Journal of Machine Learning Research 21 (2020) 1-42 Submitted 2/18; Published 5/20

Regularized Gaussian Belief Propagation with Nodes of Arbitrary Size

Francois Kamper FRANCOISK@SUN.AC.ZA
Sarel J. Steel SJST@SUN.AC.ZA
Department of Statistics and Actuarial Science
Stellenbosch University
Stellenbosch, South Africa

Johan A. du Preez DUPREEZ@SUN.AC.ZA
Department of Electrical and Electronic Engineering
Stellenbosch University
Stellenbosch, South Africa

Editor: Manfred Opper

Abstract

Gaussian belief propagation (GaBP) is a message-passing algorithm that can be used to perform approximate inference on a pairwise Markov graph (MG) constructed from a multivariate Gaussian distribution in canonical parameterization. The output of GaBP is a set of approximate univariate marginals for each variable in the pairwise MG. An extension of GaBP (labeled GaBP-m), allowing for the approximation of higher-dimensional marginal distributions, was explored by Kamper et al. (2019). The idea is to create an MG in which each node is allowed to receive more than one variable. As in the univariate case, the multivariate extension does not necessarily converge in loopy graphs and, even if convergence occurs, is not guaranteed to provide exact inference. To address the problem of convergence, we consider a multivariate extension of the principle of node regularization proposed by Kamper et al. (2018). We label this algorithm slow GaBP-m (sGaBP-m), where the term "slow" relates to the damping effect of the regularization on the message passing. We prove that, given sufficient regularization, this algorithm will converge and provide the exact marginal means at convergence, regardless of the way variables are assigned to nodes. The selection of the degree of regularization is addressed through the use of a heuristic, which is based on a tree representation of sGaBP-m. As a further contribution, we extend other GaBP variants in the literature to allow for higher-dimensional marginalization. We show that our algorithm compares favorably with these variants, both in terms of convergence speed and inference quality.

Keywords: belief propagation, Gaussian distributions, regularization, inference quality, higher-dimensional marginals

1. Introduction

In this paper we deal with the problem of finding the marginal distributions of $\mathbf{X}_i : d_i \times 1$ for $i = 1, 2, \ldots, p$ where these are mutually exclusive and exhaustive subvectors of $\mathbf{X} : k \times 1$. We restrict our focus to the case where $\mathbf{X}$ follows a multivariate Gaussian distribution in canonical parameterization with precision matrix $\mathbf{S} : k \times k$ and potential vector $\boldsymbol{b} : k \times 1$.

©2020 Francois Kamper, Sarel J. Steel, and Johan A. du Preez.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v21/18-101.html.

For the purpose of this marginalization, we consider applying belief propgation (BP) to a graphical model. BP was introduced by Pearl (1988) as an inference algorithm, and was later found to be equivalent to the sum-product algorithm for decoding LDPC codes (Gallager, 1963; Frey and Kschischang, 1996; Aji and McEliece, 2000). BP applied to a pairwise MG constructed from a Gaussian distribution in canonical parameterization is often labeled GaBP. This corresponds to our marginalization objective, with $p = k$ and $d_i = 1$ for all i. The output of this algorithm is a set of approximate marginal distributions for each of the variables represented in the pairwise MG. A limitation of GaBP is that it cannot be used to approximate higher-dimensional marginals. Another issue with GaBP is that it does not necessarily converge when applied to loopy pairwise MGs. Moreover, even if convergence occurs, the precisions provided by GaBP are not necessarily exact (however, the means provided are exact). There are several GaBP-based algorithms in the literature aimed at improving on the convergence behavior of the basic algorithm (Johnson et al., 2009; Liu, 2010; El-Kurdi et al., 2012a; Ruozzi and Tatikonda, 2013; Liu et al., 2012; Kamper et al., 2018).

Due to the way a multivariate Gaussian distribution in canonical parameterization is marginalized, GaBP implicitly solves a system of linear equations. This role of GaBP is considered in the literature (Bickson, 2008; Shental et al., 2008; El-Kurdi et al., 2012b). Other types of application include channel estimation in communication systems (Montanari et al., 2006; Guo and Ping, 2008; Guo and Huang, 2011), sparse Bayesian learning in large-scale compressed sensing problems (Seeger and Wipf, 2010), estimation on Gaussian graphical models (Chandrasekaran et al., 2008; Liu et al., 2012) and the detection of F-formations in free-standing conversational groups (Kamper, 2017).

Kamper et al. (2019) proposed an extension of the GaBP algorithm, allowing for the approximation of higher-dimensional marginal distributions (GaBP-m). This algorithm operates on a higher-dimensional extension of a pairwise MG. Therefore, GaBP-m can be used to approximate the marginals of $\mathbf{X}_i : i = 1, 2, \ldots, p$, where each $\mathbf{X}_i$ is allowed to be higher-dimensional. As in the case of GaBP, GaBP-m is not guaranteed to converge when applied to loopy graphs. Assuming convergence, GaBP-m provides the correct marginal means of $\mathbf{X}_i$, while the precision matrices provided are not necessarily exact. The main reason for using GaBP-m over GaBP is that it allows for the approximation of higher-dimensional marginals. This is useful when fast approximations to diagonal blocks of $\mathbf{S}^{-1}$ are required. Consider the marginal distribution of $\mathbf{X}_1$ with mean vector $\boldsymbol{\mu}_1$ and precision matrix $\boldsymbol{\Theta}_1$. We note that, under the assumption of convergence, both GaBP-m and GaBP provide the exact marginal means and hence both can be used to find $\boldsymbol{\mu}_1$. A major advantage of GaBP-m over GaBP is that it can be used to approximate $\boldsymbol{\Theta}_1$. Another advantage is that GaBP-m can converge in cases where GaBP does not (faster convergence is also possible). GaBP-m can also be used to perform univariate marginalization. The idea is to first approximate the marginal distribution of $\mathbf{X}_1$ using GaBP-m, and then apply direct matrix inversion to approximate the univariate marginals. This method can yield better univariate marginal approximations than GaBP (Kamper et al., 2019). However, doing this comes at an increased computational cost compared to GaBP.

In this paper, we consider a new algorithm (sGaBP-m), which can be regarded as either a convergent extension of GaBP-m or a multivariate extension of sGaBP. The main motivation for using sGaBP-m over GaBP-m is that sGaBP-m can converge for arbitrary precision matrices. This is the main theoretical contribution of this paper, viz. sGaBP-m will converge, given sufficient regularization. There are also other advantages of using sGaBP-m over GaBP-m. These include that sGaBP-m can accelerate the convergence speed of GaBP-m, and can provide superior inference quality in terms of the approximated precisions provided. These are some of the conclusions made in the empirical study presented in Section 5.1. The main motivation for using sGaBP-m over sGaBP is that sGaBP-m can provide approximations of higher-dimensional marginals. Since both these algorithms provide the exact marginal means at convergence, this improvement lies in the fact that sGaBP-m can approximate higher-dimensional marginal precision matrices. Again, there are other reasons for using sGaBP-m over sGaBP. Here we note that sGaBP-m can also be used for univariate marginal approximation. This is done by first computing the higher-dimensional precision estimate, and then using direct matrix inversion to approximate the univariate precisions. In Section 5.3 we present an empirical study in which sGaBP-m outperforms sGaBP in terms of inference quality.

We also extend relaxed GaBP (El-Kurdi et al., 2012a) and convergence fix GaBP (Johnson et al., 2009) to allow for higher-dimensional marginalization. These algorithms are labeled RGaBP and CFGaBP respectively. In the empirical study of Section 5.1 we show that sGaBP-m compares favorably to these algorithms, both in terms of convergence speed and inference quality.

The construction of this paper is as follows:

1. Section 2. We give an overview of some of the mathematical concepts that feature in this paper.

2. Section 3. We provide a proof of convergence of sGaBP-m, given sufficient regularization. This section contains most of the theoretical contributions of this paper. The main theoretical novelty with respect to Kamper et al. (2018) is the use of computation trees to derive asymptotic expressions for the precision components of sGaBP-m. The remainder of this section contains multivariate extensions of some of the theoretical results associated with sGaBP, and hence some of the proofs will be similar to those of Kamper et al. (2018). We will indicate these proofs explicitly, and discuss differences with their univariate analogs.

3. Section 4. We discuss a tree representation of sGaBP-m that can be used to unfold all the computations done by sGaBP-m (including the means, which is not discussed in Section 3). This is used to derive a heuristic for selecting λ.

4. Section 5. sGaBP-m is compared to other algorithms empirically.

2. Preliminaries

In this section we discuss the concepts needed to understand the theoretical work covered in this paper.

2.1. Terminology

BP is an iterative message-passing algorithm that operates on a graph. Direct communication only occurs between nodes linked in the graph. Bickson (2008) describes two conventional types of message-update rules. In synchronous message passing, new messages are formed using messages from the previous round only and therefore are not influenced by the message scheduling. This is in contrast to the asynchronous case, where messages updated in the current round are used to compute new messages. Although asynchronous updates tend to outperform the synchronous approach (Koller and Friedman, 2009), our focus is on the synchronous case. We do this since one of the more attractive properties of BP is its application in distributed computing settings, which is far more compatible with synchronous message updates.

At each iteration, the approximate marginal distribution of a node can be found by multiplying all its incoming messages with the node potential. We label the approximate distribution constructed by a node i at iteration n, the posterior distribution of node i at iteration n. In the Gaussian case, the posterior distribution of node i at iteration n is characterized by a mean vector and a precision matrix. We call these the posterior mean and the posterior precision associated with node i at iteration n respectively.

2.2. Higher-dimensional MG

As in the case of GaBP-m, sGaBP-m operates on a higher-dimensional extension of a pairwise MG. For the purpose of our discussion, we assume, without loss of generality, that $\mathbf{X} = (\mathbf{X}_1', \mathbf{X}_2', \ldots, \mathbf{X}_p')'$. Let $\mathcal{C}_i$ denote the set of variables contained in $\mathbf{X}_i$ for $i = 1, 2, \ldots, p$. We sometimes refer to $\mathcal{C}_i$ as cluster i. The higher-dimensional MG consists of p nodes, where we assign to node i the variables in $\mathcal{C}_i$. There exists an edge between node i and node j if, and only if, there is a variable in cluster i linked to a variable in cluster j in the original (univariate) pairwise MG. We use MG to refer to the higher-dimensional extension of the pairwise MG.

Let $\mathbf{S}_{ij}$ be the submatrix of $\mathbf{S}$ corresponding to the variables in $\mathcal{C}_i$ (rows) and $\mathcal{C}_j$ (columns). The set of edges is $\mathcal{E} = \{(i,j) : i < j, \mathbf{S}_{ij} \neq \mathbf{0} : d_i \times d_j\}$. The density function of $\mathbf{X}$ can be written as

$$f(\mathbf{x}) = \frac{1}{Z} \prod_{i=1}^{p} \exp\left[-\frac{1}{2}\mathbf{x}_i'\mathbf{S}_{ii}\mathbf{x}_i + \mathbf{x}_i'\boldsymbol{b}_i \right] \prod_{(i,j)\in\mathcal{E}} \exp\left[-\mathbf{x}_i'\mathbf{S}_{ij}\mathbf{x}_j \right], \tag{1}$$

where $\boldsymbol{b}_i$ is the subvector of $\boldsymbol{b}$ corresponding to the variables in $\mathcal{C}_i$, $\mathbf{x} = (\mathbf{x}_1', \mathbf{x}_2', \ldots, \mathbf{x}_p')'$ corresponds to the decomposition $\mathbf{X} = (\mathbf{X}_1', \mathbf{X}_2', \ldots, \mathbf{X}_p')'$, and Z is a normalization constant.

The neighborhood of cluster i is defined as $\mathcal{N}_i = \{j \neq i : \mathbf{S}_{ij} \neq \mathbf{0} : d_i \times d_j\}$. If $j \in \mathcal{N}_i$, then $\mathcal{N}_i \setminus j$ denotes $\mathcal{N}_i$ with j removed.

2.3. Convergence of GaBP and GaBP-m

When GaBP is applied to a tree-structured pairwise MG, it will converge and provide exact inference at convergence. Convergence is not guaranteed for loopy pairwise MGs, although there are some guarantees. Weiss and Freeman (2001) showed that GaBP will converge if the precision matrix is diagonally dominant. A weaker condition for convergence is the walk-summability of the precision matrix (Malioutov et al., 2006). The spectral radius of a matrix $\mathbf{A} : k \times k$ is $\rho(\mathbf{A}) = \max\{|\sigma_1|, |\sigma_2|, \ldots, |\sigma_k|\}$, where $|\sigma_i|$ is the modulus of the ith eigenvalue of $\mathbf{A}$. We give the following definition:

Definition 1 (Walk-summability) *Consider a precision matrix* $\mathbf{S} : k \times k = [s_{ij}]$, *and suppose* $\mathbf{D} = diag(\frac{1}{\sqrt{s_{11}}}, \frac{1}{\sqrt{s_{22}}}, \ldots, \frac{1}{\sqrt{s_{kk}}})$. *The matrix* $\mathbf{S}$ *is considered to be walk-summable if* $\rho(|\mathbf{I}_k - \mathbf{DSD}|) < 1$

Note that $|\mathbf{A}|$ contains the absolute values of the elements of $\mathbf{A}$ (we use $\det(\mathbf{A})$ for the determinant). For more studies on the convergence of GaBP, the reader can consult Su and Wu (2015); Sui et al. (2015); Li and Wu (2019a,b). Note that other types of Gaussian message-passing are considered in some of these studies. Different types of Gaussian message-passing correspond to different factorizations of the underlying Gaussian density. In this paper, we only consider node regularization applied to factorizations of the type in Equation (1), although extensions to other factorizations are an interesting avenue for further research.

Kamper et al. (2019) derived an extension of the walk-summability convergence condition for GaBP to GaBP-m. GaBP-m will converge if the precision matrix is preconditioned walk-summable. We give the following definitions:

Definition 2 (Valid Preconditioner) *We call a matrix* $\mathbf{\Lambda} : k \times k$ *a valid preconditioner with respect to the clusters* $\mathcal{C}_i : i = 1, 2, \ldots, p$ *if* $\mathbf{\Lambda}_{ii} : d_i \times d_i$ *is positive definite and* $\mathbf{\Lambda}_{ij} = \mathbf{0} : d_i \times d_j$.

Definition 3 (Preconditioned Walk-summability) *Consider a precision matrix* $\mathbf{S} : k \times k$ *and clusters* $\mathcal{C}_i : i = 1, 2, \ldots, p$. *The precision matrix* $\mathbf{S}$ *is preconditioned walk-summable if there exists a valid preconditioner* $\mathbf{\Lambda}$ *such that* $\mathbf{\Lambda S \Lambda}$ *is walk-summable.*

We note here that walk-summable precision matrices are always preconditioned walk-summable; however, the converse is not true. This provides theoretical justification for the fact that GaBP-m can converge in cases where GaBP does not. For an explicit example, see Kamper et al. (2019).

2.4. Derivation of Synchronous sGaBP-m

We restrict our focus to a sum-product-based derivation of sGaBP-m. We note that Kamper et al. (2018) partially derive the sGaBP-m updates based on a max-sum formulation (they

do not consider the need for damping) and these formulations (with damping for both) are equivalent. In GaBP-m, the synchronous message updates are

$$m_{ij}^{(n+1)}(\mathbf{x}_j) = \int_{\mathbf{x}_i} \exp\Big[-\frac{1}{2}\mathbf{x}_i'\mathbf{S}_{ii}\mathbf{x}_i + \mathbf{x}_i'\boldsymbol{b}_i\Big]\exp\Big[-\mathbf{x}_i'\mathbf{S}_{ij}\mathbf{x}_j\Big] \prod_{t\in\mathcal{N}_i\setminus j} m_{ti}^{(n)}(\mathbf{x}_i)d\mathbf{x}_i,$$

for all i and all $j \in \mathcal{N}_i$. sGaBP-m incorporates a regularization parameter that encourages $\mathbf{x}_i$ to use values close to $\boldsymbol{\mu}_i^{(n-1)}$ for the purpose of constructing messages at iteration $n+1$. The synchronous message updates for sGaBP-m are

$$m_{ij}^{(n+1)}(\mathbf{x}_j) = \int_{\mathbf{x}_i} \exp\Big[-\frac{1}{2}\mathbf{x}_i'\mathbf{S}_{ii}\mathbf{x}_i+\mathbf{x}_i'\boldsymbol{b}_i-\frac{\lambda}{2}||\mathbf{x}_i-\boldsymbol{\mu}_i^{(n-1)}||_2^2\Big]\exp\Big[-\mathbf{x}_i'\mathbf{S}_{ij}\mathbf{x}_j\Big] \prod_{t\in\mathcal{N}_i\setminus j} m_{ti}^{(n)}(\mathbf{x}_i)d\mathbf{x}_i, \tag{2}$$

for all i and all $j \in \mathcal{N}_i$. We can interpret the regularization in Equation (2) as encouraging the posterior distribution of node i at iteration n to consider values closer to the mean of the posterior distribution at iteration $n-1$ for the purpose of constructing new messages, hence the principle of node regularization. If $m_{ti}^{(n)}(\mathbf{x}_i) \propto \exp\Big[-\frac{1}{2}\mathbf{x}_i'\mathbf{Q}_{ti}^{(n)}\mathbf{x}_i + \mathbf{x}_i'\mathbf{v}_{ti}^{(n)}\Big]$, where $\mathbf{Q}_{ti}^{(n)} : d_i \times d_i$ and $\mathbf{v}_{ti}^{(n)} : d_i \times 1$ for all $t \in \mathcal{N}_i$, then it can be shown that:

$$m_{ij}^{(n+1)}(\mathbf{x}_j) \propto \exp\Big[-\frac{1}{2}\mathbf{x}_i'\mathbf{Q}_{ti}^{(n+1)}\mathbf{x}_i + \mathbf{x}_i'\mathbf{v}_{ti}^{(n+1)}\Big]. \tag{3}$$

The quantities in Equation (3) can be evaluated as:

$$\mathbf{Q}_{ij}^{(n+1)} = -\mathbf{S}_{ji}[\mathbf{P}_{ij}^{(n)}(\lambda)]^{-1}\mathbf{S}_{ij} \tag{4}$$

$$\mathbf{v}_{ij}^{(n+1)} = -\mathbf{S}_{ji}[\mathbf{P}_{ij}^{(n)}(\lambda)]^{-1}[\lambda\boldsymbol{\mu}_i^{(n-1)} + \boldsymbol{b}_i + \sum_{t\in\mathcal{N}_i\setminus j} \mathbf{v}_{ti}^{(n)}], \tag{5}$$

where $\mathbf{P}_{ij}^{(n)}(\lambda) = \lambda\mathbf{I}_{d_i} + \mathbf{S}_{ii} + \sum_{t\in\mathcal{N}_i\setminus j} \mathbf{Q}_{ti}^{(n)}$. To obtain the exact marginal means at convergence it is necessary to perform damping on the progression of the posterior means (see Theorem 1):

$$\boldsymbol{\mu}_i^{(n+1)} = [\mathbf{P}_i^{(n+1)}(\lambda)]^{-1}[\lambda\boldsymbol{\mu}_i^{(n)} + \boldsymbol{b}_i + \sum_{t\in\mathcal{N}_i} \mathbf{v}_{ti}^{(n)}], \tag{6}$$

where $\mathbf{P}_i^{(n+1)}(\lambda) = \lambda\mathbf{I}_{d_i} + \mathbf{S}_{ii} + \sum_{t\in\mathcal{N}_i} \mathbf{Q}_{ti}^{(n+1)}$. To see why this is a form of damping, set $\hat{\mathbf{P}}_i^{(n)} = \mathbf{S}_{ii} + \sum_{t\in\mathcal{N}_i} \mathbf{Q}_{ti}^{(n+1)}$ and consider:

$$\begin{aligned}
\boldsymbol{\mu}_i^{(n)} &= [\mathbf{P}_i^{(n)}(\lambda)]^{-1}[\lambda\boldsymbol{\mu}_i^{(n-1)} + \mathbf{z}_i^{(n)}] \\
&= \lambda[\mathbf{P}_i^{(n)}(\lambda)]^{-1}\boldsymbol{\mu}_i^{(n-1)} + [\mathbf{P}_i^{(n)}(\lambda)]^{-1}\mathbf{z}_i^{(n)} \\
&= \lambda[\mathbf{P}_i^{(n)}(\lambda)]^{-1}\boldsymbol{\mu}_i^{(n-1)} + [\mathbf{P}_i^{(n)}(\lambda)]^{-1}\hat{\mathbf{P}}_i^{(n)}[\hat{\mathbf{P}}_i^{(n)}]^{-1}\mathbf{z}_i^{(n)}.
\end{aligned}$$

