Stat Comput (2018) 28:653–672
https://doi.org/10.1007/s11222-017-9753-7

# Regularized Gaussian belief propagation

**Francois Kamper**[1] · **Johan A. du Preez**[2] · **Sarel J. Steel**[1] · **Stephan Wagner**[3]

Received: 17 January 2017 / Accepted: 3 May 2017 / Published online: 12 May 2017


**Abstract** Belief propagation (BP) has been applied in a variety of inference problems as an approximation tool. BP does not necessarily converge in loopy graphs, and even if it does, is not guaranteed to provide exact inference. Even so, BP is useful in many applications due to its computational tractability. In this article, we investigate a regularized BP scheme by focusing on loopy Markov graphs (MGs) induced by a multivariate Gaussian distribution in canonical form. There is a rich literature surrounding BP on Gaussian MGs (labelled Gaussian belief propagation or GaBP), and this is known to experience the same problems as general BP on graphs. GaBP is known to provide the correct marginal means if it converges (this is not guaranteed), but it does not provide the exact marginal precisions. We show that our adjusted BP will always converge, with sufficient tuning, while maintaining the exact marginal means. As a further contribution we show, in an empirical study, that our GaBP variant can accelerate GaBP and compares well with other GaBP-type competitors in terms of convergence speed and accuracy of approximate marginal precisions. These improvements suggest that the principle of regularized BP should be investigated in other inference problems. The selection of the degree of regularization is addressed through the use of two heuristics. A by-product of GaBP is that it can be used to solve linear systems of equations; the same is true for our variant and we make an empirical comparison with the conjugate gradient method.

**Keywords** Belief propagation · Approximate inference · Gaussian distributions · Regularization · Convergence

✉ Francois Kamper
15339017@sun.ac.za

Johan A. du Preez
jadupreez@gmail.com

Sarel J. Steel
sjst@sun.ac.za

Stephan Wagner
swagner@sun.ac.za

[1] Department of Statistics and Actuarial Science, University of Stellenbosch, Stellenbosch, South Africa

[2] Department of Electrical and Electronic Engineering, University of Stellenbosch, Stellenbosch, South Africa

[3] Department of Mathematical Sciences, University of Stellenbosch, Stellenbosch, South Africa

## 1 Introduction

Belief propagation (BP) is a message-passing algorithm used to marginalize distributions to variables contained in nodes of a graph. BP operates by sending messages between nodes which are linked in a graph and these messages are updated iteratively. At initialization each node receives a set of random variables and an associated distribution function (often called the potential of the node). At any stage of BP we can instruct nodes to collect all incoming messages, these messages are processed and used by nodes to update their potentials. From this point onwards, we refer to these updated potentials as posterior distributions. A message from a node $i$ to a neighbour $j$ is updated by node $i$ collecting all incoming messages, excluding the message from node $j$, computing the posterior distribution and then processing this posterior distribution as a message to node $j$. The way posterior distributions are converted to messages depends on the type of the graph and the goal of the propagation procedure (usually marginalization or determining the mode).

When applied to loopy graphs BP may not converge and if it does, it does not necessarily converge to the correct marginals. Even though BP does not necessarily supply true marginal distributions, it is still useful as an approximate inference tool due to its computational tractability. In this article, we propose a regularized message-passing scheme on general Markov graphs (MGs). The goal here is to improve on the performance of message passing in loopy graphs. The performance of a message-passing scheme in the BP context can be measured by whether or not it converges, the rate at which it converges and the accuracy of the converged posterior distributions as approximations to the true marginals. We illustrate how regularized message passing can be used to address all three of these issues by implementing the scheme on a Gaussian MG under canonical parametrization. This type of belief propagation is often referred to as Gaussian belief propagation (GaBP). For GaBP involving synchronous message passing, each iteration can be performed in at most $\mathcal{O}(p^2)$ computations, assuming $p$ variables distributed among $p$ nodes. The number of computations can be substantially lower than $\mathcal{O}(p^2)$ depending on the degree of sparsity in the precision matrix and further acceleration can be obtained through distributive computing.

A by-product of this inference algorithm is the implicit solving of linear systems, $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$, where $\mathbf{S} : p \times p$ is the precision matrix and $\boldsymbol{b}$ the potential vector of a multivariate Gaussian in canonical form. In the literature (Bickson 2008), GaBP has been compared favourably to the Jacobi and Gauss-Seidel methods as a solver of large and sparse linear systems, but faces tough competition from other methods such as the conjugate gradient (CG) and preconditioned conjugate gradient (PCG) algorithms. GaBP does not converge for all positive definite precision matrices and, in the case of convergence, the posterior precisions are not necessarily equal to the marginal precisions (Malioutov et al. 2006). However, if GaBP converges, the posterior means are equal to the marginal means (Weiss and Freeman 2001). We label the application of our regularized message passing on a Gaussian MG in canonical form slow Gaussian belief propagation (sGaBP). We show that sGaBP will converge given sufficient regularization and provide posterior means equal to the marginal means. This article includes empirical comparisons with other GaBP variants as well as with the CG solver. The results indicate that sGaBP converges faster than variants of GaBP and provides more accurate approximations to the true marginals. Our simulations show that sGaBP can be comparable to CG (which provides no precision estimates); we make some suggestions on how to further accelerate sGaBP and comment on the use of preconditioning for both methods. In our concluding remarks, we discuss regularized message passing in a broader class of optimization and marginalization problems (beyond the Gaussian context).

## 2 Literature review

In the context of error-correcting codes, the roots of BP can be traced back to the development of the sum-product algorithm as a decoding algorithm for LDPC codes (Gallager 1963). Belief propagation (Probability propagation) for Bayesian networks was introduced by Pearl (1988), Shachter (1988), Shafer and Shenoy (1990), Lauritzen and Spiegelhalter (1988) and later found to be equivalent to the sum-product algorithm (Aji and McEliece 2000; Frey and Kschischang 1996). BP is known to provide exact inference on tree-structured graphs but may fail to converge or may converge to incorrect marginals in the case of loop graphs (Pearl 1988; Weiss 2000). However, BP can still be a useful tool as an approximate inference algorithm on loopy structures (Weiss 2000). Early work on GaBP in loopy graphs can be found in Weiss and Freeman (2001). Important contributions made here include:

1. If GaBP converges, the posterior node potentials contain the correct marginal means.
2. An interesting representation of the computations in loopy GaBP as GaBP applied on a tree-structured precision matrix (known as unwrapped or computation trees).
3. A precision matrix $\mathbf{S} : p \times p$ is called diagonally dominant if $S_{ii} > \sum_{j \neq i} |S_{ji}|$ for $i = 1, 2, \ldots, p$. Diagonal dominance of a precision matrix is a sufficient condition for the convergence of GaBP.

The spectral radius of a matrix, $\mathbf{S} : p \times p$, with eigenvalues $\tau_i : i = 1, 2, , p$ is defined to be,

$$\rho(\mathbf{S}) = \max_i \{|\tau_i|\}. \tag{1}$$

Suppose $\mathbf{S}$ is symmetric, positive definite and normalized to have only ones along its diagonal, that is $\mathbf{S} = \mathbf{I} - \mathbf{R}$ where $\text{diag}(\mathbf{R}) = \mathbf{0}$. Let $|\mathbf{R}|$ be the matrix with entries equal to the absolute values of the entries of $\mathbf{R}$. The matrix $\mathbf{S}$ is walk-summable if and only if the spectral radius of $|\mathbf{R}|$ is less than one. The class of precision matrices for which GaBP converges was expanded to include positive definite symmetric matrices which are walk-summable, but may still converge for other precision matrices (Malioutov et al. 2006). In general, the converged posterior distributions do not give the exact marginal precisions. In the walk-summable case, this is because the computation trees do not cover all the walks present in the expansion $\mathbf{S}^{-1} = (\mathbf{I} - \mathbf{R})^{-1} = \sum_{k=0}^{\infty} \mathbf{R}^k$ (Malioutov et al. 2006). The posterior precisions can still be useful approximations for the marginal precisions. Several variants of GaBP have been proposed in the literature (Johnson et al. 2009; El-Kurdi et al. 2012a) to improve on the convergence performance of the original GaBP. These

methods are aimed at accelerating GaBP and/or achieving convergence in cases where ordinary GaBP diverges. These variants emphasize the distributive implementation of GaBP for solving large systems of linear equations. A variety of sufficient conditions for the convergence of GaBP has been proposed in the literature (Weiss and Freeman 2001; Malioutov et al. 2006). A recent work provides necessary and sufficient conditions for the convergence of synchronous GaBP under a specified initialization set (Su and Wu 2015). Furthermore, necessary and sufficient conditions are established for damped synchronous GaBP and they include the allowable range for the damping factor. A further contribution is the theoretical confirmation that damping can improve the convergence behaviour of GaBP. Applications of GaBP include MMSE multi-user detection, equalization and channel estimation in communication systems (Montanari et al. 2006; Guo and Huang 2011; Guo and Li 2008), a fast solver for systems of linear equations (El-Kurdi et al. 2012b; Shental et al. 2008), sparse Bayesian learning in large-scale compressed sensing problems (Seeger and Wipf 2010), and estimation on Gaussian graphical models (Chandrasekaran et al. 2008; Liu et al. 2012).

## 3 Message update rules

Before turning to our high-level approach, we make some comments on message update rules within the BP context. 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 are therefore 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 main focus will be on the synchronous case. We do this since one of the more attractive properties of GaBP is its application in distributive settings which is far more compatible with synchronous message updates. Synchronous implementation also allows us to compare different GaBP algorithms without considering the effects of different message schedulings. We do, however, include a section with comments on asynchronous message updates.

