VECTOR APPROXIMATE MESSAGE PASSING FOR NOT SO LARGE N.I.I.D. GENERALIZED I/O LINEAR MODELS

Zilu Zhao, Fangqing Xiao, Dirk Slock
Communication Systems Department, EURECOM, France
zilu.zhao@eurecom.fr, fangqing.xiao@eurecom.fr, dirk.slock@eurecom.fr

***Abstract*—Many signal processing problems involve a Generalized Linear Model (GLM), which is a type of linear model where the unknowns may be non-identically independently distributed (n.i.i.d.). Vector Approximate Message Passing for Generalized Linear Models (GVAMP) is a computationally efficient belief propagation technique used for Bayesian inference. However, the posterior variances obtained from GVAMP with limited complexity are only exact under the assumption of an independent and identically distributed (i.i.d.) prior, owing to the averaging operations involved. In numerous problems, it is beneficial not just to estimate the unknowns but also to obtain accurate posterior distributions. While VAMP, and especially AMP, are applicable to high-dimensional problems, many applications involve dimensions that are not excessively high, allowing for more complex operations. Furthermore, in finite dimensions, the asymptotic regime that leads to correct variances under certain measurement matrix model assumptions is not applicable. To overcome these challenges, we propose a revised version of GVAMP, named reGVAMP. This method provides a multivariate Gaussian posterior approximation, which includes inter-parameter correlations, and yields accurate posterior marginals requiring only the extrinsic distributions to become Gaussian.**

I. Introduction

Recovering the input signal in the context of a *generalized linear model* (GLM) [1] is a fundamental challenge in the field of signal processing. Applications involving this problem include statistics regression [2], wireless communication [3], and machine learning [4]. For instance, within the realm of wireless communication, $p_{\boldsymbol{y}|\boldsymbol{x}}$ often characterizes receiver hardware and inference mechanisms, in the symbol detection where $\boldsymbol{x}$ represents a vector of discrete symbols to be recovered, or in the channel estimation and localization, where $\boldsymbol{x}$ comprises propagation channel parameters, $\mathbf{A}$ could embody the propagation features, modulation/demodulation schemes, or pilot symbols, depending on the scenario. Viewing this from a Bayesian perspective, the estimation of the random vector $\boldsymbol{x} \in \mathbb{R}^N$ from observed measurements $\boldsymbol{y} \in \mathbb{R}^M$ is often tackled through methods such as maximum a posteriori (MAP) estimation and minimum mean square error (MMSE) estimation. However, in certain scenarios, these methods become unviable due to the optimization complexities inherent in MAP or the intractable integrals associated with MMSE.

To tackle this challenge, one established method is approximate inference. Generalized Approximate Message Passing (GAMP) [5] stands out as a prominent and computationally efficient approach within the realm of GLM. Expanding upon the foundation of Approximate Message Passing (AMP) [6] used in standard linear models (SLM), GAMP showcases its prowess in recovering high-dimensional signals. The dynamics of GAMP align with those of AMP, both governed by a state evolution. Nevertheless, GAMP encounters convergence issues when faced with an ill-conditioned measurement matrix.

A. Prior Work

In response, a solution comes as an extension to the Vector Approximate Message Passing (VAMP) [7] algorithm, tailored for GLM scenarios. This approach, named GVAMP [8], accommodates the challenge of an ill-conditioned matrix. It applies a vector-level Expectation Propagation [9] algorithm on a specially designed factorization scheme.

Compared to AMP which requires i.i.d. measurement matrix to converge, VAMP shines particularly for its converging behavior in the scenario of a right rotationally invariant measurement matrix, with a scalar state evolution having been rigorously established. It is worth noting that GLM-VAMP and its variants often assume a high system dimension, which drives efforts to avoid complex matrix inversions and necessitates additional approximations for efficiency.

However, the original (G)VAMP algorithm solely provides averaged variances. This limitation serves as the impetus for introducing the reVAMP (revisited- Vector Approximate Message Passing) [10] for a standard linear model with arbitrary additive Gaussian noise. it gives a Gaussian approximated posterior distribution including inter-parameter correlations.

B. Main Contribution

In this paper, reGVAMP (revisited- Generalized Vector Approximate Message Passing) algorithm is extended from reVAMP to cover GLM. We derive reGVAMP by utilizing the multivariate Gaussian marginalization under EP framework. In reGVAMP, each likelihood and prior is approximated by a Gaussian distribution. Like reVAMP, it results in an approximated posterior within the Gaussian family and gives a covariance matrix with inter-parameter correlations.

II. Derivation under EP Framework

We consider here a generalized linear model

$$p_{\boldsymbol{x}}(\boldsymbol{x})=\prod_{i=1}^{N} p_{x_i}(x_i),\ \mathbf{z}=\mathbf{A}\boldsymbol{x},\ p_{\boldsymbol{y}|\mathbf{z}}(\mathbf{z}) = \prod_{j=1}^{M} p_{y_j|z_j}(z_j), \tag{1}$$

where the ratio N/M is a constant. We interpret the linear mixing as a conditional probability

$$p_{\mathbf{z}|\boldsymbol{x}}(\boldsymbol{x},\mathbf{z}) = \delta(\mathbf{z}-\mathbf{A}\boldsymbol{x}). \tag{2}$$

We obtain the posterior probability by

$$\begin{aligned} &p_{\boldsymbol{x},\mathbf{z}|\boldsymbol{y}}(\boldsymbol{x},\mathbf{z}) \propto p_{\boldsymbol{x},\boldsymbol{y},\mathbf{z}}(\boldsymbol{x},\boldsymbol{y},\mathbf{z}) \\ &= \textstyle\prod_j p_{y_j|z_j}(z_j) p_{\mathbf{z}|\boldsymbol{x}}(\boldsymbol{x},\mathbf{z}) \prod_i p_{x_i}(x_i). \end{aligned} \tag{3}$$