Note that $\mathbf{P}_i^{(n)}(\lambda) = \lambda\mathbf{I}_{d_i} + \hat{\mathbf{P}}_i^{(n)}$. If we set $\boldsymbol{\Upsilon}_i^{(n)}(\lambda) = \lambda[\mathbf{P}_i^{(n)}(\lambda)]^{-1}$, then

$$\boldsymbol{\mu}_i^{(n)} = \boldsymbol{\Upsilon}_i^{(n)}(\lambda)\boldsymbol{\mu}_i^{(n-1)} + (\mathbf{I}_{d_i} - \boldsymbol{\Upsilon}_i^{(n)}(\lambda))[\hat{\mathbf{P}}_i^{(n)}]^{-1}\mathbf{z}_i^{(n)}.$$

We can interpret $[\hat{\mathbf{P}}_i^{(n)}]^{-1}\mathbf{z}_i^{(n)}$ as the posterior mean for iteration n, which we would have computed if no damping was applied. Hence, the posterior mean at iteration n can be interpreted as a damped value of the posterior mean of the previous iteration and the mean suggested by the current messages. The damping is done through a matrix $\boldsymbol{\Upsilon}_i^{(n)}(\lambda)$, which depends on λ and the current posterior precision. In contrast to methods such as RGaBP (relaxed GaBP), sGaBP-m automatically computes damping matrices based on the regularization parameter λ. This damping is essential to preserve the exactness of the converged posterior means as the true marginal means. This result is summarized in the following theorem (for a proof see Appendix A):

Theorem 1 *Suppose the iterative updates given in (4) - (6) have converged. The converged posterior means solve the linear system of equations* $\mathbf{Sx} = \boldsymbol{b}$.

Note that, in Theorem 1, we use the notation:

$$\ddot{\mathbf{P}}_i^{(n)}(\lambda) = [\mathbf{P}_i^{(n)}(\lambda)]^{-1}$$
$$\ddot{\mathbf{P}}_{ij}^{(n)}(\lambda) = [\mathbf{P}_{ij}^{(n)}(\lambda)]^{-1}.$$

The posterior distribution associated with node i at iteration n is a normal distribution with mean vector $\boldsymbol{\mu}_i^{(n)}$ and precision matrix $\hat{\mathbf{P}}_i^{(n)}$ (not $\mathbf{P}_i^{(n)}(\lambda)$). An efficient implementation of the updates given in Equations (4) - (6) is given in Algorithm 1. We note that the definition of convergence in Algorithm 1 depends only on how close the posterior means are to solving the linear system $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$. This is because the convergence of the posterior means requires the convergence of the precision components of sGaBP-m (convergence of the posterior means typically implies convergence of the precision components). By precision components of sGaBP-m we mean $\mathbf{Q}_{ij}^{(n)}$, $\ddot{\mathbf{P}}_i^{(n)}(\lambda)$ and $\ddot{\mathbf{P}}_{ij}^{(n)}(\lambda)$.

2.5. Computation Trees for GaBP-m

The idea behind a computation tree is to represent the computations done by GaBP as inference on a tree-structured pairwise MG. This representation leads to analytical formulas for the different components of the GaBP algorithm. Computation trees for GaBP were introduced by Weiss and Freeman (2001) and extended by Kamper et al. (2019) for GaBP-m.

In order to avoid confusion between nodes in the higher-dimensional MG and its computation tree, we will refer to nodes in the MG as clusters. This terminology will be used for the remainder of the paper.

We will highlight the basics of the computation tree analysis and how it applies to this paper. For an example, refer to Appendix D. Each cluster i receives its own computation tree, with topology denoted by $\mathcal{T}_i^{(n)}$. The superscript refers to the depth of the computation tree and each node in the computation tree has a reference to a specific cluster. The first layer of $\mathcal{T}_i^{(n)}$ consist of a single node with a reference to cluster i. The second layer consists of mutually unconnected nodes for each of the clusters in $\mathcal{N}_i$, with the root node as their parent. For $n \geq 2$, $\mathcal{T}_i^{(n+1)}$ is constructed from $\mathcal{T}_i^{(n)}$ by applying the following process to each of the terminal nodes of $\mathcal{T}_i^{(n)}$:

Algorithm 1 Synchronous sGaBP-m

1. Provide a precision matrix $\mathbf{S} : k \times k$, a potential vector $\boldsymbol{b} : k \times 1$ and clusters $\mathcal{C}_i : i = 1, 2, \ldots, p$ as inputs.
2. Specify a tolerance ϵ, a maximum number of iterations m and a regularization parameter λ.
3. Initialize $\mathbf{Q}_{ij}^{(0)} = \mathbf{0} : d_j \times d_j$, $\mathbf{v}_{ij}^{(0)} = \mathbf{0} : d_j \times 1$, $\boldsymbol{\mu}_i^{(-1)} = \boldsymbol{b}_i$ for all i and all $j \in \mathcal{N}_i$.
4. Set Err = Inf and $n = 0$.
5. While Err $> \epsilon$
   (a) Compute $\mathbf{P}_i^{(n)}(\lambda) = \lambda \mathbf{I}_{d_i} + \mathbf{S}_{ii} + \sum_{j \in \mathcal{N}_i} \mathbf{Q}_{ji}^{(n)}$ and $\mathbf{z}_i^{(n)} = \boldsymbol{b}_i + \sum_{j \in \mathcal{N}_i} \mathbf{v}_{ji}^{(n)}$ for $i = 1, 2, \ldots, p$.
   (b) Set $\boldsymbol{\mu}_i^{(n)} = [\mathbf{P}_i^{(n)}(\lambda)]^{-1}[\lambda \boldsymbol{\mu}_i^{(n-1)} + \mathbf{z}_i^{(n)}]$, $\boldsymbol{e}_i^{(n)} = \sum_j \mathbf{S}_{ij} \boldsymbol{\mu}_j^{(n)} - \boldsymbol{b}_i$ and Err $= \max_i \{||\boldsymbol{e}_i^{(n)}||_\infty\}$.
   (c) If Err $> \epsilon$, do for all $i \in \{1, 2, \ldots, p\}$ and all $j \in \mathcal{N}_i$: $\mathbf{Q}_{ij}^{(n+1)} = -\mathbf{S}_{ji}[\mathbf{P}_i^{(n)}(\lambda) - \mathbf{Q}_{ji}^{(n)}]^{-1}\mathbf{S}_{ij}$ and $\mathbf{v}_{ij}^{(n+1)} = -\mathbf{S}_{ji}[\mathbf{P}_i^{(n)}(\lambda) - \mathbf{Q}_{ji}^{(n)}]^{-1}[\lambda \boldsymbol{\mu}_i^{(n-1)} + \mathbf{z}_i^{(n)} - \mathbf{v}_{ji}^{(n)}]$.
   (d) Increment n.
   (e) If $n = m$, break.
6. End.

1. We note the reference of the terminal node (say t) and the reference of its parent in layer $n - 1$ (say s).
2. Create mutually unconnected nodes in the terminal layer of $\mathcal{T}_i^{(n+1)}$, one for each of the clusters in $\mathcal{N}_t \setminus s$.
3. The terminal node under consideration is the parent of these nodes in $\mathcal{T}_i^{(n+1)}$.

Each computation tree $\mathcal{T}_i^{(n)}$ receives a precision matrix $\mathbf{T}_{ii}^{(n)}$. This precision matrix is constructed as follows:

1. The submatrix of $\mathbf{T}_{ii}^{(n)}$ corresponding to a node in the computation tree with reference to cluster t in $\mathcal{T}_i^{(n)}$ is $\mathbf{S}_{tt}$.
2. Consider two linked nodes in $\mathcal{T}_i^{(n)}$ with references to clusters s and t respectively. The corresponding submatrix of $\mathbf{T}_{ii}^{(n)}$ is $\mathbf{S}_{ts}$.
3. All other entries of $\mathbf{T}_{ii}^{(n)}$ are zero.

The computation tree $\mathcal{T}_i^{(n)}$ can be converted into a line topology, $\mathcal{L}_i^{(n)}$. This is done by collecting all nodes in a given layer into a single node.

Consider a $j \in \mathcal{N}_i$. Define $\mathcal{T}_{ji}^{(n)}$ to be the subtree of $\mathcal{T}_i^{(n)}$, rooted at the node in the second layer corresponding to cluster j. We note here that Kamper et al. (2019) use the same notation for a different computation tree (it contains an additional node with a reference to cluster i linked to the root node of our $\mathcal{T}_{ji}^{(n)}$); however, we need our definition for the theoretical results of this paper. Let $\mathbf{T}_{ji}^{(n)}$ be the submatrix of $\mathbf{T}_{ii}^{(n)}$ corresponding to the nodes in the subtree of $\mathcal{T}_{ji}^{(n)}$. We have the following proposition:

Proposition 1 *The following formulas hold:*

$$\ddot{\mathbf{P}}_i^{(n-1)}(0) = [\mathbf{P}_i^{(n-1)}(0)]^{-1} = [\mathbf{G}_{ii}^{(n)}]'[\mathbf{T}_{ii}^{(n)}]^{-1}\mathbf{G}_{ii}^{(n)}$$
$$\ddot{\mathbf{P}}_{ij}^{(n-2)}(0) = [\mathbf{P}_{ij}^{(n-2)}(0)]^{-1} = [\mathbf{G}_{ij}^{(n)}]'[\mathbf{T}_{ij}^{(n)}]^{-1}\mathbf{G}_{ij}^{(n)},$$

where $[\mathbf{G}_{ii}^{(n)}]' = \begin{bmatrix}\mathbf{I}_{d_i} & \mathbf{0}\end{bmatrix}$ *and* $[\mathbf{G}_{ij}^{(n)}]' = \begin{bmatrix}\mathbf{I}_{d_i} & \mathbf{0}\end{bmatrix}$*, both of a suitable dimension. Moreover, convergence is guaranteed if* $\mathbf{S}$ *is preconditioned walk-summable.*

Proof The proof of this proposition follows from Kamper et al. (2019), although they do not consider the convergence of $\ddot{\mathbf{P}}_{ij}^{(n-2)}(0)$ explicitly. However, the convergence of $\ddot{\mathbf{P}}_{ij}^{(n-2)}(0)$, under preconditioned walk-summability, can be proven using similar arguments to the proof of convergence of $\ddot{\mathbf{P}}_i^{(n-1)}(0)$. ■

We note that computation trees can be constructed for any matrix $\mathbf{A}$, even if this matrix is not symmetrical. Next, we summarize Lemma 5 of Kamper et al. (2019):

Lemma 1 *Let* $\mathbf{T}_{ii}^{(n)}(\mathbf{A})$ *be the precision matrix of the computation tree constructed for cluster* i *based on the matrix* $\mathbf{A}$*. We have that:*

$$||\mathbf{T}_{ii}^{(n)}(\mathbf{A})||_\infty \leq ||\mathbf{A}||_\infty.$$

Proposition 1 and Lemma 1 will be used to derive asymptotic expressions for $\ddot{\mathbf{P}}_i(\lambda) = \lim_{n\to\infty} \ddot{\mathbf{P}}_i^{(n)}(\lambda)$ and $\ddot{\mathbf{P}}_{ij}(\lambda) = \lim_{n\to\infty} \ddot{\mathbf{P}}_{ij}^{(n)}(\lambda)$ as $\lambda \to \infty$. These expressions play a key role in the proof of convergence of sGaBP-m.

2.6. Notes on Subscripting

Care should be taken when interpreting the subscripts of matrices used in this paper, although their meaning should be clear from the context. For instance, the subscripts of the matrix $\mathbf{T}_{ii}^{(n)}$ refer to the cluster for which the computation tree is designed (cluster i in this case). This is in contrast to $\mathbf{S}_{ij}$, which denotes the submatrix of $\mathbf{S}$ associated with the variables in cluster i and cluster j for the rows and columns respectively.

3. Convergence of sGaBP-m

This section is dedicated to proving the convergence of sGaBP-m, given sufficient regularization. The construction of this section is as follows:

- Section 3.1. We prove the convergence of the precision components given sufficient regularization.
- Section 3.2. Asymptotic expressions for $\ddot{\mathbf{P}}_i(\lambda)$ and $\ddot{\mathbf{P}}_{ij}(\lambda)$ as $\lambda \to \infty$ are derived.
- Section 3.3. We derive the convergence of the mean components of sGaBP-m, given sufficient regularization. This proof is done under the assumption that the precision components have converged. The mean components are the $\boldsymbol{\mu}_i^{(n)}$ and $\mathbf{v}_{ij}^{(n)}$ in Algorithm 1.
- Section 3.4. We prove overall convergence of sGaBP-m, given sufficient regularization.

3.1. Convergence of Precision Components

Note that it is sufficient to prove the convergence of $\ddot{\mathbf{P}}_i^{(n)}(\lambda)$ and $\ddot{\mathbf{P}}_{ij}^{(n)}(\lambda)$, since $\mathbf{Q}_{ij}^{(n+1)} = -\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}^{(n)}(\lambda)\mathbf{S}_{ij}$. The proof of convergence of the precision components is simple, due to the following proposition:

Proposition 2 *$\ddot{\mathbf{P}}_i^{(n)}(\lambda)$ and $\ddot{\mathbf{P}}_{ij}^{(n)}(\lambda)$ obtained from sGaBP-m applied to $\mathbf{S}$ are equivalent to $\ddot{\mathbf{P}}_i^{(n)}(0)$ and $\ddot{\mathbf{P}}_{ij}^{(n)}(0)$ respectively, obtained from GaBP-m applied to $\mathbf{S} + \lambda\mathbf{I}_k$.*

Note that Proposition 2 is evident from Algorithm 1. We now have the following theorem:

Theorem 2 *Consider sGaBP-m applied to $\mathbf{S}$ with a regularization parameter λ, then $\ddot{\mathbf{P}}_i^{(n)}(\lambda)$ and $\ddot{\mathbf{P}}_{ij}^{(n)}(\lambda)$ will converge if $\mathbf{S} + \lambda\mathbf{I}_k$ is preconditioned walk-summable.*

Proof The proof follows directly from Propositions 1 and 2. ■

The following corollary to Theorem 2 also holds:

Corollary 1 *Consider sGaBP-m applied to $\mathbf{S}$ with a regularization parameter λ, then $\ddot{\mathbf{P}}_i^{(n)}(\lambda)$ and $\ddot{\mathbf{P}}_{ij}^{(n)}(\lambda)$ will converge if $\mathbf{S} + \lambda\mathbf{I}_k$ is diagonally dominant.*

Proof This follows from the fact that a diagonally dominant $\mathbf{S} + \lambda\mathbf{I}_k$ is walk-summable, and hence also preconditioned walk-summable. ■

Although the proof of convergence of the precision components of sGaBP-m is relatively simple, the challenge is deriving associated asymptotic expressions as $\lambda \to \infty$. We consider this in the next section.

3.2. Asymptotic Expressions

Consider the following lemma:

Lemma 2 *We have the following asymptotic expressions:*

$$\lambda\ddot{\mathbf{P}}_i(\lambda) = \mathbf{I}_{d_i} - \frac{1}{\lambda}\mathbf{S}_{ii} + \mathcal{O}(\frac{1}{\lambda^2})$$
$$\mathbf{U}_{ij}(\lambda) = -\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}(\lambda) = -\frac{1}{\lambda}\mathbf{S}_{ji} + \mathcal{O}(\frac{1}{\lambda^2}),$$

for all $i \in \mathcal{V}$ and all $j \in \mathcal{N}_i$.