## 4 High-level approach

Our high-level approach is based on the max-product belief propagation algorithm. We will restrict our discussion to synchronous message updates. Suppose we want to find the mode of a density function $f(\mathbf{x})$ with the expansion,

$$f(\mathbf{x}) = e^K \prod_{i=1}^{p} \delta_i(\mathbf{x}_i) \times \prod_{i \neq j}^{p} g_{ij}(\mathbf{x}_i, \mathbf{x}_j), \tag{2}$$

where $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p)$, $\mathbf{x}_i$ may be higher dimensional and $e^K$ is a normalization constant. The max-product algorithm operates on

$$l(\mathbf{x}) = \log(f(\mathbf{x})) = K + \sum_{i=1}^{p} \phi_i(\mathbf{x}_i) + \sum_{i \neq j}^{p} h_{ij}(\mathbf{x}_i, \mathbf{x}_j), \tag{3}$$

where $\phi_i = \log(\delta_i)$ and $h_{ij} = \log(g_{ij})$. Equations 2 and 3 correspond to a Markov graph with $p$ nodes $\mathcal{H}_i : i = 1, 2, \ldots, p$. We assign to node $\mathcal{H}_i$ the vector $\mathbf{x}_i$. Node $i$ and node $j$ are linked if $h_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ is not zero for certain $\mathbf{x}_i, \mathbf{x}_j$. Let $\mathcal{N}_i$ be the set containing the neighbours of node $i$ (we do not include $i$ in this set), that is all nodes to which $i$ has a link. Suppose we are at stage $n$ of a synchronous max-product belief propagation algorithm with messages $m_{ij}^{(n)}(.)$ for $i = 1, 2, \ldots, p$ and $j \in \mathcal{N}_i$. The updated messages are

$$m_{ij}^{(n+1)}(\mathbf{x}_j) = \max_{\mathbf{x}_i} \left\{ \phi_i(\mathbf{x}_i) + h_{ij}(\mathbf{x}_i, \mathbf{x}_j) + \sum_{k \in \mathcal{N}_i/j} m_{ki}^{(n)}(\mathbf{x}_i) \right\}, \tag{4}$$

where by $\mathcal{N}_i/j$ we mean the set $\mathcal{N}_i$ with $j$ removed. Suppose at stage $n-1$, the posterior mode is $\boldsymbol{\mu}^{(n-1)} = (\boldsymbol{\mu}_1^{(n-1)}, \boldsymbol{\mu}_2^{(n-1)}, \ldots, \boldsymbol{\mu}_p^{(n-1)})'$. By node regularization, we mean that the optimization problem in Eq. 4 is replaced by,

$$m_{ij}^{(n+1)}(\mathbf{x}_j; \lambda) = \max_{\mathbf{x}_i} \left\{ \phi_i(\mathbf{x}_i) + h_{ij}(\mathbf{x}_i, \mathbf{x}_j) + \sum_{k \in \mathcal{N}_i/j} m_{ki}^{(n)}(\mathbf{x}_i) - \frac{\lambda}{2} \left[ \| \mathbf{x}_i - \boldsymbol{\mu}_i^{(n-1)} \|_q \right]^q \right\}, \tag{5}$$

with $\lambda$ a scalar and $\| . \|_q$ the $L_q$ norm of a vector. The use of $\boldsymbol{\mu}^{(n-1)}$ instead of $\boldsymbol{\mu}^{(n)}$ when propagating messages at stage $n$ (that is computing the stage $n+1$ messages) is important. To understand the rationale behind node regularization, one needs to understand one of the fundamental problems in belief propagation, that is the problem of loopy graphs. Consider a node $i$ in a Markov graph and suppose there is at least one path $P_i$ through the graph back to node $i$. Sending messages through this path means that the prior potential of node $i$ is cycled back to this node. This causes node $i$ to continuously increase belief in its prior mode and this may cause either divergence or incorrect convergence. The idea behind the penalty in (5) is to slow down this increase in belief, hence

the name slow Gaussian belief propagation. Although the formulation in (5) seems to make sense for $\lambda \geq 0$ there is also a role for negative values. Negative values of $\lambda$ correspond to a relaxation effect on the message propagation where nodes are encouraged to favour their own prior beliefs. In the context of GaBP, negative values of $\lambda$ relate sGaBP to RGaBP (relaxed GaBP). In the empirical section, we provide an indication of when it is appropriate to use a negative $\lambda$ and provide a comparison of RGaBP and sGaBP in this context.

## 5 Slow Gaussian belief propagation message updates

We use (5) to derive the message updates in the Gaussian context. It is natural to use $q = 2$ in (5) since this preserves the conjugacy of the messages. For the Gaussian distribution, we have

$$\begin{aligned} \phi_i(\mathbf{x}_i) &= -\frac{1}{2}\mathbf{x}_i'\mathbf{S}_{ii}\mathbf{x}_i + \mathbf{x}_i'\boldsymbol{b}_i. \\ h_{ij}(\mathbf{x}_i) &= -\mathbf{x}_i\mathbf{S}_{ij}\mathbf{x}_j. \end{aligned} \tag{6}$$

We assume that the messages are of the form,

$$m_{ij}^{(n)}(\mathbf{x}_j) = -\frac{1}{2}\mathbf{x}_j'\mathbf{Q}_{ij}^{(n)}\mathbf{x}_j + \mathbf{x}_j'\mathbf{v}_{ij}^{(n)} + C_{ij}^{(n)}, \tag{7}$$

for all $i \neq j$ and $C_{ij}^{(n)}$ is a constant. Applying (7) to (5) we obtain,

$$\begin{aligned} m_{ij}^{(n+1)}(\mathbf{x}_j;\lambda) = \max_{\mathbf{x}_i}\Bigg\{ &-\frac{1}{2}\mathbf{x}_i'(\mathbf{S}_{ii} + \sum_{k\in\mathcal{N}_i/j}\mathbf{Q}_{ki}^{(n)})\mathbf{x}_i \\ &+ \mathbf{x}_i'(\boldsymbol{b}_i - \mathbf{S}_{ij}\mathbf{x}_j + \sum_{k\in\mathcal{N}_i/j}\mathbf{v}_{ki}^{(n)}) \\ &- \frac{\lambda}{2}(\mathbf{x}_i - \boldsymbol{\mu}_i^{(n-1)})'(\mathbf{x}_i - \boldsymbol{\mu}_i^{(n-1)}) \\ &+ \sum_{k\in\mathcal{N}_i/j} C_{ki}^{(n)}\Bigg\}. \end{aligned} \tag{8}$$

Since $\frac{\lambda}{2}(\mathbf{x}_i - \boldsymbol{\mu}_i^{(n-1)})'(\mathbf{x}_i - \boldsymbol{\mu}_i^{(n-1)}) = \frac{\lambda}{2}\mathbf{x}_i'\mathbf{x}_i - \lambda\mathbf{x}_i'\boldsymbol{\mu}_i^{(n-1)} + \frac{1}{2}||\boldsymbol{\mu}_i^{(n-1)}||^2$,

$$\begin{aligned} m_{ij}^{(n+1)}(\mathbf{x}_j;\lambda) = \max_{\mathbf{x}_i}\Bigg\{ &-\frac{1}{2}\mathbf{x}_i'(\lambda\mathbf{I} + \mathbf{S}_{ii} + \sum_{k\in\mathcal{N}_i/j}\mathbf{Q}_{ki}^{(n)})\mathbf{x}_i \\ &+ \mathbf{x}_i'(\boldsymbol{b}_i + \lambda\boldsymbol{\mu}_i^{(n-1)} - \mathbf{S}_{ij}\mathbf{x}_j \\ &+ \sum_{k\in\mathcal{N}_i/j}\mathbf{v}_{ki}^{(n)}) + \tilde{C}_{ij}\Bigg\}, \end{aligned} \tag{9}$$

where $\tilde{C}_{ij}$ is a constant. For convenience we set $\mathbf{A}_{ij}^{(n)} = \lambda\mathbf{I} + \mathbf{S}_{ii} + \sum_{k\in\mathcal{N}_i/j}\mathbf{Q}_{ki}^{(n)}$ and $\mathbf{a}_{ij}^{(n)} = \boldsymbol{b}_i + \lambda\boldsymbol{\mu}_i^{(n-1)} + \sum_{k\in\mathcal{N}_i/j}\mathbf{v}_{ki}^{(n)}$. We now need to compute

$$\begin{aligned} &m_{ij}^{(n+1)}(\mathbf{x}_j;\lambda) \\ &\quad= \max_{\mathbf{x}_i}\left\{-\frac{1}{2}\mathbf{x}_i'\mathbf{A}_{ij}^{(n)}\mathbf{x}_i + \mathbf{x}_i'(\mathbf{a}_{ij}^{(n)} - \mathbf{S}_{ij}\mathbf{x}_j) + \tilde{C}_{ij}\right\}. \end{aligned} \tag{10}$$

It is easy to show that substituting $\mathbf{x}_i = [\mathbf{A}_{ij}^{(n)}]^{-1}(\mathbf{a}_{ij}^{(n)} - \mathbf{S}_{ij}\mathbf{x}_j)$ into $-\frac{1}{2}\mathbf{x}_i'\mathbf{A}_{ij}^{(n)}\mathbf{x}_i + \mathbf{x}_i'(\mathbf{a}_{ij}^{(n)} - \mathbf{S}_{ij}\mathbf{x}_j) + \tilde{C}_{ij}$ gives (10). We now make the assumption that $\mathbf{Q}_{ij}^{(n)}$ is symmetric for all $i \neq j$. We have,

$$\begin{aligned} m_{ij}^{(n+1)}(\mathbf{x}_j;\lambda) &= \frac{1}{2}(\mathbf{a}_{ij}^{(n)} - \mathbf{S}_{ij}\mathbf{x}_j)'[\mathbf{A}_{ij}^{(n)}]^{-1}(\mathbf{a}_{ij}^{(n)} - \mathbf{S}_{ij}\mathbf{x}_j) \\ &\quad + \tilde{C}_{ij} \\ &= \frac{1}{2}\mathbf{x}_j'\mathbf{S}_{ji}[\mathbf{A}_{ij}^{(n)}]^{-1}\mathbf{S}_{ij}\mathbf{x}_j \\ &\quad - \mathbf{x}_j'\mathbf{S}_{ji}[\mathbf{A}_{ij}^{(n)}]^{-1}\mathbf{a}_{ij}^{(n)} + C_{ij}^{(n+1)} \\ &= -\frac{1}{2}\mathbf{x}_j'\mathbf{Q}_{ij}^{(n+1)}\mathbf{x}_j + \mathbf{x}_j'\mathbf{v}_{ij}^{(n+1)} + C_{ij}^{(n+1)}, \end{aligned} \tag{11}$$

where $\mathbf{Q}_{ij}^{(n+1)} = -\mathbf{S}_{ji}[\mathbf{A}_{ij}^{(n)}]^{-1}\mathbf{S}_{ij}$, $\mathbf{v}_{ij}^{(n+1)} = -\mathbf{S}_{ji}[\mathbf{A}_{ij}^{(n)}]^{-1}\mathbf{a}_{ij}^{(n)}$ and $C_{ij}^{(n+1)}$ does not depend on $\mathbf{x}_j$. In the literature, it is common practice to ignore the update of the constant-components of messages since they are not needed to update $\mathbf{Q}_{ij}^{(n+1)}$ and $\mathbf{v}_{ij}^{(n+1)}$ and we will follow this convention. We note that all the assumptions made in this section are recurring and can therefore be ensured by appropriate initialization. Our focus will be on one-dimensional nodes. Here, all the matrices are replaced by scalars and our message updates are performed by

$$Q_{ij}^{(n+1)} = \frac{-S_{ij}^2}{\lambda + S_{ii} + \sum_{k\in\mathcal{N}_i/j}Q_{ki}^{(n)}}. \tag{12}$$

$$V_{ij}^{(n+1)} = \frac{Q_{ij}^{(n+1)}}{S_{ij}}\left[\lambda\mu_i^{(n-1)} + b_i + \sum_{k\in\mathcal{N}_i/j}V_{ki}^{(n)}\right]. \tag{13}$$

In order to ensure the convergence of sGaBP, while preserving the exactness of the posterior means, the implementation of the algorithm requires some additional steps (beyond the message passing). The implementation of these steps and the convergence properties of sGaBP are discussed in the next section.

