Rectified Gaussian Scale Mixtures and the Sparse Non-Negative Least Squares Problem
Alican Nalci, Igor Fedorov, Maher Al-Shoukairi, Thomas T. Liu, and Bhaskar D. Rao

Abstract—In this paper, we develop a Bayesian evidence maximization framework to solve the sparse non-negative least squares problem (S-NNLS). We introduce a family of scale mixtures referred to as Rectified Gaussian Scale Mixture (R-GSM) to model the sparsity enforcing prior distribution for the signal of interest. Through proper choice of the mixing density, the R-GSM prior encompasses a wide variety of heavy-tailed distributions such as the rectified Laplacian and rectified Student-t distributions. Utilizing the hierarchical representation induced by the scale mixture prior, an evidence maximization or Type II estimation method based on the expectation-maximization (EM) framework is developed to estimate the hyper-parameters and to obtain a point estimate of the parameter of interest. In the proposed method, called rectified Sparse Bayesian Learning (R-SBL), we provide four alternative approaches that offer a range of options to trade-off computational complexity to quality of the E-step computation. The methods include the Markov Chain Monte Carlo EM, linear minimum mean square estimation, approximate message passing and a diagonal approximation. Through numerical experiments, we show that the proposed R-SBL method outperforms existing S-NNLS solvers in terms of both signal and support recovery.

Index Terms—Non-negative Least Squares, Sparse Bayesian learning, Sparse Signal Recovery, Rectified Gaussian Scale Mixtures

I. INTRODUCTION

We consider the linear system of equations of the following form:

$$y = \Phi x + v,$$

where the solution vector $x \in \mathbb{R}_+^M$ is assumed to be non-negative. The matrix $\Phi \in \mathbb{R}^{N \times M}$ is fixed and obtained from the physics of the underlying problem, $y \in \mathbb{R}^N$ is the measurement vector, and $v$ is the additive measurement noise modeled as a zero mean Gaussian random vector with uncorrelated entries i.e. $v \sim \mathcal{N}(0, \sigma_v^2 I)$. Recovering the optimal solution to Eq. (1) is known as solving the non-negative least squares (NNLS) problem. NNLS has received considerable attention in the context of methods for solving systems of linear equations [1], density estimation [2], compressive non-negative imaging [3], and non-negative matrix factorization [4–6]. Non-negative data is also a natural occurrence in many application areas, such as text mining [7], image processing [8], speech enhancement [9], and spectral decomposition [10, 11] and requires special consideration. The solution for the NNLS problem can be recovered by solving the optimization problem:

$$\min_{x \geq 0} \| y - \Phi x \|_2. \quad (2)$$

In many applications, $N < M$ and Eq. (1) is under-determined. As a result, a unique solution to Eq. (2) may not exist. Recovering a unique solution is possible if more information is known a-priori about the solution vector. A useful assumption, and one that has been recently become very popular is that the solution is sparse (i.e. has few non-zero elements) [12, 13, 14]. In this case, the optimal sparsest solution (e.g. assuming a noiseless scenario) can be recovered by modifying Eq. (2) to:

$$\min_{x \geq 0} \| y - \Phi x \|_0 \quad (3)$$

where the norm of $\| x \|_0$ is the $\ell_0$ pseudo-norm, which is the number of non-zero elements in $x$ (e.g. this number is also referred to as cardinality of the solution). Therefore, the objective is to minimize the number of non-zero elements in $x$, while still satisfying the optimization constraints. We refer to Eq. (3) as the sparse NNLS (S-NNLS) problem.

Directly solving Eq. (3) is not tractable, since the $\ell_0$ pseudo-norm is not convex and the problem is NP-hard [15]. Therefore, various ‘greedy’ algorithms have been been proposed to approximate the solution [15, 16]. One example is the class of algorithms known as Orthogonal Matching Pursuit (OMP) [17], which successively select non-zero elements of $x$ in a greedy fashion. In order to adapt OMP to the S-NNLS problem, the criterion by which a new non-zero element of $x$ is selected is modified to select the one having the largest positive value [18]. Another approach in this class of algorithms first finds an $x$ such that $\| y - \Phi x \|_2 \leq \epsilon$ and $x \geq 0$ using the active-set Lawson-Hanson algorithm [11] and then prunes $x$ in a greedy fashion until $\| x \|_0 \leq K$, where $K$ is a pre-specified desired cardinality [4].

Greedy algorithms are computationally attractive to solve NP-hard problems, but may lead to sub-optimal solutions [15]. Therefore, convex relaxations of the $\ell_0$ penalty have been proposed [15]. One simple alternative is to replace the $\ell_0$ penalty on $x$ and modify the problem in Eq. (3) as:

$$\min_{x \geq 0} \| y - \Phi x \|_2 + \lambda \| x \|_1 \quad (4)$$

where $\lambda > 0$ is a suitably chosen regularization parameter to account for the measurement noise. The advantage of this
formulation is that, it is in the form of a constrained convex optimization problem and can be solved by a number of methods [19], [20]. One approach is to estimate $x$ through a projected gradient descent procedure [21]. In fact, the $\ell_1$ norm penalty in Eq. (4) can be replaced by an arbitrary sparsity inducing surrogate function $g(x)$, thereby leading to alternative methods based on solving the following optimization problem:

$$\minimize_{x \geq 0} \| y - \Phi x \|_2 + \lambda g(x).$$

(5)

For instance, a surrogate function $g(x) = \sum_{i=1}^M \log (x_i^2 + \beta)$ has been considered [22], [23], which leads to an iterative reweighted optimization approach.

An alternative and perhaps a promising view on the S-NNLS problem is to cast the entire problem in a Bayesian framework and consider the maximum a-posteriori (MAP) estimate of $x$ given the data $y$:

$$x_{MAP} = \arg \max_x p(x|y).$$

(6)

There is a strong connection between the MAP framework in Eq. (6) and deterministic formulations like the one in Eq. (5). It has recently been shown that formulations of the form in Eq. (3) are equivalent to the formulation in Eq. (6) with the proper choice of prior $p(x)$ [24]. For example, considering a separable $p(x)$ of the form

$$p(x) = \prod_{i=1}^M p(x_i),$$

(7)

the $\ell_1$ regularization approach in Eq. (4) is equivalent to the Bayesian approach in Eq. (6) with an exponential prior for $x_i$. In this paper, our emphasis will be on Bayesian approaches for solving Eq. (1) for $x$. We will introduce a novel class of priors called a rectified Gaussian scale mixture for $x_i$, and detail our rectified sparse Bayesian learning (R-SBL) approach to solve the S-NNLS problem.

A. Organization of the paper

In the following sections, we provide details of the proposed R-SBL algorithm. In Section II we discuss the advantages of scale mixture priors for $p(x)$ and introduce the Rectified Gaussian Scale Mixture (R-GSM) prior. In Section III we define the Type I and Type II Bayesian approaches to solve the S-NNLS problem and introduce the main R-SBL framework with the R-GSM prior. We provide details of an evidence maximization based estimation procedure that utilizes the EM algorithm in Section III-B. We present experimental results comparing the proposed R-SBL algorithm to existing methods in Section V. Finally, we conclude the work in Section VI.