Proof We start by defining the following:

1. Let $\mathcal{T}_i^{(n)}(\lambda)$ and $\mathcal{T}_{ij}^{(n)}(\lambda)$ be the computation trees constructed from $\mathbf{S}+\lambda\mathbf{I}_k$, with $\mathbf{T}_{ii}^{(n)}(\lambda)$ and $\mathbf{T}_{ij}^{(n)}(\lambda)$ the associated precision matrices respectively.
2. Set $\mathbf{S}=[s_{ij}]$, $\mathbf{D}=\text{diag}(s_{11},s_{22},\ldots,s_{kk})$ and $\mathbf{R}=\mathbf{D}-\mathbf{S}$.
3. Let $\bar{\mathbf{S}}(\lambda)=(\lambda\mathbf{I}_k+\mathbf{D})^{-0.5}(\lambda\mathbf{I}_k+\mathbf{S})(\lambda\mathbf{I}_k+\mathbf{D})^{-0.5}$ and set $\bar{\mathbf{R}}(\lambda)=\mathbf{I}_k-\bar{\mathbf{S}}(\lambda)$.
4. $\bar{\mathbf{T}}_{ii}^{(n)}(\lambda)$ and $\bar{\mathbf{T}}_{ij}^{(n)}(\lambda)$ are defined analogous to $\mathbf{T}_{ii}^{(n)}(\lambda)$ and $\mathbf{T}_{ij}^{(n)}(\lambda)$ respectively; however, they are constructed from $\bar{\mathbf{S}}(\lambda)$.
5. $\bar{\mathcal{R}}_{ii}^{(n)}(\lambda)=\mathbf{I}_{m_{n;i}}-\bar{\mathbf{T}}_{ii}^{(n)}(\lambda)$, where $\bar{\mathbf{T}}_{ii}^{(n)}(\lambda)$ is $m_{n;i}\times m_{n;i}$. The matrix $\bar{\mathcal{R}}_{ii}^{(n)}(\lambda)$ contains only zeros on its diagonal.
6. Note that $\bar{\mathcal{R}}_{ii}^{(n)}(\lambda)$ is the precision matrix of the computation tree for cluster i, with a depth of n, constructed from the matrix $\bar{\mathbf{R}}(\lambda)$.
7. It can be shown that:

$$[\mathbf{G}_{ii}^{(n)}]'[\mathbf{T}_{ii}^{(n)}(\lambda)]^{-1}\mathbf{G}_{ii}^{(n)}=(\lambda\mathbf{I}_{d_i}+\mathbf{D}_{ii})^{-0.5}[\mathbf{G}_{ii}^{(n)}]'[\bar{\mathbf{T}}_{ii}^{(n)}(\lambda)]^{-1}[\mathbf{G}_{ii}^{(n)}](\lambda\mathbf{I}_{d_i}+\mathbf{D}_{ii})^{-0.5}.$$

By Lemma 1 we see that:

$$||\bar{\mathcal{R}}_{ii}^{(n)}(\lambda)||_\infty\leq||\bar{\mathbf{R}}(\lambda)||_\infty. \tag{7}$$

A further consequence of Equation (7) is:

$$||(\bar{\mathcal{R}}_{ii}^{(n)}(\lambda))^t||_\infty\leq||\bar{\mathbf{R}}(\lambda)||_\infty^t.$$

Consider:

$$\begin{aligned}\bar{\mathbf{R}}(\lambda)&=\mathbf{I}_k-\bar{\mathbf{S}}(\lambda)\\&=(\lambda\mathbf{I}_k+\mathbf{D})^{-0.5}(\mathbf{D}-\mathbf{S})(\lambda\mathbf{I}_k+\mathbf{D})^{-0.5}\\&=(\lambda\mathbf{I}_k+\mathbf{D})^{-0.5}\mathbf{R}(\lambda\mathbf{I}_k+\mathbf{D})^{-0.5}.\end{aligned} \tag{8}$$

From (8) we have:

$$\begin{aligned}||\bar{\mathbf{R}}(\lambda)||_\infty&=||(\lambda\mathbf{I}_k+\mathbf{D})^{-0.5}\mathbf{R}(\lambda\mathbf{I}_k+\mathbf{D})^{-0.5}||_\infty\\&\leq||(\lambda\mathbf{I}_k+\mathbf{D})^{-0.5}||_\infty||\mathbf{R}||_\infty||(\lambda\mathbf{I}_k+\mathbf{D})^{-0.5}||_\infty\\&=\frac{||\mathbf{R}||_\infty}{\lambda+\min_l\{S_{ll}\}}.\end{aligned} \tag{9}$$

The bound in (9) shows that there will always be a selection of λ such that $\bar{\mathbf{S}}(\lambda)$ is strictly diagonally dominant. Furthermore, $\rho(\bar{\mathcal{R}}_{ii}^{(n)}(\lambda))\leq||\bar{\mathcal{R}}_{ii}^{(n)}(\lambda)||_\infty\leq||\bar{\mathbf{R}}(\lambda)||_\infty\leq r_\lambda$, where $r_\lambda=\frac{||\mathbf{R}||_\infty}{\lambda+\min_l\{S_{ll}\}}$. From this point onwards we assume $\lambda>||\mathbf{R}||_\infty-\min_l\{S_{ll}\}$ such that

$r_\lambda < 1$. Hence, $\rho(\bar{\mathcal{R}}_{ii}^{(n)}(\lambda)) < 1$, and we can apply the Neumann power series $[\mathbf{I}_{m_n} - \bar{\mathcal{R}}_{ii}^{(n)}(\lambda)]^{-1} = \sum_{t=0}^{\infty}[\bar{\mathcal{R}}_{ii}^{(n)}(\lambda)]^t$ to obtain:

$$\begin{aligned}[\mathbf{G}_{ii}^{(n)}]'[\mathbf{I}_{m_n} - \bar{\mathcal{R}}_{ii}^{(n)}(\lambda)]^{-1}\mathbf{G}_{ii}^{(n)} &= \sum_{t=0}^{\infty}[\mathbf{G}_{ii}^{(n)}]'[\bar{\mathcal{R}}_{ii}^{(n)}(\lambda)]^t\mathbf{G}_{ii}^{(n)} \\ &= \mathbf{I}_{d_i} + \bar{\mathbf{R}}_{ii}(\lambda) + \sum_{l\in\{\mathcal{N}_i\cup i\}}\bar{\mathbf{R}}_{il}(\lambda)\bar{\mathbf{R}}_{li}(\lambda) + \mathbf{\Omega}_{ii}^{(n)}(\lambda), \qquad (10)\end{aligned}$$

where $\mathbf{\Omega}_{ii}^{(n)}(\lambda) = \sum_{t=3}^{\infty}[\mathbf{G}_{ii}^{(n)}]'[\bar{\mathcal{R}}_{ii}^{(n)}(\lambda)]^t\mathbf{G}_{ii}^{(n)}$.

We note three important consequences of (10):

1. Since $\bar{\mathbf{S}}(\lambda)$ is strictly diagonally dominant, $\lim_{n\to\infty}[\mathbf{G}_{ii}^{(n)}]'[\mathbf{I}_{m_n} - \bar{\mathcal{R}}_{ii}^{(n)}(\lambda)]^{-1}\mathbf{G}_{ii}^{(n)}$ exists by Corollary 1, and therefore $\lim_{n\to\infty}\mathbf{\Omega}_{ii}^{(n)}(\lambda) = \mathbf{\Omega}_{ii}(\lambda)$ for a specified matrix $\mathbf{\Omega}_{ii}(\lambda)$.

2. We see that:

$$\begin{aligned}||\mathbf{\Omega}_{ii}^{(n)}(\lambda)||_\infty &\leq \sum_{t=3}^{\infty}||[\mathbf{G}_{ii}^{(n)}]'||_\infty||[\bar{\mathcal{R}}_{ii}^{(n)}(\lambda)]^t||_\infty||\mathbf{G}_{ii}^{(n)}||_\infty \\ &= \sum_{t=3}^{\infty}||[\bar{\mathcal{R}}_{ii}^{(n)}(\lambda)]^t||_\infty \\ &\leq \sum_{t=3}^{\infty} r_\lambda^t \\ &= \frac{r_\lambda^3}{1-r_\lambda} = \mathcal{O}\left(\frac{1}{\lambda^3}\right).\end{aligned}$$

3. Points (1) and (2) guarantee that $\mathbf{\Omega}_{ii}(\lambda) = \mathcal{O}(\frac{1}{\lambda^2})$ (see Lemma 4 in Appendix D).

Consider further simplification of (8). We note that $(\mathbf{I}_k + \frac{\mathbf{D}}{\lambda})^{-0.5}$ is a diagonal matrix with the lth diagonal entry equal to:

$$\left(1 + \frac{S_{ll}}{\lambda}\right)^{-0.5} = 1 + \sum_{t=1}^{\infty}\binom{-0.5}{t}\frac{S_{ll}^t}{\lambda^t} = 1 + \mathcal{O}\left(\frac{1}{\lambda}\right), \qquad (11)$$

where we assume that $\lambda > S_{ll}$ and $\binom{\alpha}{t} = \frac{\alpha(\alpha-1)\ldots(\alpha-t+1)}{t!}$ denote the generalized binomial coefficients. As a consequence, we have:

$$\left(\mathbf{I}_k + \frac{\mathbf{D}}{\lambda}\right)^{-0.5} = \mathbf{I}_k + \mathcal{O}\left(\frac{1}{\lambda}\right),$$

and

$$\begin{aligned}\bar{\mathbf{R}}(\lambda) &= (\lambda\mathbf{I}_k + \mathbf{D})^{-0.5}\mathbf{R}(\lambda\mathbf{I}_k + \mathbf{D})^{-0.5} \\ &= \frac{1}{\lambda}\left(\mathbf{I}_k + \frac{\mathbf{D}}{\lambda}\right)^{-0.5}\mathbf{R}\left(\mathbf{I}_k + \frac{\mathbf{D}}{\lambda}\right)^{-0.5} \\ &= \frac{1}{\lambda}\left(\mathbf{I}_k + \mathcal{O}\left(\frac{1}{\lambda}\right)\right)\mathbf{R}\left(\mathbf{I}_k + \mathcal{O}\left(\frac{1}{\lambda}\right)\right) \\ &= \frac{1}{\lambda}\mathbf{R} + \mathcal{O}\left(\frac{1}{\lambda^2}\right). \end{aligned} \tag{12}$$

From (12) we see that:

$$\mathbf{I}_{d_i} + \bar{\mathbf{R}}_{ii}(\lambda) + \sum_{l\in\{\mathcal{N}_i\cup i\}} \bar{\mathbf{R}}_{il}(\lambda)\bar{\mathbf{R}}_{li}(\lambda) = \mathbf{I}_{d_i} + \frac{1}{\lambda}\mathbf{R}_{ii} + \mathcal{O}\left(\frac{1}{\lambda^2}\right),$$

where no terms involve the iteration number n. Consider:

$$\begin{aligned}\lambda\ddot{\mathbf{P}}_i^{(n-1)}(\lambda) &= \lambda[\mathbf{G}_{ii}^{(n)}]'[\mathbf{T}_{ii}^{(n)}(\lambda)]^{-1}\mathbf{G}_{ii}^{(n)} \\ &= \lambda(\lambda\mathbf{I}_{d_i} + \mathbf{D}_{ii})^{-0.5}[\mathbf{G}_{ii}^{(n)}]'[\bar{\mathbf{T}}_{ii}^{(n)}(\lambda)]^{-1}[\mathbf{G}_{ii}^{(n)}](\lambda\mathbf{I}_{d_i} + \mathbf{D}_{ii})^{-0.5} \\ &= \lambda(\lambda\mathbf{I}_{d_i} + \mathbf{D}_{ii})^{-0.5}\left[\mathbf{I}_{d_i} + \frac{1}{\lambda}\mathbf{R}_{ii} + \mathcal{O}(\frac{1}{\lambda^2}) + \mathbf{\Omega}_{ii}^{(n)}(\lambda)\right](\lambda\mathbf{I}_{d_i} + \mathbf{D}_{ii})^{-0.5} \\ &\to \left(\mathbf{I}_{d_i} + \frac{\mathbf{D}_{ii}}{\lambda}\right)^{-0.5}\left[\mathbf{I}_{d_i} + \frac{1}{\lambda}\mathbf{R}_{ii} + \mathcal{O}(\frac{1}{\lambda^2})\right]\left(\mathbf{I}_{d_i} + \frac{\mathbf{D}_{ii}}{\lambda}\right)^{-0.5} \end{aligned} \tag{13}$$

as $n \to \infty$, and where it should be noted that $\mathbf{\Omega}_{ii}(\lambda) = \mathcal{O}(\frac{1}{\lambda^2})$. A second-order expansion of (11) yields:

$$\left(1 + \frac{S_{ll}}{\lambda}\right)^{-0.5} = 1 - 0.5\frac{S_{ll}}{\lambda} + \sum_{t=2}^{\infty}\binom{-0.5}{t}\frac{S_{ll}^t}{\lambda^t} = 1 - 0.5\frac{S_{ll}}{\lambda} + \mathcal{O}\left(\frac{1}{\lambda^2}\right),$$

which gives $\left(\mathbf{I}_{d_i} + \frac{\mathbf{D}_{ii}}{\lambda}\right)^{-0.5} = \mathbf{I}_{d_i} - 0.5\frac{\mathbf{D}_{ii}}{\lambda} + \mathcal{O}(\frac{1}{\lambda^2})$. Equation (13) becomes:

$$\begin{aligned}&\left(\mathbf{I}_{d_i} + \frac{\mathbf{D}_{ii}}{\lambda}\right)^{-0.5}\left[\mathbf{I}_{d_i} + \frac{1}{\lambda}\mathbf{R}_{ii} + \mathcal{O}\left(\frac{1}{\lambda^2}\right)\right]\left(\mathbf{I}_{d_i} + \frac{\mathbf{D}_{ii}}{\lambda}\right)^{-0.5} \\ &= \left(\mathbf{I}_{d_i} - 0.5\frac{\mathbf{D}_{ii}}{\lambda} + \mathcal{O}\left(\frac{1}{\lambda^2}\right)\right)\left[\mathbf{I}_{d_i} + \frac{1}{\lambda}\mathbf{R}_{ii} + \mathcal{O}\left(\frac{1}{\lambda^2}\right)\right]\left(\mathbf{I}_{d_i} - 0.5\frac{\mathbf{D}_{ii}}{\lambda} + \mathcal{O}\left(\frac{1}{\lambda^2}\right)\right) \\ &= \left(\mathbf{I}_{d_i} - 0.5\frac{\mathbf{D}_{ii}}{\lambda}\right)\left[\mathbf{I}_{d_i} + \frac{1}{\lambda}\mathbf{R}_{ii}\right]\left(\mathbf{I}_{d_i} - 0.5\frac{\mathbf{D}_{ii}}{\lambda}\right) + \mathcal{O}\left(\frac{1}{\lambda^2}\right) \\ &= \mathbf{I}_{d_i} + \frac{1}{\lambda}(\mathbf{R}_{ii} - \mathbf{D}_{ii}) + \mathcal{O}\left(\frac{1}{\lambda^2}\right) \\ &= \mathbf{I}_{d_i} - \frac{1}{\lambda}\mathbf{S}_{ii} + \mathcal{O}\left(\frac{1}{\lambda^2}\right). \end{aligned}$$

Finally we obtain:

$$\lambda\ddot{\mathbf{P}}_i(\lambda) = \lim_{n\to\infty} \lambda\ddot{\mathbf{P}}_i^{(n-1)}(\lambda) = \mathbf{I}_{d_i} - \frac{1}{\lambda}\mathbf{S}_{ii} + \mathcal{O}(\frac{1}{\lambda^2}). \tag{14}$$

Similar to the derivation of (14), it can be shown that:

$$\ddot{\mathbf{P}}_{ij}(\lambda) = \frac{1}{\lambda}\mathbf{I}_{d_i} + \mathcal{O}\left(\frac{1}{\lambda^2}\right).$$

Defining $\mathbf{U}_{ij}(\lambda) = -\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}(\lambda)$, we see that:

$$\mathbf{U}_{ij}(\lambda) = -\frac{1}{\lambda}\mathbf{S}_{ji} + \mathcal{O}\left(\frac{1}{\lambda^2}\right).$$

■

In the next section, we show that, after the precision components of sGaBP-m have converged, the updates of the mean components become linear. The linear update matrix is determined by the matrices $\mathbf{U}_{ij}(\lambda)$ and $\lambda\ddot{\mathbf{P}}_i(\lambda)$ for $i \in \mathcal{V}$ and $j \in \mathcal{N}_i$, and hence they play a crucial role in the convergence behavior of sGaBP-m. Of particular interest is the behavior of the spectral radius of the linear update matrix as $\lambda \to \infty$. The asymptotic expressions derived in this section allow us to show that this spectral radius will be less than 1 for large enough λ.

3.3. Convergence of Mean Components

We now turn our attention to proving the convergence of the mean components, given the convergence of the precision components. We will also assume, without loss of generality, that we are dealing with a fully connected higher-dimensional MG.

3.3.1. Convergence to Linear Updates

By Theorem 2 we know that, if λ is sufficiently large, then $\ddot{\mathbf{P}}_{ij}^{(n)}(\lambda) \to \ddot{\mathbf{P}}_{ij}(\lambda)$ and $\ddot{\mathbf{P}}_i^{(n)}(\lambda) \to \ddot{\mathbf{P}}_i(\lambda)$ as $n \to \infty$ for specified $\ddot{\mathbf{P}}_{ij}(\lambda)$ and $\ddot{\mathbf{P}}_i(\lambda)$. For the remainder of Section 3, we will write

$$\ddot{\mathbf{P}}_i = \ddot{\mathbf{P}}_i(\lambda)$$
$$\mathbf{U}_{ij} = \mathbf{U}_{ij}(\lambda),$$

and proceed under the assumption that the precision components of sGaBP-m have converged. Under this assumption, the remaining components of sGaBP-m are updated through the equations:

$$\mathbf{v}_{ij}^{(n+1)} = \mathbf{U}_{ij}[\lambda\boldsymbol{\mu}_i^{(n-1)} + \boldsymbol{b}_i + \sum_{t\neq i,j} \mathbf{v}_{ti}^{(n)}] \tag{15}$$

$$\boldsymbol{\mu}_i^{(n+1)} = \ddot{\mathbf{P}}_i[\lambda\boldsymbol{\mu}_i^{(n)} + \boldsymbol{b}_i + \sum_{t\neq i} \mathbf{v}_{ti}^{(n+1)}] \tag{16}$$

for all i and $j \neq i$. We now show that these updates can be done through a linear transformation matrix. Define

$$\boldsymbol{\gamma}_i^{(n+1)} = (\mathbf{v}_{1i}^{(n+1)'}, \mathbf{v}_{2i}^{(n+1)'}, \ldots, \mathbf{v}_{(i-1);i}^{(n+1)'}, \mathbf{v}_{(i+1);i}^{(n+1)'}, \ldots, \mathbf{v}_{pi}^{(n+1)'})',$$

and set $\boldsymbol{\gamma}^{(n+1)} = (\boldsymbol{\gamma}_1^{(n+1)'}, \boldsymbol{\gamma}_2^{(n+1)'}, \ldots, \boldsymbol{\gamma}_p^{(n+1)'}, \boldsymbol{\mu}_1^{(n)'}, \ldots, \boldsymbol{\mu}_p^{(n)'})'$. Note that the size of $\boldsymbol{\gamma}_i^{(n+1)}$ is $(p-1)d_i$. Set $m_1 = (p-1)\sum_i d_i = k(p-1)$ and $m_2 = m_1 + k = kp$. We are going to show that there is a matrix $\mathbf{L} : m_2 \times m_2$, such that:

$$\boldsymbol{\gamma}^{(n+1)} = \boldsymbol{\gamma}_0 + \mathbf{L}\boldsymbol{\gamma}^{(n)}, \tag{17}$$

where $\boldsymbol{\gamma}_0$ is an $m_2 \times 1$ vector. Consider $\mathbf{U}_{ij} : d_j \times d_i$ and set

$$\boldsymbol{\gamma}_{0i} = (\boldsymbol{b}_1'\mathbf{U}_{1i}', \boldsymbol{b}_2'\mathbf{U}_{2i}', \ldots, \boldsymbol{b}_{i-1}'\mathbf{U}_{(i-1);i}', \boldsymbol{b}_{i+1}'\mathbf{U}_{(i+1);i}', \ldots, \boldsymbol{b}_p'\mathbf{U}_{pi}')'.$$

Let $\boldsymbol{\gamma}_0 = (\boldsymbol{\gamma}_{01}', \boldsymbol{\gamma}_{02}', \ldots, \boldsymbol{\gamma}_{0p}', \boldsymbol{b}_1'\ddot{\mathbf{P}}_1', \ldots, \boldsymbol{b}_p'\ddot{\mathbf{P}}_p')'$. For the linear update matrix, consider the decomposition

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{11} : m_1 \times m_1 & \mathbf{L}_{12} : m_1 \times k \\ \mathbf{L}_{21} : k \times m_1 & \mathbf{L}_{22} : k \times k \end{bmatrix}.$$

The construction of $\mathbf{L}$ is as follows:

1. Consider first the matrix $\mathbf{L}_{11}$. We can decompose $\mathbf{L}_{11}$ into blocks, where each block corresponds to a row message and a column message. Consider block s,t of $\mathbf{L}_{11}$ and assign to this block a row index and column index, which are to be obtained from the first m_1 components of $\boldsymbol{\gamma}_0$. To obtain the row and column message indices of this block, we move to entry s and entry t of $\boldsymbol{\gamma}_0$. If entry s is $\mathbf{U}_{ji}\boldsymbol{b}_j$ and entry t is $\mathbf{U}_{ru}\boldsymbol{b}_r$, then the row and column indices of block s,t are (j,i) (message from j to i) and (r,u) (message from r to u) respectively.