## 6 Convergence analysis

The synchronous implementation of sGaBP is given in Algorithm 1 where certain steps are discussed in this section. We start by discussing the computation of the posterior distributions after each iteration. These computations are important since they are sufficient to ensure convergence of sGaBP (for large enough $\lambda$) while preserving the posterior means as the exact marginal means. This is followed by a study of the convergence behaviour of the precision components of the messages, i.e. the behaviour of $\mathbf{Q}^{(n)}$ in Algorithm 1. We then proceed to the mean/potential components of the messages by assuming convergence of the precision components. In our convergence analysis, we assume that the precision matrix has been preconditioned to have 1 on the diagonals. If the precision matrix is $\mathbf{S}$ (before preconditioning), this can be achieved by setting $\mathbf{D} = \text{diag}(\frac{1}{\sqrt{S_{11}}}, \frac{1}{\sqrt{S_{22}}}, \ldots, \frac{1}{\sqrt{S_{pp}}})$ and computing $\mathbf{DSD}$. This type of preconditioning does not entail any loss of information in the sense that both the marginal means and precisions of the distribution in its original scale can be recovered.

### 6.1 Computation of posterior distributions

In order to ensure convergence of sGaBP and to have the posterior means (at convergence) equal to the correct marginal means, it is necessary to adjust the manner in which posterior distributions are computed. Consider the computation of the posterior distribution of node $i$ at stage $n$. As a first step, we instruct node $i$ to collect all incoming messages, which can be characterized by the parameters $\sum_{t\neq i} Q_{ti}^{(n)}$ (precision components) and $\sum_{t\neq i} V_{ti}^{(n)}$ (mean/potential components). We suggest keeping the posterior precisions as in normal belief propagation, that is $1+\sum_{t\neq i} Q_{ti}^{(n)}$. Later, we investigate the role of $\lambda$ in the tuning of the posterior precisions to better approximate the marginal precisions. The posterior mean of node $i$ at stage $n$ is given by

$$\mu_i^{(n)} = \frac{\lambda\mu_i^{(n-1)} + z_i^{(n)}}{\lambda + q_i^{(n)}} = \gamma_i^{(n)}\mu_i^{(n-1)} + (1-\gamma_i^{(n)})\frac{z_i^{(n)}}{q_i^{(n)}} \tag{14}$$

where $z_i^{(n)} = b_i + \sum_{t\neq i} V_{ti}^{(n)}$, $q_i^{(n)} = 1 + \sum_{t\neq i} V_{ti}^{(n)}$ and $\gamma_i^{(n)} = \frac{\lambda}{\lambda+q_i^{(n)}}$. Note that $\frac{z_i^{(n)}}{q_i^{(n)}}$ is the posterior mean we would have computed if no adjustment was made to the computation of the posterior distribution. Hence, we can interpret (14) as damping between the posterior mean, under normal belief propagation, and the posterior mean computed in the previous round. What is nice here is that these damping factors are computed automatically (using $\lambda$ and the current posterior precisions) and no additional parameters are required.

**Algorithm 1** Synchronous sGaBP.

1. Provide $\mathbf{S}: p\times p$, $\boldsymbol{b}: p\times 1$, $\lambda$, $m$ and $\epsilon$ as inputs to the algorithm. Here, we wish to solve $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$ where $\mathbf{S}$ is positive definite and symmetric. The parameters $\lambda$, $m$ and $\epsilon$ denote the degree of diagonal loading, the maximum number of iterations allowed and the tolerance used to define convergence, respectively.
2. Initiate $\mathbf{Q}^{(0)} = \text{diag}(1,1,\ldots,1)$, $\mathbf{V}^{(0)} = \text{diag}(b_1, b_2, \ldots, b_p)$ and $\boldsymbol{\mu}^{(-1)} = \mathbf{0}$.
3. Set Err = Inf and $n = 0$.
4. while Err $> \epsilon$
   (a) Compute $q_i^{(n)} = 1+\sum_{j\in\mathcal{N}_i} Q_{ji}^{(n)}$ and $z_i^{(n)} = b_i+\sum_{j\in\mathcal{N}_i} V_{ji}^{(n)}$ for $i = 1,2,\ldots,p$.
   (b) Set $\mu_i^{(n)} = \frac{\lambda\mu_i^{(n-1)}+z_i^{(n)}}{\lambda+q_i^{(n)}}$ for $i = 1,2,\ldots,p$.
   (c) For all $i$ and all $j\in\mathcal{N}_i$, set $Q_{ij}^{(n+1)} = \frac{-S_{ij}^2}{\lambda+q_i^{(n)}-Q_{ji}^{(n)}}$ and $V_{ij}^{(n+1)} = \frac{Q_{ij}^{(n+1)}}{S_{ij}}(\lambda\mu_i^{(n-1)} + z_i^{(n)} - V_{ji}^{(n)})$.
   (d) Set Err $= \sqrt{\frac{\sum_k(\mu_k^{(n)}-\mu_k^{(n-1)})^2}{\sum_k(\mu_k^{(n)})^2}}$ and increment $n$.
   (e) If $m = n$ break.
5. End.

The values, $\gamma_i^{(n)}: i = 1,2,\ldots,p$, can also be relaxation factors which correspond to negative $\lambda$. We now show that these adjustments are sufficient for the convergence and the preservation of the (converged) posterior means as the exact marginal means.

### 6.2 The precision components

The convergence analysis of the precision components is the simpler of the two since we can apply results found in the literature (Malioutov et al. 2006; Bickson 2008). Suppose $\mathbf{Q}^{(n)}(\lambda)$ holds the precision components of the messages at iteration $n$. The analysis of $\mathbf{Q}^{(n)}(\lambda)$ is simple since it is identical to the precision components provided by ordinary GaBP applied on the matrix $\lambda\mathbf{I}+\mathbf{S}$. Therefore, we only need to select $\lambda$ large enough such that $\lambda\mathbf{I}+\mathbf{S}$ is walk-summable (although smaller selections of $\lambda$ can also suffice). From this point onwards, we use the symbol $\tilde{\rho}(\mathbf{S}) = \rho(\mathbf{I}-\mathbf{S})$, we also refer to $\tilde{\rho}(\mathbf{S})$ as the zero-diagonal spectral radius of $\mathbf{S}$. Selecting $\lambda > \tilde{\rho}(|\mathbf{S}|) - 1 = \rho(|\mathbf{R}|) - 1$, where $\mathbf{S} = \mathbf{I}-\mathbf{R}$, is sufficient for the precision components in Algorithm 1 to converge. In addition to this, we prove Theorem 1 in "Appendix 1".

**Theorem 1** *The following properties recur indefinitely (as a set) in sGaBP.*

1. $Q_{ij}^{(n)} \leq 0$ *for all* $i,\ j\in\mathcal{N}_i$.
2. $|Q_{ij}^{(n)}| > |Q_{ij}^{(n-1)}|$ *for all* $i,\ j\in\mathcal{N}_i$.
3. $\delta_i^{(n)} = \sum_{t\in\mathcal{N}_i}|Q_{ti}^{(n)}| \leq \delta_i$ *for a* $0\leq\delta_i<1+\lambda$ *and all* $i$.
4. $\sum_{t\in\mathcal{N}_i}\frac{S_{ti}^2}{1+\lambda-\delta_t+|Q_{it}^{(n)}|} \leq \delta_i$ *for all* $i$.

If these conditions hold, then $Q_{ij}^{(n)}$ are monotone decreasing and bounded from below by $\frac{-S_{ij}^2}{1+\lambda-\delta_i}$ and will therefore converge. Consider the case where $\delta_i = \delta$ for all $i$ and suppose we want the conditions in Theorem 1 to hold from $n = 0$. Since $Q_{ij}^{(0)} = 0$ we need a $0 \leq \delta < 1 + \lambda$ satisfying $\sum_{i \neq j} \frac{S_{ij}^2}{1+\lambda-\delta} \leq \delta$ for all $j$. This inequality is equivalent to a quadratic inequality with roots,

$$\frac{(1+\lambda) \pm \sqrt{(1+\lambda)^2 - 4\sum_{i \neq j} S_{ij}^2}}{2}. \tag{15}$$

If we select $(1+\lambda)^2 - 4 \times \max_j \left\{ \sum_{i \neq j} S_{ij}^2 \right\} \geq 0$ or $\lambda \geq 2\sqrt{\max_j \left\{ \sum_{i \neq j} S_{ij}^2 \right\}} - 1$, then we can select $\delta = \frac{1+\lambda}{2}$ to guarantee monotone convergence of all the precisions. For this selection, the bounds on the precisions are

$$-\frac{2S_{ij}^2}{1+\lambda} \leq Q_{ij}^{(n)} \leq 0. \tag{16}$$

An important consequence of (16) is that

$$\lim_{\lambda \to \infty} Q_{ij}^{(n)} = 0, \tag{17}$$

for all $i \neq j$. Here, we emphasize the role of $\lambda$ in the tuning of the posterior precisions. Note that we can tune the converged precisions, $Q_{ij}$, to any value in the interval $[-\frac{2S_{ij}^2}{1+\lambda_0}; 0]$ (this interval can be much larger) where $\lambda_0 = 2\sqrt{\max_j \left\{ \sum_{i \neq j} S_{ij}^2 \right\}} - 1$, although there is dependence among the $Q_{ij}$'s in terms of $\lambda$. This can in turn be used to tune the posterior precisions, $1 + \sum_{t \neq i} Q_{ti}$, under certain restrictions. The tuning can be made more flexible by introducing multiple tuning parameters.