II. RECTIFIED GAUSSIAN SCALE MIXTURES

In this work, we assume separable priors of the form in Eq. (7) and focus on the choice of $p(x_i)$. The choice of prior is very important because it plays a central role in the Bayesian inference procedure [25], [26], [27]. For the problem at hand, the prior must be sparsity inducing and satisfy the non-negativity constraints. Consequently, we consider the scale mixture prior:

$$p(x_i) = \int_0^\infty p(x_i|\gamma)p(\gamma)d\gamma.$$

(8)

Scale mixture priors were first considered in the context of Gaussian Scale Mixtures (GSM) where $p(x_i|\gamma) = N(x_i; 0, \gamma)$ [28]. It is well known that super-gaussian densities are suitable priors for promoting sparsity [25], [29] and such priors can be mostly represented in the form shown in Eq. (3), with the proper choice of $p(\gamma)$ [30], [31], [32], [33], [34]. This has made GSM based priors a valuable form of prior for the general sparse signal recovery problem. Another advantage of the scale mixture prior is that it establishes a Markovian structure of the form,

$$\gamma \rightarrow x \rightarrow y$$

(9)

where inference can be performed in the $x$ domain (referred as to MAP or Type I) and also in the $\gamma$ domain (referred as to Type II). The Markovian structure provides additional flexibility and opportunity to explore algorithm development. Experimental results in the standard sparse signal recovery problem show that performing inference in the $\gamma$ domain consistently achieves superior performance [24], [25], [35].

The Type II procedure involves finding a MAP estimate of $\gamma$ and approximating the posterior $p(x|y)$ by $p(x|y; \gamma_{MAP})$. The performance gains can be understood by noting that $\gamma$ is deeper in the Markovian chain than $x$, so the influence of errors in performing inference in the $\gamma$ domain may be diminished [24], [35], [36]. At the same time, $\gamma$ is close enough to $y$ in the Markovian chain such that meaningful inference about $\gamma$ can still be performed mitigating the problem of local minima that is more prevalent when seeking a MAP estimate of $x$, i.e Type I [35].

Although priors of the form shown in Eq. (8) have been used widely in the compressed sensing literature (where the signal model is identical to Eq. (1) without the non-negativity constraint) [24], this framework has not been extended to the S-NNLS problem. In [37], a rectified Gaussian (RG) prior is considered for $p(x_i)$ within a variational Bayesian inference framework. Our goal in this work is to develop a more flexible and general class of priors. Considering the findings that the scale mixture prior has been useful for the development of sparse signal recovery algorithms [24], [35], we propose a R-GSM prior for the S-NNLS problem, where $p(x_i|\gamma_i)$ in Eq. (8) is given by the RG distribution. We refer to the proposed Type II inference framework as R-SBL.

The RG distribution is defined as

$$N^R(x; \mu, \gamma) = \sqrt{\frac{2}{\pi\gamma}} e^{-\frac{2\gamma}{\mu^2}} u(x),$$

(10)

where $\mu$ is the location parameter (i.e. and is not the mean), $\gamma$ is the scale parameter, $u(x)$ is the unit step function, and
erfc(x) is the complementary error function, defined as
\[
erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt.
\] (11)

When \( \mu = 0 \), the RG density becomes:
\[
\mathcal{N}^R(x;0,\gamma) = \sqrt{\frac{2}{\pi \gamma}} e^{-\frac{x^2}{2\gamma}} u(x).
\] (12)

As noted in [38], [39], closed form inference computations using scale mixtures of RG’s is tractable only if the location parameter is zero. Thus, in this work, we focus on R-GSM with \( \mu = 0 \) to provide a general class of priors that better promote sparse non-negative solutions. R-GSM priors have the form
\[
p(x) = \int_0^\infty \mathcal{N}^R(x;0,\gamma)p(\gamma)d\gamma.
\] (13)

Different choices of \( p(\gamma) \) lead to different choices of priors and some examples are presented below.

A. Examples of R-GSM representation of sparse prior

A random variable modeled by a R-GSM can also be viewed as generating a random variable with a GSM and then taking its absolute value. This allows us to leverage the GSM prior literature to develop the class of R-GSM priors. For instance, consider the rectified Laplacian, prior for \( x \):
\[
p_e(x) = \lambda e^{-\lambda |x|} u(x).
\]

By using an exponential prior for \( \gamma \),
\[
p(\gamma) = \frac{\lambda^2}{2} e^{-\frac{\lambda^2}{2}} u(\gamma),
\] (14)

we can express \( p_e(x) \) in the R-GSM form [40] as
\[
p_e(x) = 2u(x) \int_0^\infty \mathcal{N}(x|0,\gamma_i)\frac{\lambda^2}{2} e^{-\frac{\lambda^2}{2}} u(\gamma)d\gamma
= \lambda e^{-\lambda |x|} u(x).
\]

Likewise, by considering the R-GSM with \( p(\gamma) \) given by the Gamma\((a,b)\) distribution, we get a rectified Student-t distribution for \( p(x) \). Then, Eq. (8) simplifies as in [28] to
\[
p(x) = 2u(x) \int_0^\infty \mathcal{N}(x|0,\gamma)\frac{\lambda^2}{a^b \Gamma(a)} e^{-\frac{\lambda^2}{2}} u(\gamma)d\gamma
= \frac{2b \Gamma(a + \frac{1}{2})}{(\pi)^{\frac{1}{2}} \Gamma(a)} \left( b + \frac{x^2}{2} \right)^{-a-\frac{1}{2}} u(x),
\]
where \( \Gamma(a) = \int_0^\infty t^{a-1} e^{-t}dt. \)

More generally, all of the distributions represented by the GSM have a corresponding rectified version represented by a R-GSM, e.g. contaminated Normal and slash, symmetric stable and logistic, hyperbolic, etc. [28], [30], [31], [32], [33], [34].

III. BAYESIAN INFERENCE WITH SCALE MIXTURE PRIOR

There are two main Bayesian approaches for solving the S-NNLS problem with an R-GSM prior. These approaches are referred to as Type I and Type II. Though the majority of this paper is dedicated to Type II estimation because of its superior performance in sparse signal recovery problems [24], [45], we briefly introduce Type I in the following section for the sake of completeness.

A. Type I estimation

Using Type I to solve the S-NNLS problem translates into calculating the maximum a-posteriori (MAP) estimate of \( x \) given \( y \), i.e. by optimizing:
\[
\arg \min_x \| y - \Phi x \|^2 - \lambda \sum_{i=1}^M \ln p(x_i). \] (15)

Some of the \( \ell_0 \) relaxation methods described in Section I can be derived from a Type I perspective. For instance, by choosing an exponential prior for \( p(\gamma_i) \) Eq. (15) reduces to the \( \ell_1 \) regularization approach in Eq. (4) with the interpretation of \( \lambda \) as being determined by the parameters of the prior and the noise variance. Similarly, choosing a Gamma prior for \( p(\gamma_i) \) Eq. (15) reduces to:
\[
\arg \min_x \| y - \Phi x \|^2 + \lambda \sum_{i=1}^M \ln \left( b + \frac{x_i^2}{2} \right)
\]
which leads to the re-weighted \( \ell_2 \) approach to the S-NNLS problem described in [22], [23]. A unified Type I approach for the R-GSM prior can be readily derived using the approaches discussed in [24], [40].

B. Type II estimation

The Type II framework aims to infer the hyper-parameters \( \gamma \) that shape the R-GSM prior. In this framework, a MAP estimate of \( \gamma \) is computed and then the posterior density of interest \( p(x|y) \) is approximated as \( p(x|y,\gamma_{MAP}) \). Then appropriate point estimates can be obtained. We refer to this Type II framework as Rectified Sparse Bayesian Learning (R-SBL).