We want to approximate this true posterior by

$$\begin{aligned} &q_{\boldsymbol{x},\mathbf{z}|\boldsymbol{y}}(\boldsymbol{x},\mathbf{z}) \propto \textstyle\prod_j q_{z_j|\boldsymbol{y}}(z_j) \prod_i q_{x_i|\boldsymbol{y}}(x_i) \\ &\propto \textstyle\prod_j q_{y_j|z_j}(z_j) q_{\boldsymbol{y}_{\bar{j}},z_j}(z_j) \prod_i q_{x_i}(x_i) q_{\boldsymbol{y}|x_i}(x_i), \end{aligned} \tag{4}$$

where for all the events E, q_E is the Gaussian approximation for p_E. We shall also see later that $q_{\boldsymbol{y}_{\bar{j}},z_j}$ and $q_{\boldsymbol{y}|x_i}$ are fully determined by $q_{y_j|z_j}$ and q_{x_i}. Ideally, we wish to find an approximation by minimizing

$$\begin{aligned} &KLD[p_{\boldsymbol{x},\mathbf{z}|\boldsymbol{y}} \| q_{\boldsymbol{x},\mathbf{z}|\boldsymbol{y}}] \\ &= \sum_j KLD[p_{\boldsymbol{x},\mathbf{z}|\boldsymbol{y}} \| q_{z_j|\boldsymbol{y}}] + \sum_i KLD[p_{\boldsymbol{x},\mathbf{z}|\boldsymbol{y}} \| q_{x_i|\boldsymbol{y}}] + c \\ &= \sum_j KLD[p_{z_j|\boldsymbol{y}} \| q_{z_j|\boldsymbol{y}}] + \sum_i KLD[p_{x_i|\boldsymbol{y}} \| q_{x_i|\boldsymbol{y}}] + c, \end{aligned} \tag{5}$$

where we define $KLD(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx$ to be the Kullback-Leibler divergence. We can minimize alternatingly w.r.t. the factors $q_{z_j|\boldsymbol{y}}$ and $q_{x_i|\boldsymbol{y}}$. The true posteriors for z_j can be written as

$$\begin{aligned} &p_{z_j|\boldsymbol{y}}(z_j) \propto \textstyle\int p_{\boldsymbol{x},\boldsymbol{y},\mathbf{z}}(\boldsymbol{x},\boldsymbol{y},\mathbf{z}) d\boldsymbol{x} d\mathbf{z}_{\bar{j}} \\ &= p_{y_j|z_j}(z_j) \textstyle\int p_{\boldsymbol{y}_{\bar{j}}|\mathbf{z}_{\bar{j}}}(\mathbf{z}_{\bar{j}}) p_{\mathbf{z}|\boldsymbol{x}}(\boldsymbol{x},\mathbf{z}) p_{\boldsymbol{x}}(\boldsymbol{x}) d\boldsymbol{x} d\mathbf{z}_{\bar{j}} \\ &= p_{y_j|z_j}(z_j) p_{\boldsymbol{y}_{\bar{j}}|z_j}(z_j) p_{z_j}(z_j) = p_{y_j|z_j}(z_j) p_{\boldsymbol{y}_{\bar{j}},z_j}(z_j), \end{aligned} \tag{6}$$

where we define the notation $\forall \mathbf{z} \forall j, \mathbf{z}_{\bar{j}}$ denotes all the elements in $\mathbf{z}$ except the j-th element.

A similar procedure can be done to derive the true posterior for x_i,

$$\begin{aligned} &p_{x_i|\boldsymbol{y}}(x_i) \propto p_{x_i}(x_i) \textstyle\int p_{\boldsymbol{y}|\mathbf{z}}(\mathbf{z}) p_{\mathbf{z}|\boldsymbol{x}}(\boldsymbol{x},\mathbf{z}) p_{\boldsymbol{x}_{\bar{i}}}(\boldsymbol{x}_{\bar{i}}) d\boldsymbol{x}_{\bar{i}} d\mathbf{z} \\ &= p_{x_i}(x_i) p_{\boldsymbol{y}|x_i}(x_i). \end{aligned} \tag{7}$$

To make the computation tractable, we approximate the extrinsic $p_{\boldsymbol{y}_{\bar{j}},z_j}$ and $p_{\boldsymbol{y}|x_i}(x_i)$ as Gaussian. If Gaussian distributions are close approximations for $p_{\boldsymbol{y}_{\bar{j}},z_j}$ and $p_{\boldsymbol{y}|x_i}$, then due to the Central Limit Theory, we have at convergence

$$\begin{aligned} &KLD[p_{z_j|\boldsymbol{y}} \| q_{z_j|\boldsymbol{y}}] \simeq KLD\Big[p_{y_j|z_j} \cdot q_{\boldsymbol{y}_{\bar{j}},z_j} \| q_{y_j|z_j} \cdot q_{\boldsymbol{y}_{\bar{j}},z_j}\Big] \\ &= \mathbb{E}\left[\ln \tfrac{p_{y_j|z_j}(z_j)}{q_{y_j|z_j}(z_j)}\right] \\ &\textstyle\sum_i KLD[p_{x_i|\boldsymbol{y}} \| q_{x_i|\boldsymbol{y}}] \simeq KLD\big[p_{x_i} \cdot q_{\boldsymbol{y}|x_i} \| q_{x_i} \cdot q_{\boldsymbol{y}|x_i}\big] \\ &= \mathbb{E}\left[\ln \tfrac{p_{x_i}(x_i)}{q_{x_i}(x_i)}\right], \end{aligned} \tag{8}$$