### 6.3 The mean components

In the previous section, we saw that the precision components of the messages will converge for sufficiently large choices of $\lambda$. In this section, we proceed under the assumption that the precision components have converged. We denote the converged precision message-components, posterior precisions and damping factors by $Q_{ij}$, $q_i$ and $\gamma_i$, respectively. The updates of the mean components are

$$V_{ij}^{(n+1)} = \frac{Q_{ij}}{S_{ij}} \left[ \lambda \mu_i^{(n-1)} + b_i + \sum_{t \in \mathcal{N}_i/j} V_{ti}^{(n)} \right]. \tag{18}$$

We define $\boldsymbol{\theta}^{(n+1)}$ to be the vector obtained by stacking the columns of $\mathbf{V}^{(n+1)}$, removing the diagonal entries and appending $\boldsymbol{\mu}^{(n)}$ (after the columns of $\mathbf{V}^{(n+1)}$). This vector can be expressed as,

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta} + \mathbf{L}\boldsymbol{\theta}^{(n)}, \tag{19}$$

for a matrix $\mathbf{L} : p^2 \times p^2$ and a vector of constants $\boldsymbol{\theta} : p^2 \times 1$. The first $p^2 - p$ entries of $\boldsymbol{\theta}$ can be obtained by constructing the matrix $\mathbf{C} = [\frac{Q_{ij}}{S_{ij}} b_i]$, with the understanding that the diagonals are zero, and stacking the columns in the same way as we did with the mean precision components. The final $p$ entries of $\boldsymbol{\theta}$ are $\frac{1-\gamma_i}{q_i} b_i$ in order $i = 1, 2, \ldots, p$. The construction of $\mathbf{L}$ is more complex. Consider,

$$\mathbf{L} : p^2 \times p^2 = \begin{bmatrix} \mathbf{L}_{11} : l \times l & \mathbf{L}_{12} : l \times p \\ \mathbf{L}_{21} : p \times l & \mathbf{L}_{22} : p \times p \end{bmatrix}, \tag{20}$$

where $l = p^2 - p$. Consider one of the first $l$ elements of $\boldsymbol{\theta}^{(n+1)}$, say $m$. This element corresponds to an entry in the matrix $\mathbf{V}^{(n+1)}$, say $V_{ij}^{(n+1)}$. The next step is to identify the neighbours of $i$, that is the set $\mathcal{N}_i$. For each $k \in \mathcal{N}_i/j$, we find the element in $\boldsymbol{\theta}^{(n)}$ corresponding to $V_{ki}^{(n)}$ and note its position. The entry in row $m$ of $\mathbf{L}$ in this position is $\frac{Q_{ij}}{S_{ij}}$. This accounts for the matrix $\mathbf{L}_{11}$ with the understanding that all elements not accessed are zero. Continuing with this notation, the entry in row $m$ of $\mathbf{L}_{12}$ in position $i$ is $\frac{\lambda Q_{ij}}{S_{ij}}$ and all other elements in this row are zero. We see that $\mathbf{L}_{22}$ is a diagonal matrix with entries $\gamma_i$ in order $i = 1, 2, \ldots, p$. Consider the matrix $\mathbf{L}_{21}$. The first step is to identify the neighbours of node $i$, that is $\mathcal{N}_i$. We then move along the vector $\boldsymbol{\theta}^{(n)}$ and identify all the positions corresponding to $V_{ti}^{(n)} : t \in \mathcal{N}_i$. In row $i$ of $\mathbf{L}_{21}$, we place the value $\frac{1-\gamma_i}{q_i}$ in the identified positions, the rest of the entries are zero.

Our goal is to analyse the spectral radius of $\mathbf{L}$. We note that the eigenvalues of $\mathbf{L}$ can possibly be complex. In the case of complex eigenvalues, the spectral radius of $\mathbf{L}$ is defined to be the largest modulus among the eigenvalues of $\mathbf{L}$. If the spectral radius of $\mathbf{L}$ is less than 1, sGaBP will converge (assuming that the precisions converge). The value of the spectrum has a heavy influence on the convergence speed of sGaBP and can play a role in deciding on how to select $\lambda$. A natural way to select the amount of regularization is to seek $\lambda$ such that the spectral radius (of $\mathbf{L}$) is a minimum. We make some comments on the form of the spectrum later in this section. For the purpose of this article, we consider the asymptotic behaviour of the spectral radius and show that the spectral approaches 1 from below as $\lambda \to \infty$. The selection of $\lambda$ is considered in the section on heuristic measures. Theorem 2 provides information on the asymptotic behaviour of the spectrum, the proof is given in "Appendix 1".

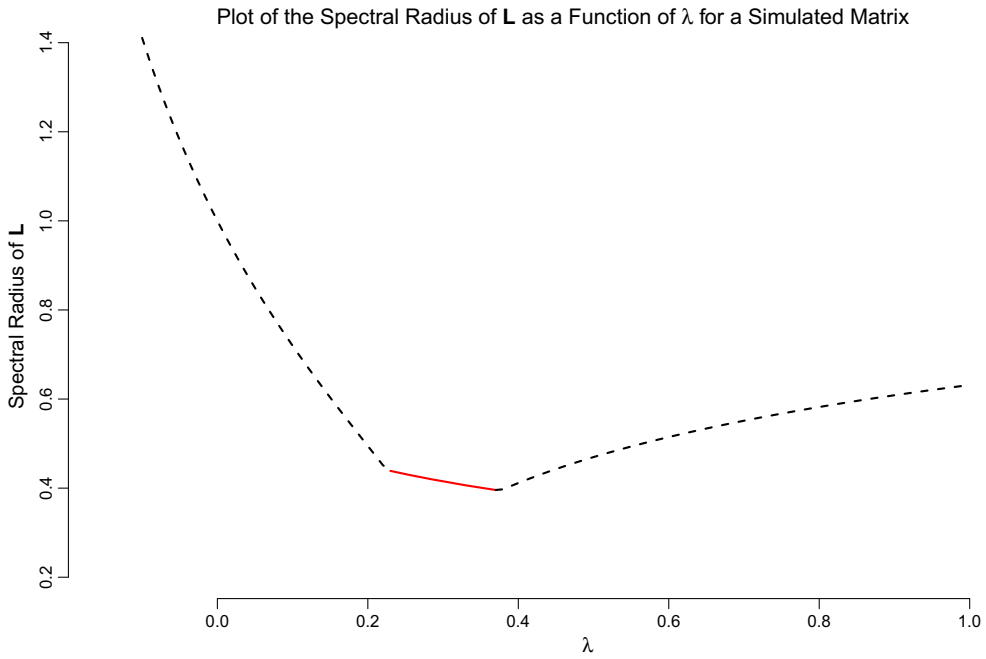


**Fig. 1** Plot of the spectral radius of the linear-update matrix as a function of λ for a simulated $10 \times 10$ matrix with a zero-diagonal spectral radius equal to one. The *red solid line* corresponds to a complex spectral radius. The *black broken line* corresponds to a real spectral radius

**Theorem 2** *Consider sGaBP applied to a multivariate Gaussian with potential* $\boldsymbol{b}$ *and precision matrix* $\mathbf{S}$. *The eigenvalues of the linear-update matrix can be characterized as,*

$$1 - \frac{\sigma_i}{\lambda} + \mathcal{O}\left(\frac{1}{\lambda^2}\right); \quad i = 1, 2, \ldots, p \tag{21}$$

$$\frac{\pm S_{ij}}{\lambda} + \mathcal{O}\left(\frac{1}{\lambda^2}\right); \quad i \neq j, \tag{22}$$

*where* $\sigma_i, i = 1, 2, \ldots, p$ *represent the eigenvalues of* $\mathbf{S}$.

In particular, we see that if $\mathbf{S}$ is positive definite, the maximum of the eigenvalues in Theorem 2 tends to 1 from below as $\lambda \to \infty$. We see that the precisions will converge for $\lambda$ large enough and will eventually generate a linear-update matrix with a spectral radius less than 1, that is sGaBP will converge for large enough $\lambda$. In "Appendix 1", we show that the posterior means provided by sGaBP (under the assumption of convergence) provide the exact marginal means.

We now make some comments on the behaviour of the spectrum (eigenvalues) of $\mathbf{L}$. One interesting aspect of the spectrum is when sGaBP is applied to tree-structured precision matrices. We generated a few of these and in each case we found the matrix $\mathbf{L}$ to be nilpotent when $\lambda = 0$. This relates to BP as an efficient and exact marginalization algorithm on tree structures. The use of values of $\lambda$ other than zero is nonsensical in this case. A typical plot of the spectral radius as a function of $\lambda$ is given in Fig. 1. In this case, the spectral radius has a global minimum at a value of $\lambda$ just under 0.4. The spectral radius can correspond to either a complex or a real eigenvalue and the graph of the spectral radius seems to change curvature when the eigenvalue responsible for the spectral radius changes from real to complex (and vice versa). This can be seen in Fig. 1 with the red solid line corresponding to a complex spectral radius and the black broken line to a real spectral radius. We also see that the spectral radius eventually becomes real, which is consistent with Theorem 2. Another important observation is that the value of $\lambda$ which minimizes the spectral radius seems to occur at a point where the eigenvalue responsible switches from real to complex or vice versa. Furthermore, there can be more than one point where this change occurs. Our simulations show similar results for other precision matrices. The interaction between complex and real eigenvalues could prove useful in the minimization of the spectral radius and should be considered in further research.

### 6.4 The converged posteriors

Having proved convergence of sGaBP, we now turn to the posterior distributions as approximations of the marginal distributions. In Theorem 3, we prove that the posterior means are the exact marginal means, the proof is provided in "Appendix 1". A consequence of Theorem 3 is that sGaBP can be used to solve linear systems, $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$, as long as $\mathbf{S}$ is a valid precision matrix. Unfortunately, the posterior precisions are not necessarily equal to the true marginal precisions. In the experimental section, we show that the posterior precisions provided by sGaBP can be useful as approximate

**Algorithm 2** Heuristic Selection of $\lambda$.

1. In Algorithm 1, step 1, add the specification of a lag ($d$) and a step size ($\alpha$).
2. In Algorithm 1, step 2, add the initialization $e^{\text{best}} = \max_i |b_i|$.
3. Before Algorithm 1, step 4a, add the following.
   (a) If mod($n$,$d$) = 0
   – If $e^{\text{best}} >$ Err then $\lambda \leftarrow \lambda - \alpha$ and $e^{\text{best}} \leftarrow$ Err , else $\lambda \leftarrow \lambda + \alpha$.

quantities in the sense that the KL distance between the posterior and marginal distributions can be small. Included in our empirical study are two variants of GaBP namely Relaxed Gaussian belief propagation (RGaBP) and Convergence Fix Gaussian belief propagation (CF). All three methods (sGaBP, RGaBP and CF) require the specification of at least one hyper-parameter. We compare the precisions provided by these 3 methods by finding the values of the hyper-parameters yielding the fastest convergence. For RGaBP, the value of the hyper-parameter is irrelevant in terms of comparing precisions since the posterior precisions provided are identical to those provided by ordinary GaBP. Within this set-up, we show empirically that sGaBP can provide substantially more accurate approximations of the marginal precisions (compared to RGaBP and CF) while also converging at faster rates.