Several strategies exist for estimating \( \gamma \). The first strategy considers the problem of forming a MAP estimate of \( \gamma \) given \( y \) by directly minimizing the appropriate posterior density. In our case, \( p(\gamma | y) \) does not admit a closed form. The second strategy, investigated here, aims to estimate \( \gamma \) by utilizing an EM framework. In the EM approach, we treat \( (x,y,\gamma) \) as the complete data and \( x \) as the hidden variable. Utilizing the current estimate \( \gamma^t \), where \( t \) refers to the iteration index, the expectation step (E-step) involves determining the expectation of the log-likelihood \( Q(\gamma, \gamma^t) \) defined as:
\[
Q(\gamma, \gamma^t) = E_{x|y,\gamma^t} \left[ \ln p(y|x) + \ln p(x|\gamma) + \ln p(\gamma) \right]
\] (16)
\[
d\equiv \sum_{i=1}^M E_{x|y,\gamma^t} \left[ -\frac{1}{2} \ln \gamma_i - \frac{x_i^2}{2\gamma_i} + \ln p(\gamma_i) \right]
\] (17)
where \( d \) refers to the fact that constant terms and terms which don’t depend on \( \gamma \) have been dropped since they do not effect
the M-step. For simplicity, we assume a non-informative prior on $\gamma$ [25]. In the M-step, we maximize $Q(\gamma, \gamma^t)$ with respect to $\gamma$ by taking the derivative and setting it equal to zero, which yields an estimate of $\gamma$:

$$\gamma^{t+1} = E[x_i|y, \gamma^t, \sigma^2] =: \langle x_i \rangle$$  \hspace{1cm} (18)

To compute $\langle x_i^2 \rangle$, we first consider the multivariate posterior density $p(x|y, \gamma, \sigma^2)$ which can be shown to have the form (e.g. see Appendix VII-B for details):

$$p(x|y, \gamma) = c(y)e^{-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}} u(x)$$  \hspace{1cm} (19)

where $\mu$ and $\Sigma$ are given by [25], [41], [42] as

$$\mu = \Gamma \Phi^T (\sigma^2 I + \Phi \Gamma \Phi^T)^{-1} y$$  \hspace{1cm} (20)

$$\Sigma = \Gamma - \Gamma \Phi^T (\sigma^2 I + \Phi \Gamma \Phi^T)^{-1} \Phi \Gamma$$  \hspace{1cm} (21)

and $\Gamma = \text{diag}(\gamma)$. The posterior in Eq. (19) is known as a multivariate RG (or a multivariate truncated normal [43]). The normalizing constant $c(y)$ in the multivariate RG is a function of $y$ and does not have an immediate closed form expression. Note that the M-step in Eq. (18) only requires the marginal density. However, the marginals of a multivariate RG are not univariate-RG’s, and also do not admit a closed form expression [43], which unfortunately means no closed form for marginal moments.

However, we can approximate the first and the second moments $\langle x_i \rangle$ and $\langle x_i^2 \rangle$ of the exact multivariate RG posterior. In the next section we propose four different approaches that offer a trade-off between computational complexity and theoretical accuracy.

1) Markov Chain Monte Carlo EM (MCMC-EM):

Advances in numerical methods make it possible to sample from complex multivariate distributions [44], [45], [46]. Numerical methods are particularly useful when the first and second order statistics of a posterior density do not have a closed form expression. In this case, the E-step of the EM approach can be performed by drawing samples using MCMC methods and then calculating the sample statistics from those samples. This approach is usually referred as Monte Carlo EM (MCMC-EM) [47], [48]. We first consider the classical Gibbs sampling approach [49], [50]. We use hat notation to refer to the estimates of various parameters (e.g. $\hat{\Sigma}$).

By introducing the transformation, $w = L^{-1}(x - \hat{\mu})$ where $L$ is the lower triangular Cholesky decomposition of $\Sigma$, it can be shown that $w$ has a truncated normal (TN) distribution $\text{TN}(w; 0, L, -\hat{\mu}, \infty)$ as in [50],

$$\text{TN}(w; \hat{\mu}, \hat{\Sigma}, \hat{\alpha}_L, \hat{\alpha}_U) = \left( \begin{array}{c} c_{tn} e \frac{(x - \hat{\mu})^T\Sigma^{-1}(x - \hat{\mu})}{2} 1_{\alpha_L \leq Lw \leq \alpha_U} \end{array} \right)$$  \hspace{1cm} (22)

and $1_{(\cdot)}$ is the indicator function and $c_{tn}$ is the normalizing constant. The Gibbs sampler then proceeds by iteratively drawing samples from the conditional distribution $p(w_i|y, \hat{\gamma}, \sigma^2, w_{-i})$, where $w_{-i}$ refers to the vector containing all but the $i$th element of $w$. Given a set of samples drawn from $w$, we can generate samples from the original distribution of interest by inverting the transformation: $\{x^n\}_{n=1}^N = \{L w^n + \hat{\mu}\}_{n=1}^N$. Then, the first and second empirical moments can be calculated from the drawn samples using:

$$\langle \hat{x}_i \rangle = \frac{1}{N} \sum_{n=1}^N (x^n_i),$$  \hspace{1cm} (24)

$$\langle \hat{x}_i^2 \rangle = \frac{1}{N} \sum_{n=1}^N (x^n_i)^2,$$  \hspace{1cm} (25)

and the EM step can be iterated for convergence by updating $\hat{\gamma}_i^{t+1} = \langle \hat{x}_i^2 \rangle$.

After convergence of the MCMC-EM, a point estimate for $x$ is needed as the solution. The optimal estimator of $x$ in the mean-square-error (MSE) sense is simply $\hat{x}_{\text{mean}} = \langle \hat{x}_i \rangle$. An alternative point estimate is to use $\hat{x}_{\text{mode}}$ given by

$$\hat{x}_{\text{mode}} = \arg \max_x p(x|y, \hat{\gamma}, \sigma^2)$$  \hspace{1cm} (26)

$$= \arg \min_{x \geq 0} ||y - \Phi x||^2 + \lambda \sum_{i=1}^M \hat{x}_i^2$$

where Eq. (26) can be solved by any NNLS solver. $\hat{x}_{\text{mode}}$ could be a favorable point estimate because it chooses the peak of $p(x|y, \hat{\gamma}, \sigma^2)$, which could be multi-modal and not characterized well by its mean particularly when a sparse solution is of interest.

For the sparse recovery problem at hand, we experienced very slow convergence with the Gibbs sampling. Convergence was particularly slower for higher problem dimensions and at larger cardinalities. This was expected as the sparse solution is harder to recover in those cases. Therefore, we resorted to Hamiltonian Monte Carlo (HMC) sampling designed specifically for target spaces constrained by linear or quadratic constraints [46]. HMC incorporates Hamiltonian dynamics to improve the MCMC mixing performance by incorporating the gradient information of the target distribution [47].

Despite the use of state of the art MCMC techniques, MCMC-EM might still converge to local minima solutions resulting in sub-optimal performance [51], [52], [53]. Particularly, performance may be even poorer for under-determined problems with $N < M$. Though MCMC-EM is not thoroughly investigated for sparse recovery problems, here we list four major issues below for the MCMC-EM technique for our sparse recovery problem for consideration:

I. Convergence: MCMC-EM based algorithms can get stuck in a local minima depending on the problem dimensions and complexity of the search space. This is true even for well-posed problems [48], [54]. In the case of under-determined problems the solution set in Eq. (2) contains many local minima and thus, a good MCMC-EM implementation should try to avoid sub-optimal solutions.

II. Computational Limits: Current MCMC sampling techniques for multivariate Rectified Gaussian distributions
are not optimal for sampling large samples sizes $n$ from high dimensional multivariate posterior densities, and therefore the number of available samples is often limited by computational constraints [44], [45], [46].