where

$$\begin{aligned} q_{\boldsymbol{y}_{\bar{j}},z_j}(z_j) &\propto \frac{\iint q_{\boldsymbol{y}|\mathbf{z}}(\mathbf{z}) \delta(\mathbf{z} - \mathbf{A}\boldsymbol{x}) q_{\boldsymbol{x}}(\boldsymbol{x}) d\boldsymbol{x} d\mathbf{z}_{\bar{j}}}{q_{y_j|z_j}(z_j)}; \\ q_{\boldsymbol{y}|x_i}(x_i) &\propto \frac{\iint q_{\boldsymbol{x}}(\boldsymbol{x}) q_{\boldsymbol{y}|\mathbf{z}}(\mathbf{z}) \delta(\mathbf{z} - \mathbf{A}\boldsymbol{x}) d\mathbf{z} d\boldsymbol{x}_{\bar{i}}}{q_{x_i}(x_i)}, \end{aligned} \tag{9}$$

and the two expectations are calculated with respect to the tilted posteriors

$$q_{z_j|\boldsymbol{y}}(z_j) \propto p_{y_j|z_j}(z_j) q_{\boldsymbol{y}_{\bar{j}},z_j}(z_j) \tag{10}$$

$$q_{x_i|\boldsymbol{y}}(x_i) \propto p_{x_i}(x_i) q_{\boldsymbol{y}|x_i}(x_i) \tag{11}$$

respectively. It is worth noticing that these tilted posteriors can be identified as marginal posterior with Gaussian approximated extrinsic.

A. Output Node

At the output node, we optimize the first expression in (8). We denote

$$\begin{aligned} q_{\boldsymbol{y}|\mathbf{z}}(\mathbf{z}) &= \textstyle\prod_j q_{y_j|z_j}(z_j) := \mathcal{N}(\mathbf{z}|\mathbf{m}_{\mathbf{z}}, \mathbf{D}_{\boldsymbol{\tau}_{\mathbf{z}}}) \\ q_{\boldsymbol{x}}(\boldsymbol{x}) &= \textstyle\prod_i q_{x_i}(x_i) := \mathcal{N}(\boldsymbol{x}|\mathbf{m}_{\boldsymbol{x}}, \mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}), \end{aligned} \tag{12}$$

where

$$\mathbf{D}_{\boldsymbol{\tau}_{\mathbf{z}}} = \text{diag}[\boldsymbol{\tau}_{\mathbf{z}}];\ \mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}} = \text{diag}[\boldsymbol{\tau}_{\boldsymbol{x}}]. \tag{13}$$

The diag operation either transforms a vector into a diagonal matrix or extracts the diagonal elements from a matrix into a vector identical to the MATLAB function.

The optimization of the first KLD in (8) is equivalent to matching the first and second order moment with respect to z_j of the tilted posterior given by (10). We first compute the marginalization step of the tilted posterior

$$\begin{aligned} &q_{z_j|\boldsymbol{y}}(z_j) \propto \textstyle\int\int \frac{p_{y_j|z_j}(z_j)}{q_{y_j|z_j}(z_j)} q_{\boldsymbol{y}|\mathbf{z}}(\mathbf{z}) \delta(\mathbf{z} - \mathbf{A}\boldsymbol{x}) q_{\boldsymbol{x}}(\boldsymbol{x}) d\boldsymbol{x} d\mathbf{z}_{\bar{j}} \\ &= \tfrac{p_{y_j|z_j}(z_j)}{q_{y_j|z_j}(z_j)} \textstyle\int q_{\boldsymbol{y}|\mathbf{z}}(\mathbf{z}) \int \delta(\mathbf{z} - \mathbf{A}\boldsymbol{x}) \mathcal{N}(\boldsymbol{x}|\mathbf{m}_{\boldsymbol{x}}, \mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}) d\boldsymbol{x} d\mathbf{z}_{\bar{j}} \\ &= \tfrac{p_{y_j|z_j}(z_j)}{q_{y_j|z_j}(z_j)} \textstyle\int \mathcal{N}(\mathbf{z}|\mathbf{m}_{\mathbf{z}}, \mathbf{D}_{\boldsymbol{\tau}_{\mathbf{z}}}) \mathcal{N}(\mathbf{z}|\mathbf{A}\mathbf{m}_{\boldsymbol{x}}, \mathbf{A}\mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}\mathbf{A}^T) d\mathbf{z}_{\bar{j}} \end{aligned} \tag{14}$$

From the second line to the third line, we consider an auxiliary joint probability

$$\tilde{p}_{\boldsymbol{x},\mathbf{z}}(\boldsymbol{x},\mathbf{z}) = p_{\mathbf{z}|\boldsymbol{x}}(\boldsymbol{x},\mathbf{z}) \tilde{p}_{\boldsymbol{x}}(\boldsymbol{x}) := \delta(\mathbf{z} - \mathbf{A}\boldsymbol{x}) \mathcal{N}(\boldsymbol{x}|\mathbf{m}_{\boldsymbol{x}}, \mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}). \tag{15}$$

Since $\mathbf{z} = \mathbf{A}\boldsymbol{x}$ and $\boldsymbol{x} \sim \mathcal{N}(\mathbf{m}_{\boldsymbol{x}}, \mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}})$, we have $\mathbf{z} \sim \mathcal{N}(\mathbf{A}\mathbf{m}_{\boldsymbol{x}}, \mathbf{A}\mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}\mathbf{A}^T)$. The integral in the last line can be viewed as a multivariate Gaussian marginalization of Gaussian joint posterior (with $\boldsymbol{x}$ marginalized out)

$$\mathcal{N}(\mathbf{z}|\mathbf{m}_{\mathbf{z}}, \mathbf{D}_{\boldsymbol{\tau}_{\mathbf{z}}}) \mathcal{N}(\mathbf{z}|\mathbf{A}\mathbf{m}_{\boldsymbol{x}}, \mathbf{A}\mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}\mathbf{A}^T) \propto \mathcal{N}(\mathbf{z}|\mathbf{m}_{\widehat{\mathbf{z}}'}, \mathbf{C}_{\widehat{\mathbf{z}}'}), \tag{16}$$