2. Consider block s,t of $\mathbf{L}_{11}$ with row indices (j,i) and column indices (r,u). If $u = j$ and $r \neq j,i$; then this block is $\mathbf{U}_{ji}$, otherwise the block is a matrix of zeros.

3. The matrix $\mathbf{L}_{22}$ has a decomposition according to the last k components of $\boldsymbol{\gamma}_0$. Block s,t of $\mathbf{L}_{22}$ is associated with $\boldsymbol{b}_s$ (row index is s) and $\boldsymbol{b}_t$ (column index is t). A block of $\mathbf{L}_{22}$, corresponding to a row and column index of s and t respectively, is $\mathbf{0}$ if $s \neq t$ and $\lambda\ddot{\mathbf{P}}_t$ otherwise. Therefore, $\mathbf{L}_{22}$ is block-diagonal.

4. The matrix $\mathbf{L}_{12}$ has a decomposition according to the row indices of $\mathbf{L}_{11}$ and the column indices of $\mathbf{L}_{22}$. Block s,t has a row index (j,i) and a column index u. This block is $\lambda\mathbf{U}_{ji}$ if $u = j$, and a matrix of zeros otherwise.

5. $\mathbf{L}_{21}$ has a block decomposition with row indices equal to the row indices of $\mathbf{L}_{22}$ and the column indices equal to the column indices of $\mathbf{L}_{11}$. Block s,t has a row index u and a column index (j,i). This block is $\ddot{\mathbf{P}}_u$ if $i = u$ and $j \neq u$, otherwise it is a matrix of zeros.

Let us look at an example for $p = 3$ with the following precision matrix:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{S}_{23} \\ \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{S}_{33} \end{bmatrix}.$$

After the convergence of the precision components, we have the the matrix

$$\begin{bmatrix} \ddot{\mathbf{P}}_1 & \mathbf{U}_{12} & \mathbf{U}_{13} \\ \mathbf{U}_{21} & \ddot{\mathbf{P}}_2 & \mathbf{U}_{23} \\ \mathbf{U}_{31} & \mathbf{U}_{32} & \ddot{\mathbf{P}}_3 \end{bmatrix},$$

which is to be used in the construction of the linear update matrix. The vectors $\boldsymbol{\gamma}^{(n+1)}$ and $\boldsymbol{\gamma}_0$, given in Equation (17), are

$$\boldsymbol{\gamma}^{(n+1)} = \begin{bmatrix} \mathbf{v}_{21}^{(n+1)} \\ \mathbf{v}_{31}^{(n+1)} \\ \mathbf{v}_{12}^{(n+1)} \\ \mathbf{v}_{32}^{(n+1)} \\ \mathbf{v}_{13}^{(n+1)} \\ \mathbf{v}_{23}^{(n+1)} \\ \boldsymbol{\mu}_1^{(n)} \\ \boldsymbol{\mu}_2^{(n)} \\ \boldsymbol{\mu}_3^{(n)} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\gamma}_0 = \begin{bmatrix} \mathbf{U}_{21}\boldsymbol{b}_2 \\ \mathbf{U}_{31}\boldsymbol{b}_3 \\ \mathbf{U}_{12}\boldsymbol{b}_1 \\ \mathbf{U}_{32}\boldsymbol{b}_3 \\ \mathbf{U}_{13}\boldsymbol{b}_1 \\ \mathbf{U}_{23}\boldsymbol{b}_2 \\ \ddot{\mathbf{P}}_1\boldsymbol{b}_1 \\ \ddot{\mathbf{P}}_2\boldsymbol{b}_2 \\ \ddot{\mathbf{P}}_3\boldsymbol{b}_3 \end{bmatrix}$$

respectively. The linear update matrix, with the row and column indices as discussed, is

$$\mathbf{L} = \begin{array}{c} \\ 2\to1 \\ 3\to1 \\ 1\to2 \\ 3\to2 \\ 1\to3 \\ 2\to3 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 2\to1 & 3\to1 & 1\to2 & 3\to2 & 1\to3 & 2\to3 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{U}_{21} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda\mathbf{U}_{21} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{U}_{31} & \mathbf{0} & \mathbf{0} & \lambda\mathbf{U}_{31} \\ \mathbf{0} & \mathbf{U}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda\mathbf{U}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{U}_{32} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda\mathbf{U}_{32} \\ \mathbf{U}_{13} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda\mathbf{U}_{13} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{U}_{23} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda\mathbf{U}_{23} & \mathbf{0} \\ \ddot{\mathbf{P}}_1 & \ddot{\mathbf{P}}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda\ddot{\mathbf{P}}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddot{\mathbf{P}}_2 & \ddot{\mathbf{P}}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda\ddot{\mathbf{P}}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddot{\mathbf{P}}_3 & \ddot{\mathbf{P}}_3 & \mathbf{0} & \mathbf{0} & \lambda\ddot{\mathbf{P}}_3 \end{bmatrix} \end{array}.$$

Note how the row and column messages follow the subscripts of the messages in $\boldsymbol{\gamma}^{(n+1)}$. For this example, we see that Equation (17) performs the updates given in (15) and (16).

Returning to the general case, the vector $\boldsymbol{\gamma}^{(n+1)} \to (\mathbf{I} - \mathbf{L})^{-1}\boldsymbol{\gamma}_0$ as $n \to \infty$ if, and only if, $\rho(\mathbf{L}) < 1$.

Let $\boldsymbol{\Delta} = \text{diag}(\mathbf{1}_{m_1}, \lambda \mathbf{1}_k)$, where $\mathbf{1}_i$ is an $i \times 1$ vector containing only ones. To study the spectral radius of $\mathbf{L}$, we consider the following matrix:

$$\tilde{\mathbf{L}} = \boldsymbol{\Delta}\mathbf{L}\boldsymbol{\Delta}^{-1} = \begin{bmatrix} \tilde{\mathbf{L}}_{11} : m_1 \times m_1 & \tilde{\mathbf{L}}_{12} : m_1 \times k \\ \tilde{\mathbf{L}}_{21} : k \times m_1 & \tilde{\mathbf{L}}_{22} : k \times k \end{bmatrix},$$

where $\lambda > 0$, $\tilde{\mathbf{L}}_{11} = \mathbf{L}_{11}$, $\tilde{\mathbf{L}}_{12} = \frac{1}{\lambda}\mathbf{L}_{12}$, $\tilde{\mathbf{L}}_{21} = \lambda \mathbf{L}_{21}$ and $\tilde{\mathbf{L}}_{22} = \mathbf{L}_{22}$. Define $\mathbf{L}_{st}$ as the block of $\mathbf{L}$ corresponding to (s, t). A consequence of this scaling is that all non-zero blocks in a given row block are identical. Note that σ is an eigenvalue of $\tilde{\mathbf{L}}$ if, and only if, it is an eigenvalue of $\mathbf{L}$. Hence, our objective is to show that the spectral radius of $\tilde{\mathbf{L}}$ is less than 1 when λ is sufficiently large. To this end, we consider the asymptotic behavior of $\tilde{\mathbf{L}}$ as $\lambda \to \infty$.

3.3.2. The Asymptotic Linear Update Matrix

Consider a matrix $\mathbf{H}$ defined as

$$\mathbf{H} : m_1 \times k = \begin{array}{c} \begin{array}{cccc} 1 & 2 & \dots & \text{p} \end{array} \\ \begin{bmatrix} \mathbf{I}_{d_1} & \dots & \dots & \dots \\ \mathbf{I}_{d_1} & \dots & \dots & \dots \\ \vdots & \dots & \dots & \dots \\ \mathbf{I}_{d_1} & \dots & \dots & \dots \\ \dots & \mathbf{I}_{d_2} & \dots & \dots \\ \dots & \mathbf{I}_{d_2} & \dots & \dots \\ \dots & \vdots & \dots & \dots \\ \dots & \mathbf{I}_{d_2} & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \mathbf{I}_{d_p} \\ \dots & \dots & \dots & \mathbf{I}_{d_p} \\ \dots & \dots & \dots & \mathbf{I}_{d_p} \\ \dots & \dots & \dots & \mathbf{I}_{d_p} \end{bmatrix} \end{array}, \tag{18}$$

where each column block (indexed by i) contains exactly $(p-1)$ $\mathbf{I}_{d_i}$ matrices stacked in consecutive rows; the remainder of the column block contains zeros. The way the stacking is done is shown clearly in Equation (18), where the $\dots$ and $\vdots$ indicate that the corresponding part of the matrix is filled with zeros. For our example we have

$$\mathbf{H} = \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} \mathbf{I}_{d_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{d_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{d_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{d_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_3} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_3} \end{bmatrix} \end{array},$$

where each $\mathbf{0}$ is of a suitable dimension. We now have the following expressions:

$$\tilde{\mathbf{L}}_{12} = \frac{1}{p-2}\mathbf{L}_{11}\mathbf{H}$$
$$\tilde{\mathbf{L}}_{21} = \mathbf{L}_{22}\mathbf{H}'.$$

Set $\delta = \frac{1}{\lambda}$ and consider the following matrices:

1. $\mathbf{M}_{11}$ has an identical construction to $\mathbf{L}_{11}$, using $-\mathbf{S}_{ji}$ instead of $\mathbf{U}_{ij}$.
2. $\mathbf{M}_{22}$ has an identical construction to $\mathbf{L}_{22}$, using $\mathbf{S}_{ii}$ instead of $\lambda\ddot{\mathbf{P}}_i$.

By Lemma 2 we have that:

$$\mathbf{L}_{11} = \delta\mathbf{M}_{11} + \mathcal{O}(\delta^2)$$
$$\mathbf{L}_{22} = \mathbf{I}_k - \delta\mathbf{M}_{22} + \mathcal{O}(\delta^2).$$

Therefore, we have the following asymptotic expression for the scaled linear update matrix:

$$\tilde{\mathbf{L}} = \begin{bmatrix} \delta\mathbf{M}_{11} & \frac{\delta}{p-2}\mathbf{M}_{11}\mathbf{H} \\ (\mathbf{I}_k - \delta\mathbf{M}_{22})\mathbf{H}' & (\mathbf{I}_k - \delta\mathbf{M}_{22}) \end{bmatrix} + \mathcal{O}(\delta^2).$$

The behavior of the eigenvalues of $\mathbf{L}$ (which is equivalent to the behavior of the eigenvalues of $\tilde{\mathbf{L}}$) as $\lambda \to \infty$ is given in the following theorem.

Theorem 3 *Consider applying sGaBP-m to a precision matrix $\mathbf{S} : k \times k$ and a potential vector $\boldsymbol{b} : k \times 1$, where variables are assigned to nodes according to clusters $\mathcal{C}_i : i = 1, 2, \ldots, p$. Suppose that $\mathbf{L}$ is the linear update matrix obtained after the precision components have converged. Let $\nu_{stu} \geq 0 : u = 1, 2, \ldots, d_s$ $(s \neq t)$ be the eigenvalues of $\mathbf{S}_{st}\mathbf{S}_{ts}$ and $\sigma_1, \sigma_2, \ldots, \sigma_k$ the eigenvalues of $\mathbf{S}$. The eigenvalues of $\mathbf{L}$ can be characterized as:*

$$1 - \frac{\sigma_i}{\lambda} + \mathcal{O}(\frac{1}{\lambda^2}) \text{ for } i = 1, 2, \ldots, k$$
$$\pm\frac{\sqrt{\nu_{stu}}}{\lambda} + \mathcal{O}(\frac{1}{\lambda^2}) \text{ for } s \neq t \text{ and } u = 1, 2, \ldots, d_s.$$

The proof of Theorem 3 (see Appendix C) is similar to the proof of Theorem 2 of Kamper et al. (2018); however, there are some differences. The differences lie in the asymptotic behavior of the eigenvalues that converge to zero. For sGaBP (univariate version of sGaBP-m), these eigenvalues are characterized by $\pm\frac{s_{ij}}{\lambda} + \mathcal{O}(\frac{1}{\lambda^2})$, for $i, j = 1, 2, \ldots, k$ and $i \neq j$. This means that the convergence rate of sGaBP-m and sGaBP will be identical for large λ, but will likely differ for moderate λ.

The following corollary follows directly from Theorem 3:

Corollary 2 *The spectral radius of $\mathbf{L}$ defined in Theorem 3 can be characterized as*

$$\rho(\mathbf{L}) = 1 - \frac{\sigma_{min}}{\lambda} + \mathcal{O}(\frac{1}{\lambda^2}),$$

where σ_{min} is the smallest eigenvalue of $\mathbf{S}$.

Therefore, we can always find a λ sufficiently large such that the mean components converge, assuming the convergence of the precision components.

3.4. Overall Convergence

In the previous section we proved convergence of the mean components given convergence of the precision components. We now extend this result by dropping the assumption of convergence of the precision components. To this end, we consider the following lemma.

Lemma 3 *Consider the recursion* $\mathbf{a}_{n+1} = \mathbf{b}_n + \mathbf{C}_n\mathbf{a}_n$*, where* $lim_{n\to\infty}\mathbf{b}_n = \mathbf{b}$ *and* $lim_{n\to\infty}\mathbf{C}_n = \mathbf{C}$*, such that* $\rho(\mathbf{C}) < 1$*. We have the following:*

$$\lim_{n\to\infty} \mathbf{a}_n = (\mathbf{I} - \mathbf{C})^{-1}\mathbf{b}.$$

The proof of Lemma 3 is given in Section VI of Moallemi and Van Roy (2009). The following corollary to Lemma 3 shows the convergence of sGaBP-m, given sufficient regularization.

Corollary 3 *There exists a constant* λ_0 *such that sGaBP-m will converge if* $\lambda > \lambda_0$*.*

Proof Consider the following:

1. $\mathbf{L}_n$ is constructed as $\mathbf{L}$ by using $\mathbf{U}_{ij}^{(n)} = -\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}^{(n)}$ and $\ddot{\mathbf{P}}_i^{(n)}$ instead of $\mathbf{U}_{ij}$ and $\ddot{\mathbf{P}}_i$ respectively.
2. $\boldsymbol{\gamma}^{(n+1)}$ remains as before, while $\boldsymbol{\gamma}_0^{(n)}$ is constructed as $\boldsymbol{\gamma}_0$ by using $\mathbf{U}_{ij}^{(n)} = -\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}^{(n)}$ instead of $\mathbf{U}_{ij}$.

We now have the following updating equation:

$$\boldsymbol{\gamma}^{(n+1)} = \boldsymbol{\gamma}_0^{(n)} + \mathbf{L}_n\boldsymbol{\gamma}^{(n)}.$$

The following holds:

1. From Corollary 1 we know that there exists a λ_1 such that $\lim_{n\to\infty}\mathbf{L}_n = \mathbf{L}$ and $\lim_{n\to\infty}\boldsymbol{\gamma}_0^{(n)} = \boldsymbol{\gamma}_0$ when $\lambda > \lambda_1$.
2. By Corollary 2 there is a λ_2 such that $\rho(\mathbf{L}) < 1$ when $\lambda > \lambda_2$.

By taking $\lambda > \lambda_0 = \max(\lambda_1, \lambda_2)$, the conditions of Lemma 3 are satisfied, and hence $\boldsymbol{\gamma}^{(n+1)}$ will converge to $(\mathbf{I} - \mathbf{L})^{-1}\boldsymbol{\gamma}_0$. Note that by selecting $\lambda > \lambda_0$, the precision components will converge, and hence the conditions for Theorem 1 are satisfied. ■

4. Heuristic Regularization

In this section we discuss a heuristic for selecting the degree of regularization. The selection of the degree of regularization through this heuristic is adaptive, i.e. it changes from iteration to iteration.

In order to develop this heuristic, we consider a tree representation of sGaBP-m. The role of this tree representation is similar to the computation trees for GaBP-m, that is we want to represent the computations done by sGaBP-m as inference on a tree-structured

MG. We can interpret this tree representation as unfolding the computations of sGaBP-m.

The tree representation of sGaBP-m, discussed in this section, could be a useful basis for further research into the convergence behavior of sGaBP-m. From this point onward, we use computation tree to refer to the computation tree of sGaBP-m.

4.1. Tree Representation of sGaBP-m

Assume we want to use the computation tree for node i to determine an analytical formula for $\boldsymbol{\mu}_i^{(n)}(\lambda)$, i.e. the posterior mean associated with cluster i after n iterations where the dependence on λ is emphasized. In order to do this, we need to adjust the way in which we assign a precision matrix and potential vector to the computation tree, compared to the method for GaBP-m.

For the precision matrix associated with the computation tree, we assign to a node with reference to cluster j the precision matrix $\lambda\mathbf{I}_{d_j} + \mathbf{S}_{jj}$. The precision matrices between nodes in the computation tree remain as for GaBP-m. The precision matrix assigned to the computation tree for cluster i is $\mathbf{T}_{ii}^{(n)}(\lambda)$ (precision matrix of computation tree constructed from $\mathbf{S}(\lambda)$).

To assign a potential to a node in the computation tree, we require the cluster reference of the node and the layer number in which it occurs. In addition, we require the history of the posterior means, that is $\boldsymbol{\mu}^{(s)}(\lambda)$ (posterior mean at iteration l) for all $s < n-1$. Consider a node in layer l with a reference to cluster j. To this node we assign the potential $\boldsymbol{b}_j + \lambda\boldsymbol{\mu}_j^{(n-l-1)}(\lambda)$. We do this with the understanding that $\boldsymbol{\mu}_j^{(-1)}(\lambda)$ is an initial value for the posterior mean associated with cluster i. In our application of sGaBP-m, we used $\boldsymbol{\mu}_j^{(-1)}(\lambda) = \boldsymbol{b}_j$. We use $\mathbf{t}_i^{(n)}(\lambda)$ to denote the potential of the computation tree for cluster i.

If we marginalize the computation tree with the above precision matrix and potential vector and extract the marginal mean at the root node, we obtain $\boldsymbol{\mu}_i^{(n-1)}(\lambda)$. An illustration of this procedure is given in Appendix D. In the next section, we introduce matrix notation for the tree representation of sGaBP-m.

4.2. Matrix Notation

The different types of matrices considered in Section 3.1 of Kamper et al. (2019) will be used here. For some examples we refer to Appendix D. Consider the row-extractor matrix, $\mathbf{E}_{ii}^{(n)}$, for cluster i and a computation tree depth of n. Let the rows of $\mathbf{E}_{ii}^{(n)}$ be decomposed according to the different layers of the computation tree:

$$\mathbf{E}_{ii}^{(n)} = \begin{bmatrix} \tilde{\mathbf{F}}_{1i} \\ \tilde{\mathbf{F}}_{2i} \\ \vdots \\ \tilde{\mathbf{F}}_{ni} \end{bmatrix}. \tag{19}$$

A node in the computation tree with a reference to cluster t receives a row-extractor matrix $\mathbf{F}_t$. This matrix has the property that $\mathbf{F}_t\mathbf{S}$ are the rows of $\mathbf{S}$ corresponding to the variables in $\mathcal{C}_t$. The matrix $\tilde{\mathbf{F}}_{mi}$ is obtained by row stacking all the row-extractor matrices in layer m of the computation tree.