III. Quality of the Sufficient Statistics Estimates: Since the MCMC samples are determined by random sampling at each iteration, the estimates of $\gamma$, $\mu$, and $\Sigma$ depend highly on the quality of the MCMC estimates $\hat{x}$, which in turn affects the quality of future MCMC samples. This may lead EM algorithm to converge to a sub-optimal solution.

IV. Structure of Empirical $\hat{\Sigma}$: When $M$ is large and the dimensions of the empirical scale matrix is also large, $\Sigma$ may be no longer a good estimate [52], [53], [54]. This could happen when the problem is inherently ill-posed (e.g. $N < M$), and reveals itself as $\Sigma$ being close to singular. Therefore, regularization methods for $\Sigma$ are often needed to alleviate this problem [52], [53].

The scale matrix $\Sigma$ has direct control over the search space for MCMC and spurious off-diagonal values tend to increase the number of local-minima. Therefore, an affine simple LMMSE estimate of $x$ is then calculated as:

$$\hat{x} = \mu_x + R_x \Phi^T (\Phi R_x \Phi^T + \sigma^2 I)^{-1} (y - \Phi \mu_x),$$  
(27)

where $R_x$ is the covariance matrix of $x$ and is a diagonal matrix in this case. The estimation error covariance matrix is given by [57]

$$R_e = R_x - R_x \Phi^T (\Phi R_x \Phi^T + \sigma^2 I)^{-1} \Phi R_x.$$  
(28)

To elaborate, in the E-step where $\gamma$ is fixed at $\gamma^t$, the entries for $x$ are independent entries, and the prior mean and the prior covariance will be equal to the mean and variance of the independent univariate RG distributions $p(x_i|\gamma_i) = N_R(0, \gamma_i)$. The mean of a univariate rectified Gaussian density with zero location parameter is given by [58]

$$\mu_{x,i} = \sqrt{\frac{2\gamma_i}{\pi}},$$  
(29)

and the variances, which are the diagonal entries of the diagonal matrix $R_x$, is given by

$$R_{x,ii} = \gamma_i (1 - 2/\pi).$$  
(30)

Using the values of $\mu_x$ and $R_x$ from Eq. (29) and Eq. (30) in Eq. (27) we obtain the LMMSE point estimate for the solution vector. Similarly, the update for $\gamma$ (M-step) is given by:

$$\gamma_i = \hat{x}_i^2 + R_{e,ii}.$$  
(31)

This is sufficient to implement the EM algorithm. Upon convergence of values the final mean point estimate is simply $\hat{x}_{\text{mean}} = \hat{x}$, and the mode point estimate can be obtained by utilizing the converged values $\gamma_i$ in Eq. (26).

3) Generalized approximate message passing (GAMP):

In this section, we present an EM implementation using the generalized approximate message passing algorithm (GAMP) [59]. A GAMP based non-negative Gaussian mixture approach was previously used in [31]. However, the model in [31] is different in the sense that it uses an i.i.d. Bernoulli non-negative Gaussian mixture prior, with a fixed and known mixture order that is independent of $M$. A detailed comparison of the R-GSM prior with the alternate choices will be subject of our future work. GAMP is a low complexity iterative inference algorithm. The low complexity is achieved by applying quadratic and Taylor series approximations to loopy belief propagation. Under assumptions of the prior $p(x)$ and the likelihood $p(y|x)$, GAMP can approximate the minimum mean square error (MMSE) estimate when used in the sum-product mode, or it can approximate the MAP estimate when used in its max-sum mode.

The sum-product version of the algorithm computes the mean and variance of approximate marginal posteriors on $x_i$, the approximate posteriors are given by:

$$p(x_i|r_i; \tau_{r_i}) \propto p(x_i)N(x_i; r_i, \tau_{r_i}),$$  
(32)

where $r_i$ approximates a corrupted version (AWGN) of the true $x_i$,

$$r_i = x_i + \tilde{r}_i$$  
(33)

$$\tilde{r}_i \sim N(0, \tau_{r_{r_i}})$$  
(34)

This approximation was shown to be exact in the large system limit, when $\Phi$ is i.i.d sub-Gaussian [59], [60]. Therefore, in sum-product mode, GAMP sets $\hat{x}_i$ at the conditional mean in Eq. (33) that corresponds to the MMSE given $r_i$, and it sets $\tau_{r_{r_i}}$ at the conditional variance in Eq. (34) which corresponds to the MMSE given $r_i$:

$$\hat{x}_i = \mathbb{E}\{x_i|r_i; \tau_{r_i}\}$$  
(35)

$$\tau_{x_i} = \text{var}\{x_i|r_i; \tau_{r_i}\}$$  
(36)
Similarly, the max-sum version of GAMP algorithm sets $\hat{x}_i$ using the proximal operator in Eq. (37) at the MAP estimate given $r_i$, and it sets $\tau_{x_i}$ at the corresponding sensitivity of the proximal thresholding given $r_i$ in Eq. (38).

$$\hat{x}_i = \text{prox}_{-\ln p(x_i)}(r_i; \tau_{x_i})$$  (37)

$$\tau_{x_i} = \tau_{x} \text{prox}'_{-\ln p(x_i)}(r_i; \tau_{x_i})$$  (38)

$$\text{prox}_f(\hat{u}, \tau^0) \triangleq \arg \min_{x \in \mathbb{R}} f(x) + \frac{1}{2\tau^0} |x - \hat{u}|^2$$  (39)

When implementing the EM algorithm, the approximate posterior computed by the sum-product GAMP can be used to efficiently approximate the E-step 61. Moreover, in the case of max-sum GAMP, even though the algorithm does not provide marginal distributions, because in the large system limit and under i.i.d sub-Gaussian $\Phi$, the assumption on $r_i$ being an AWGN corrupted version of the true $x_i$ holds 60. Then similar to the approach in 33, an extra step can be added to compute marginal distributions using Eq. (32), and the computed marginals can be used to approximate the E-step. For the assumed rectified Gaussian scale mixture prior $p(x|\gamma)$ the details of finding $\hat{x}_i$ and $\tau_{x_i}$ estimates in both the sum-product and max-sum cases are shown in Appendix VII-A.

Upon the convergence of the GAMP algorithm, the approximate E-step of the EM algorithm is complete, and we can proceed to evaluate the M-step in Eq. (18) as follows:

$$\langle x_i^2 \rangle = \int_{x_i} x_i^2 p(x_i; \tau_{x_i}) = \tau_{x_i} + \hat{x}_i^2$$  (40)

The R-SBL GAMP algorithm based on the E and M-steps described above is summarized Table I. We note that for the AWGN case, the output function of the GAMP algorithm which is used to evaluate $s$ and $\tau_s$ in Table I is the same for both sum-product and max-sum GAMP and is stated in 59. Moreover, to overcome the known convergence issues with the GAMP algorithm when a non-i.i.d. transformation matrix $\Phi$ is used 62, 63, 64, we incorporate the damping technique used in 64 into the proposed R-SBL GAMP algorithm.

In Table I all vector squares, divisions and multiplications are taken element wise. $K_{\max}$ is the maximum allowed number of GAMP iterations, $\epsilon_{\text{gamp}}$ is the GAMP normalized tolerance parameter. $I_{\max}$ is the maximum allowed number of EM iterations and $\epsilon_{\text{em}}$ is the EM normalized tolerance parameter.