where

$$\begin{aligned} &\mathbf{C}_{\widehat{\mathbf{z}}'} = (\mathbf{D}_{\boldsymbol{\tau}_{\mathbf{z}}}^{-1} + (\mathbf{A}\mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}\mathbf{A}^T)^{-1})^{-1} = \mathbf{A}(\mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}^{-1} + \mathbf{A}^T\mathbf{D}_{\boldsymbol{\tau}_{\mathbf{z}}}^{-1}\mathbf{A})^{-1}\mathbf{A}^T; \\ &\mathbf{m}_{\widehat{\mathbf{z}}'} = \mathbf{C}_{\widehat{\mathbf{z}}'}(\mathbf{D}_{\boldsymbol{\tau}_{\mathbf{z}}}^{-1}\mathbf{m}_{\mathbf{z}} + (\mathbf{A}\mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{m}_{\boldsymbol{x}}) \\ &= \mathbf{A}(\mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}^{-1} + \mathbf{A}^T\mathbf{D}_{\boldsymbol{\tau}_{\mathbf{z}}}^{-1}\mathbf{A})^{-1}(\mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}^{-1}\mathbf{m}_{\boldsymbol{x}} + \mathbf{A}^T\mathbf{D}_{\boldsymbol{\tau}_{\mathbf{z}}}^{-1}\mathbf{m}_{\mathbf{z}}). \end{aligned} \tag{17}$$

Denote

$$\boldsymbol{\tau}_{\widehat{\mathbf{z}}'} = \text{diag}[\mathbf{C}_{\widehat{\mathbf{z}}'}], \tag{18}$$

we then obtain the extrinsic/cavity distribution $q_{\boldsymbol{y}_{\bar{j}},z_j}(z_j)$ by multivariate Gaussian marginalization

$$\mathcal{N}(z_j|m_{p_j}, \tau_{p_j}) \propto \tfrac{\int \mathcal{N}(\mathbf{z}|\mathbf{m}_{\widehat{\mathbf{z}}'}, \mathbf{C}_{\widehat{\mathbf{z}}'}) d\mathbf{z}_{\bar{j}}}{q_{y_j|z_j}(z_j)} \propto \tfrac{\mathcal{N}(z_j|m_{\hat{z}_j'}, \tau_{\hat{z}_j'})}{\mathcal{N}(z_j|m_{z_j}, \tau_{z_j})}, \tag{19}$$

where

$$\begin{aligned} &\tfrac{1}{\tau_{p_j}} = \tfrac{1}{\tau_{\hat{z}_j'}} - \tfrac{1}{\tau_{z_j}}; \\ &m_{p_j} = \tau_{p_j}\left(\tfrac{m_{\hat{z}_j'}}{\tau_{\hat{z}_j'}} - \tfrac{m_{z_j}}{\tau_{z_j}}\right). \end{aligned} \tag{20}$$

Thus, the tilted posterior is proportional to

$$q_{z_j|\boldsymbol{y}}(z_j) \propto p_{y_j|z_j}(z_j)\mathcal{N}(z_j|m_{p_j}, \tau_{p_j}). \tag{21}$$

Matching the first and second order moment of the two parameters in the first KLD in (8), we obtain

$$q_{y_j|z_j}(z_j)q_{\boldsymbol{y}_{\overline{j}},z_j}(z_j) \propto \mathcal{N}(z_j|m_{\widehat{z}_j}, \tau_{\widehat{z}_j}), \tag{22}$$

where $\widehat{z}_j$ and τ_{z_j} are calculated as the mean and variances of the tilted posterior

$$\begin{aligned} m_{\widehat{z}_j} &= \mathbb{E}_{q_{z_j|\boldsymbol{y}}}[z_j] = g_z(m_{p_j}, \tau_{p_j}); \\ \tau_{\widehat{z}_j} &= \mathbb{E}_{q_{z_j|\boldsymbol{y}}}[(z_j - m_{\widehat{z}_j})^2] = \tau_{p_j}\, g'_z(m_{p_j}, \tau_{p_j}), \end{aligned} \tag{23}$$

where we denote

$$g_z(m_{p_j}, \tau_{p_j}) = \frac{\int z_j p_{y_j|z_j}(z_j)\mathcal{N}(z_j|m_{p_j}, \tau_{p_j})dz_j}{\int p_{y_j|z_j}(z_j)\mathcal{N}(z_j|m_{p_j}, \tau_{p_j})dz_j}, \tag{24}$$

and the derivative is with respect to the first parameter m_{p_j}. The new approximated likelihood is then derived as

$$q_{y_j|z_j}(z_j) \propto \frac{\mathcal{N}(z_j|m_{\widehat{z}_j}, \tau_{\widehat{z}_j})}{q_{\boldsymbol{y}_{\overline{j}},z_j}(z_j)} = \frac{\mathcal{N}(z_j|m_{\widehat{z}_j}, \tau_{\widehat{z}_j})}{\mathcal{N}(z_j|m_{p_j}, \tau_{p_j})}. \tag{25}$$

It is a quotient of two Gaussian, and thus, the approximated likelihood is updated by

$$q_{y_j|z_j}(z_j) = \mathcal{N}(z_j|m^+_{z_j}, \tau^+_{z_j}), \tag{26}$$

where

$$\begin{aligned} \tfrac{1}{\tau^+_{z_j}} &= \tfrac{1}{\tau_{\widehat{z}_j}} - \tfrac{1}{\tau_{p_j}}; \\ m^+_{z_j} &= \tau^+_{z_j}\left(\tfrac{m_{\widehat{z}_j}}{\tau_{\widehat{z}_j}} - \tfrac{m_{p_j}}{\tau_{p_j}}\right). \end{aligned} \tag{27}$$

