x_i, $i \neq n$. When $V_{x_q \to g_m}(x_n) \propto \mathcal{N}(x_q; \mu_{qm}, v_{qm})$, we use the following fact:

$$\prod_q \mathcal{N}(x; \mu_q, v_q) \propto \mathcal{N}(x; \frac{\sum_q \mu_q v_q^{-1}}{\sum_q v_q^{-1}}, \frac{1}{\sum_q v_q^{-1}}) \tag{8}$$

And the mean and variance of a message from the factor node g_m to the variable node x_n becomes:

$$V_{g_m \to x_n} \propto \mathcal{N}(A_{mn} x_n; z_{nm}, c_{nm}); \tag{9}$$

Where

$$z_{nm} = y_m - \sum_{q \neq n} A_{mq} \mu_{qm} \tag{10}$$

$$c_{nm} = \sigma^2 + \sum_{q \neq n} |A_{mq}|^2 v_{qm} \tag{11}$$

Next we compute the message from a variable node x_n to a factor node g_m. And by using (8) again, this message corresponds to a Gaussian pdf, and we only need to evaluate its mean and variance.

$$V_{x_n \to g_m} \propto V_{f_n \to x_n} \prod_{l \neq m} V_{g_l \to x_n} \tag{12}$$

$$\prod_{l \neq m} V_{g_l \to x_n} \propto \mathcal{N}(x_n; \frac{\sum_{l \neq m} A_{ln} z_{ln}/c_{ln}}{\sum_{l \neq m} |A_{lm}|^2}, \frac{1}{\sum_{l \neq m} |A_{lm}|^2}) \tag{13}$$

In the large system limit we approximate c_{ln} by:

$$c_{ln} \approx c_n \triangleq \frac{1}{M} \sum_{m=1}^{M} c_{mn} \tag{14}$$

And the mean and variance of $V_{x_n \to g_m}$ are given by:

$$\mu_{nm} = \sum_{l \neq m} A_{ln} z_{ln} (\frac{\gamma_n}{c_n + \gamma_n}) \tag{15}$$

$$v_{nm} = \frac{c_n \gamma_n}{c_n + \gamma_n} \tag{16}$$

Finally, we can estimate the posterior $P(x_n \mid \boldsymbol{y})$ using:

$$P(x_n \mid \boldsymbol{y}) \propto V_{f_n \to x_n} \prod_{l=1}^{M} V_{g_l \to x_n} \propto \mathcal{N}(x_n; \mu_n, v_n) \tag{17}$$

$$\mu_n = \sum_{l=1}^{M} A_{ln} z_{ln} (\frac{\gamma_n}{c_n + \gamma_n}) \tag{18}$$

$$v_n = \frac{c_n \gamma_n}{c_n + \gamma_n} \tag{19}$$

The analysis above can be considered a simplified version of the analysis in [11] which assumes a Bernoulli-Gaussian prior on $\boldsymbol{x}$. We then use the same Taylor series approximation given by the original AMP algorithm, and summarize one iteration of the AMP-SBL algorithm in Table1. This process is repeated, until we reach a stable solution, or until a predetermined maximum number of iterations are completed.

As we discussed earlier, we will need to learn $\boldsymbol{\gamma}$ from the data, and to do this we use the EM algorithm which is also used in the original SBL. The EM updates can be found by:

$$\gamma_n^{(i+1)} = \underset{\gamma_n}{\operatorname{argmax}} \mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(i)},\sigma^2}[P(\boldsymbol{y},\boldsymbol{x};\boldsymbol{\gamma},\sigma^2)] \tag{20}$$

$$= \underset{\gamma_n}{\operatorname{argmax}} \mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(i)},\sigma^2}[P(\boldsymbol{y} \mid \boldsymbol{x};\sigma^2)P(\boldsymbol{x};\boldsymbol{\gamma})] \tag{21}$$

$$= \mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}^{(i)},\sigma^2}[x_n^2] \tag{22}$$

$$= \mu_n^2 + v_n \tag{23}$$

The EM algorithm will also be used to find the noise variance σ^2 in the case that it is unknown.

$$\sigma^{2^{(i+1)}} = \underset{\sigma^2}{\operatorname{argmax}} \mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma},\sigma^{2(i)}}[P(\boldsymbol{y},\boldsymbol{x};\boldsymbol{\gamma},\sigma^2)] \tag{24}$$

$$\tag{25}$$

The derivation is very similar to the original SBL one [2] and yields a very similar result:

$$\sigma^{2^{(i+1)}} = \frac{\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 - \sigma^2 \sum_{n=1}^{N}(1 - (\gamma_n)^{-1} v_n)}{M} \tag{26}$$