If $\lambda = 0$, then we would have assigned $\mathbf{E}_{ii}^{(n)}\boldsymbol{b}$ to the potential of the computation tree. In the $\lambda \neq 0$ case, different layers of the computation tree correspond to different posterior means. To allow for this, we define the matrix

$$\mathbf{J}_{ii}^{(n)} = \begin{bmatrix} \tilde{\mathbf{F}}_{1i} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{F}}_{2i} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \tilde{\mathbf{F}}_{ni} \end{bmatrix}$$

and the vector

$$\boldsymbol{\phi}_{n-1}(\lambda) = (\boldsymbol{\mu}^{(n-2)}(\lambda)', \boldsymbol{\mu}^{(n-3)}(\lambda)', \dots, \boldsymbol{\mu}^{(0)}(\lambda)', \boldsymbol{\mu}^{(-1)}(\lambda)')',$$

where $\boldsymbol{\mu}^{(-1)}(\lambda) = \boldsymbol{b}$. The potential assigned to the computation tree becomes $\mathbf{E}_{ii}^{(n)}\boldsymbol{b} + \lambda\mathbf{J}_{ii}^{(n)}\boldsymbol{\phi}_{n-1}(\lambda)$. Consider a node in layer l with a reference to cluster j. We state the following formula (it should be noted that this formula was validated empirically):

$$\boldsymbol{\mu}_i^{(n-1)}(\lambda) = (\mathbf{G}_{ii}^{(n)})'(\mathbf{T}_{ii}^{(n)}(\lambda))^{-1}[\mathbf{E}_{ii}^{(n)}\boldsymbol{b} + \lambda\mathbf{J}_{ii}^{(n)}\boldsymbol{\phi}_{n-1}(\lambda)]. \tag{20}$$

In the next section, we use the formula given in Equation (20) to derive a recursive representation of the posterior means.

4.3. Recursive Representation of the Posterior Means

Let $\mathbf{V}_n(\lambda) : k \times k$ and $\mathbf{W}_n(\lambda) : k \times nk$ be defined as:

$$\mathbf{V}_n(\lambda) = \begin{bmatrix} (\mathbf{G}_{11}^{(n)})'(\mathbf{T}_{11}^{(n)}(\lambda))^{-1}\mathbf{E}_{11}^{(n)} \\ (\mathbf{G}_{22}^{(n)})'(\mathbf{T}_{22}^{(n)}(\lambda))^{-1}\mathbf{E}_{22}^{(n)} \\ \vdots \\ (\mathbf{G}_{pp}^{(n)})'(\mathbf{T}_{pp}^{(n)}(\lambda))^{-1}\mathbf{E}_{pp}^{(n)} \end{bmatrix}$$

$$\mathbf{W}_n(\lambda) = \begin{bmatrix} (\mathbf{G}_{11}^{(n)})'(\mathbf{T}_{11}^{(n)}(\lambda))^{-1}\mathbf{J}_{11}^{(n)} \\ (\mathbf{G}_{22}^{(n)})'(\mathbf{T}_{22}^{(n)}(\lambda))^{-1}\mathbf{J}_{22}^{(n)} \\ \vdots \\ (\mathbf{G}_{pp}^{(n)})'(\mathbf{T}_{pp}^{(n)}(\lambda))^{-1}\mathbf{J}_{pp}^{(n)} \end{bmatrix}.$$

Equation (20) implies

$$\boldsymbol{\mu}^{(n-1)}(\lambda) = \mathbf{V}_n(\lambda)\boldsymbol{b} + \lambda\mathbf{W}_n(\lambda)\boldsymbol{\phi}_{n-1}(\lambda).$$

Define the following matrices,

$$\tilde{\mathbf{V}}_n(\lambda) : (n+1)k \times k = \begin{bmatrix} \mathbf{V}_n(\lambda) \\ \mathbf{0} : nk \times k \end{bmatrix}$$
$$\tilde{\mathbf{W}}_n(\lambda) = \begin{bmatrix} \mathbf{W}_n(\lambda) \\ \mathbf{I}_{kn} \end{bmatrix},$$

which give:

$$\boldsymbol{\phi}_n(\lambda) = \tilde{\mathbf{V}}_n(\lambda)\boldsymbol{b} + \lambda\tilde{\mathbf{W}}_n(\lambda)\boldsymbol{\phi}_{n-1}(\lambda). \tag{21}$$

Equation (21) can be used to obtain a recursive formula for $\boldsymbol{\mu}^{(n)}(\lambda)$ and can, in principle, be used to derive convergence conditions for sGaBP-m. However, this recursive formula is complicated due to the varying nature of $\tilde{\mathbf{V}}_n(\lambda)$ and $\tilde{\mathbf{W}}_n(\lambda)$ (in terms of varying dimensionality and its dependence on n), and we leave the study of this formula for further research. The results of this section can be useful in the derivation of heuristics for the selection of λ, which we discuss in the next section.

4.4. Heuristic Regularization

The computation tree analysis discussed in the previous section can be regarded as a method of unfolding the computations done by sGaBP-m. We now show how the computation trees can be adapted to develop a heuristic for selecting the degree of regularization. Instead of using a single regularization parameter for all blocks in the line topology, we vary λ from layer to layer (we call this a varying computation tree). For the precision matrix associated with the varying computation tree for cluster i, we assign to a node in layer l, with reference to cluster j, the precision matrix $\lambda^{(n-l)}\mathbf{I}_{d_j} + \mathbf{S}_{jj}$. We call this precision matrix $\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)})$, where we only emphasize its dependence on $\lambda^{(n-1)}$ ($\lambda^{(n-l)}$ is assumed to be fixed for $l \geq 2$). The potential associated with a node, with reference to cluster j, is $\boldsymbol{b}_j + \lambda^{(n-l)}\boldsymbol{\mu}_j^{(n-l-1)}$. Note that we assume $\boldsymbol{\mu}_j^{(n-l)}$ to be fixed for $l \geq 2$, and hence no dependence on any regularization is indicated. The precision matrix between nodes remains as in Section 4.1.

We now discuss a heuristic aimed at varying the regularization between layers such that convergence is achieved at a faster rate. Let $\mathcal{D}_{ni}(\lambda^{(n-1)})$ denote a diagonal matrix in which the diagonal entries corresponding to layer l of the varying computation tree are $\lambda^{(n-l)}$. Similar to Equation (20), we see that:

$$\boldsymbol{\mu}_i^{(n-1)}(\lambda^{(n-1)}) = (\mathbf{G}_{ii}^{(n)})'(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}[\mathbf{E}_{ii}^{(n)}\boldsymbol{b} + \mathcal{D}_{ni}(\lambda^{(n-1)})\mathbf{J}_{ii}^{(n)}\boldsymbol{\phi}_{n-1}], \tag{22}$$

where $\boldsymbol{\phi}_{n-1}$ contains all the posterior means until stage $n-2$. Note that Equation (22) was validated empirically. Since we have completed the updates until stage $n-2$, the entries of $\boldsymbol{\phi}_{n-1}$ will be fixed. Furthermore,

$$\begin{aligned}
\frac{\partial \mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)})}{\partial \lambda^{(n-1)}} &= \frac{\partial \mathcal{D}_{ni}(\lambda^{(n-1)})}{\partial \lambda^{(n-1)}} = \mathbf{G}_{ii}^{(n)}(\mathbf{G}_{ii}^{(n)})' \\
\frac{\partial (\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}}{\partial \lambda^{(n-1)}} &= -(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}\frac{\partial \mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)})}{\partial \lambda^{(n-1)}}(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1} \\
&= -(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}\mathbf{G}_{ii}^{(n)}(\mathbf{G}_{ii}^{(n)})'(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}.
\end{aligned}$$

Differentiating (22) with respect to $\lambda^{(n-1)}$, we obtain

$$\begin{aligned}\frac{\partial \boldsymbol{\mu}_i^{(n-1)}(\lambda^{(n-1)})}{\partial \lambda^{(n-1)}} &= -(\mathbf{G}_{ii}^{(n)})'(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}\mathbf{G}_{ii}^{(n)}(\mathbf{G}_{ii}^{(n)})'(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}\mathbf{E}_{ii}^{(n)}\boldsymbol{b} \\ &+ (\mathbf{G}_{ii}^{(n)})'(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}\mathbf{G}_{ii}^{(n)}(\mathbf{G}_{ii}^{(n)})'\mathbf{J}_{ii}^{(n)}\boldsymbol{\phi}_{n-1} \\ &- (\mathbf{G}_{ii}^{(n)})'(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}\mathbf{G}_{ii}^{(n)}(\mathbf{G}_{ii}^{(n)})'(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}\mathcal{D}_{ni}(\lambda^{(n-1)})\mathbf{J}_{ii}^{(n)}\boldsymbol{\phi}_{n-1} \\ &= (\mathbf{G}_{ii}^{(n)})'(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}\mathbf{G}_{ii}^{(n)}\Big[(\mathbf{G}_{ii}^{(n)})'\mathbf{J}_{ii}^{(n)}\boldsymbol{\phi}_{n-1} \\ &- (\mathbf{G}_{ii}^{(n)})'(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}(\mathbf{E}_{ii}^{(n)}\boldsymbol{b} + \mathcal{D}_{ni}(\lambda^{(n-1)})\mathbf{J}_{ii}^{(n)}\boldsymbol{\phi}_{n-1})\Big].\end{aligned}$$

Since $(\mathbf{G}_{ii}^{(n)})'\mathbf{J}_{ni}\boldsymbol{\phi}_{n-1} = \boldsymbol{\mu}_i^{(n-2)}$ (which is fixed), we have

$$\begin{aligned}\frac{\partial \boldsymbol{\mu}_i^{(n-1)}(\lambda^{(n-1)})}{\partial \lambda^{(n-1)}} &= (\mathbf{G}_{ii}^{(n)})'(\mathbf{T}_{ii}^{(n)}(\lambda^{(n-1)}))^{-1}\mathbf{G}_{ii}^{(n)}\Big[\boldsymbol{\mu}_i^{(n-2)} - \boldsymbol{\mu}_i^{(n-1)}(\lambda^{(n-1)})\Big] \\ &= (\mathbf{P}_i^{(n-1)}(\lambda^{(n-1)}))^{-1}\Big[\boldsymbol{\mu}_i^{(n-2)} - \boldsymbol{\mu}_i^{(n-1)}(\lambda^{(n-1)})\Big], \qquad (23)\end{aligned}$$

where $\mathbf{P}_i^{(n-1)}(\lambda^{(n-1)})$ indicates the posterior precision at iteration $n-1$, and this only depends on $\lambda^{(n-1)}$. We can obtain the derivative of $\boldsymbol{\mu}^{(n)}(\lambda^{(n)})$ with respect to $\lambda^{(n)}$ by applying Equation (23) (where $n \leftarrow n+1$) to all nodes i. Set $j = \text{argmax}_i\{|\mathbf{s}_i'\boldsymbol{\mu}^{(n)}(\lambda^{(n)}) - \boldsymbol{b}|\}$, where $\mathbf{s}_i'$ is the ith row of $\mathbf{S}$, and let $\text{div}(\lambda^{(n)}) = \mathbf{s}_j'\nabla\boldsymbol{\mu}^{(n)}(\lambda^{(n)})$, where $\nabla\boldsymbol{\mu}^{(n)}(\lambda^{(n)})$ is the gradient of $\boldsymbol{\mu}^{(n)}(\lambda^{(n)})$ with respect to $\lambda^{(n)}$. Consider λ_0 as a candidate for $\lambda^{(n)}$. To evaluate $\text{div}(\lambda_0)$, we need to perform the message updates using λ_0 as the value for the regularization parameter. After the message updates, we see that using $\lambda_0 - \alpha\text{sign}(\text{div}(\lambda_0))$ instead of λ_0 would have been better (for sufficiently small α) in the sense that it would have given posterior means that are closer to solving the linear system, $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$. If we assume that $\lambda^{(n)}$ was decided upon at iteration $n-1$, we can make the retrospective adjustment $\lambda^{(n+1)} = \lambda^{(n)} - \alpha\text{sign}(\text{div}(\lambda^{(n)}))$. We test this heuristic measure in the empirical section by varying α over different values.

We note that this approach can also be used to derive further heuristics, by changing the way in which we vary the regularization. For instance, we could set $\lambda^{(n-1)} = \lambda^{(n-2)} = \lambda$ and differentiate the computation tree for a new heuristic. We leave this for further research. Appendix D contains some examples of the considerations of this section.

5. Empirical Results

In this section we present three empirical studies of the sGaBP-m algorithm. In the first, we compare sGaBP-m to the multivariate extensions of RGaBP and convergence fix GaBP (CF-GaBP) by considering both convergence speed and inference quality. We describe RGaBP and CFGaBP for nodes of any size in Algorithms 2 and 3 respectively. In the literature, these algorithms are formulated for univariate nodes, but they can easily be extended to the multivariate case. The second study is dedicated to testing the heuristic described in

Algorithm 2 Synchronous RGaBP

1. Specify a tolerance ϵ, a maximum number of iterations m and a relaxation parameter τ.
2. Initialize $\mathbf{Q}_{ij}^{(0)} = \mathbf{0} : d_j \times d_j$, $\mathbf{v}_{ij}^{(0)} = \mathbf{0} : d_j \times 1$, $\boldsymbol{\mu}^{(-1)} = \boldsymbol{b}$ for all i and all $j \in \mathcal{N}_i$.
3. Set Err = Inf and $n = 0$.
4. While Err $> \epsilon$
   (a) Compute $\mathbf{P}_i^{(n)}(0) = \mathbf{S}_{ii} + \sum_{j \in \mathcal{N}_i} \mathbf{Q}_{ji}^{(n)}$ and $\mathbf{z}_i^{(n)} = \boldsymbol{b}_i + \sum_{j \in \mathcal{N}_i} \mathbf{v}_{ji}^{(n)}$ for $i = 1, 2, \ldots, p$.
   (b) Update $\mathbf{z}_i^{(n)} \leftarrow \tau \mathbf{z}_i^{(n)} + (1-\tau)\mathbf{P}_i^{(n)}(0)\boldsymbol{\mu}_i^{(n-1)}$, set $\boldsymbol{\mu}_i^{(n)} = [\mathbf{P}_i^{(n)}(0)]^{-1}\mathbf{z}_i^{(n)}$, $\boldsymbol{e}_i^{(n)} = \sum_j \mathbf{S}_{ij}\boldsymbol{\mu}_j^{(n)} - \boldsymbol{b}_i$ and Err $= \max_i\{||\boldsymbol{e}_i^{(n)}||_\infty\}$.
   (c) If Err $> \epsilon$, do for all $i \in \{1, 2, \ldots, p\}$ and all $j \in \mathcal{N}_i$: $\mathbf{Q}_{ij}^{(n+1)} = -\mathbf{S}_{ji}[\mathbf{P}_i^{(n)}(0) - \mathbf{Q}_{ji}^{(n)}]^{-1}\mathbf{S}_{ij}$ and $\mathbf{v}_{ij}^{(n+1)} = -\mathbf{S}_{ji}[\mathbf{P}_i^{(n)}(0) - \mathbf{Q}_{ji}^{(n)}]^{-1}[\mathbf{z}_i^{(n)} - \mathbf{v}_{ji}^{(n)}]$.
   (d) Increment n.
   (e) If $n = m$, break.
5. End.

the previous section, while the third involves a performance comparison between sGaBP and sGaBP-m.

5.1. Comparison of sGaBP-m with Other Methods

We simulated data using the following procedure:

1. Select a $\tilde{\rho}$ uniformly from the interval $[1; 1.3]$.
2. Using the method from Kamper et al. (2018), we generate a 100×100 precision matrix $\mathbf{S}$ with zero-diagonal spectral radius equal to $\tilde{\rho}$, along with a 100×1 potential vector $\boldsymbol{b}$. The zero-diagonal spectral radius of $\mathbf{S}$ is defined as the spectral radius of $\mathbf{I}_k - \mathbf{S}$ after $\mathbf{S}$ has been scaled to have all diagonal entries equal to one.
3. The 100 variables are assigned randomly to 10 clusters each of size 10.
4. For each of sGaBP-m, RGaBP and CFGaBP, we determine the hyperparameters yielding convergence in the minimum number of iterations using a line search with increments of 0.01. These parameters are then used to initialize the methods.

This process was repeated 1 000 times. For each simulation, we record the number of iterations required for convergence and the posterior precisions for each cluster supplied by each method. For sGaBP-m and RGaBP, the precision estimates are computed as

$$\hat{\mathbf{P}}_i = \mathbf{S}_{ii} + \sum_{t \in \mathcal{N}_i} \mathbf{Q}_{ti}.$$

Algorithm 3 Synchronous CFGaBP

1. Specify a tolerance ϵ, a maximum number of iterations m and a diagonal loading λ.
2. Initialize $\boldsymbol{\mu}_{\text{work}} = \mathbf{0}$.
3. Set Err = Inf.
4. While Err $> \epsilon$
   (a) Compute $\mathbf{h} = \boldsymbol{b} - \mathbf{S}\boldsymbol{\mu}_{\text{work}}$.
   (b) Apply ordinary GaBP-m using the precision matrix $\mathbf{S} + \lambda\mathbf{I}_k$ and the potential vector $\mathbf{h}$. This can be done by setting $\lambda = 0$ or $\tau = 1$ in Algorithm 1 or Algorithm 2 respectively.
   (c) Let $\boldsymbol{\xi}$ be the posterior means supplied in Step (4b). Set $\boldsymbol{\mu}_{\text{work}} \leftarrow \boldsymbol{\mu}_{\text{work}} + \boldsymbol{\xi}$ and let Err $= ||\mathbf{S}\boldsymbol{\mu}_{\text{work}} - \boldsymbol{b}||_\infty$.
   (d) Increment n by the number of iterations performed by GaBP-m in Step (4b).
   (e) If $n \geq m$, break.
5. End.

Because we are supplying $\mathbf{S} + \lambda\mathbf{I}_k$ to the inner loop of CFGaBP, we propose computing the precision estimate of CFGaBP for cluster i as

$$\hat{\mathbf{P}}_i = \mathbf{S}_{ii} - \lambda\mathbf{I}_{d_i} + \sum_{t \in \mathcal{N}_i} \mathbf{Q}_{ti}.$$

Note that, if we use the same λ for sGaBP-m and CFGaBP, then the precision estimates will be the same. They are likely to differ in the simulations, since the λ yielding the convergence in the smallest number of iterations will differ between the methods. We note the following practical considerations for the CFGaBP algorithm:

1. The converged posterior precisions are the posterior precisions obtained from the first inner-loop application of GaBP-m in Algorithm 3.
2. This is because, in the later stages of the outer-loop of Algorithm 3, $\mathbf{h} = \boldsymbol{b} - \mathbf{S}\boldsymbol{\mu}_{\text{work}} \approx \mathbf{0}$, and this could cause the inner-loop application of GaBP-m to terminate before the convergence of the precision components.