4) Diagonal approximation (DA):

We know a-priori that the posterior density in Eq. (19) does not admit a closed form expression. However, to successively implement an EM framework for R-SBL we only need the marginals of the posterior. We first note that if $\Sigma$ was a diagonal matrix then we could evaluate the normalizing constant $c(y)$ in closed form since the multivariate RG posterior can be written as a product of marginals (e.g. see Appendix VII-B for details). In the diagonal approximation approach, we resort to approximating the posterior in Eq. (19) with a suitable posterior density $\tilde{p}(x|y, \gamma)$ which could be written as a product of independent marginal densities (i.e. $\tilde{p}(x_i|y_i, \gamma_i)$). This density is derived in Appendix VII-B as:

$$\tilde{p}(x | y, \gamma) = \prod_{i=1}^{M} \tilde{p}(x_i | y_i, \gamma_i)$$  (41)

$$= \prod_{i=1}^{M} \sqrt{\frac{2}{\pi \Sigma_{ii}}} \frac{e^{-\frac{(x_i - \mu_i)^2}{2\Sigma_{ii}}}}{\text{erfc} \left( - \frac{\mu_i}{\sqrt{2\Sigma_{ii}}} \right)}$$  (42)

where $\mu_i$ is the $i$th element of $\mu$ and $\Sigma_{ii}$ is the $i$th diagonal element of $\Sigma$ obtained using Eqs. (20) and (21). The marginal density $\tilde{p}(x_i|y_i, \gamma_i)$ in Eq. (42) is exact the univariate RG density defined in Eq. (16). In other words we have:

$$\tilde{p}(x_i|y_i, \gamma_i) = N(x_i; \mu_i, \Sigma_{ii}).$$  (43)

Then, the RG marginals are well-characterized by their first and second moments given in 53, with the first moment given as:

$$\langle x_i \rangle = \mu_i + \sqrt{\frac{2\Sigma_{ii}}{\pi}} \frac{e^{-\frac{x_i^2}{2\Sigma_{ii}}}}{\text{erfc} \left( - \frac{\mu_i}{\sqrt{2\Sigma_{ii}}} \right)}$$  (44)

and the second moment is given as:

$$\langle x_i^2 \rangle = \mu_i^2 + \Sigma_{ii} + \mu_i \sqrt{\frac{2\Sigma_{ii}}{\pi}} \frac{e^{-\frac{x_i^2}{2\Sigma_{ii}}}}{\text{erfc} \left( - \frac{\mu_i}{\sqrt{2\Sigma_{ii}}} \right)}.$$  (45)

\begin{table}
\centering
\begin{tabular}{|l|}
\hline
\textbf{TABLE I: R-SBL GAMP Algorithm} \\
\hline
\end{tabular}
\end{table}
Note that the moments of \( \tilde{p}(x_i|y_i, \gamma_i) \) are approximations to the moments of the exact marginals. However, since the approximate marginals admit closed form expressions, we can perform EM to approximate the true solution. EM can be carried out by setting \( \gamma_i^{t+1} = \langle x_i^2 \rangle \) and iterating over \( t \). After convergence of \( \gamma_i \)'s to their final values, a point estimate for \( x \) is needed from \( \tilde{p}(x|y, \gamma, \sigma^2) \). The mean point estimate is given as \( \hat{x}_{\text{mean}} = \langle x_i \rangle \), where this is computed using the converged values of \( \mu_i \) and \( \Sigma_{ii} \). The mode point estimate \( \hat{x}_{\text{mode}} \) can be calculated by using converged values of \( \gamma_i \)'s in Eq. (26). We refer to this EM approach as the diagonal approximation or DA for short.

We again emphasize that the DA is an approximate EM method for the true posterior density \( p(x_i|y_i, \gamma_i) \). DA is an exact EM solution to the R-SBL if \( \Sigma \) is already diagonal. Furthermore, if diagonal elements of \( \Sigma \) are large valued, or becomes large over EM iterations as compared to off-diagonals, then DA is expected to work well since the multivariate density will behave more like a multivariate Gaussian. Note that assuming a diagonal \( \Sigma \) was also motivated by previous work [52], [55], [56], [65], [66] for various applications. In this work, we empirically report that DA has very good sparse recovery performance and has low complexity.

To further support the DA approximation, we present empirical findings regarding the structure of \( \Sigma \). We performed sparse recovery simulations using Eq. (1) with the MCMC-EM approach (i.e. without regularizing MCMC estimates of \( \Sigma \)). We assumed that \( x \sim \mathcal{N}^R(0, I) \) was a multivariate rectified Gaussian vector of size 50 with 10 non-zero elements and the dictionary \( \Phi \in \mathbb{R}^{50 \times 200} \) columns were normally distributed \( \Phi \sim \mathcal{N}(0, I) \). We solved this problem for 1,000 simulations and overlay plots of the average absolute value of the off-diagonals of \( \Sigma \) as a function of MCMC-EM iteration in the top plot of Fig. 1.

We see that the average off-diagonals decreased exponentially, and the average of this behavior (over 1,000 simulations; red line) has a final value of \( 10^{-4} \) after 10 iterations. This indicates the off-diagonals of \( \Sigma \) of the true posterior (with MCMC sampling) indeed become very close to zero. Moreover, in the bottom plot of Fig. 1, we plot the Frobenius norm of the difference between \( \Sigma \) and \( \Sigma_D \), where \( \Sigma_D \) indicates a diagonal matrix consisting of diagonal elements from \( \Sigma \). This shows that as MCMC-EM converges, the true \( \Sigma \) becomes closer to a diagonal matrix. These results suggest, that if there is flexibility in choosing the sensing matrix \( \Phi \) as in compressed sensing, then proper choice of \( \Phi \) can lead to the DA approach producing high quality approximate marginals \( \tilde{p}(x_i|y_i, \gamma_i) \) that are close to the true marginals \( p(x_i|y_i, \gamma_i) \).

**IV. Experiment Design**

In this section, we provide experiments to compare the performance of the proposed R-SBL variants LMMSE, GAMP, MCMC and DA to the currently available baseline S-NNLS solvers such as SLEP-\( \ell_1 \) [67], and NN-OMP [68].

In the first set of experiments we simulate a ‘noiseless’ sparse recovery scenario where the noise is distributed as \( v \sim \mathcal{N}(0, \sigma^2 I) \) with \( \sigma^2 = 10^{-6} \), the solution vector is \( x \sim \mathcal{N}^R(0, I) \), and the dictionary \( \Phi \) columns are normally distributed \( \Phi \sim \mathcal{N}(0, I) \). We generate sparse vectors \( x^{\text{gen}} \in \mathbb{R}^{100} \) such that \( \|x^{\text{gen}}\|_0 = K \) and random dictionaries \( \Phi \in \mathbb{R}^{100 \times 400} \). We normalize the columns of \( \Phi \) to have unit \( \ell_2 \) norm. For a given \( \Phi \) and \( x^{\text{gen}} \), we calculate the measurements \( y = \Phi x^{\text{gen}} \) and use the baseline algorithms and the proposed R-SBL variants to approximate \( x^{\text{gen}} \) by \( \hat{x} \). We perform 1,000 trials for all possible dictionary and sparse vector combination for cardinalities \( K = \{10, 20, 30, 40, 50\} \). Note that we use the MCMC approach only in the first set of experiments to establish the high quality of the parameter estimates obtained with the low complexity approximate approaches (DA, LMMSE, GAMP) and omit it in other experiments due to computational constraints. Also note that problem dimensions cardinality settings (i.e. \( K = \{10, 20, 30, 40, 50\} \)) and number of trials are the same for the other experiments.