## 7 Heuristic measures

In this section, we propose some heuristic measures for the selection of $\lambda$. These measures vary in degree of complexity and we discuss some of their advantages and disadvantages.

### 7.1 Search heuristic

This heuristic is basically the same as the one proposed by El-Kurdi et al. (2012a), adjusted for sGaBP, and is given in Algorithm 2. The main advantage of this heuristic is that it is easy to implement. There are some drawbacks to this measure arising from the monotone way in which the tuning is adjusted. When the current tuning provides posterior means with a smaller (larger) error, the heuristic will always decrement (increment) the tuning. The heuristic seeks tuning for which the spectral radius of $\mathbf{L}$ is less than one and not necessarily tuning for which the value of the spectral radius is a minimum.

### 7.2 Gradient descent heuristic (GDH)

GDH is more complex to implement, but does not have the monotonicity of SH as described in Sect. 7.1. The heuristic is aimed at determining the direction in which the tuning needs to be adjusted to achieve the smallest possible spectral radius. The tuning is then adjusted in this direction in step sizes which should not be overly large.

Suppose we have completed iteration $n$ of sGaBP and we are preparing to perform the next round of updates. We wish to adjust the value of the tuning parameter in a direction which yields faster convergence. At this point, we have the posterior precisions $q_i^{(n)} : i = 1, 2, \ldots, p$ and posterior means $\mu_i^{(n)}$. The posterior means were computed using

$$\begin{aligned}\mu_i^{(n)} &= \frac{q_i^{(n)}}{\lambda + q_i^{(n)}} \frac{b_i + \sum_{j\in\mathcal{N}_j} V_{ji}^{(n)}}{q_i^{(n)}} + \frac{\lambda}{\lambda + q_i^{(n)}} \mu_i^{(n-1)} \\ &= [1 - \gamma_i^{(n)}(\lambda)]\tilde{\mu}_i^{(n)} + \gamma_i^{(n)}(\lambda)\mu_i^{(n-1)}. \end{aligned} \tag{23}$$

Although this is not technically correct, we assume that $q_i^{(n)}$, $\mu_i^{(n-1)}$ and $\tilde{\mu}_i^{(n)}$ are constant and do not depend on $\lambda$. The GDH starts by instructing each node to send its posterior mean to its neighbours, each node then computes $e_j = \sum_{i\in\tilde{\mathcal{N}}_j} S_{ji}\mu_i^{(n)} - b_j$, where $\tilde{\mathcal{N}}_j$ is $\mathcal{N}_j$ with node $j$ included. Let $k = \text{argmax}_j\{|e_j|\}$. The node $k$ and each of its neighbours are instructed to compute the derivative of their own mean (can be done in parallel) by differentiating (23) relative to $\lambda$ and evaluating this at the current value of the tuning, say $\lambda_0$. The neighbours of node $k$ send these derivatives to node $k$ and this node computes $d_k = \sum_{j\in\{k,\mathcal{N}_k\}} S_{kj} \bigtriangledown\mu_j^{(n)}$, where $\bigtriangledown\mu_j^{(n)}$ is the derivative received from node $j$. Node $k$ is then instructed to adjust the tuning $\lambda_0 \leftarrow \lambda_0 - \alpha\text{sign}(d_k)$, for a specified step size $\alpha$, and to send this new tuning value to the other nodes.

### 7.3 Comparing SH and GDH: a concrete example

We use simulation to illustrate the possible benefits of using GDH instead of SH. We start by simulating a $100 \times 100$ precision matrix, $\mathbf{S}$, and potential vector, $\boldsymbol{b}$. We use the method in "Appendix 2" to regulate the zero-diagonal spectral radius of the precision matrix to 1. We defined convergence to occur when the error is less than $10^{-14}$. Using a line search in increments of 0.01, we observed that initializing the tuning of sGaBP with values 0.33, 0.34 and 0.35 yielded the fastest convergence and that this occurred after 28 iterations. We found that the spectral radius of $\mathbf{L}$ is 1 when $\lambda = 0$ (this is typical when the zero-diagonal spectral radius of $\mathbf{S}$ is 1). The values of the spectral radius (of $\mathbf{L}$) corresponding to $\lambda = -0.01$ and $\lambda = 0.01$ are 1.029343 and 0.971501, respectively. Assuming convergence of the precision components of the messages, we observed the error to be increasing for negative tuning and decreasing for positive tuning. If SH is used, there is the risk that the heuristic tuning will vary around the tuning corresponding to a spectral radius of one. This is because SH seeks tuning for which the spectral radius of $\mathbf{L}$ is less than one and not necessarily tuning which minimizes the

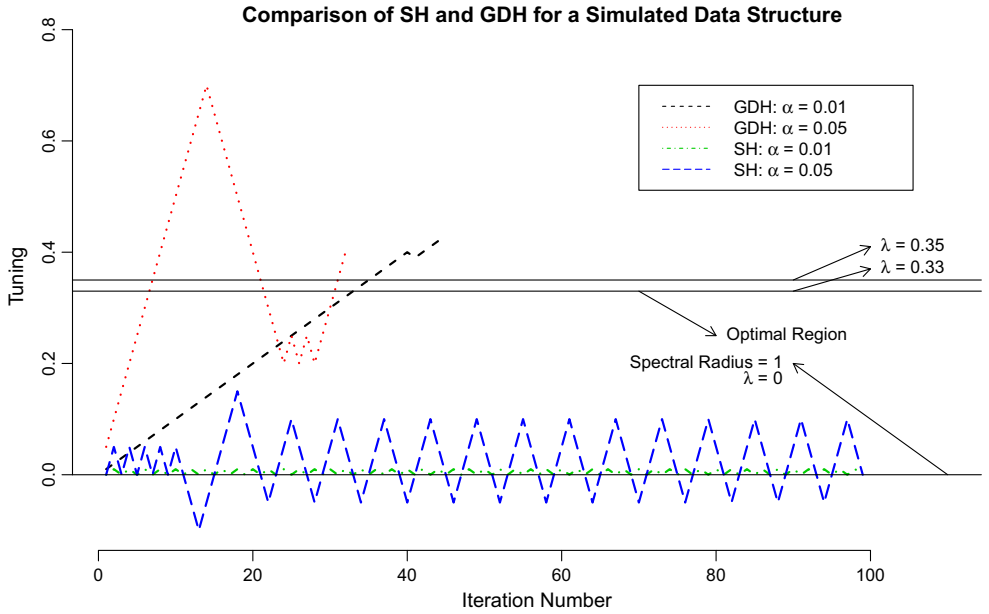


**Fig. 2** Comparison of SH and GDH for a simulated data structure. The precision matrix was regulated to have a zero-diagonal spectral radius equal to one. Some relevant quantities are given in the display. We see that the tuning provided by SH is stuck around $\lambda = 0$. Selecting $\lambda = 0$ a priori gives a spectral radius equal to one. GDH provides tuning closer to the values yielding the fastest convergence (determined using a line search in increments of 0.01). GDH converges faster than SH

spectral radius of $\mathbf{L}$. This is illustrated in Fig. 2. The y-axis shows the level of tuning used at the iteration number given on the x-axis. Figure 2 contains two lines for each of GDH and SH. The two lines for each of GDH and SH correspond to the step sizes 0.01 and 0.05. The tuning suggested by SH varies around $\lambda = 0$, the level of tuning corresponding to a spectral radius of 1. GDH is not restricted around a spectral radius of one and is able to make better adjustments on the tuning. Notice that the two graphs corresponding to GDH are terminated at iteration 45 and 32 corresponding to step sizes 0.01 and 0.05, respectively. This was done to indicate that sGaBP has converged after these numbers of iterations. Both applications of SH failed to converge after 100 iterations. This is not to say that SH cannot be effective, indeed the simplicity of implementation is an advantage over GDH, but rather that SH is more sensitive to the initialization of $\lambda$, particularly when this starting value is close to the level of tuning yielding a spectral radius of one.

## 8 Asynchronous message updates

We have referred to the use of asynchronous message updates as opposed to the synchronous version. In general, it is believed that asynchronous message updates can provide better convergence behaviour in applications of BP in the sense that they may induce convergence where synchronous updates diverge or require the passing of a smaller number of messages to converge (Koller and Friedman 2009). The major shortcoming of asynchronous updates is loss of distributive applicability. Another problem posed by asynchronous updates is the problem of deciding upon the order in which messages are passed, since this can have a significant effect on the convergence speed. In the context of GaBP, this problem is compounded by the fact that synchronous messages operate in iterations with $\mathcal{O}(p^2)$ computations, which discounts complicated heuristics used in other applications of BP to decide on the message scheduling. Progress can be made by deciding on the message scheduling in advance. There are other considerations as well, such as deciding on the degree of regularization. This should be considered from the viewpoint that the degree of regularization yielding optimal convergence should naturally provide useful posterior precisions. An example of the advantages of asynchronous message passing can be found in the diabetes data (Efron et al. 2004). The diabetes data were used to illustrate the advantages of the least angle regression algorithm in settings involving a high degree of collinearity among the explanatory variables. Estimating the linear coefficients of the diabetes data is challenging for GaBP since:

1. The number of explanatory variables is small.
2. The zero-diagonal spectral radius of the sample correlation matrix is high (3.024214).
3. There is significant variation among the sample correlations.

**Algorithm 3** Asynchronous sGaBP.

1. Provide $\mathbf{S}: p \times p, \boldsymbol{b}: p \times 1, \lambda, m$ and $\epsilon$ as inputs to the algorithm. Here, we wish to solve $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$ where $\mathbf{S}$ is positive definite and symmetric. The parameters $\lambda$, $m$ and $\epsilon$ denote the degree of diagonal loading, the maximum number of iterations allowed and the tolerance used to define convergence, respectively.
2. Initiate $\mathbf{Q} = \text{diag}(1, 1, \ldots, 1)$, $\mathbf{V} = \text{diag}(b_1, b_2, \ldots, b_p)$ , $\boldsymbol{\mu} = \mathbf{0}$, $\mathbf{q} = (1, 1, \ldots, 1)'$ and $\mathbf{z} = \boldsymbol{b}$.
3. Set Err = Inf and $n = 0$.
4. while Err $> \epsilon$
   (a) For $i = 1, 2, \ldots, p$ set $\mu_i^{\text{old}} = \mu_i$.
   (b) For $j = 1, 2, \ldots, p$
   For $i = 1, 2, \ldots, p$
      i. If $j = i$ or $S_{ij} = 0$ then next.
      ii. Set $a_1 = Q_{ij}, a_2 = V_{ij}$.
      iii. Update $Q_{ij} = \frac{-S_{ij}^2}{\lambda + q_i - Q_{ji}}$ and $V_{ij} = \frac{Q_{ij}}{S_{ij}}(\lambda\mu_i + z_i - V_{ji})$.
      iv. Update $q_j = q_j - a_1 + Q_{ij}$ and $z_j = z_j - a_2 + V_{ij}$.
      v. Update $\mu_j = \frac{\lambda\mu_j + z_j}{\lambda + q_j}$.
   (c) Set Err $= \sqrt{\frac{\sum_k (\mu_k - \mu_k^{\text{old}})^2}{\sum_k (\mu_k)^2}}$ and increment $n$.
   (d) If $m = n$ break.
5. End.