Definitions $F_n(K_n, c) = K_n(\frac{\gamma_n}{c+\gamma_n})$ $G_n(K_n, c) = (\frac{c.\gamma_n}{c+\gamma_n})$ $F'_n(K_n, c) = (\frac{\gamma_n}{c+\gamma_n})$
Message updates $K_n = \sum_{m=1}^{M} A^*_{mn} z_m + \mu_n$ $\mu_n = F_n(K_n, c)$ $v_n = G_n(K_n, c)$ $c = \sigma^2 + \frac{1}{M}\sum_{n=1}^{N} v_n$ $z_m^{(t)} = y_m - \sum_{n=1}^{N} A_{mn}\mu_n + \frac{z_m}{M}\sum_{n=1}^{N} F'_n(\mu_n, c)$
Parameter updates $\gamma_n = V_n + \mu_n^2$ $\sigma^{2^{(i+1)}} = \frac{\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2 - \sigma^2 \sum_{n=1}^{N}(1-(\gamma_n)^{-1} v_n)}{M}$

TABLE I: AMP SBL Algorithm

III. MMV AMP-TSBL

The original single vector SBL can be extended for the case of multiple measurement vectors (MMV). In [10] the case of independent measurements was studied. However, since the extension to the independent measurement case is straight forward, and it can be considered a special case of the time correlated multiple measurements that share a common support, we will only discuss the time correlated case in this section. The above factor graph can be extended in accordance to the following model:

$$\boldsymbol{y^{(t)}} = \boldsymbol{A}\boldsymbol{x^{(t)}} + \boldsymbol{e^{(t)}}, t = 1, 2...T \tag{27}$$

$$\boldsymbol{x^{(t)}} \in \mathbb{C}^{\mathbb{N}},\ \boldsymbol{y^{(t)}} \in \mathbb{C}^{\mathbb{M}} \tag{28}$$

The model can be restated as:

$$\bar{\boldsymbol{y}} = \boldsymbol{D}(\boldsymbol{A})\bar{\boldsymbol{x}} + \bar{\boldsymbol{e}} \tag{29}$$

Where $\bar{\boldsymbol{y}} \triangleq [\boldsymbol{y^{(1)}}^{\top}, \boldsymbol{y^{(2)}}^{\top} .., \boldsymbol{y^{(T)}}^{\top}]$, $\bar{\boldsymbol{x}} \triangleq [\boldsymbol{x^{(1)}}^{\top}, \boldsymbol{x^{(2)}}^{\top} .., \boldsymbol{x^{(T)}}^{\top}]$, $\bar{\boldsymbol{e}} \triangleq [\boldsymbol{e^{(1)}}^{\top}, \boldsymbol{e^{(2)}}^{\top} .., \boldsymbol{e^{(T)}}^{\top}]$ and $\boldsymbol{D(A)}$ is a diagonal matrix constructed from T replicas of $\boldsymbol{A}$.

$$P(\bar{\boldsymbol{x}} \mid \bar{\boldsymbol{y}}) \ \propto \ \prod_{t=1}^{T}\left[\prod_{m=1}^{M} P(y_m^{(t)} \mid \boldsymbol{x^{(t)}}) \prod_{n=1}^{N} P(x_n^{(t)} \mid x_n^{(t-1)})\right] \tag{30}$$

We can now construct the factor graph in figure 2.

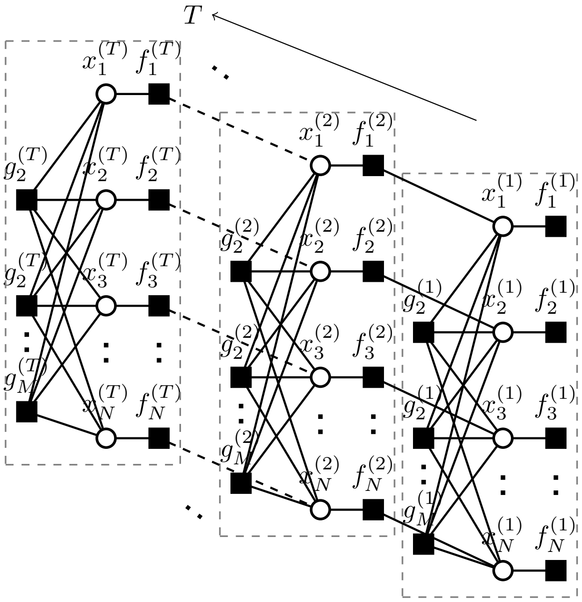

Fig. 2: AMP TSBL Factor Graph

Where:

$$g_m^{(t)}(\boldsymbol{x}) = P(y_m^{(t)} \mid \boldsymbol{x^{(t)}}) = \mathcal{N}(y_m^{(t)}; a_m^H \boldsymbol{x^{(t)}}, \sigma^2) \tag{31}$$

and we model the temporal correlation between measurement vectors by an AR(1) process, with correlation coefficient β:

$$f_n^{(t)}(x_n) = P(x_n^{(t)} \mid x_n^{(t-1)}) = \mathcal{N}(x_n^{(t)}; \beta x_n^{(t-1)}, (1-\beta^2)\gamma_n) \tag{32}$$

The derivation of messages within each measurement vector is very similar to the SMV case and will not be shown here due to space considerations. We will only show the derivation of messages passed between different measurement vectors. Again in this factor graph, all of the messages are Gaussian, and we only need to keep track of the mean and variance of each message. We start by finding the messages from factor nodes to neighboring variable node in the forward direction:

$$V_{f_n^{(1)} \to x_n^{(1)}} \propto \mathcal{N}(x_n^{(1)}; 0, \gamma_n) \tag{33}$$

$$V_{f_n^{(t)} \to x_n^{(t)}} \propto \mathcal{N}(x_n^{(t)}; \eta_n^{(t)}, \psi_n^{(t)}) \tag{34}$$

$$\propto \int \left(\prod_{l=1}^{M} V_{g_l^{(t-1)} \to x_n^{(t-1)}}\right) V_{f_n^{(t-1)} \to x_n^{(t-1)}} P(x_n^{(t)} \mid x_n^{(t-1)})\, \mathrm{d}x_n^{(t-1)} \tag{35}$$

$$\propto \int\Big(\prod_{l=1}^{M}\mathcal{N}(x_n^{(t-1)};\mu_n^{(t-1)},v_n^{(t-1)})\Big)\,\mathcal{N}(x_n^{(t-1)};\eta_n^{(t-1)},\psi_n^{(t-1)})$$
$$\times\mathcal{N}(x_n^{(t)};\beta x_n^{(t-1)},(1-\beta^2)\gamma_n)\,\mathrm{d}x_n^{(t-1)} \quad (36)$$

Where:

$$\mu_n^{(t)}=\sum_{l=1}^{M}A_{ln}^{*}z_{ln}^{(t)},\ v_n^{(t)}=c_n^{(t)}=\frac{1}{M}\sum_{m=1}^{M}c_{mn}^{(t)} \quad (37)$$

Using rules for Gaussian pdf multiplication and convolution we get:

$$\eta_n^{(t)}=\beta\left(\frac{\mu_n^{(t-1)}}{v_n^{(t-1)}}+\frac{\eta_n^{(t-1)}}{\psi_n^{(t-1)}}\right)\left(\frac{v_n^{(t-1)}\psi_n^{(t-1)}}{v_n^{(t-1)}+\psi_n^{(t-1)}}\right) \quad (38)$$
$$\psi_n^{(t)}=\beta^2\left(\frac{v_n^{(t-1)}\psi_n^{(t-1)}}{v_n^{(t-1)}+\psi_n^{(t-1)}}\right)+(1-\beta^2)\gamma_n \quad (39)$$

We now find the message updates between vectors in the backward direction:

$$V_{f_n^{(t+1)}\to x_n^{(t)}}\propto\mathcal{N}(x_n^{(t)};\theta_n^{(t)},\phi_n^{(t)}) \quad (40)$$

$$\propto\int\left(\prod_{l=1}^{M}V_{g_l^{(t+1)}\to x_n^{(t+1)}}\right)V_{f_n^{(t+2)}\to x_n^{(t+1)}}P(x_n^{(t+1)}\mid x_n^{(t)})\,\mathrm{d}x_n^{(t+1)} \quad (41)$$

$$\propto \int\Big(\prod_{l=1}^{M}\mathcal{N}(x_n^{(t+1)};\mu_n^{(t+1)},v_n^{(t+1)})\Big)\,\mathcal{N}(x_n^{(t+1)};\theta_n^{(t+1)},\phi_n^{(t+1)})$$
$$\times\mathcal{N}(x_n^{(t+1)};\beta x_n^{(t)},(1-\beta^2)\gamma_n)\,\mathrm{d}x_n^{(t+1)} \quad (42)$$