If sequential update is used, the j-th entry of $\mathbf{m_z}$ and $\boldsymbol{\tau}_\mathbf{z}$ is replaced with $m^+_{z_j}$ and $\tau^+_{z_j}$ before the proceeding to the update of next variable node z_{j+1}. The computation of (17) can be simplified as a rank one update. To simplify the computation of the covariance, we define

$$\tfrac{1}{\Delta_{\tau_{z_j}}} = \tfrac{1}{\tau^+_{z_j}} - \tfrac{1}{\tau_{z_j}}. \tag{28}$$

The updated covariance according to matrix inverse lemma is

$$\mathbf{C}^+_{\widehat{\mathbf{z}}'} = \left(\mathbf{C}^{-1}_{\widehat{\mathbf{z}}'} + \boldsymbol{e}_j\Delta^{-1}_{\tau_{z_j}}\boldsymbol{e}_j^T\right)^{-1} = \mathbf{C}_{\widehat{\mathbf{z}}'} - \tfrac{\mathbf{C}_{\widehat{\mathbf{z}}'}\boldsymbol{e}_j\boldsymbol{e}_j^T\mathbf{C}_{\widehat{\mathbf{z}}'}}{\Delta_{\tau_{z_j}} + \boldsymbol{e}_j^T\mathbf{C}_{\widehat{\mathbf{z}}'}\boldsymbol{e}_j}. \tag{29}$$

Similarly, we define

$$\Delta_{m_{z_j}} = \Delta_{\tau_{z_j}}\left(\frac{m^+_{z_j}}{\tau^+_{z_j}} - \frac{m_{z_j}}{\tau_{z_j}}\right), \tag{30}$$

and the mean update is obtained as

$$\mathbf{m}^+_{\widehat{\mathbf{z}}'} = \mathbf{m}_{\widehat{\mathbf{z}}'} + \frac{\Delta_{m_{z_j}} - \boldsymbol{e}_j^T\mathbf{m}_{\widehat{\mathbf{z}}'}}{\Delta_{\tau_{z_j}} + \boldsymbol{e}_j^T\mathbf{C}_{\widehat{\mathbf{z}}'}\boldsymbol{e}_j}\mathbf{C}_{\widehat{\mathbf{z}}'}\boldsymbol{e}_j. \tag{31}$$

Thus, the complexity for updating each z_j approximated prior is $O(N^2)$ due to (29). The complexity for approximation at the output node is dominated by $O(N^3)$ happening at the start of a sweep where we need to compute $\mathbf{C}_{\widehat{\mathbf{z}}'}$ according to (17) for the first time. Moreover, there are $O(N)$ variable nodes to be updated. In other words, the sweep over vector $\mathbf{z}$ has complexity $O(N^3) + O(N)O(N^2) = O(N^3)$.

B. Input Node

At the input node, we optimize the second expression in (8) with respect to $q_{x_i}(x_i)$. This KLD's optimization is equivalent to matching the first and second order moment of the tilted posterior since q_{x_i} belongs to the Gaussian family.

With the notations defined in (12), the tilted posterior is given by

$$\begin{aligned} q_{x_i|\boldsymbol{y}}(x_i) &\propto \int\int \tfrac{p_{x_i}(x_i)}{q_{x_i}(x_i)} q_{\boldsymbol{x}}(\boldsymbol{x})q_{\boldsymbol{y}|\mathbf{z}}(\mathbf{z})\delta(\mathbf{z} - \mathbf{A}\boldsymbol{x})d\mathbf{z}d\boldsymbol{x}_{\overline{i}} \\ &= \tfrac{p_{x_i}(x_i)}{q_{x_i}(x_i)} \int q_{\boldsymbol{x}}(\boldsymbol{x}) \int \delta(\mathbf{z} - \mathbf{A}\boldsymbol{x})\mathcal{N}(\mathbf{z}|\mathbf{m}_\mathbf{z}, \mathbf{D}_{\boldsymbol{\tau}_\mathbf{z}})d\mathbf{z}d\boldsymbol{x}_{\overline{i}} \\ &= \tfrac{p_{x_i}(x_i)}{q_{x_i}(x_i)} \int \mathcal{N}(\boldsymbol{x}|\mathbf{m}_{\boldsymbol{x}}, \mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}})\mathcal{N}(\mathbf{A}\boldsymbol{x}|\mathbf{m}_\mathbf{z}, \mathbf{D}_{\boldsymbol{\tau}_\mathbf{z}})d\boldsymbol{x}_{\overline{i}}. \end{aligned} \tag{32}$$

Again, the last integral is proportional to a multi-variate Gaussian marginalization of the Gaussian joint posterior (with $\mathbf{z}$ marginalized out), and hence, we compute the Gaussian joint posterior as

$$\mathcal{N}(\boldsymbol{x}|\mathbf{m}_{\boldsymbol{x}}, \mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}})\mathcal{N}(\mathbf{A}\boldsymbol{x}|\mathbf{m}_\mathbf{z}, \mathbf{D}_{\boldsymbol{\tau}_\mathbf{z}}) \propto \mathcal{N}(\boldsymbol{x}|\mathbf{m}_{\widehat{\boldsymbol{x}}'}, \mathbf{C}_{\widehat{\boldsymbol{x}}'}), \tag{33}$$

where

$$\begin{aligned} \mathbf{C}_{\widehat{\boldsymbol{x}}'} &= \left(\mathbf{D}^{-1}_{\boldsymbol{\tau}_{\boldsymbol{x}}} + \mathbf{A}^T\mathbf{D}^{-1}_{\boldsymbol{\tau}_\mathbf{z}}\mathbf{A}\right)^{-1}; \\ \mathbf{m}_{\widehat{\boldsymbol{x}}'} &= \mathbf{C}_{\widehat{\boldsymbol{x}}'}\left(\mathbf{D}^{-1}_{\boldsymbol{\tau}_{\boldsymbol{x}}}\mathbf{m}_{\boldsymbol{x}} + \mathbf{A}^T\mathbf{D}^{-1}_{\boldsymbol{\tau}_\mathbf{z}}\mathbf{m}_\mathbf{z}\right). \end{aligned} \tag{34}$$