A line search in increments of 0.01 revealed that $\lambda = 1.29$ yields the fastest convergence for synchronous sGaBP (using a tolerance of $10^{-10}$) and convergence occurred after 574 iterations. The 574 iterations required for convergence is substantial when compared to the number of explanatory variables (which is 10). A further complication is that synchronous sGaBP with $\lambda = 1.29$ yields negative posterior precisions for certain variables, although this can be addressed by increasing $\lambda$ at the cost of slower convergence. We now apply asynchronous sGaBP, which is formulated in Algorithm 3. Notice that the outer-loop of the message updates iterates over $j$ indicating that the inner-loop iterates over messages to node $j$. This was done because we found that iterating over incoming messages first was more efficient in our simulations. Each round of message updates requires $\mathcal{O}(p^2)$ computations (fewer with sparsity) as in the synchronous case. Unlike the synchronous case, it is not necessary to store old messages and Algorithm 3 performs damping throughout the double-loop (instead of after). The optimal tuning value was determined as 2.01 (line search as for the synchronous case) and convergence occurred after 131 iterations. All posterior precisions were positive. We see that asynchronous message passing improved on the convergence speed and accuracy of the posterior distributions in the case of the diabetes data. In general, our simulations showed that asynchronous outperforms synchronous sGaBP in terms of convergence behaviour. This is further compounded by the fact that the asynchronous message passing does not require old messages to be stored, resulting in a lower computational burden on each iteration and lower memory requirements. We leave further aspects of asynchronous sGaBP, such as proof of convergence and heuristic measures, for further research.

## 9 Empirical work

In this section, we provide empirical comparisons of sGaBP with other GaBP variants in the literature as well as with the CG solver. Our empirical work will be summarized using two quantities, that is the number of iterations required by a specified method to converge and, if relevant, the KL distance of the posterior distributions to the true marginal distributions. All quantities are summarized using boxplots , the blue boxplots representing sGaBP and the red boxplots the method it is being compared with. Each figure corresponds to a set of zero-diagonal spectral radii which is indicated on the x-axis. For every zero-diagonal spectral radius indicated on the x-axis, we generate 100 data structures each consisting of a precision matrix and potential vector. We use the method described in "Appendix 2" to regulate the zero-diagonal spectral radius of the precision matrix to the appropriate value. We then apply sGaBP and the method it is being compared with on these data structures. With the exception of the CG solver all other methods require the specification of hyper-parameter(s). We initialize these methods by finding the value(s) of the hyper-parameter(s) yielding the fastest convergence through a line (grid) search in increments of 0.01. We refer to sGaBP (for instance), initialized with the optimal hyper-parameter determined through the line search, as optimal sGaBP. Similar labels are used for the other methods.

We now have 100 data structures for every zero-diagonal spectral radius given on the x-axis of the figures. We apply optimal sGaBP and the (optimized) competitor and record the number of iterations required by each method to converge. The blue boxplot is constructed from the number of iterations required by sGaBP to converge and the red boxplot from the number of iterations required by the competitor. The KL distances are slightly more complicated since for each precision matrix we get multiple marginals. For each application of sGaBP (and its competitor), we determine the KL distance of all the posterior distributions to their respective marginals, a given data structure is represented by the mean of all these distances. Boxplots are then constructed in a similar way to those of the iterations. To account for differences in the scaling of quantities provided by different methods, it may be necessary to focus (or zoom in) on certain parts of a figure.

### 9.1 Relaxed Gaussian belief propagation

El-Kurdi et al. (2012a) illustrate the advantages of RGaBP on large ill-conditioned and weakly diagonally dominant inverse covariance matrices. RGaBP does not allow tuning of the

precision components and can therefore only be applied in settings where the precision components of ordinary GaBP converge. The relaxation is applied on the mean components by setting $z_i^{(n)} = \gamma(b_i + \sum_{j\in\mathcal{N}_j} V_{ji}^{(n)}) + (1-\gamma)q_i^{(n)}\mu_i^{(n-1)}$. Setting $\gamma = 1$ gives ordinary GaBP (similar to setting $\lambda = 0$ for sGaBP). Although El-Kurdi et al. (2012a) focus on relaxation factors ($\gamma > 1$), RGaBP can also be used to perform damping ($\gamma < 1$). There is an interesting relationship between RGaBP and sGaBP with regard to how posterior means are computed:

$$\text{RGaBP}: \mu_i^{(n)} = \gamma \frac{(b_i + \sum_{j\in\mathcal{N}_j} V_{ji}^{(n)})}{q_i^{(n)}} + (1-\gamma)\mu_i^{(n-1)} \tag{24}$$

$$\text{sGaBP}: \mu_i^{(n)} = \frac{q_i^{(n)}}{\lambda + q_i^{(n)}} \frac{b_i + \sum_{j\in\mathcal{N}_j} V_{ji}^{(n)}}{q_i^{(n)}} + \frac{\lambda}{\lambda + q_i^{(n)}}\mu_i^{(n-1)}. \tag{25}$$

We see that $\gamma = 1 - \frac{\lambda}{\lambda + q_i^{(n)}}$. In contrast to RGaBP, sGaBP computes adaptive damping/relaxation factors using the tuning parameter $\lambda$ and the posterior precisions. In particular, we see that relaxation, $\gamma > 1$, and damping, $\gamma < 1$, correspond to negative and positive $\lambda$, respectively. This would imply that there is a role to play for negative $\lambda$. Part of our comparison is to give an indication of when to use relaxation versus damping. It is also worthwhile to emphasize that the posterior precisions provided by RGaBP are the posterior precisions provided by ordinary GaBP. Another important contribution in this regard is to provide empirical evidence that sGaBP can provide posterior precisions closer to the true marginal precisions when compared to ordinary GaBP.

In Fig. 3, the convergence speed of optimal sGaBP and optimal RGaBP are compared. For smaller zero-diagonal spectral radii, the methods are very comparable with sGaBP holding a slight advantage. As the zero-diagonal spectral radius approaches 1.5, the convergence speed of RGaBP starts to destabilize. When considering the boxplot corresponding to a zero-diagonal spectral radius of 1.5 we see that sGaBP can converge up to 16 times faster than RGaBP. It is also worthwhile to note that outliers were suppressed in these boxplots.

In Fig. 4, the KL distances of optimal sGaBP and optimal RGaBP are compared. In the simulations, sGaBP provided far more accurate posterior distributions. The simulations provide evidence, even in cases where the optimal convergence speeds are comparable, that it is better to use sGaBP instead of RGaBP, since sGaBP provides posterior precisions closer to the true marginal precisions.

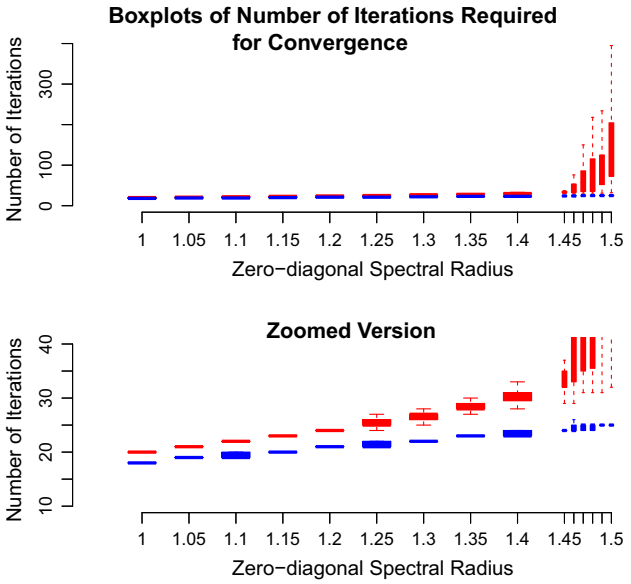


**Fig. 3** Comparison of the convergence speed of optimal sGaBP and optimal RGaBP over different zero-diagonal spectral radii. sGaBP outperformed RGaBP in these simulations, the relative convergence speed of RGaBP tending to decrease as the zero-diagonal spectral radius increases. (Color figure online)

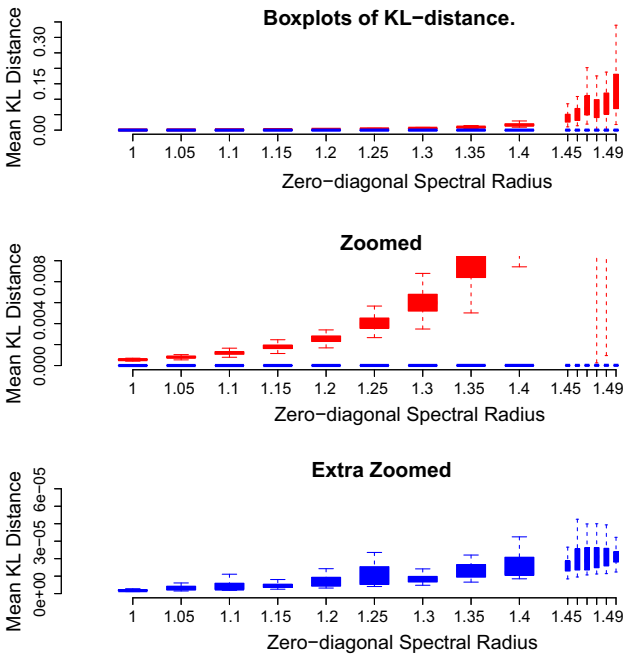


**Fig. 4** This is similar to Fig. 3, but the *boxplots* now represent the mean KL distance between the posterior marginals (provided by each method) and the true marginals. In these simulations, sGaBP provided more accurate approximations to the true marginals

An interesting sub-plot is the role of relaxation versus damping in the acceleration of GaBP. Relaxation corresponds to $\gamma > 1$, or negative $\lambda$, while damping occurs when $\gamma < 1$, or positive $\lambda$. The zero-diagonal spectral radius of a pre-