Using rules for Gaussian pdf multiplication and convolution again we get:

$$\theta_n^{(t)}=\frac{1}{\beta}\left(\frac{\mu_n^{(t+1)}}{v_n^{(t+1)}}+\frac{\theta_n^{(t+1)}}{\phi_n^{(t+1)}}\right)\left(\frac{v_n^{(t+1)}\phi_n^{(t+1)}}{v_n^{(t+1)}+\phi_n^{(t+1)}}\right) \quad (43)$$
$$\phi_n^{(t)}=\frac{1}{\beta^2}\left(\frac{v_n^{(t+1)}\phi_n^{(t+1)}}{v_n^{(t+1)}+\phi_n^{(t+1)}}+(1-\beta^2)\gamma_n\right) \quad (44)$$

Following the same framework in [6] Taylor series approximations are made to reduce the number of updates.

Finally, we use the EM algorithm to update the values of $\boldsymbol{\gamma}$, σ^2 and β. Table II summarizes one iteration of the AMP-TSBL algorithm. While the formula for the EM updates for β is not shown in the table, it follows standard EM derivation, and requires the solution of a cubic equation.

Definitions $F_n(K_n^{(t)},c^{(t)})=\frac{\frac{K_n^{(t)}}{c^{(t)}}+\frac{\eta_n^{(t)}}{\psi_n^{(t)}}+\frac{\theta_n^{(t)}}{\phi_n^{(t)}}}{\frac{1}{c^{(t)}}+\frac{1}{\psi_n^{(t)}}+\frac{1}{\phi_n^{(t)}}}$ $G_n(K_n^{(t)},c^{(t)})=\frac{1}{\frac{1}{c^{(t)}}+\frac{1}{\psi_n^{(t)}}+\frac{1}{\phi_n^{(t)}}}$ $F'_n(K_n^{(t)},c^{(t)})=\frac{\frac{1}{c^{(t)}}}{\frac{1}{c^{(t)}}+\frac{1}{\psi_n^{(t)}}+\frac{1}{\phi_n^{(t)}}}$
Message Updates $\eta_n^{(1)}=0$ $\psi_n^{(1)}=\gamma_n$ $for t=2:T$ $\eta_n^{(t)}=\beta(\frac{K_n^{(t-1)}}{c^{(t-1)}}+\frac{\eta_n^{(t-1)}}{\psi_n^{(t-1)}})(\frac{\psi_n^{(t-1)}c^{(t-1)}}{\psi_n^{(t-1)}+c^{(t-1)}})$ $\psi_n^{(t)}=\beta^2(\frac{\psi_n^{(t-1)}c^{(t-1)}}{\psi_n^{(t-1)}+c^{(t-1)}})+(1-\beta^2)\gamma_n$ $K_n^{(t)}=\sum_{m=1}^{M}A_{mn}^{*}z_m^{(t)}+\mu_n^{(t)}$ $\mu_n^{(t)}=F_n(K_n^{(t)},c^{(t)})$ $v_n^{(t)}=G_n(K_n^{(t)},c^{(t)})$ $c^{(t)}=\sigma^2+\frac{1}{M}\sum_{n=1}^{N}v_n^{(t)}$ $z_m^{(t)}=y_m-\sum_{n=1}^{N}A_{mn}\mu_n^{(t)}+\frac{z_m^{(t)}}{M}\sum_{n=1}^{N}F'_n(\mu_n^{(t)},c^{(t)})$ $for t=T-1:1$ $\theta_n^{(t)}=\frac{1}{\beta}(\frac{K_n^{(t+1)}}{c^{(t+1)}}+\frac{\theta_n^{(t+1)}}{\phi_n^{(t+1)}})(\frac{\phi_n^{(t+1)}c^{(t+1)}}{\theta_n^{(t+1)}+c^{(t+1)}})$ $\phi_n^{(t)}=\frac{1}{\beta^2}(\frac{\phi_n^{(t+1)}c^{(t+1)}}{\phi_n^{(t+1)}+c^{(t+1)}}+(1-\beta^2)\gamma_n)$
Parameter updates $\gamma_n=\frac{1}{T}[\sum_{t=1}^{T}V_n^{(t)}+(\mu_n^{(t)})^2-\beta^2\sum_{t=2}^{T}V_n^{(t-1)}+(\mu_n^{(t-1)})^2]$ $\sigma^2=\frac{1}{M\times T}\sum_{t=1}^{T}[\|\boldsymbol{y^{(t)}-Ax^{(t)}}\|^2-\sigma^2\sum_{n=1}^{N}(1-\frac{V_n^{(t)}}{\gamma_n})]$