To obtain the marginal distribution, we denote

$$\boldsymbol{\tau}_{\widehat{\boldsymbol{x}}'} = \mathrm{diag}[\mathbf{C}_{\widehat{\boldsymbol{x}}'}]. \tag{35}$$

The extrinsic/cavity distribution $q_{\boldsymbol{y}|x_i}(x_i)$ can be computed as a quotient of two Gaussian distributions

$$\mathcal{N}(x_i|m_{r_i}, \tau_{r_i}) \propto \tfrac{\int \mathcal{N}(\boldsymbol{x}|\mathbf{m}_{\widehat{\boldsymbol{x}}'}, \mathbf{C}_{\widehat{\boldsymbol{x}}'})d\boldsymbol{x}_{\overline{i}}}{q_{x_i}(x_i)} \propto \tfrac{\mathcal{N}(x_i|m_{\widehat{x}'_i}, \tau_{\widehat{x}'_i})}{\mathcal{N}(x_i|m_{x_i}, \tau_{x_i})} \tag{36}$$

where

$$\begin{aligned} \tfrac{1}{\tau_{r_i}} &= \tfrac{1}{\tau_{\widehat{x}'_i}} - \tfrac{1}{\tau_{x_i}}; \\ m_{r_i} &= \tau_{r_i}\left(\tfrac{m_{\widehat{x}'_i}}{\tau_{\widehat{x}'_i}} - \tfrac{m_{x_i}}{\tau_{x_i}}\right). \end{aligned} \tag{37}$$

The tilted posterior is proportional to the product of Gaussian extrinsic and the prior of x_i,

$$q_{x_i|\boldsymbol{y}}(x_i) \propto p_{x_i}(x_i)\mathcal{N}(x_i|m_{r_i}, \tau_{r_i}). \tag{38}$$

Analog to the discussion (22) to (24), the approximated posterior is obtained by

$$q_{x_i}(x_i)q_{\boldsymbol{y}|x_i}(x_i) \propto \mathcal{N}(x_i|m_{\widehat{x}_i}, \tau_{\widehat{x}_i}), \tag{39}$$

where

$$\begin{aligned} m_{\widehat{x}_i} &= \mathbb{E}_{q_{x_i|\boldsymbol{y}}}[x_i] = g_x(m_{r_i}, \tau_{r_i}); \\ \tau_{\widehat{x}_i} &= \mathbb{E}_{q_{x_i|\boldsymbol{y}}}[(x_i - m_{\widehat{x}_i})^2] = \tau_{r_i}\, g'_x(m_{r_i}, \tau_{r_i}), \end{aligned} \tag{40}$$

with $g_x(m_{r_i}, \tau_{r_i})$ defined by

$$g_x(m_{r_i}, \tau_{r_i}) = \frac{\int x_i p_{x_i}(x_i)\mathcal{N}(x_i|m_{r_i}, \tau_{r_i})dx_i}{\int p_{x_i}(x_i)\mathcal{N}(x_i|m_{r_i}, \tau_{r_i})dx_i}, \tag{41}$$

and the derivative $g'_x(m_{r_i}, \tau_{r_i})$ is once again with respect to the first parameter m_{r_i}. Hence, the approximated prior can be updated by

$$q_{x_i}(x_i) \propto \frac{\mathcal{N}(x_i|m_{\widehat{x}_i}, \tau_{x,j})}{q_{\boldsymbol{y}|x_i}(x_i)} = \frac{\mathcal{N}(x_i|m_{\widehat{x}_i}, \tau_{\widehat{x}_i})}{\mathcal{N}(x_i|m_{r_i}, \tau_{r_i})}. \tag{42}$$

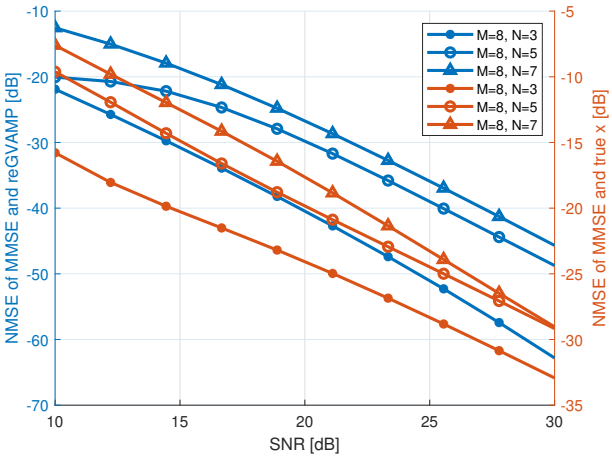

Fig. 1. Performance of GLM-reVAMP algorithm with different variable length N.