In the second set of experiments, for a wide variety of different distributions for \( x \) and \( \Phi \), we present results of
LMMSE, GAMP, DA, SLEP-ℓ₁ and NN-OMP. Specifically, we draw the nonzero elements of \( x^{\text{gen}} \) randomly according to the following probability distributions:

I. NN-Cauchy (Location: 0, Scale: 1)
II. NN-Laplace (Location: 0, Scale: 1)
III. NN-Gamma (Location: 1, Scale: 2)
IV. Chi-square with \( \nu = 2 \)
V. Bernoulli with \( p(0.5) = 1/2 \) and \( p(1.5) = 1/2 \)

where the prefix ‘NN’ stands for non-negative. The non-negative distributions are obtained by taking the absolute value of the respective probability densities, i.e. NN-\( X = |X| \). We generate random dictionaries as in first set of experiments according to the following densities

I. Gaussian (Location: 0, Scale: 1)
II. \( \pm 1 \) with \( p(1) = 1/2 \) and \( p(-1) = 1/2 \)

In the third set, we set the noise variance \( \sigma^2 \) for \( \nu \) such that the signal to noise ratio (SNR) is 20 dB and assess the robustness of R-SBL variants under noisy conditions. In the last set of experiments we introduce correlation among the columns of the original dictionary with \( \Phi = \mathcal{N}(0, I) \) and report the R-SBL recovery performance in correlated dictionaries. This is achieved by multiplying the original dictionary \( \Phi \) with a correlation matrix \( C \) to obtain a new correlated dictionary \( \Phi_r \). Where \( C \) is simply the Cholesky factor of the Toeplitz(\( \rho \)) matrix with a parameter \( \rho \). We experimented with various correlation values by selecting \( \rho = \{0.60, 0.65, 0.70, ..., 0.95, 0.98\} \).

The noiseless case provides a fair comparison between the recovery performance of the S-NNLS solvers, since in this scenario algorithm parameters are easy to select. In the noisy setting, tuning of various algorithm parameters leads to differing behavior of the algorithms, making it difficult to draw wide-reaching conclusions about the relative performance of the algorithms. Therefore, noisy simulations are meant for robustness analysis of R-SBL rather than for comparison with other methods.

### A. Performance metrics

To evaluate the performance of various S-NNLS algorithms, we use the mean square recovery error (MSE) and the probability of error in the recovered support set (PE), which are two commonly used performance metrics in the compressive sensing literature [15]. We measure the MSE between the recovered signal \( \hat{x} \) and the ground truth \( x^{\text{gen}} \) using

\[
\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\hat{x}_i - x_i^{\text{gen}})^2.
\]

Denoting the support of the true solution as \( S \) and the support of \( \hat{x} \) as \( \hat{S} \), PE is calculated using

\[
\text{PE} = \frac{\max(|S|, |\hat{S}|) - |S \cap \hat{S}|}{\max(|S|, |\hat{S}|)}.
\]

A value of PE = 0 indicates that the ground truth and recovered supports are the same, whereas PE = 1 indicates no overlap between supports. Averaging the PE over multiple trials (i.e. 1,000 trials) gives us the empirical probability of making errors in the support. We calculate and report the average values of MSE and PE. In the Experiment Results section we use MSE and PE to indicate the averaged values.

### B. MCMC implementation & parameter selection

The MCMC implementation we used for the experiments was presented in [46], with Matlab and R codes available in [https://github.com/aripakman/hmc-tmg](https://github.com/aripakman/hmc-tmg). Parameters explained in Section [III-B1] are selected as follows, the off-diagonal pruning of the empirical scale parameter \( \hat{\Sigma} \) was performed with a threshold of \( T_p = 5 \times 10^{-2} \). Diagonal scaling was performed with a factor of \( \beta = 1.7 \), and shrinkage parameter \( \lambda = 0.5 \). These values were heuristically determined to minimize the MSE for the first set of experiments.

### V. Experiment Results

In this section, we detail the experimental results and show the benefits of R-SBL variants as compared to the baseline...
methods. We will extensivly show that in all of the sparse recovery experiments described above, the R-SBL variants we propose perform significantly better than its SLEP-ℓ₁ and NN-OMP counterparts both in terms of MSE and PE.

In Fig. 2 we show the sparse recovery performance of the R-SBL variants and the baseline solvers as a function of the cardinality for the noiseless scenario. The dictionary elements were drawn from a normal distribution and the ground truth solution increases (e.g. after $K = 30$) the performances of NN-OMP and SLEP-ℓ₁ deteriorate both in terms of MSE and PE. On the other hand, R-SBL variants are quite robust with almost no recovery error. At the largest cardinality of $K = 50$ we see that R-SBL variants still outperform the baseline algorithms by a large margin. We further notice that MCMC and DA perform better than LMMSE recovery and slightly better than GAMP. Furthermore, the DA variant performs nearly identical to MCMC in terms of MSE and PE. This is an expected behavior since MCMC prunes off-diagonal elements of the scale matrix $\Sigma$ iteratively when they drop below a certain threshold. In this case, MCMC might converge to the DA solution.

We see that the R-SBL variants outperform the baseline methods regardless of the distribution of $x^{gen}$. In Table II we summarize the MSE and PE for all distributions of $x^{gen}$ and for $\Phi \sim N(0,I)$ for a fixed cardinality of $K = 50$. We select $K = 50$ because it is the most difficult experimental setting for the sparse recovery problem considered in this paper. The first row in Table II shows that the MSE values for R-SBL variants are significantly lower as compared to baseline solvers, NN-OMP performs the worst among all methods, SLEP-ℓ₁ is slightly better than NN-OMP and R-SBL variants can perform 3 to 20 times better depending on the variant and distribution of $x^{gen}$. For Bernoulli distribution we noticed $\Phi \sim \pm 1$ with equal probability the recovery performances were very similar to that of Table II.

### A. Recovery performance under noisy conditions

We compared the recovery performance of the proposed R-SBL variants and the baseline solvers in a noisy setting. In this case the observations were contaminated with additive white Gaussian noise to have a signal to noise ratio (SNR) of 20 dB. Fig. 3 shows the MSE and PE recovery performance versus the cardinality. The dictionary elements were drawn from a normal distribution and the ground truth vector is drawn from $N_R^R(0,I)$. Compared with the noiseless case in Fig. 2 the performances of all of the methods noticeably reduced. However, the proposed R-SBL variants performed much better as compared to the baseline solvers. Moreover,

| Dist. of $x^{gen}$ | NN-OMP | SLEP-ℓ₁ | R-SBL (LMMSE) | R-SBL (GAMP) | R-SBL (DA) |
|---------------------|---------|---------|---------------|-------------|-----------|
| $N(0, I)$           | 0.4217  | 0.1502  | 0.0591        | 0.0374      | 0.0279    |
| Cauchy              | 0.0084  | 0.0086  | 0.0005        | 0.0004      | 0.0004    |
| Gamma               | 0.1342  | 0.0686  | 0.0075        | 0.0055      | 0.0035    |
| Chi-square          | 0.1234  | 0.0466  | 0.0066        | 0.0053      | 0.0039    |
| Bernoulli           | 0.1338  | 0.0680  | 0.0075        | 0.0052      | 0.0041    |

#### Table II: MSE and PE results for various distributions for $x^{gen}$ for the noiseless case. The dictionary is normally distributed. MSE and PE values for the R-SBL variants are lower as compared to baseline solvers. DA and GAMP perform very close to each other and best among other methods.
GAMP, LMMSE and DA performed very similar to each other in terms of MSE and PE. This shows that in the noisy setting considered here R-SBL performed much better than NN-OMP and SLEP-$\ell_1$.