To compare inference quality, we consider the Kullback-Leibler (KL) divergence of the posterior marginal of a cluster from its corresponding exact marginal. Suppose that $f_i(\mathbf{y})$ and $\hat{f}_i(\mathbf{y})$ are the exact and posterior marginals associated with cluster i. We calculate the KL divergence of the posterior marginal from the exact marginal as:

$$D_{KL}(f_i||\hat{f}_i) = \int_{\mathbf{y} \in \Re^{d_i}} f_i(\mathbf{y}) \log\left(\frac{f_i(\mathbf{y})}{\hat{f}_i(\mathbf{y})}\right) d\mathbf{y}.$$

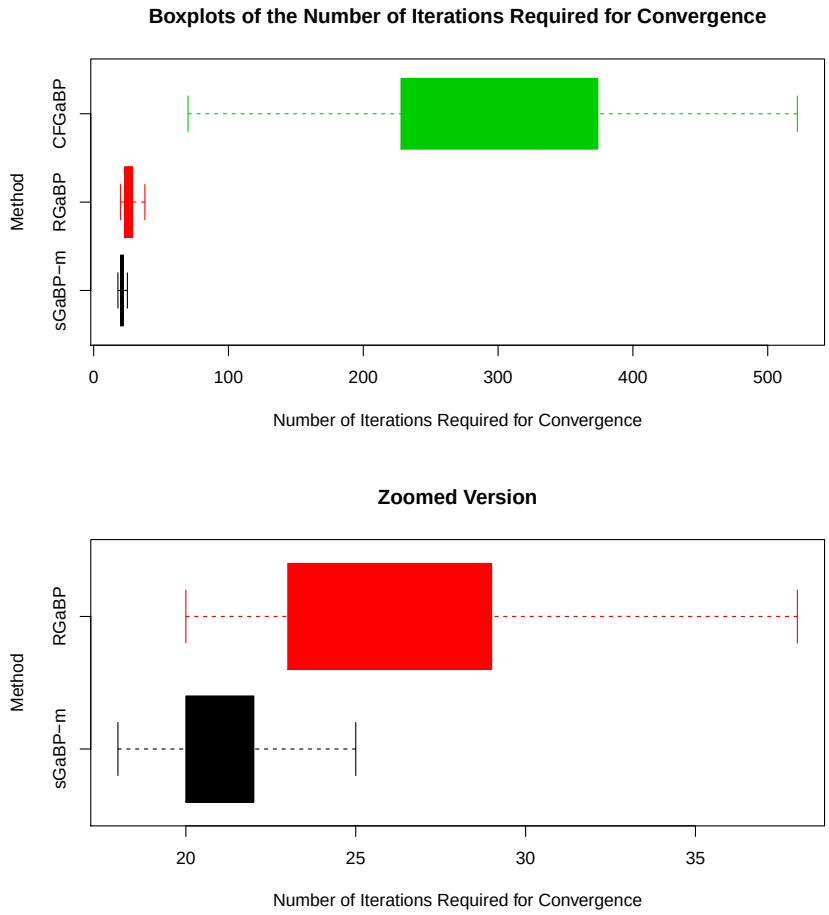

Figure 1: Visualization of the results of our simulations comparing the number of iterations required for convergence by sGaBP-m, RGaBP and CFGaBP. The bottom panel zooms in on the boxplots corresponding to sGaBP-m and RGaBP. CFGaBP required the greatest number of iterations to converge. The number of iterations required for convergence by sGaBP-m and RGaBP are more comparable, with sGaBP-m tending to require a smaller number of iterations to converge.

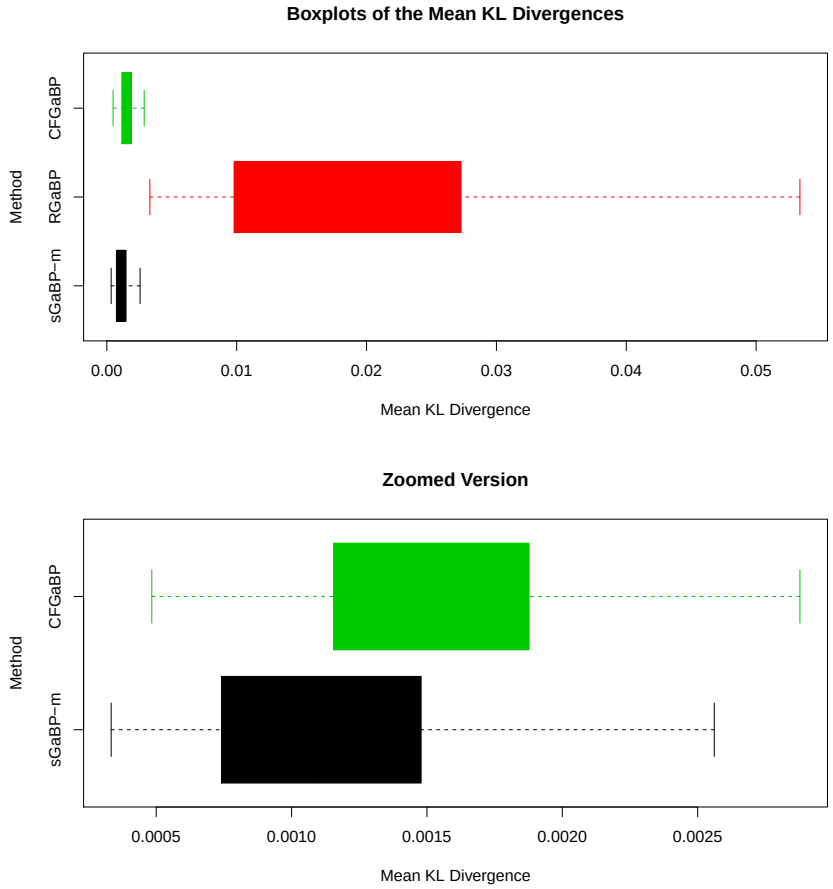

Figure 2: Visualization of the results of our simulations comparing the inference quality of sGaBP-m, RGaBP and CFGaBP. The bottom panel zooms in on the boxplots corresponding to sGaBP-m and CFGaBP. The inference quality of RGaBP is poor compared to that of the other methods. This is because RGaBP computes precision estimates in the same manner as ordinary GaBP-m. Clearly, sGaBP-m performed the best of the methods in terms of inference quality.

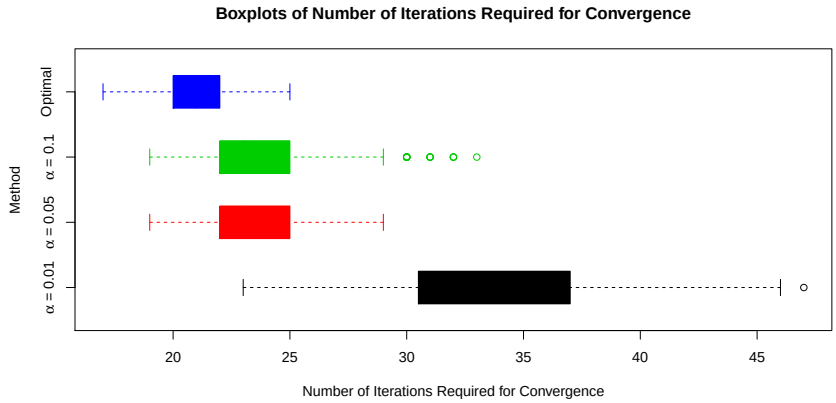

Figure 3: Visualization of the results of our simulations of the convergence speed of the heuristic method, with different initializations, compared to optimal regularization. We see that the heuristics with $\alpha = 0.05$ or $\alpha = 0.1$ compare well with the optimal regularization, although they tend to provide slower convergence. The $\alpha = 0.01$ heuristic does not compare well with the other regularizations, indicating that the adjustments are done too slowly.

Because all the methods considered yield the exact marginal means at convergence, the KL divergence of the exact marginal of a cluster to its corresponding posterior distribution will only be influenced by the precisions of the respective distributions. For a specific simulation, each method is represented by the mean of all the KL divergences of the exact marginals to their corresponding posterior distributions.

The results for the convergence speed (as measured by the number of iterations required for convergence) and inference quality (as measured by the mean KL divergence) are summarized in Figures 1 and 2 respectively. The convergence speed of CFGaBP is slow compared to that of the other methods. This is caused by the double-loop implementation in Algorithm 3, Step (b). sGaBP-m tends to converge faster than RGaBP. In terms of inference quality, the performance of RGaBP is poor compared to that of the other methods. The best inference quality is provided by sGaBP-m. Clearly, sGaBP-m outperformed the competitors in our simulations.

It is possible to measure inference quality by computing the KL divergence of the exact marginal from the posterior marginal (this is the opposite direction used in the simulations). We note that changing the order of the KL divergence does not affect the conclusions drawn from Figure 2.

As a final comment, we note that a comparison of sGaBP-m with GaBP-m was made implicitly. This is because the approximate precisions of RGaBP is equivalent to the approximate precisions of GaBP-m. Hence, sGaBP-m can provide superior inference quality

compared to GaBP-m. Furthermore, GaBP-m is equivalent to sGaBP-m with $\lambda = 0$, and since all simulations involved selected a $\lambda > 0$ (recall that λ was chosen to yield the fastest convergence), we see that sGaBP-m can accelerate the convergence of GaBP-m.

5.2. Performance of Heuristic Regularization

In this section we investigate how well the heuristic regularization approaches optimal regularization. For this purpose, we use the data from the previous section and compare optimal sGaBP-m with different initializations of the heuristic. For each application of the heuristic, we start with $\lambda = 0$ and consider using $\alpha = 0.01, 0.05$ and 0.1. The different methods are compared in terms of the number of iterations required for convergence. The results are given in Figure 3. We see that the heuristics with $\alpha = 0.05$ and $\alpha = 0.1$ compare well with the optimal method (sGaBP-m initialized to have fastest convergence), but they tend to converge at a slower speed. The heuristic with $\alpha = 0.01$ does not compare well with the other methods. We see that the heuristic makes some progress in shifting the regularization towards the optimal level, but it is sensitive to the selection of α. This simulation study shows that our heuristic can play a role in the selection of the regularization parameter, given appropriate initialization.

5.3. Performance Comparison of sGaBP-m and sGaBP

In this empirical study, we compare sGaBP-m to sGaBP in terms of univariate inference quality. We use $p = 100$ in our simulations and clusters $\mathcal{C}_i = \{10(i-1)+1, \ldots, 10i\}$ for $i = 1, 2, \ldots, 10$. The following simulation procedure was used:

1. Generate a precision matrix $\mathbf{S}$ with zero-diagonal spectral radius equal to 0.8 and a potential vector $\boldsymbol{b}$ using the method from Kamper et al. (2018).
2. Select a ρ uniformly from $[1.2; 1.3]$.
3. For $i = 1, 2, \ldots, 10$, generate a 10×10 precision matrix with zero-diagonal spectral radius equal to ρ and override $\mathbf{S}_{ii}$ with this precision matrix.

This simulation procedure is an example of where the partial correlations are stronger within clusters than between clusters. The above procedure was applied 1 000 times. For each of these simulated examples, sGaBP and sGaBP-m were given 50 iterations to provide approximate univariate marginals. For sGaBP-m, this implies first approximating the higher-dimensional marginals, and then applying a direct method to these in order to obtain approximate univariate marginals. The inference quality of a method applied to a simulated example was measured by calculating the mean KL divergence of the posterior marginal from the exact marginal. The results are illustrated in Figure 4.

We see that the univariate inference quality of sGaBP-m is far superior to that provided by sGaBP (about 32 times more accurate on average). The reason for this superior inference quality is as follows:

1. sGaBP-m tends to provide more accurate univariate precision approximations.
2. The posterior means of sGaBP-m tend to converge faster in terms of iteration count.

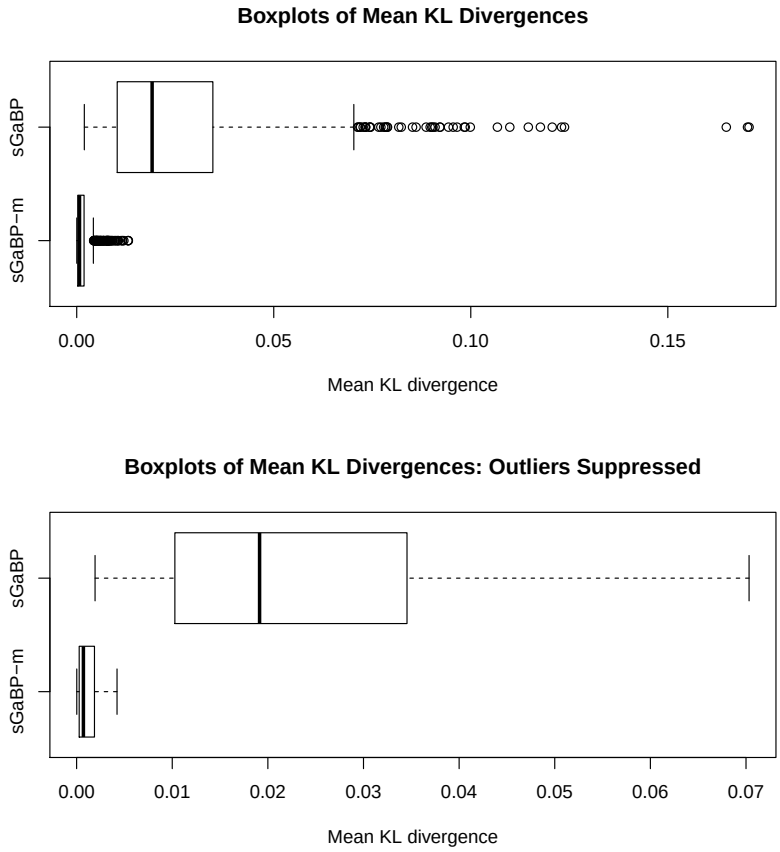

Figure 4: Visualizations of the results of the performance comparison between sGaBP and sGaBP-m. We see that the inference quality of sGaBP-m is superior to that provided by sGaBP. This is due to more accurate approximate precisions and faster convergence of the posterior means in terms of iteration count.

Although the main motivation for using sGaBP-m over sGaBP is that it can be used to approximate higher-dimensional marginals, we also see that it can be useful for univariate marginal approximation.

6. Conclusion and Further Research

This paper was concerned with the application of node regularization to a higher-dimensional extension of a pairwise MG. This extension allows for the approximation of multivariate marginals through the use of BP. The main result was a proof of the convergence of sGaBP-m given sufficient regularization. The proof is based on asymptotic expressions for the precision components of sGaBP-m, which were derived through the use of computation trees. Under the assumption of convergence of the precision components, we then showed that the updates are linear and that the linear-update matrix depends on the precision components

of sGaBP-m. We proved that the spectral radius of the linear-update matrix approaches one from below as $\lambda \to \infty$. We completed the proof by showing that the above conditions are sufficient for the overall convergence of sGaBP-m, given sufficient regularization. A proof that sGaBP-m provides the exact marginal means at convergence was also given. The selection of the level of regularization was addressed through the use of a heuristic. The heuristic was derived from a novel computation tree-type representation of sGaBP-m and based on a gradient-descent principle. The performance of sGaBP-m was considered empirically. The empirical study showed that sGaBP-m compares favorably to certain competitors, both in terms of inference quality and convergence speed. The advantages of using the heuristic were also illustrated empirically.

The computation tree representation of sGaBP-m could be useful in deriving sufficient conditions for the convergence of this algorithm (in terms of λ) and in establishing theoretical guarantees for the heuristic measure. The accuracy of the posterior precisions as approximations for the true marginal precisions in terms of λ needs to be analyzed. In this paper, we did not consider the possibility of using multiple regularization parameters. If we can find an effective way of selecting these parameters, it would most likely result in improved inference quality and faster convergence. We also believe that node regularization should be applied to other types of Gaussian message-passing. All these considerations are left for further research.

Acknowledgments

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the authors and are not necessarily to be attributed to the NRF.

Appendix A. Proof of Theorem 1

Proof We assume, without loss of generality, that we are dealing with a fully connected MG. We write $\lim_{n\to\infty} \ddot{\mathbf{P}}_{ij}^{(n)}(\lambda) = \ddot{\mathbf{P}}_{ij}$ and $\lim_{n\to\infty} \ddot{\mathbf{P}}_{i}^{(n)}(\lambda) = \ddot{\mathbf{P}}_{i}$ (we are assuming convergence of sGaBP-m). Convergence implies the following conditions:

$$\begin{aligned}
\mathbf{Q}_{ij} &= -\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}\mathbf{S}_{ji} \\
\mathbf{v}_{ij} &= -\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}[\lambda\boldsymbol{\mu}_i + \mathbf{z}_i - \mathbf{v}_{ji}] \\
\boldsymbol{\mu}_i &= \ddot{\mathbf{P}}_{i}[\lambda\boldsymbol{\mu}_i + \mathbf{z}_i],
\end{aligned}$$

where $\ddot{\mathbf{P}}_{ij}^{-1} = \lambda\mathbf{I}_{d_i} + \mathbf{S}_{ii} + \sum_{t\neq i,j}\mathbf{Q}_{ti}$, $\ddot{\mathbf{P}}_{i}^{-1} = \ddot{\mathbf{P}}_{ij}^{-1} + \mathbf{Q}_{ji}$ and $\mathbf{z}_i = \boldsymbol{b}_i + \sum_{t\neq i}\mathbf{v}_{ti}$. We now show that these equations imply that $\sum_i \mathbf{S}_{ji}\boldsymbol{\mu}_i = \boldsymbol{b}_j$ for all j.

Note the following for all $i \neq j$:

$$\begin{aligned}
\ddot{\mathbf{P}}_{ji} &= [\ddot{\mathbf{P}}_{j}^{-1} - \mathbf{Q}_{ij}]^{-1} \\
&= [\ddot{\mathbf{P}}_{j}^{-1} + \mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}\mathbf{S}_{ji}]^{-1} \\
&= \ddot{\mathbf{P}}_{j} - \ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}[\ddot{\mathbf{P}}_{ij}^{-1} + \mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}]^{-1}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j},
\end{aligned} \tag{24}$$

and

$$\begin{aligned}
\ddot{\mathbf{P}}_{i} &= [\ddot{\mathbf{P}}_{ij}^{-1} + \mathbf{Q}_{ji}]^{-1} \\
&= [\ddot{\mathbf{P}}_{ij}^{-1} - \mathbf{S}_{ij}\ddot{\mathbf{P}}_{ji}\mathbf{S}_{ji}]^{-1} \\
&= \ddot{\mathbf{P}}_{ij} + \ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}[\ddot{\mathbf{P}}_{ji}^{-1} - \mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}]^{-1}\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij} \\
&= \ddot{\mathbf{P}}_{ij} + \ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}[\ddot{\mathbf{P}}_{ji}^{-1} + \mathbf{Q}_{ij}]^{-1}\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij} \\
&= \ddot{\mathbf{P}}_{ij} + \ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}.
\end{aligned} \tag{25}$$

Consider

$$\begin{aligned}
\mathbf{S}_{ji}\ddot{\mathbf{P}}_{i}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{ji} &= \mathbf{S}_{ji}[\ddot{\mathbf{P}}_{ij} + \ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}]\mathbf{S}_{ij}\ddot{\mathbf{P}}_{ji} \text{ (see Equation (25))} \\
&= \mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{ji} + \mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{ji} \\
&= \mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}[\ddot{\mathbf{P}}_{j} - \ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}[\ddot{\mathbf{P}}_{ij}^{-1} + \mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}]^{-1}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}] \\
&+ \mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}[\ddot{\mathbf{P}}_{j} - \ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}[\ddot{\mathbf{P}}_{ij}^{-1} + \mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}]^{-1}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}].
\end{aligned}$$

Setting $\mathbf{O}_{ij} = [\ddot{\mathbf{P}}_{ij}^{-1} + \mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}]^{-1}$, we obtain (after some simplification)

$$\begin{aligned}
\mathbf{S}_{ji}\ddot{\mathbf{P}}_{i}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{ji} &= -\mathbf{Q}_{ij}\ddot{\mathbf{P}}_{j} \\
&+ \mathbf{Q}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}[\mathbf{I}_{d_i} - \ddot{\mathbf{P}}_{ij}\mathbf{O}_{ij}^{-1} + \ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}]\mathbf{O}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}.
\end{aligned}$$

Consider

$$\begin{aligned}
\mathbf{I}_{d_i} - \ddot{\mathbf{P}}_{ij}\mathbf{O}_{ij}^{-1} + \ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji} &= \mathbf{I}_{d_i} - \ddot{\mathbf{P}}_{ij}[\ddot{\mathbf{P}}_{ij}^{-1} + \mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji}] + \ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji} \\
&= \mathbf{I}_{d_i} - \mathbf{I}_{d_i} - \ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji} + \ddot{\mathbf{P}}_{ij}\mathbf{S}_{ij}\ddot{\mathbf{P}}_{j}\mathbf{S}_{ji} \\
&= \mathbf{0} : d_i \times d_i,
\end{aligned}$$

and hence, for all $i \neq j$, we have $\mathbf{S}_{ji}\ddot{\mathbf{P}}_i\mathbf{S}_{ij}\ddot{\mathbf{P}}_{ji} = -\mathbf{Q}_{ij}\ddot{\mathbf{P}}_j$.