This Gaussian quotient can be calculated by

$$q_{x_i}(x_i) = \mathcal{N}(x_i|m^+_{x_i}, \tau^+_{x_i}) \propto \frac{\mathcal{N}(x_i|m_{\widehat{x}_i}, \tau_{\widehat{x}_i})}{\mathcal{N}(x_i|m_{r_i}, \tau_{r_i})}, \tag{43}$$

where

$$\begin{aligned} \tfrac{1}{\tau^+_{x_i}} &= \tfrac{1}{\tau_{\widehat{x}_i}} - \tfrac{1}{\tau_{r_i}}; \\ m^+_{x_i} &= \tau^+_{x_i}\left(\tfrac{m_{\widehat{x}_i}}{\tau_{\widehat{x}_i}} - \tfrac{m_{r_i}}{\tau_{r_i}}\right). \end{aligned} \tag{44}$$

We use a similar method as described in (28) till (31) to simplify the LMMSE step (34). Define

$$\frac{1}{\Delta_{\tau_{x_i}}} = \frac{1}{\tau^+_{x_i}} - \frac{1}{\tau_{x_i}}; \; \Delta_{m_{x_i}} = \Delta_{\tau_{x_i}}\left(\frac{m^+_{x_i}}{\tau^+_{x_i}} - \frac{m_{x_i}}{\tau_{x_i}}\right). \tag{45}$$

The update of (34) with the new estimated prior is

$$\begin{aligned} \mathbf{C}^+_{\widehat{\boldsymbol{x}}'} &= \mathbf{C}_{\widehat{\boldsymbol{x}}'} - \tfrac{\mathbf{C}_{\widehat{\boldsymbol{x}}'}\boldsymbol{e}_i\boldsymbol{e}_i^T\mathbf{C}_{\widehat{\boldsymbol{x}}'}}{\Delta_{\tau_{x_i}} + \boldsymbol{e}_i^T\mathbf{C}_{\widehat{\boldsymbol{x}}'}\boldsymbol{e}_i}; \\ \mathbf{m}^+_{\widehat{\boldsymbol{x}}'} &= \mathbf{m}_{\widehat{\boldsymbol{x}}'} + \tfrac{\Delta_{m_{x_i}} - \boldsymbol{e}_i^T\mathbf{m}_{\widehat{\boldsymbol{x}}'}}{\Delta_{\tau_{x_i}} + \boldsymbol{e}_i^T\mathbf{C}_{\widehat{\boldsymbol{x}}'}\boldsymbol{e}_i}\mathbf{C}_{\widehat{\boldsymbol{x}}'}\boldsymbol{e}_i. \end{aligned} \tag{46}$$

Therefore, at the start of a sweep, the algorithm has a complexity of $O(N^3)$ due to the matrix inverse operation. After that, during the sweep, the algorithm still has a complexity of $O(N^3)$ because there are $O(N)$ messages each of which has a complexity of $O(N^2)$. This algorithm displays a nice symmetry between the input and output nodes. Considering both sweeps for the input and output node, the algorithm has an overall complexity of $O(N^3)$. We conclude the discussion in Algorithm reGVAMP.

Algorithm 1 reGVAMP under EP Framework

Require: $\boldsymbol{y}$, $\mathbf{z} = \mathbf{A}\boldsymbol{x}$, $p_{\boldsymbol{x}}(\boldsymbol{x})$, $p_{\boldsymbol{y}|\mathbf{z}}(\boldsymbol{y}|\mathbf{z})$

Initialize: $\mathbf{m}_{\mathbf{z}}$, $\boldsymbol{\tau}_{\mathbf{z}}$, $\mathbf{m}_{\boldsymbol{x}}$, $\boldsymbol{\tau}_{\boldsymbol{x}}$
repeat[For iteration step t]
  [Update the Output Nodes]
  $\mathbf{C}_{\widehat{\mathbf{z}}'} = \mathbf{A}(\mathbf{D}^{-1}_{\boldsymbol{\tau}_{\boldsymbol{x}}} + \mathbf{A}^T\mathbf{D}^{-1}_{\boldsymbol{\tau}_{\mathbf{z}}}\mathbf{A})^{-1}\mathbf{A}^T$
  $\mathbf{m}_{\widehat{\mathbf{z}}'} = \mathbf{C}_{\widehat{\mathbf{z}}'}(\mathbf{D}^{-1}_{\boldsymbol{\tau}_{\mathbf{z}}}\mathbf{m}_{\mathbf{z}} + (\mathbf{A}\mathbf{D}_{\boldsymbol{\tau}_{\boldsymbol{x}}}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{m}_{\boldsymbol{x}})$
  repeat[For $j = 1$]
    $\tau_{\widehat{z}'_j} = [\mathbf{C}_{\widehat{\mathbf{z}}'}]_{jj}$
    $\tau_{p_j} = \frac{\tau_{z_j}\tau_{\widehat{z}'_j}}{\tau_{z_j} - \tau_{\widehat{z}'_j}}$; $m_{p_j} = \frac{m_{\widehat{z}'_j}\tau_{z_j} - m_{z_j}\tau_{\widehat{z}'_j}}{\tau_{z_j} - \tau_{\widehat{z}'_j}}$
    $m_{\widehat{z}_j} = g_z(m_{p_j}, \tau_{p_j})$; $\tau_{\widehat{z}_j} = \tau_{p_j}\, g'_z(m_{p_j}, \tau_{p_j})$
    $\tau^+_{z_j} = \frac{\tau_{p_j}\tau_{\widehat{z}_j}}{\tau_{p_j} - \tau_{\widehat{z}_j}}$; $m^+_{z_j} = \frac{m_{\widehat{z}_j}\tau_{p_j} - m_{p_j}\tau_{\widehat{z}_j}}{\tau_{p_j} - \tau_{\widehat{z}_j}}$
    Update $\mathbf{C}^+_{\widehat{\mathbf{z}}'}$ and $\mathbf{m}^+_{\widehat{\mathbf{z}}'}$ according to (29), (31)
  until $j = M$
  [Update the Input Nodes]
  $\mathbf{C}_{\widehat{\boldsymbol{x}}'} = \left(\mathbf{D}^{-1}_{\boldsymbol{\tau}_{\boldsymbol{x}}} + \mathbf{A}^T\mathbf{D}^{-1}_{\boldsymbol{\tau}_{\mathbf{z}}}\mathbf{A}\right)^{-1}$
  $\mathbf{m}_{\widehat{\boldsymbol{x}}'} = \mathbf{C}_{\widehat{\boldsymbol{x}}'}\left(\mathbf{D}^{-1}_{\boldsymbol{\tau}_{\boldsymbol{x}}}\mathbf{m}_{\boldsymbol{x}} + \mathbf{A}^T\mathbf{D}^{-1}_{\boldsymbol{\tau}_{\mathbf{z}}}\mathbf{m}_{\mathbf{z}}\right)$
  repeat[For $i = 1$]
    $\tau_{\widehat{x}'_i} = [\mathbf{C}_{\widehat{\boldsymbol{x}}'}]_{ii}$
    $\tau_{r_i} = \frac{\tau_{x_i}\tau_{\widehat{x}'_i}}{\tau_{x_i} - \tau_{\widehat{x}'_i}}$; $m_{r_i} = \frac{m_{\widehat{x}'_i}\tau_{x_i} - m_{x_i}\tau_{\widehat{x}'_i}}{\tau_{x_i} - \tau_{\widehat{x}'_i}}$
    $m_{\widehat{x}_i} = g_x(m_{r_i}, \tau_{r_i})$; $\tau_{\widehat{x}_i} = \tau_{r_i}\, g'_x(m_{r_i}, \tau_{r_i})$
    $\tau^+_{x_i} = \frac{\tau_{\widehat{x}_i}\tau_{r_i}}{\tau_{r_i} - \tau_{\widehat{x}_i}}$; $m^+_{x_i} = \frac{m_{\widehat{x}_i}\tau_{r_i} - m_{r_i}\tau_{\widehat{x}_i}}{\tau_{r_i} - \tau_{\widehat{x}_i}}$
    Update $\mathbf{C}^+_{\widehat{\boldsymbol{x}}'}$ and $\mathbf{m}^+_{\widehat{\boldsymbol{x}}'}$ according to (46)
  until $i = N$
until Convergence