TABLE II: AMP TSBL Algorithm

IV. Numerical Results

We conduct numerical experiments to compare the performance of the proposed algorithms to the original SBL [2], TSBL [3] and AMP-MMV [8] algorithms. The elements of $\boldsymbol{A}$ are generated according to $A_{mn}\sim\mathcal{N}(0,M^{-1})$. The source vector $\boldsymbol{x}_o$ was generated with K nonzero elements, and the source matrix $\boldsymbol{X}_o$ was generated with K nonzero rows. While the locations of the nonzero elements/rows were randomly chosen. We use two performance measures. The first measure is the normalized MSE. In the SMV case $NMSE\triangleq\mathbb{E}\{\|\hat{\boldsymbol{x}}-\boldsymbol{x}_o\|^2\}/\mathbb{E}\{\|\boldsymbol{x}_o\|^2\}$ and in the MMV case $NMSE\triangleq\mathbb{E}\{\|\hat{\boldsymbol{X}}-\boldsymbol{X}_o\|_{\mathcal{F}}^2\}/\mathbb{E}\{\|\boldsymbol{X}_o\|_{\mathcal{F}}^2\}$. We also use the runtime of each algorithm as a measure of complexity. In each experiment we run 100 iterations, with $N=200,\ M=100$ and $\lambda=\frac{K}{N}=.2$.

Figure 3 shows a comparison between NMSE of the original SBL and the proposed AMP-SBL over a range of SNR values. We can see that AMP-SBL performs slightly better than SBL. However, checking the complexity (runtime) in figure 4, we can see that a significant reduction in complexity is offered by AMP-SBL. The complexity difference will be even more significant at larger problem sizes, since for each iteration AMP-SBL requires $\mathcal{O}(M+N)$ linear operations, while the original SBL requires an inversion of a $M\times M$ matrix.

In figure 5 we run a comparison between AMP-TSBL, TSBL and AMP-MMV over a range of SNR values and correlation coefficient β values. The AMP-MMV algorithm uses a Bernoulli-Gaussian prior and requires several approximations to make all messages on the factor graph Gaussian. And it also requires more complicated message scheduling compared to AMP-TSBL because the factor graph it constructs contains more loops. From figure 5 we can see that the performance of AMP-TSBL is often comparable and sometimes better than the other two algorithms. However, comparing runtimes (figure 6) we can see a significant reduction in the case on AMP-TSBL compared to the other two algorithms. It is worth noting that the problem size we are using here is not very large, due to runtime limitations of the original TSBL. As the problem size grows both the AMP-TSBL and AMP-MMV complexities will grow linearly, which is not the case for TSBL that requires matrix inversions.