B. Dictionary coherence

We further performed simulations by making the columns of the dictionary $\Phi$ increasingly coherent. This is important since in some applications the dictionary can be coherent. Furthermore, since the DA variant of R-SBL only considers the diagonals of the scale matrix $\Sigma$ it is important to see if coherence affects its recovery performance. In both columns of Fig. 4 we start with a Toeplitz parameter value $\rho = 0.50$ and increase the value of $\rho$ up to a value of 0.98 along the x-axis.

The cardinality is fixed at $K = 50$ across all $\rho$ values. For the setting of $\rho = 0.50$, we consider the dictionary as mildly coherent. As a result compared to the uncorrelated dictionary results in Fig. 2 for $K = 50$, the MSE performances are slightly worse and the R-SBL variants still perform robustly and better than the baseline methods. As $\rho$ is increased beyond $0.50$, MSE performance of NN-OMP starts to fail, whereas R-SBL and SLEP-$\ell_1$ are quite robust up to $\rho = 0.80$. After this value, SLEP-$\ell_1$ starts to degrade in performance whereas LMMSE and DA variants perform significantly better until $\rho = 0.98$. In terms of PE, R-SBL variants are significantly better than NN-OMP and SLEP-$\ell_1$. These results demonstrate that the proposed R-SBL variants are robust for the coherence scenario considered in this paper and they achieve superiority over baseline solvers especially in terms of PE.
C. Recovery time performance

Complexity is an important issue in sparse recovery problems. In this section we provide an analysis of average execution times to solve the S-NNLS problem. We also provide a simple way to speed up the proposed algorithms by pruning the problem size when the elements of $\gamma$ become smaller than a threshold. For example, when an index of the vector $\gamma$ becomes smaller than i.e. $\gamma_i \leq \epsilon_\gamma$, we ignore the computations regarding that index in the next iterations. This effectively reduces the problem dimensions and improves execution time.

In Fig. 5 we included the average execution times of the various algorithms discussed in the units of seconds. In the left hand side of Fig. 5 we see that MCMC is the slowest method. Note that for display purposes we divided the actual MCMC execution time values by 25. The LMMSE approach takes about 2 to 3.5 seconds (e.g. in average over 1,000 simulations) to recover the optimal solution for a problem size of 100x400. SLEP-$\ell_1$ is the fastest approach. NN-OMP is similar to SLEP-$\ell_1$ but it’s execution time increases for larger cardinalities. GAMP is slower than SLEP-$\ell_1$ and NN-OMP but its execution time is more linear with respect to the increasing cardinality. As a result, it performs faster than NN-OMP at higher cardinalities. On the other hand, DA runs slower than SLEP-$\ell_1$, NN-OMP and GAMP.

In the right hand side of Fig. 5 we show the average execution times with the pruning method discussed above. The pruning threshold was selected for all methods as $\epsilon_\gamma = 10^{-5}$. Pruning significantly reduces the average execution times for all R-SBL variants. Especially the GAMP approach is now very fast similar to that of SLEP-$\ell_1$. This demonstrates that message passing with pruning can recover the solution very fast. Considering the faster recovery speed and good recovery performance of GAMP in Figs.2 and 3, GAMP is a very good option for time sensitive sparse recovery applications and for large size problems.

VI. CONCLUSION

In this paper, we introduced a hierarchical Bayesian method to solve the S-NNLS problem. We developed the rectified Gaussian scale mixture model as a general and versatile prior to promote sparsity in the recovered solutions. Since the marginals of the posterior was not tractable, we constructed the R-SBL algorithm using the EM framework using four different approaches. We demonstrated that our R-SBL approaches outperformed the available S-NNLS solvers by a large margin both in terms of signal and support recovery. The performance gains achieved by R-SBL variants are consistent across different non-negative data distributions for $x$, different sensing matrix distributions for $\Phi$, in noisy conditions and for coherent dictionaries. In terms of execution time we found the GAMP to be as fast as SLEP-$\ell_1$ with pruning, and DA is a fairly easy to implement S-NNLS solver with simple closed form moment expressions.

VII. APPENDIX

We detail the GAMP derivations in Appendix VII-A. We detail the DA derivations including the approximate posterior density $\tilde{p}(x \mid y, \gamma)$ and the first and second order moments based on this approximate density in Appendix VII-B.

A. Full derivation of GAMP

Using the assumed rectified Gaussian scale mixture $p(x \mid \gamma)$ we evaluate Eq. (35) and Eq. (36) to find the first two moments of the approximate marginal posterior distributions under sum-product GAMP implementation:

$$\hat{x}_i = \mathbb{E}\{x_i | r_i; \tau_{r_i}\} = \int x_i p(x_i | r_i; \tau_{r_i}) \, dx_i$$

$$= \int x_i \mathcal{N}^R(x_i | 0, \gamma_i) \mathcal{N}(x_i, r_i, \tau_{r_i}) \, dx_i$$

using Gaussian pdf multiplication rule, and finding the mean of resulting rectified Gaussian:

$$\hat{x}_i = \eta_i + \sqrt{\tau_i} h(\eta_i)$$

$$\eta_i = \frac{r_i \gamma_i}{\tau_i + \gamma_i}$$

$$\nu_i = \frac{\tau_{r_i} \gamma_i}{\tau_i + \gamma_i}$$

$$h(a) = \frac{\varphi(a)}{\Phi_c(a)}$$

where $\varphi$ refers to the pdf and $\Phi_c$ refers to the complementary cdf of a zero-mean and unit-variance Gaussian distribution.

$$\tau_{x_i} = \text{var}\{x_i | r_i; \tau_{r_i}\} = \int x_i^2 p(x_i | r_i; \tau_{r_i}) - \hat{x}_i^2$$

$$= \int x_i^2 \mathcal{N}^R(x_i | 0, \gamma_i) \mathcal{N}(x_i, r_i, \tau_{r_i}) - \hat{x}_i^2$$

using Gaussian pdf multiplication rule, and finding the variance of the resulting rectified Gaussian:

$$\tau_{x_i} = \nu_i g(\eta_i)$$

$$g(a) = 1 - h(a) (h(a) - a)$$

In the case of max-sum GAMP implementation, we evaluate Eq. (57) and Eq. (58):

$$\hat{x}_i = \text{arg min}_{\hat{x}_i \geq 0} \frac{1}{2\gamma_i} + \frac{1}{2\tau_i} |\hat{x}_i - r_i|$$

$$\hat{x}_i = \begin{cases} \frac{r_i \gamma_i}{\tau_i + \gamma_i} = \eta_i & \text{if } \hat{x}_i \geq 0 \\ 0 & \text{if } \hat{x}_i < 0 \end{cases}$$

Using Eq. (38)

$$\tau_{x_i} = \begin{cases} \tau_{r_i} \gamma_i \tau_{r_i} + \gamma_i = \nu_i & \text{if } \hat{x}_i \geq 0 \\ 0 & \text{if } \hat{x}_i < 0 \end{cases}$$

Upon convergence of the max-sum GAMP implementation, the approximate marginal distributions can be obtained using Eq. (50) and Eq. (56).