Setting $\ddot{\mathbf{Q}}_{ij} = -\mathbf{S}_{ji}\ddot{\mathbf{P}}_i\mathbf{S}_{ij}$, we see that $\ddot{\mathbf{Q}}_{ij}\ddot{\mathbf{P}}_{ji} = \mathbf{Q}_{ij}\ddot{\mathbf{P}}_j$ and $\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij} = \mathbf{S}_{ji}\ddot{\mathbf{P}}_i + \ddot{\mathbf{Q}}_{ij}[\ddot{\mathbf{P}}_{ji}^{-1} - \ddot{\mathbf{Q}}_{ij}]^{-1}\mathbf{S}_{ji}\ddot{\mathbf{P}}_i$ by Equation (24). Furthermore,

$$\begin{aligned}
\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij} &= [\mathbf{I}_{d_j} + \ddot{\mathbf{Q}}_{ij}[\ddot{\mathbf{P}}_{ji}^{-1} - \ddot{\mathbf{Q}}_{ij}]^{-1}]\mathbf{S}_{ji}\ddot{\mathbf{P}}_i \\
&= [\ddot{\mathbf{P}}_{ji}^{-1} - \ddot{\mathbf{Q}}_{ij} + \ddot{\mathbf{Q}}_{ij}][\ddot{\mathbf{P}}_{ji}^{-1} - \ddot{\mathbf{Q}}_{ij}]^{-1}\mathbf{S}_{ji}\ddot{\mathbf{P}}_i \\
&= \ddot{\mathbf{P}}_{ji}^{-1}[\ddot{\mathbf{P}}_{ji}^{-1} - \ddot{\mathbf{Q}}_{ij}]^{-1}\mathbf{S}_{ji}\ddot{\mathbf{P}}_i.
\end{aligned}$$

Consider

$$\begin{aligned}
\mathbf{v}_{ij} &= -\mathbf{S}_{ji}\ddot{\mathbf{P}}_{ij}[\lambda\boldsymbol{\mu}_i + \mathbf{z}_i - \mathbf{v}_{ji}] \\
&= -\ddot{\mathbf{P}}_{ji}^{-1}[\ddot{\mathbf{P}}_{ji}^{-1} - \ddot{\mathbf{Q}}_{ij}]^{-1}\mathbf{S}_{ji}\ddot{\mathbf{P}}_i[\lambda\boldsymbol{\mu}_i + \mathbf{z}_i - \mathbf{v}_{ji}],
\end{aligned}$$

which implies that

$$-[\ddot{\mathbf{P}}_{ji}^{-1} - \ddot{\mathbf{Q}}_{ij}]\ddot{\mathbf{P}}_{ji}\mathbf{v}_{ij} = \mathbf{S}_{ji}\boldsymbol{\mu}_i - \mathbf{S}_{ji}\ddot{\mathbf{P}}_i\mathbf{v}_{ji}, \tag{26}$$

since $\boldsymbol{\mu}_i = \ddot{\mathbf{P}}_i[\lambda\boldsymbol{\mu}_i + \mathbf{z}_i]$. From Equation (26), we see that

$$\mathbf{S}_{ji}\boldsymbol{\mu}_i = -\mathbf{v}_{ij} + \ddot{\mathbf{Q}}_{ij}\ddot{\mathbf{P}}_{ji}\mathbf{v}_{ij} + \mathbf{S}_{ji}\ddot{\mathbf{P}}_i\mathbf{v}_{ji}.$$

Since $\mathbf{S}_{ji}\ddot{\mathbf{P}}_i\mathbf{v}_{ji} = -\mathbf{S}_{ji}\ddot{\mathbf{P}}_i\mathbf{S}_{ij}\ddot{\mathbf{P}}_{ji}[\lambda\boldsymbol{\mu}_j + \mathbf{z}_j - \mathbf{v}_{ji}] = \ddot{\mathbf{Q}}_{ij}\ddot{\mathbf{P}}_{ji}[\lambda\boldsymbol{\mu}_j + \mathbf{z}_j - \mathbf{v}_{ji}] = \ddot{\mathbf{Q}}_{ij}\ddot{\mathbf{P}}_{ji}[\lambda\boldsymbol{\mu}_j + \mathbf{z}_j] - \ddot{\mathbf{Q}}_{ij}\ddot{\mathbf{P}}_{ji}\mathbf{v}_{ij} = \mathbf{Q}_{ij}\ddot{\mathbf{P}}_j[\lambda\boldsymbol{\mu}_j + \mathbf{z}_j] - \ddot{\mathbf{Q}}_{ij}\ddot{\mathbf{P}}_{ji}\mathbf{v}_{ij} = \mathbf{Q}_{ij}\boldsymbol{\mu}_j - \ddot{\mathbf{Q}}_{ij}\ddot{\mathbf{P}}_{ji}\mathbf{v}_{ij}$, we have

$$\mathbf{S}_{ji}\boldsymbol{\mu}_i = -\mathbf{v}_{ij} + \mathbf{Q}_{ij}\boldsymbol{\mu}_j \tag{27}$$

for all $i \neq j$. From Equation (27),

$$\begin{aligned}
\sum_i \mathbf{S}_{ji}\boldsymbol{\mu}_i &= \mathbf{S}_{jj}\boldsymbol{\mu}_j - \sum_{i\neq j}\mathbf{v}_{ij} + \left[\sum_{i\neq j}\mathbf{Q}_{ij}\right]\boldsymbol{\mu}_j \\
&= \mathbf{S}_{jj}\boldsymbol{\mu}_j + \boldsymbol{b}_j - \mathbf{z}_j + \left[\ddot{\mathbf{P}}_j^{-1} - \lambda\mathbf{I}_{d_j} - \mathbf{S}_{jj}\right]\boldsymbol{\mu}_j \\
&= \boldsymbol{b}_j - \mathbf{z}_j + \ddot{\mathbf{P}}_j^{-1}\boldsymbol{\mu}_j - \lambda\boldsymbol{\mu}_j \\
&= \boldsymbol{b}_j - \mathbf{z}_j + \lambda\boldsymbol{\mu}_j + \mathbf{z}_j - \lambda\boldsymbol{\mu}_j \\
&= \boldsymbol{b}_j
\end{aligned}$$

for all j. In particular, we see that the means implied by the stationary conditions satisfy $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$ and are therefore the correct marginal means. ■

Appendix B. Proof of Lemma 4

Lemma 4 *Consider a sequence of matrices $\mathbf{A}_n(\lambda)$ with the following properties:*

1. *There exists a constant λ_0 such that $lim_{n\to\infty}\mathbf{A}_n(\lambda) = \mathbf{A}(\lambda)$ for all $\lambda > \lambda_0$.*
2. *$||\mathbf{A}_n(\lambda)||_\infty \leq g(\lambda) = \mathcal{O}(\frac{1}{\lambda^3})$.*

These properties imply that $\mathbf{A}(\lambda) = \mathcal{O}(\frac{1}{\lambda^2})$.

Proof Consider $A_{ij}^{(n)}(\lambda)$ and $\lim_{n\to\infty} A_{ij}^{(n)}(\lambda) = A_{ij}(\lambda)$ for any i, j. Since $||\mathbf{A}_n(\lambda)||_\infty \leq g(\lambda) = \mathcal{O}(\frac{1}{\lambda^3})$, we also have:

$$|A_{ij}^{(n)}(\lambda)| \leq g(\lambda). \tag{28}$$

Pre-multiplying (28) by λ^2 and taking the limit as $n \to \infty$, we see that:

$$|\lambda^2 A_{ij}(\lambda)| \leq \lambda^2 g(\lambda).$$

Since $\lim_{\lambda\to\infty} \lambda^2 g(\lambda) = 0$, we see from the squeeze theorem,

$$\lim_{\lambda\to\infty} \lambda^2 A_{ij}(\lambda) = 0. \tag{29}$$

Equation (29) implies that $A_{ij}(\lambda) = \mathcal{O}(\frac{1}{\lambda^2})$. ■

Appendix C. Proofs Leading to Theorem 3

Lemma 5 *There exists a constant $K > 0$ such that, for sufficiently small δ, each eigenvalue x of $\tilde{\mathbf{L}}$ either satisfies $|x| \leq K\delta$ or $|x - 1| \leq K\delta$.*

Proof We reason by contradiction and assume that there is an eigenvalue for which $|x| > K\delta$ and $|x - 1| > K\delta$. Consider $||\tilde{\mathbf{L}}_{11}||_\infty = ||\mathbf{L}_{11}||_\infty = \delta||\mathbf{M}_{11}||_\infty + \mathcal{O}(\delta^2)$. If we choose K large enough (e.g. $K \geq 1 + ||\mathbf{M}_{11}||_\infty$), then $||\mathbf{L}_{11}||_\infty < K\delta + \mathcal{O}(\delta^2)$ and $||\mathbf{L}_{11}||_\infty < |x|$ for sufficiently small δ. Therefore, x is not an eigenvalue of $\mathbf{L}_{11}$ and $x\mathbf{I}_{m_1} - \mathbf{L}_{11}$ will be invertible. We can now apply the Schur complement on $\tilde{\mathbf{L}}$ to obtain:

$$\begin{aligned}\det(x\mathbf{I}_{m_2} - \tilde{\mathbf{L}}) = \det(x\mathbf{I}_{m_1} - \mathbf{L}_{11}) \\ \times \det(x\mathbf{I}_k - \mathbf{L}_{22} - \frac{1}{p-2}\mathbf{L}_{22}\mathbf{H}'(x\mathbf{I}_{m_1} - \mathbf{L}_{11})^{-1}\mathbf{L}_{11}\mathbf{H}).\end{aligned} \tag{30}$$

It remains to show that the second determinant is not equal to zero:

$$\begin{aligned}&x\mathbf{I}_k - \mathbf{L}_{22} - \frac{1}{p-2}\mathbf{L}_{22}\mathbf{H}'(x\mathbf{I}_{m_1} - \mathbf{L}_{11})^{-1}\mathbf{L}_{11}\mathbf{H} \\ &= (x-1)\mathbf{I}_k - (\mathbf{L}_{22} - \mathbf{I}_k) - \frac{1}{x(p-2)}\mathbf{L}_{22}\mathbf{H}'(\mathbf{I}_{m_1} - \frac{1}{x}\mathbf{L}_{11})^{-1}\mathbf{L}_{11}\mathbf{H} \\ &= (x-1)\mathbf{I}_k + \mathbf{H}_1 + \frac{1}{x}\mathbf{H}_2,\end{aligned} \tag{31}$$

where $\mathbf{H}_1 = -(\mathbf{L}_{22} - \mathbf{I}_k)$ and $\mathbf{H}_2 = -\frac{1}{(p-2)}\mathbf{L}_{22}\mathbf{H}'(\mathbf{I}_{m_1} - \frac{1}{x}\mathbf{L}_{11})^{-1}\mathbf{L}_{11}\mathbf{H}$. Consider

$$\begin{aligned}
||\mathbf{H}_1||_\infty &= ||\mathbf{L}_{22} - \mathbf{I}_k||_\infty \\
&= ||\delta\mathbf{M}_{22} + \mathcal{O}(\delta^2)||_\infty \\
&= \delta||\mathbf{M}_{22}||_\infty + \mathcal{O}(\delta^2) \\
&\leq \delta\kappa_1,
\end{aligned}$$

where δ is sufficiently small and, e.g., $\kappa_1 \geq ||\mathbf{M}_{22}||_\infty + 1$. Furthermore,

$$\begin{aligned}
(\mathbf{I} - \frac{1}{x}\mathbf{L}_{11})^{-1}\mathbf{L}_{11} &= (\mathbf{I} - \frac{\delta}{x}\mathbf{M}_{11} + \mathcal{O}(\delta^2))^{-1}(\delta\mathbf{M}_{11} + \mathcal{O}(\delta^2)) \\
&= \delta(\mathbf{I} - \frac{\delta}{x}\mathbf{M}_{11})^{-1}\mathbf{M}_{11} + \mathcal{O}(\delta^2) \\
&= \delta(\mathbf{I} + \mathcal{O}(\delta))\mathbf{M}_{11} + \mathcal{O}(\delta^2) \\
&= \delta\mathbf{M}_{11} + \mathcal{O}(\delta^2).
\end{aligned}$$

Since $\mathbf{L}_{22} = \mathbf{I}_k + \mathcal{O}(\delta)$, we see that

$$-\mathbf{H}_2 = \frac{\delta}{p-2}\mathbf{H}'\mathbf{M}_{11}\mathbf{H} + \mathcal{O}(\delta^2),$$

and $||\mathbf{H}_2||_\infty \leq \kappa_2\delta$ for sufficiently small δ and, e.g., $\kappa_2 \geq \frac{1}{p-2}||\mathbf{H}'\mathbf{M}_{11}\mathbf{H}||_\infty + 1$. Suppose that $|x| \geq 0.5$, then

$$||\mathbf{H}_1 + \frac{1}{x}\mathbf{H}_2||_\infty \leq \delta\kappa_1 + \delta\frac{\kappa_2}{|x|} \leq \delta(\kappa_1 + 2\kappa_2) < K\delta < |x - 1|$$

for δ sufficiently small and $K > \kappa_1 + 2\kappa_2$. If $|x| < 0.5$,

$$||\mathbf{H}_1 + \frac{1}{x}\mathbf{H}_2||_\infty \leq \delta\kappa_1 + \delta\frac{\kappa_2}{|x|} \leq \delta\kappa_1 + \frac{\delta\kappa_2}{\delta K} = \delta\kappa_1 + \frac{\kappa_2}{K} < 0.5 \leq |x - 1|$$

for δ sufficiently small and K sufficiently large (e.g. $K > 2\kappa_2$). We have that, for sufficiently small δ and sufficiently large K, $x - 1$ will not be an eigenvalue of $\mathbf{H}_1 + \mathbf{H}_2$, and the second determinant in Equation (30) will not be zero. Therefore, x cannot be an eigenvalue of $\tilde{\mathbf{L}}$ and, by contradiction, the statement is proved. ■

C.1. Proof of Theorem 3

Proof We first consider the eigenvalues that are close to one. Set $x = 1 - \delta t$ for some t where $|t| < K$. For sufficiently small δ, x will not be an eigenvalue of $\mathbf{L}_{11}$, and therefore $x\mathbf{I}_{m_1} - \mathbf{L}_{11}$ will be invertible. Application of the Schur complement on $\tilde{\mathbf{L}}$ yields

$$\det(\tilde{\mathbf{L}}) = \det(x\mathbf{I}_{m_1} - \mathbf{L}_{11})\det(-\delta t + \mathbf{H}_1 + \frac{1}{x}\mathbf{H}_2) \tag{32}$$

(see Equation (31)). Since $\mathbf{H}_1 = \delta\mathbf{M}_{22} + \mathcal{O}(\delta^2)$ and $\mathbf{H}_2 = -\frac{\delta}{p-2}\mathbf{H}'\mathbf{M}_{11}\mathbf{H} + \mathcal{O}(\delta^2)$, we see that the second determinant of Equation (32) becomes

$$\begin{aligned}
&\det(\delta(\mathbf{M}_{22} - t\mathbf{I}_k) - \frac{\delta}{p-2}\mathbf{H}'\mathbf{M}_{11}\mathbf{H} + \mathcal{O}(\delta^2)) \\
&= \det((\mathbf{M}_{22} - t\mathbf{I}_k) - \frac{1}{p-2}\mathbf{H}'\mathbf{M}_{11}\mathbf{H} + \mathcal{O}(\delta)) \\
&= \det(-t\mathbf{I}_k + \mathbf{M}_{22} - \frac{1}{p-2}\mathbf{H}'\mathbf{M}_{11}\mathbf{H} + \mathbf{M}) \\
&= \det(t\mathbf{I}_k - \left[\mathbf{M}_{22} - \frac{1}{p-2}\mathbf{H}'\mathbf{M}_{11}\mathbf{H}\right] + \mathbf{M}), \qquad (33)
\end{aligned}$$

where $\mathbf{M} = \mathcal{O}(\delta)$. As $\delta \to 0$ we obtain the characteristic equation of $\left[\mathbf{M}_{22} - \frac{1}{p-2}\mathbf{H}'\mathbf{M}_{11}\mathbf{H}\right]$. Its k solutions (counted with multiplicity) give rise to k branches $t_1(\delta), t_2(\delta), \ldots, t_k(\delta)$ that solve the implicit equation in Equation (33).

Considering the eigenvalues that are close to zero, we set $x = \delta t$ for some t with $|t| < K$. For δ sufficiently small, x will not be an eigenvalue of $\mathbf{L}_{22}$, and hence $x\mathbf{I}_k - \mathbf{L}_{22}$ will be invertible. A second application of the Schur complement (with respect to the diagonal block $\mathbf{L}_{22}$) gives

$$\det(\tilde{\mathbf{L}}) = \det(x\mathbf{I}_k - \mathbf{L}_{22})\det(x\mathbf{I}_{m_1} - \mathbf{L}_{11} - \frac{1}{p-2}\mathbf{L}_{11}\mathbf{H}(x\mathbf{I}_k - \mathbf{L}_{22})^{-1}\mathbf{L}_{22}\mathbf{H}'). \qquad (34)$$

First note that $x\mathbf{I}_{m_1} - \mathbf{L}_{11} = \delta t\mathbf{I}_{m_1} - \delta\mathbf{M}_{11} + \mathcal{O}(\delta^2)$ and $(x\mathbf{I}_k - \mathbf{L}_{22})^{-1} = -\mathbf{I}_k + \mathcal{O}(\delta)$. Hence $\mathbf{L}_{11}\mathbf{H}(x\mathbf{I}_k - \mathbf{L}_{22})^{-1} = \delta\mathbf{M}_{11}\mathbf{H} + \mathcal{O}(\delta^2)$ and

$$\frac{1}{p-2}\mathbf{L}_{11}\mathbf{H}(x\mathbf{I}_k - \mathbf{L}_{22})^{-1}\mathbf{L}_{22}\mathbf{H}' = -\frac{\delta}{p-2}\mathbf{M}_{11}\mathbf{H}\mathbf{H}' + \mathcal{O}(\delta^2).$$

The second determinant of Equation (34) becomes

$$\det(t\mathbf{I}_{m_1} - \mathbf{M}_{11}[\mathbf{I}_{m_1} - \frac{1}{p-2}\mathbf{H}\mathbf{H}'] + \tilde{\mathbf{M}}), \qquad (35)$$

where $\tilde{\mathbf{M}} = \mathcal{O}(\delta)$. As $\delta \to 0$, Equation (35) converges to the characteristic function of $\mathbf{M}_{11}[\mathbf{I}_{m_1} - \frac{1}{p-2}\mathbf{H}\mathbf{H}']$. This gives rise to $k^2 - k$ branches $\bar{t}_i(\delta) : i = 1, 2, \ldots, k^2 - k$, which solve the implicit equation in Equation (35).