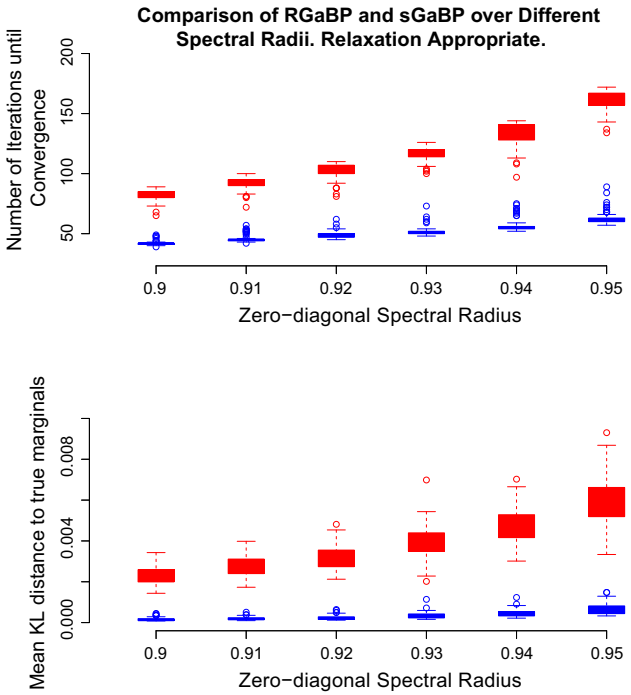


**Fig. 5** This is similar to Figs. 3 and 4, however comparisons were made over different zero-diagonal spectral radii. In contrast to the previous figures, the smallest eigenvalue was used to regulate the zero-diagonal spectral radius. In these simulations, the optimal relaxation factor is greater than one and this corresponds to negative $\lambda$ in the case of sGaBP. In these simulations, sGaBP converged faster and provided more accurate posterior marginals

cision matrix can be determined by one of two quantities, these being either the largest or the smallest eigenvalue of the precision matrix. In our simulations, we found that optimal convergence occurs with relaxation factors when the zero-diagonal spectral radius is determined by the smallest eigenvalue and otherwise damping. This indicates that relaxation can only be applied when the spectral radius is less than one, because if the zero-diagonal spectral radius is at least one and caused by the smallest eigenvalue the (standardized) precision matrix will either be singular or negative definite. Figure 5 is constructed by considering zero-diagonal spectral radii less than one and determined by the smallest eigenvalue of the precision matrix. Each application of optimal sGaBP and optimal RGaBP involved the use of relaxation factors. In terms of performance, we can make similar observations to those made on Figs. 3 and 4. In these simulations optimal sGaBP outperforms optimal rGaBP, both in terms of convergence speed and KL distances, with the relative performance improving as the zero-diagonal spectral radius approaches one. One can argue that the comparisons made in Fig. 5 are more relevant than the others made in this section since, as the name suggests, the focus of RGaBP is on relaxation factors.

Another method proposed in the literature to improve on the convergence behaviour of GaBP is based on the principle

**Algorithm 4** Compressed Inner-Loop Convergence Fix.

1. Provide $\mathbf{S}: p \times p$, $\boldsymbol{b}: p \times 1$, $\lambda$, $m$, $\epsilon$ and $s$ as inputs to the algorithm. Here, we wish to solve $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$ where $\mathbf{S}$ is positive definite and symmetric. The parameters $\lambda$, $m$ , $\epsilon$ and $s$ denote the degree of diagonal loading, the maximum number of iterations allowed, the tolerance used to define convergence and the damping factor, respectively.
2. Initiate $\mathbf{Q}^{(0)} = \text{diag}(1, 1, \ldots, 1)$, $\mathbf{V}^{(0)} = \text{diag}(b_1, b_2, \ldots, b_p)$ and $\boldsymbol{\mu}^{(-1)} = \mathbf{0}$.
3. Set Err = Inf and $n = 0$.
4. while Err $> \epsilon$
   (a) Compute $q_i^{(n)} = 1 + \lambda + \sum_{j \in \mathcal{N}_i} Q_{ji}^{(n)}$ and $z_i^{(n)} = V_{ii} + \sum_{j \in \mathcal{N}_i} V_{ji}^{(n)}$ for $i = 1, 2, \ldots, p$.
   (b) Set $\mu_i^{(n)} = \mu_i^{(n-1)} + \frac{z_i^{(n)}}{q_i^{(n)}}$ for $i = 1, 2, \ldots, p$.
   (c) For all $i \neq j$ set $Q_{ij}^{(n+1)} = \frac{-S_{ij}^2}{q_i^{(n)} - Q_{ji}^{(n)}}$ and $V_{ij}^{(n+1)} = \frac{Q_{ij}^{(n+1)}}{S_{ij}}(z_i^{(n)} - V_{ji}^{(n)})$.
   (d) Set $e_i^{(n+1)} = b_i - \sum_j S_{ij}\mu_j^{(n)}$, Err $= \sqrt{\frac{\sum_k (\mu_k^{(n)} - \mu_k^{(n-1)})^2}{\sum_k (\mu_k^{(n)})^2}}$, $V_{ii}^{(n+1)} = s \times V_{ii}^{(n)} + (1 - s)e_i^{(n+1)}$ and increment $n$.
   (e) If $m = n$ break.
5. End.

of message damping (Malioutov et al. 2006). As is mentioned by Malioutov et al. (2006), we found in our simulations that the convergence/divergence of the precision components is independent of the degree of damping applied. Furthermore, when the precisions do converge, we found that the degree of damping does not influence the actual converged posterior precisions. We also observed that RGaBP tends to outperform the message damping approach, based on optimal comparisons, and therefore did not include this in our empirical comparisons.

### 9.2 Compressed inner-loop convergence fix

The convergence fix (CF) method has been proposed in the literature as a method of solving arbitrary symmetric positive definite linear systems with GaBP (Johnson et al. 2009). The basic idea is to solve systems of the form $(\mathbf{S} + \Gamma)\boldsymbol{\mu}^{(n+1)} = \boldsymbol{b} + \Gamma\boldsymbol{\mu}^{(n)}$ using ordinary GaBP. Johnson et al. (2009) show that if $\mathbf{S} + \Gamma$ is walk-summable, CF will converge and provide the correct solution to the system $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$. We restrict our focus to the case where $\Gamma = \lambda\mathbf{I}$. Johnson et al. (2009) make a quick reference to a compressed inner-loop version of CF where each application of GaBP is limited to one iteration. Johnson et al. (2009) report that compressed inner-loop CF can be more efficient than the original method, but may require damping on the adjustment of the potential vector. The closeness of the compressed CF variant to sGaBP depends heavily on the interpretation of the description in the literature. Johnson et al. (2009) do not prove conver-

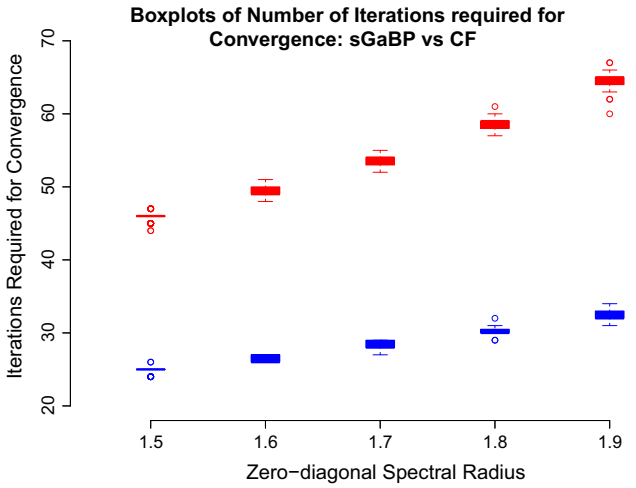


**Fig. 6** Illustration of the iterations required for convergence of optimal sGaBP (*blue*) and optimal CF (*red*). Both methods are relatively stable, however sGaBP converged faster in the simulations. The relative performance of sGaBP seems to improve with growth in the zero-diagonal spectral radius. (Color figure online)

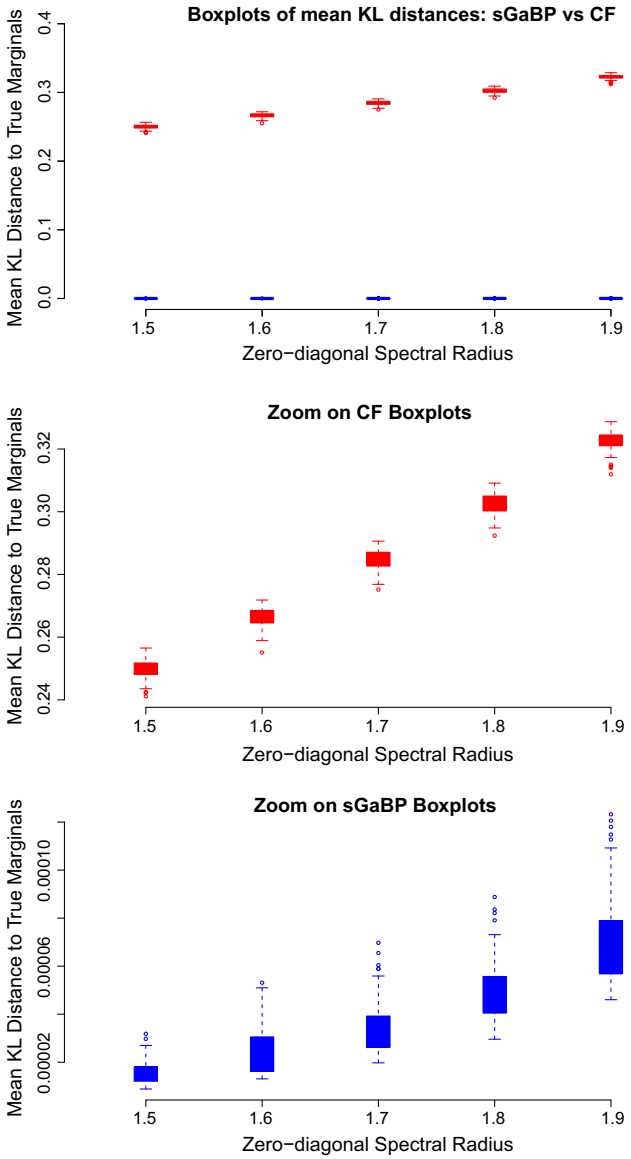


**Fig. 7** Illustration of the accuracy of the posterior distributions of optimal sGaBP (*blue*) and optimal CF (*red*) to the true marginals. In the simulations, sGaBP provided more accurate approximations. The poor performance of CF is due to the high values of the diagonal loadings it requires to converge optimally. (Color figure online)

gence of compressed CF and do not consider the potential usefulness of the diagonal loadings of the precision matrix in the tuning of the posterior precisions. We could not find any reference to compressed CF in the source code provided by Bickson (2008) and we formed our interpretation here-of by considering the source code provided for the original CF method along with the description by Johnson et al. (2009). We give this interpretation in Algorithm 4. We now compare our interpretation of compressed CF to sGaBP.