1Practically it was found that setting $\tau_{x_i} = 0$ when $\hat{x}_i < 0$ increases the chances of the algorithm getting stuck at a local minimum. Instead, we set $\tau_{x_i} = \frac{\tau_{r_i} \gamma_i}{\tau_{r_i} + \gamma_i} = \nu_i$. 
B. Approximate marginals and moments using DA

In this section we detail the derivation of the approximate moments used in the DA variant. We start with the exact posterior \( p(x \mid y, \gamma) \) and use the chain rule to write, \[ p(x \mid y, \gamma) = \frac{p(y \mid x, \gamma)p(x \mid \gamma)}{\int_x p(y \mid x, \gamma)p(x \mid \gamma) \, dx}. \tag{61} \]

Due to the Gaussian noise assumption, \( p(y \mid x, \gamma) \) is a Gaussian density. Since \( p(x \mid \gamma) \) is a rectified Gaussian density, the numerator of Eq. \((61)\) is a Gaussian multiplied by a rectified Gaussian. Then we can write,

\[
p(x \mid y, \gamma) = c(y) e^{-\frac{1}{2} \frac{(x-\mu)^T \Sigma^{-1} (x-\mu)}{u(x)}}, \tag{62}
\]

where \( c(y) \) is the normalizing constant for the posterior density and \( \mu \) and \( \Sigma \) are given by Eqs. \((20)\) and \((21)\), respectively. Now we calculate \( c(y) \). We let \( \Sigma = L L^T \) and \( r = x - \mu \), so that \( dx = dr \) and \( \Sigma^{-1} = L^{-T} L^{-1} \). Therefore, we can write

\[
1 = c(y) \int_{-\mu}^{\infty} e^{-\frac{1}{2} \frac{r^T L \Sigma^{-1} L^T r}{u(x)}} \, dr. \tag{63}
\]

Now, let \( z = L^{-1} r \), which implies that \( dr = |L| \, dz \). So that,

\[
c(y) = \frac{1}{|L|} \int_{-\mu}^{\infty} e^{-\frac{1}{2} \frac{z^T \beta^T \Sigma^{-1} \beta z}{u(x)}} \, dz. \tag{64}
\]

where \( \beta = L^{-1} \mu \) is the lower limit of the new integral in vector form. This \( \beta \) depends on the linear combination of \( \mu \)'s since \( L \) is not diagonal. The integral in the denominator of Eq. \((64)\) is not tractable as the integration limits are not separable (i.e. \( L \) is not diagonal). Thus the multidimensional integral over \( z \) is not separable as a product of one dimensional integrals over \( z_i \)'s.

Now assume that we are interested in an approximate density \( \tilde{p}(x \mid y, \gamma) \) instead of the exact posterior. We calculate an approximate \( \tilde{c}(y) \) by approximating \( \Sigma \) with its diagonal \( \Sigma_d = \text{diag}(\Sigma) \approx \Sigma \). In this case the new \( L \) will be a diagonal matrix with diagonal entries \( \sqrt{\Sigma_{ii}} \). The integral in Eq. \((64)\) is now separable, thus the approximate normalizing constant will be

\[
\tilde{c}(y) = \frac{1}{|\Sigma_d|^{1/2}} \prod_{i=1}^{M} \frac{\sqrt{\pi}}{2} \text{erfc} \left( \frac{-\mu_i}{\sqrt{2\Sigma_{ii}}} \right). \tag{65}
\]

Using \( \tilde{c}(y) \) we can write the approximate posterior density as:

\[
\tilde{p}(x \mid y, \gamma) = e^{-\frac{1}{2} \frac{(x-\mu)^T \Sigma_d^{-1} (x-\mu)}{u(x)}}, \tag{66}
\]

\[
\prod_{i=1}^{M} \frac{\sqrt{\pi \Sigma_{ii}}}{2} \text{erfc} \left( \frac{-\mu_i}{\sqrt{2\Sigma_{ii}}} \right) = \prod_{i=1}^{M} \frac{2}{\pi \Sigma_{ii}} \frac{\sqrt{\pi \Sigma_{ii}}}{2} \text{erfc} \left( \frac{-\mu_i}{\sqrt{2\Sigma_{ii}}} \right) = \prod_{i=1}^{M} \tilde{p}(x_i \mid y_i, \gamma_i). \tag{67}
\]

Eq. \((68)\) shows that multivariate \( \tilde{p}(x \mid y, \gamma) \) is separable into product of univariate densities. The univariate density \( \tilde{p}(x_i \mid y_i, \gamma_i) \) is actually the univariate RG density defined in Eq. \((10)\). In other words we have,

\[
\tilde{p}(x_i \mid y_i, \gamma_i) = \mathcal{N}(x_i; \mu_i, \Sigma_{ii}). \tag{69}
\]

The first and second moments of a univariate RG density is well-known in closed form, these moments are can be obtained as in \((19)\) and reproduced in Eqs. \((44)\) and \((45)\). These moments are used to implement the DA algorithm presented in this paper.

\[\text{REFERENCES}\]

[1] C. L. Lawson and R. J. Hanson, Solving least squares problems. SIAM, 1974, vol. 161.

[2] B. M. Jedynak and S. Khandap, “Maximum likelihood set for estimating a probability mass function,” Neural computation, vol. 17, no. 7, pp. 1305–1350, 2005.

[3] J. P. Vila and S. Schnitter, “An empirical-bayes approach to recovering linearly constrained non-negative sparse signals,” Signal Processing, IEEE Transactions on, vol. 62, no. 18, pp. 4689–4703, 2014.

[4] R. Peharz and F. Pernkopf, “Sparse nonnegative matrix factorization with 0-constraints,” Neurocomputing, vol. 80, pp. 38–48, 2012.

[5] H. Kim and H. Park, “Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis,” Bioinformatics, vol. 23, no. 12, pp. 1495–1502, 2007.

[6] H. Kim and H. Park, “Nonnegative matrix factorization based on alternating nonnegativity-constrained least squares and active set method,” SIAM Journal on Matrix Analysis and Applications, vol. 30, no. 2, pp. 713–730, 2008.

[7] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, “Text mining using non-negative matrix factorizations,” in SDM, vol. 4, pp. 452–456.

[8] V. Monga and M. K. Mihčak, “Robust and secure image hashing via non-negative matrix factorizations,” Information Forensics and Security, IEEE Transactions on, vol. 2, no. 3, pp. 376–390, 2007.

[9] P. C. Loizou, “Speech enhancement based on perceptually motivated bayesian estimators of the magnitude spectrum,” Speech and Audio Processing, IEEE Transactions on, vol. 13, no. 5, pp. 857–869, 2005.

[10] C. Févotte, N. Bertin, and J.-L. Durrieu, “Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis,” Neural computation, vol. 21, no. 3, pp. 793–830, 2009.

[11] P. Sajda, S. Du, T. R. Brown, R. Stoyanova, D. C. Shungu, X. Mao, and L. C. Parra, “Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain,” Medical Imaging, IEEE Transactions on, vol. 23, no. 12, pp. 1453–1465, 2004.

[12] L. C. Potter, E. Erkin, J. T. Parker, and M. Cetin, “Sparsity and compressed sensing in radar imaging,” Proceedings of the IEEE, vol. 98, no. 6, pp. 1006–1020, 2010.

[13] A. Hurmalainen, R. Saaidi, and T. Virtanen, “Group sparsity for speaker identity discrimination in factorisation-based speech recognition,” in INTERSPEECH, 2012.

[14] M. Lustig, J. M. Santos, D. L. Donoho, and J. M. Pauly, “kt sparse: High frame rate dynamic mri exploiting spatio-temporal sparsity,” in Proceedings of the 13th Annual Meeting of ISMRM, Seattle, vol. 2420, 2006.

[15] M. Elad, Sparse and Redundant Representations. Springer New York, 2010.

[16] Y. C. Eldar and G. Kutyniok, Compressed sensing: theory and applications. Cambridge University Press, 2012.

[17] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on. IEEE, 1993, pp. 40–44.

[18] A. M. Bruckstein, D. L. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” SIAM review, vol. 51, no. 1, pp. 34–81, 2009.

[19] J. Nocedal and S. Wright, Numerical optimization. Springer Science & Business Media, 2006.