III. Simulation Results

We present the results of numerical experiments with varying parameters. In this experiment, we set M to 8 and N to $\{3, 5, 7\}$, respectively. Both the prior distribution $p_{\boldsymbol{x}}(\boldsymbol{x})$ and the conditional distribution $p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})$ are assumed to follow mixture Gaussian distributions. Specifically, they can be expressed as:

$$p_{x_i}(x_i) = 0.5\mathcal{N}(x_i|0, \sigma^2_{x_{1i}}) + 0.5\mathcal{N}(x_i|0, \sigma^2_{x_{2i}}), \tag{47}$$

$$p_{y_j|z_j}(z_j) = 0.5\mathcal{N}(y_j|z_j, \sigma^2_{v_{1j}}) + 0.5\mathcal{N}(y_j|z_j, \sigma^2_{v_{2j}}), \tag{48}$$

where $\sigma_{v_{1j}}$, $\sigma_{v_{2j}}$, $\sigma_{x_{1i}}$ and $\sigma_{x_{2i}}$ are independently and uniformly drawn from the interval $(0, 1]$. The elements of the measurement matrix $\mathbf{A}$ are drawn independently from a Gaussian distribution $\mathcal{N}(0, \gamma)$. By adjusting γ, we can control the signal-to-noise ratio (SNR) of the system. We conducted simulations with SNR values ranging from 10 to 30 dB. Additionally, the recovery performance was evaluated using the normalized mean-squared error (NMSE) defined as $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2/\|\boldsymbol{x}\|^2$ of one realization. The NMSE of $\hat{\boldsymbol{x}}_{mmse}$ with respect to $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}_{reGVAMP}$ were chosen as performance metrics.

The simulation results are presented in Figure 1. It is evident from the figure that as SNR increases, the bias decreases. Furthermore, as N approaches the value of M, the estimation performance deteriorates, although the overall recovery remains satisfactory. These results confirm the effectiveness of the proposed algorithm.

IV. Concluding Remarks

In this paper, we proposed an iterative Bayesian estimation method for estimating the GLM input signal which enables us to obtain the posterior mean $\mathbf{m}_{\widehat{\boldsymbol{x}}'}$ and covariance matrix $\mathbf{C}_{\widehat{\boldsymbol{x}}'}$ at the cost of higher complexity. It also gives Gaussian approximation for the prior and likelihood as a byproduct. Further study is required to study its convergence behavior.

Acknowledgements EURECOM's research is partially supported by its industrial members: ORANGE, BMW, SAP, iABG, Norton LifeLock, and by the Franco-German projects CellFree6G and 5G-OPERA.

REFERENCES

[1] J. A. Nelder and R. W. Wedderburn, "Generalized linear models," *J. R. Stat. Soc. Ser. A. Stat. Soc.*, vol. 135, no. 3, pp. 370–384, 1972.

[2] J. D. Blume, L. Su, R. M. Olveda, and S. T. McGarvey, "Statistical evidence for glm regression parameters: a robust likelihood approach," *STAT. MED.*, vol. 26, no. 15, pp. 2919–2936, 2007.

[3] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1772–1793, 2002.

[4] Y. Wang, R. Wang, S. S. Du, and A. Krishnamurthy, "Optimism in reinforcement learning with generalized linear function approximation," *arXiv:1912.04136*, 2019.

[5] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *IEEE Int. Symp. Inf. Theory - Proc.*, 2011, pp. 2168–2172.

[6] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *PNAS*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[7] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6664–6684, 2019.

[8] P. Schniter, S. Rangan, and A. K. Fletcher, "Vector approximate message passing for the generalized linear model," in *IEEE ASILOMAR*, 2016, pp. 1525–1529.

[9] T. P. Minka, "Expectation propagation for approximate bayesian inference," *arXiv:1301.2294*, 2013.

[10] Z. Zhao, F. Xiao, and D. Slock, "Approximate Message Passing for not so Large NIID Generalized Linear Models," in *SPAWC 2023, 24th IEEE International Workshop on Signal Processing Advances in Wireless Communications*.