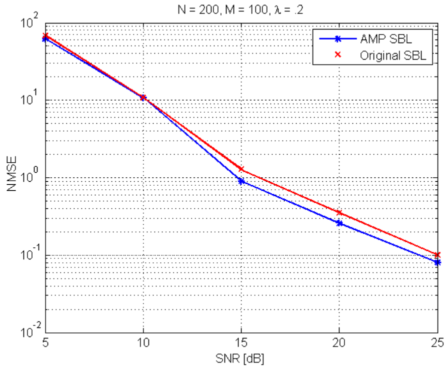

Fig. 3: MSE Comparison for SMV Algorithms

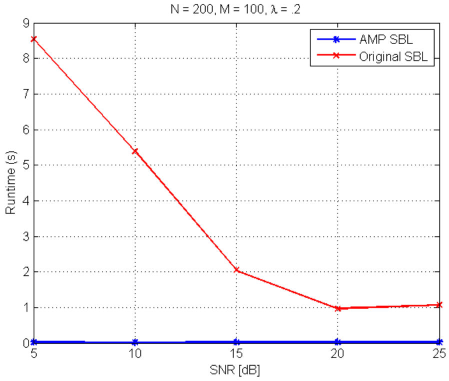

Fig. 4: Runtime Comparison for SMV Algorithms

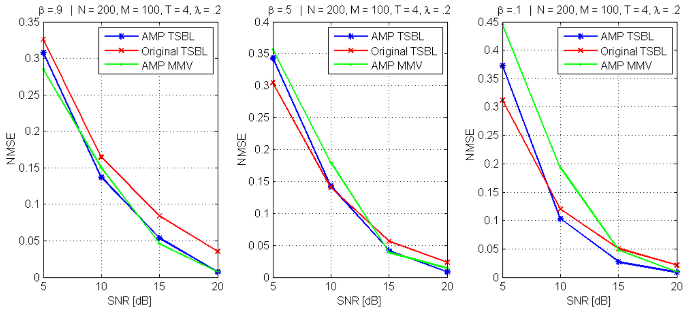

Fig. 5: MSE Comparison for MMV Algorithms

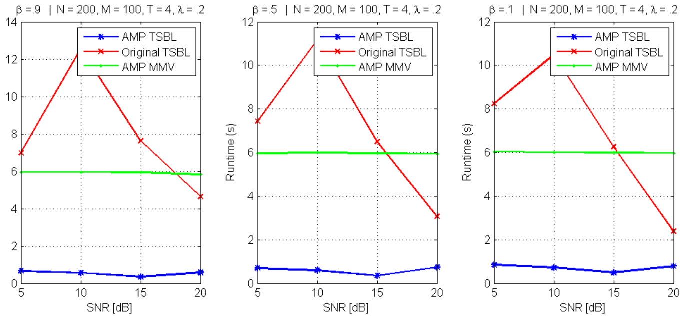

Fig. 6: Runtime Comparison for MMV Algorithms

V. Conclusion

In this paper we considered the problem of sparse signal recovery in the single measurement case and in the multiple measurement case with temporal correlation between measurements. We presented two algorithms that use the original assumptions of the SBL and TSBL algorithms. The proposed algorithms apply the approximate messaeg passing framework with these assumptions to significantly reduce the complexity of the original algorithms. We compared the proposed algorithms to the original SBL and TSBL algorithms, and to the AMP-MMV algorithm which also uses the AMP framework. We showed that while NMSE performance is comparable with previous algorithms, the new approach offers the lowest complexity level. The complexity reduction can be attributed to using the AMP framework with a Gaussian prior on $\boldsymbol{x}$ when conditioned on $\boldsymbol{\gamma}$. In addition to this complexity reduction, the proposed technique offers much simpler implementation when compared to other previously used Bayesian techniques.

Acknowledgment

This research was supported by the National Science Foundation grant CCF-1144258 and Ericsson endowed chair funds.

References

[1] D. L. Donoho , A. Maleki and A. Montanari, *"Message passing algorithms for compressed sensing,"* Proceedings of the National Academy of Sciences, vol. 106, no. 45, 2009.

[2] D.P. Wipf and B.D. Rao, *Sparse Bayesian Learning for Basis Selection,"* IEEE Transactions on Signal Processing, vol. 52, No. 8, August 2004

[3] Z. Zhang and B. D. Rao, *"Sparse Signal Recovery with Temporally Correlated Source Vectors Using Sparse Bayesian Learning,"* IEEE Journal of Selected Topics in Signal Processing, vol.5, no.5, 2011.

[4] M.E. Tipping, *Sparse Bayesian learning and the relevance vector machine"* Journal of Machine Learning, vol. 1, 2001.

[5] I. Gorodnitsky and B. D. Rao *" Sparse signal reconstruction from limited data using FOCUSS: a recursive weighted norm minimization algorithm,"* IEEE Transaction on Signal Processing, vol. 45, no. 3, 1997.

[6] D. Donoho, A. Maleki, and A. Montanari, *How to Design Message Passing Algorithms for Compressed Sensing* [Online]. Available: http://www.ece.rice.edu/mam15/bpist.pdf 2011.

[7] J. P. Vila and P. Schniter, *"Expectation-Maximization Gaussian-Mixture Approximate Message Passing,"* Proc. Conf. on Information Sciences and Systems, 2012

[8] J. Ziniel and P. Schniter, *"Efficient High-Dimensional Inference in the Multiple Measurement Vector Problem,"* IEEE Transactions on Signal Processing, vol. 61, no. 2, Jan. 2013.

[9] J.A. Palmer, K. Kreutz-Delgado, D.P. Wipf and B.D. Rao, *"Variational EM algorithms for non- gaussian latent variable models,"* Advances in Neural Information Processing Systems. MIT Press, Cambridge, 2006.

[10] D. P. Wipf and B. D. Rao, *"An empirical Bayesian strategy for solving the simultaneous sparse approximation problem,"* IEEE Transactions on Signal Processing, vol. 55, no. 7, 2007.

[11] P. Schniter, *"Turbo reconstruction of structured sparse signals,"* Proc. Conf. on Information Sciences and Systems, 2010