For our example, the matrix $\left[\mathbf{M}_{22} - \frac{1}{p-2}\mathbf{H}'\mathbf{M}_{11}\mathbf{H}\right]$ can easily be seen to be equal to $\mathbf{S}$, and this holds for the general case as well. The matrix $\mathbf{W} = \mathbf{M}_{11}[\mathbf{I}_{m_1} - \frac{1}{p-2}\mathbf{H}\mathbf{H}']$ has a more complicated construction. For our example, we see that $\mathbf{W}$ is symmetric and that $\mathbf{W}'\mathbf{W}$ is a block-diagonal matrix with a diagonal block for each of $\mathbf{S}_{ij}\mathbf{S}_{ji}$ where $i \neq j$. Again, this holds in general.

Consider Equation (33) set equal to zero,

$$\det(t\mathbf{I}_k - \mathbf{S} + \mathbf{M}) = 0, \qquad (36)$$

where $\mathbf{M} = \mathcal{O}(\delta)$. Let $\mathbf{V}$ be the matrix of eigenvectors that diagonalizes $\mathbf{S}$. Pre- and post-multiplying Equation (36) by $\mathbf{V}$ and $\mathbf{V}'$ respectively, we obtain

$$\det(t\mathbf{I}_k - \mathbf{\Lambda} + \mathbf{V}\mathbf{M}\mathbf{V}') = 0,$$

where $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{S}$ on its diagonal. Since $\mathbf{M} = \mathcal{O}(\delta)$, we have that $\mathbf{V}\mathbf{M}\mathbf{V}' = \mathcal{O}(\delta)$. This guarantees the existence of a constant κ_3 such that $||\mathbf{V}\mathbf{M}\mathbf{V}'||_\infty \leq \kappa_3\delta$ for sufficiently small δ. In order for the matrix in the determinant of Equation (36) to be singular, we must have that $|t - \Lambda_{ii}| \leq \kappa_3\delta$, otherwise this matrix will be strictly diagonally dominant. The diagonal entries of $\mathbf{\Lambda}$ are the eigenvalues $\sigma_i : i = 1, 2, \ldots, k$ of $\mathbf{S}$, and therefore $t = \sigma_i + \mathcal{O}(\delta)$.

The smaller eigenvalues can be dealt with in a similar way where we need to find the eigenvalues of $\mathbf{W}$. Since $\mathbf{W}'\mathbf{W}$ is a block-diagonal matrix and symmetric, we can find the squared values of the eigenvalues of $\mathbf{W}$ (recall that $\mathbf{W}$ is symmetric) by computing the eigenvalues of $\mathbf{S}_{ij}\mathbf{S}_{ji}$ for all $i \neq j$. Let $\nu_{iju} \geq 0 : u = 1, 2, \ldots, d_i$ $(i \neq j)$ be the eigenvalues of $\mathbf{S}_{ij}\mathbf{S}_{ji}$. We have the following asymptotic expressions for the eigenvalues:

$$\begin{aligned} &1 - \sigma_i\delta + \mathcal{O}(\delta^2) : i = 1, 2, \ldots, k. \\ &\pm\sqrt{\nu_{iju}}\delta + \mathcal{O}(\delta^2) : i \neq j \text{ and } u = 1, 2, \ldots, d_i. \end{aligned}$$

Clearly, the eigenvalue with the largest absolute value approaches one from below as $\delta \to 0$. For sufficiently large λ (small δ), the spectral radius of the linear update matrix will be less than one, and convergence will occur with this level of regularization. ■

Appendix D. Computation Tree Examples

Let us consider an example for the loopy MG given in the top panel of Figure 5. The associated precision matrix and potential vector are:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{0} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{S}_{23} & \mathbf{S}_{24} \\ \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{S}_{33} & \mathbf{S}_{34} \\ \mathbf{0} & \mathbf{S}_{42} & \mathbf{S}_{43} & \mathbf{S}_{44} \end{bmatrix}$$

and

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \end{bmatrix},$$

respectively. The computation tree for cluster 4 (with a depth of $n = 4$) is shown in the bottom panel of Figure 5. From this point onwards we will consider the matrices and vectors used in Section 4.4 using the computation tree of depth 3 for cluster 4 (the matrices corresponding to a depth of 4 are too large). This computation tree is obtained by eliminating the final layer of the tree on the bottom panel of Figure 5.

The matrix $\mathbf{T}_{ii}^{(n)}(\lambda)$ can be obtained by moving along the computation tree (first by layer and then from left to right) and assigning to a node, with reference to cluster t, the matrix $\lambda\mathbf{I}_{d_t} + \mathbf{S}_{tt}$. If two nodes in the computation tree are linked, we need to determine the references of both nodes (say s and t), and they are linked by the matrix $\mathbf{S}_{ts}$. As an example,

$$\mathbf{T}_{44}^{(3)}(\lambda) = \begin{bmatrix} \mathbf{S}_{44} + \lambda\mathbf{I}_{d_4} & \mathbf{S}_{42} & \mathbf{S}_{43} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{24} & \mathbf{S}_{22} + \lambda\mathbf{I}_{d_2} & \mathbf{0} & \mathbf{S}_{21} & \mathbf{S}_{23} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{34} & \mathbf{0} & \mathbf{S}_{33} + \lambda\mathbf{I}_{d_3} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{31} & \mathbf{S}_{32} \\ \mathbf{0} & \mathbf{S}_{12} & \mathbf{0} & \mathbf{S}_{11} + \lambda\mathbf{I}_{d_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{32} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{33} + \lambda\mathbf{I}_{d_3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{13} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{11} + \lambda\mathbf{I}_{d_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{23} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{22} + \lambda\mathbf{I}_{d_2} \end{bmatrix}.$$

The matrix $\mathbf{T}_{ji}^{(n)}(\lambda)$, $j \neq i$ can be obtained in a similar fashion. However, we only consider the subtree rooted at the node in the second layer, corresponding to cluster j of the original graph. For example,

$$\mathbf{T}_{34}^{(3)}(\lambda) = \begin{bmatrix} \mathbf{S}_{33} + \lambda\mathbf{I}_{d_3} & \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{13} & \mathbf{S}_{11} + \lambda\mathbf{I}_{d_1} & \mathbf{0} & \mathbf{S}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{23} & \mathbf{0} & \mathbf{S}_{22} + \lambda\mathbf{I}_{d_2} & \mathbf{0} & \mathbf{S}_{21} & \mathbf{S}_{24} \\ \mathbf{0} & \mathbf{S}_{21} & \mathbf{0} & \mathbf{S}_{22} + \lambda\mathbf{I}_{d_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{12} & \mathbf{0} & \mathbf{S}_{11} + \lambda\mathbf{I}_{d_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{42} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{44} + \lambda\mathbf{I}_{d_4} \end{bmatrix}.$$

Let us consider constructing a potential vector $\mathbf{t}_4^{(3)}(\lambda)$ to use alongside $\mathbf{T}_{44}^{(3)}(\lambda)$ for the computation of $\boldsymbol{\mu}_4^{(2)}(\lambda)$. Suppose we have already obtained $\boldsymbol{\mu}^{(0)}(\lambda)$ and $\boldsymbol{\mu}^{(1)}(\lambda)$. The potential is:

$$\mathbf{t}_4^{(3)}(\lambda) = \begin{bmatrix} \boldsymbol{b}_4 + \lambda\boldsymbol{\mu}_4^{(1)}(\lambda) \\ \boldsymbol{b}_2 + \lambda\boldsymbol{\mu}_2^{(0)}(\lambda) \\ \boldsymbol{b}_3 + \lambda\boldsymbol{\mu}_3^{(0)}(\lambda) \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}.$$

The row-extractor matrix for a node in the computation tree with reference to cluster j is defined as:

$$\mathbf{F}_j = \begin{bmatrix} \mathbf{0} : d_j \times d_1 & \mathbf{0} : d_j \times d_2 & \dots & \mathbf{0} : d_j \times d_{j-1} & \mathbf{I}_{d_j} & \mathbf{0} : d_j \times d_{j+1} & \dots & \mathbf{0} : d_j \times d_k \end{bmatrix}.$$

The row-extractor matrix for cluster i ($\mathbf{E}_{ii}^{(n)}$) is obtained by moving along its computation tree and stacking the row-extractor matrices of the nodes row-wise. The matrix $\tilde{\mathbf{F}}_{li}$, used in Equation (19), is obtained by moving to layer l of the computation tree associated with cluster i and then stacking the row-extractor matrices of its nodes row-wise. We now give

examples of the matrices in Section 4.2 using the MG given in Figure 5:

$$[\mathbf{G}_{44}^{(3)}]' = \begin{bmatrix} \mathbf{I}_{d_4} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{E}_{44}^{(3)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_4} \\ \mathbf{0} & \mathbf{I}_{d_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_3} & \mathbf{0} \\ \mathbf{I}_{d_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_3} & \mathbf{0} \\ \mathbf{I}_{d_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{d_2} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{J}_{44}^{(3)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_4} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_3} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_2} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

From these matrices we see that $\mathbf{t}_4^{(3)}(\lambda) = \mathbf{E}_{44}^{(3)}\boldsymbol{b} + \lambda \mathbf{J}_{44}^{(3)}\boldsymbol{\phi}_2(\lambda)$ where

$$\boldsymbol{\phi}_2(\lambda) = (\boldsymbol{\mu}^{(1)}(\lambda)', \boldsymbol{\mu}^{(0)}(\lambda)', \boldsymbol{b}')'.$$

We can obtain $\boldsymbol{\mu}_4^{(2)}(\lambda)$ by computing $[\mathbf{G}_{44}^{(3)}]'[\mathbf{T}_{44}^{(3)}(\lambda)]^{-1}\mathbf{t}_4^{(3)}(\lambda)$, which is what is given by Equation (20).

Let us consider an example for the change of Equation (20) to Equation (22) for the heuristic measure. For our example we have $n = 3$ and $i = 4$. The following changes are made:

$$\mathbf{T}_{44}^{(3)}(\lambda^{(2)}) = \begin{bmatrix} \mathbf{S}_{44} + \lambda^{(2)}\mathbf{I}_{d_4} & \mathbf{S}_{42} & \mathbf{S}_{43} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{24} & \mathbf{S}_{22} + \lambda^{(1)}\mathbf{I}_{d_2} & \mathbf{0} & \mathbf{S}_{21} & \mathbf{S}_{23} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{34} & \mathbf{0} & \mathbf{S}_{33} + \lambda^{(1)}\mathbf{I}_{d_3} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{31} & \mathbf{S}_{32} \\ \mathbf{0} & \mathbf{S}_{12} & \mathbf{0} & \mathbf{S}_{11} + \lambda^{(0)}\mathbf{I}_{d_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{32} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{33} + \lambda^{(0)}\mathbf{I}_{d_3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{13} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{11} + \lambda^{(0)}\mathbf{I}_{d_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{23} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{22} + \lambda^{(0)}\mathbf{I}_{d_2} \end{bmatrix}.$$

$$\mathcal{D}_{34}(\lambda^{(2)}) = \begin{bmatrix} \lambda^{(2)}\mathbf{I}_{d_4} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \lambda^{(1)}\mathbf{I}_{d_2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda^{(1)}\mathbf{I}_{d_3} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda^{(0)}\mathbf{I}_{d_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda^{(0)}\mathbf{I}_{d_3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda^{(0)}\mathbf{I}_{d_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda^{(0)}\mathbf{I}_{d_2} \end{bmatrix}.$$

Here, $\lambda^{(2)}$, $\lambda^{(1)}$ and $\lambda^{(0)}$ denote the regularization used for layers 1, 2 and 3 of the computation tree in Figure 5. The regularization parameter $\lambda^{(0)}$ is an initial parameter, and we used $\lambda^{(0)} = 0$ in our simulations. For the purpose of our heuristic, we are interested in adjusting $\lambda^{(2)}$ to a level where the posterior means at iteration 2 are closer to solving the system of linear equations, $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$.

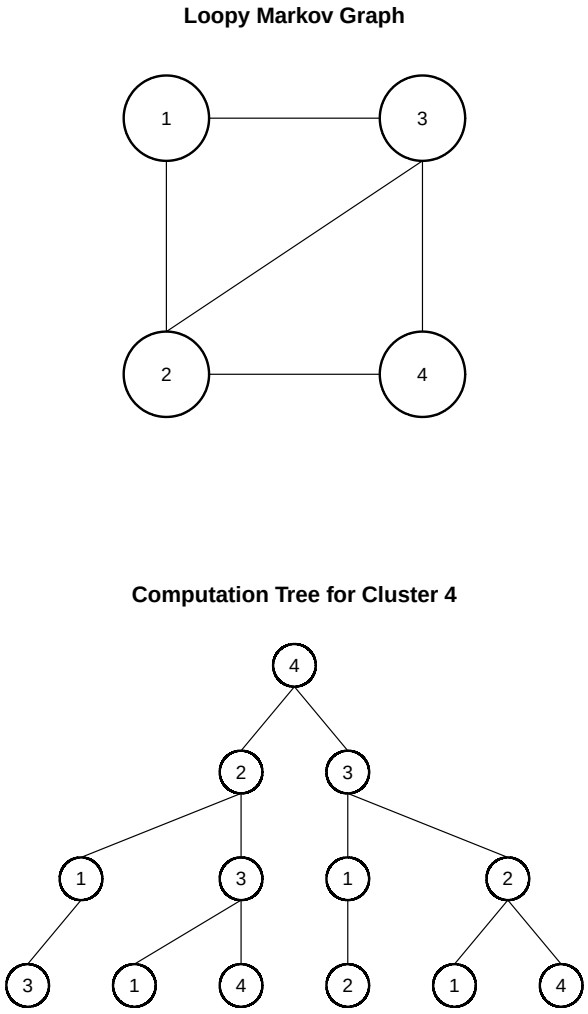

Figure 5: Loopy Markov graph and the computation tree for cluster 4 with $n = 3$ iterations.

References

Srinivas M. Aji and Robert J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46:325–343, March 2000.

Danny Bickson. *Gaussian Belief Propagation: Theory and Application*. PhD thesis, The Hebrew University of Jerusalem, October 2008.

Venkat Chandrasekaran, Jason K. Johnson, and Alan S. Willsky. Estimation in Gaussian graphical models using tractable subgraphs: a walk-sum analysis. *IEEE Transactions on Signal Processing*, 56:1916–1930, 2008.

Yousef El-Kurdi, Dennis Giannacopoulos, and Warren J. Gross. Relaxed Gaussian belief propagation. In *Proceedings IEEE International Symposium on Information Theory*, September 2012a.

Yousef El-Kurdi, Warren J. Gross, and Dennis Giannacopoulos. Efficient implementation of Gaussian belief propagation solver for large sparse diagonally dominant linear systems. *IEEE Transactions on Magnetics*, 48:471–474, February 2012b.

Brendan J. Frey and Frank R. Kschischang. Probability propagation and iterative decoding. In *Proceedings 34th Annual Allerton Conference on Communication, Control, and Computing*, Allerton House, Monticello, Illinois, October 1996.

Robert G. Gallager. *Low-Density Parity-Check Codes*. MA: MIT Press, Cambridge, 1963.

Qinghua Guo and Defeng Huang. EM-based joint channel estimation and detection for frequency selective channels using Gaussian message passing. *IEEE Transactions on Signal Processing*, 59:4030–4035, 2011.

Qinghua Guo and Li Ping. LMMSE turbo equalization based on factor graphs. *IEEE Journal on Selected Areas in Communications*, 26:311–319, 2008.

Jason K. Johnson, Danny Bickson, and Danny Dolev. Fixing convergence of Gaussian belief propagation. In *Proceedings IEEE International Symposium on Information Theory*, Seoul, South Korea, 2009.

Francois Kamper. An empirical study of Gaussian belief propagation and application in the detection of F-formations. In *Proceedings of the ACM Multimedia 2017 Workshop on South African Academic Participation*, October 2017.

Francois Kamper, Johan A. du Preez, Sarel J. Steel, and Stephan Wagner. Regularized Gaussian belief propagation. *Statistics and Computing*, 28(3):653–672, May 2018.

Francois Kamper, Sarel J. Steel, and Johan A. du Preez. On the convergence of Gaussian belief propagation with nodes of arbitrary size. *Journal of Machine Learning Research*, 20(165):1–37, 2019.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, MA, 2009.

Bin Li and Yik-Chung Wu. Convergence analysis of Gaussian belief propagation under high-order factorization and asynchronous scheduling. *IEEE Transactions on Signal Processing*, 67(11):2884–2897, June 2019a.

Bin Li and Yik-Chung Wu. Convergence of Gaussian belief propagation under general pairwise factorization: connecting Gaussian MRF with pairwise linear Gaussian model. *Journal of Machine Learning Research*, 20(144):1–30, 2019b.

Ying Liu. Feedback message passing for inference in Gaussian graphical models. Master's thesis, Massachusetts Institute of Technology, June 2010.

Ying Liu, Venkat Chandrasekaran, Animashree Anandkumar, and Alan S. Willsky. Feedback message passing for inference in Gaussian graphical models. *IEEE Transactions on Signal Processing*, 60(8):4135–4150, 2012.

Dmitry M. Malioutov, Jason K. Johnson, and Alan S. Willsky. Walk-sums and belief propagation in Gaussian graphical models. *Journal of Machine Learning Research*, 7: 2031–2064, October 2006.

Ciamac C. Moallemi and Benjamin Van Roy. Convergence of min-sum message passing for quadratic optimization. *IEEE Transactions on Information Theory*, 55(5):2413–2423, May 2009.

Andrea Montanari, Balaji Prabhakar, and David Tse. Belief propagation based multi-user detection. In *Proceedings IEEE Information Theory Workshop*, Punta del Este, Uruguay, March 2006.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann, San Francisco, CA, USA, 1988.

Nicholas Ruozzi and Sekhar Tatikonda. Message-passing algorithms for quadratic minimization. *Journal of Machine Learning Research*, 14:2287–2314, 2013.

Matthias W. Seeger and David P. Wipf. Variational Bayesian inference techniques. *IEEE Signal Processing Magazine*, 27:81–91, November 2010.

Ori Shental, Paul H. Siegel, Jack K. Wolf, Danny Bickson, and Danny Dolev. Gaussian belief propagation solver for systems of linear equations. In *Proceedings IEEE International Symposium on Information Theory*, pages 1863–1867, 2008.

Qinliang Su and Yik-Chung Wu. On convergence conditions of Gaussian belief propagation. *IEEE International Transactions on Signal Processing*, 63:1144–1155, March 2015.

Tianju Sui, Damian E. Marelli, and Minyue Fu. Convergence analysis of Gaussian belief propagation for distributed state estimation. In *Proceedings IEEE Annual Conference on Decision and Control*, Osaka, Japan, Dec. 2015.

Yair Weiss and William T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.