The visual summaries of the iterations required for convergence and the KL distances are given in Figs. 6 and 7, respectively. In the simulations, sGaBP outperformed CF in terms of convergence speed. Both methods were relatively stable in terms of the number of iterations required for convergence. The performance of CF in terms of KL distances to the true marginals was poor relative to the performance of sGaBP. In our simulations, we found that the degree of diagonal loadings required by CF to converge optimally was substantially higher than the tuning parameter required by optimal sGaBP. These simulations provide empirical evidence that sGaBP should be used instead of our interpretation of CF, both in terms of convergence speed and accuracy of the posterior distributions.

### 9.3 Conjugate gradient

One of the attractive properties of GaBP as a solver of large and sparse systems of linear equations lies in distributive computing. In general, BP algorithms are well suited to distributive implementation, under synchronous message scheduling, since no communication is required between nodes not linked in the graph. Like GaBP the CG method is a solver of linear systems and can be applied in distributive settings. A description of the CG solver can be found in Shewchuk (1994). Unlike GaBP, CG is guaranteed to converge for all symmetric and positive definite linear systems. Furthermore, CG is guaranteed to converge in at most $p$ iterations where $p$ is the number of variables in the system. This causes sGaBP to compare unfavourably with CG in small linear systems and hence our focus will be on systems with a large number of variables. In practice, the CG method converges much faster than $p$ iterations and the convergence becomes faster for linear systems with a smaller conditioning

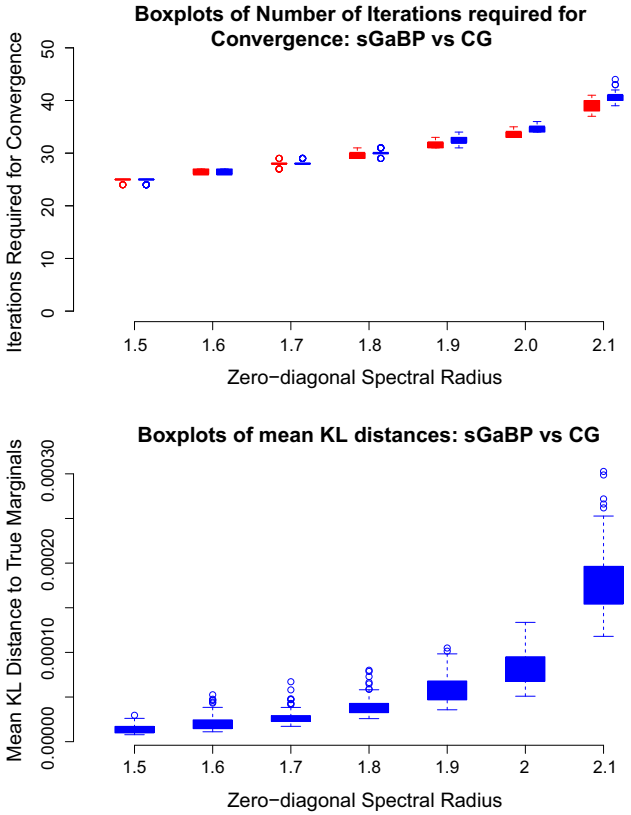


**Fig. 8** Comparison of the conjugate gradient solver with sGaBP. The CG method does not give approximations to the marginal precisions, but we do include the mean KL distances for sGaBP. In our simulations, these methods are comparable in terms of iterations required for convergence. The *boxplots* corresponding to each method are reasonably stable. The CG *boxplots* have a slight advantage for larger zero-diagonal spectral radii. Note that for each zero-diagonal spectral radius the *boxplots* corresponding to sGaBP and CG were plotted adjacent to each other

number. One of the contributions of this paper is a message-passing scheme which guarantees convergence and therefore we include a comparison with the CG method. We note here that the sGaBP and CG methods come from different areas of mathematics, those being approximate inference and linear algebra, respectively. The main advantage of the CG method is that it does not require any regularization, while sGaBP provides approximate precisions.

We now compare CG with optimal sGaBP in linear systems with 700 variables. The results are given in Fig. 8. In these simulations, we see that both methods are quite stable and very comparable, although CG has a small advantage in the simulations involving larger zero-diagonal spectral radii. The bottom plot of Fig. 8 shows the mean KL distances obtained for sGaBP. We see that these distances are small and therefore the posterior precisions can be useful as approximations of the true marginal precisions.

There are strategies which can be used to accelerate sGaBP. One approach would be to consider asynchronous message passing. The main drawback with this strategy is the loss of distributive applicability. Another approach is to use multiple tuning parameters, that is one tuning parameter for each node. This will not only improve on convergence speed, but could also be used to obtain (even) more accurate approximations to the marginal precisions. The disadvantage is that the complexity of deciding on the amount of tuning is amplified. Another interesting strategy is to increase the dimension of nodes, that is assigning more than one variable to each node. The difficulty here is deciding on which variables to cluster together in nodes and that communication between higher dimensional nodes is computationally more expensive. In certain situations, we found that the GDH can improve on optimal sGaBP in terms of convergence speed. It is also possible to extend GDH to allow for multiple tuning parameters which (hopefully) will accelerate convergence. The main problem surrounding GDH is the specification of the step size.

There are strategies to accelerate CG as well, the most prominent being that of preconditioning. Consider solving the system $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$. The idea behind preconditioning is to select a matrix $\mathbf{P}$, solve $\mathbf{PSP}'\tilde{\boldsymbol{\mu}} = \mathbf{P}\boldsymbol{b}$ and transform back to the original system using $\boldsymbol{\mu} = \mathbf{P}'\tilde{\boldsymbol{\mu}}$. The matrix $\mathbf{P}$ should be selected such that the conditioning number of $\mathbf{PSP}'$ is smaller than that of $\mathbf{S}$ and the computational cost of computing $\mathbf{PSP}'$ must be low. Here, we wish to emphasize that sGaBP can also benefit substantially from this type of preconditioning, even more so because it makes the selection of tuning parameters easier. The major loss is in terms of the accuracy of the posterior precisions as approximates for the marginal precisions. This is because the posterior precisions now approximate the marginal precisions of $\mathbf{PSP}'$ and transformation back to $\mathbf{S}$ cannot (directly) be done without knowing the off-diagonal entries of $[\mathbf{PSP}']^{-1}$. Finding a method to sensibly transform the approximate precisions back to the original scale will be very rewarding.

## 10 Concluding remarks and further research

We proposed an adjusted BP method on general MGs to address some of the problems underlying general BP. We took this high-level approach and applied it to a Gaussian MG, this type of BP is referred to as Gaussian belief propagation. We showed that sGaBP (our variant of GaBP) will always converge, with sufficient regularization, and showed how to compute posterior distributions to preserve the posterior means as exact marginal means. We provided empirical evidence that the posterior precisions provided by sGaBP are better approximations of the true marginal precisions when compared to two other variants of GaBP where hyper-parameters were initialized to yield the fastest convergence. This seems to indicate that our high-level approach should

be investigated in other MGs and perhaps also other graph structures (such as cluster graphs). Within the GaBP context, there are some questions that should be the subject of further research. The use of asynchronous message updates needs attention. The ranges of $\lambda$ which guarantee convergence need to be specified and work needs to be done on methods seeking the value of $\lambda$ which yields the fastest convergence. Some theoretical bounds on the proximity of the posterior precisions to the marginal precisions at the value of $\lambda$ corresponding to the fastest convergence would be useful. Further improvements on convergence and the accuracy of posterior distributions can be obtained through the use of multiple regularization parameters and this should be investigated. Another natural extension of our work is a generalization to multivariate nodes. Another interesting prospect is considering other loss functions in Eq. 5. For instances setting $\boldsymbol{\mu}_i^{(n-1)} = \mathbf{0}$ and using $q = 2$ relates to ridge regression under the linear model while $q = 1$ relates to the Lasso. The sGaBP implementation of the Lasso can be done without loss of the conjugacy of the messages by majorization of the $L_1$-norm by a $L_2$-norm. To ensure general convergence in the case of the Lasso it may be necessary to use a penalty of the form $\tau||\mathbf{x}_i||_1 + \frac{\lambda}{2}||\mathbf{x}_i - \boldsymbol{\mu}_i^{(n-1)}||_2^2$ while applying a working-set method on the messages being updated. The latter is necessary since majorization of an absolute with a quadratic function is not possible at the origin.

**Acknowledgements** 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 author and are not necessarily to be attributed to the NRF.

## Appendix 1: Proofs

### Proof of Theorem 1

*Proof* The proof is contained in the following list.

1. From 1, all the precision components are negative at stage $n$, hence $Q_{ij}^{(n+1)} = \frac{-S_{ij}^2}{1+\lambda+\sum_{t\in\mathcal{N}_i/j} Q_{ti}^{(n)}} = \frac{-S_{ij}^2}{1+\lambda-\sum_{t\in\mathcal{N}_i/j}|Q_{ti}^{(n)}|}$. From 3 we have that $\sum_{t\in\mathcal{N}_i/j}|Q_{ti}^{(n)}| \leq \sum_{t\in\mathcal{N}_i}|Q_{ti}^{(n)}| < \delta_i^{(n)} < 1+\lambda$ and $1+\lambda-\sum_{t\in\mathcal{N}_i/j}|Q_{ti}^{(n)}| > 0$ from which 1 follows for iteration $n+1$.
2. $|Q_{ij}^{(n+1)}| = \frac{S_{ij}^2}{1+\lambda-\sum_{t\in\mathcal{N}_i/j}|Q_{ti}^{(n)}|} \geq \frac{S_{ij}^2}{1+\lambda-\sum_{t\in\mathcal{N}_i/j}|Q_{ti}^{(n-1)}|} = |Q_{ij}^{(n)}|$ since $|Q_{ti}^{(n)}| > |Q_{ti}^{(n-1)}|$, $t \in \mathcal{N}_i$ from 2 for iteration $n$ and hence 2 is also true for $n+1$.
3. $\delta_i^{(n+1)} = \sum_{t\in\mathcal{N}_i}|Q_{ti}^{(n+1)}| = \sum_{t\in\mathcal{N}_i}\frac{S_{ti}^2}{1+\lambda-\delta_t^{(n)}+|Q_{it}^{(n)}|} \leq \sum_{t\in\mathcal{N}_i}\frac{S_{ti}^2}{1+\lambda-\delta_t+|Q_{it}^{(n)}|} \leq \delta_i < 1+\lambda$ by 4 and therefore 3 is true for $n+1$.
4. From the above we have, $\sum_{t\in\mathcal{N}_i}\frac{S_{ti}^2}{1+\lambda-\delta_t+|Q_{it}^{(n+1)}|} \leq \sum_{t\in\mathcal{N}_i}\frac{S_{ti}^2}{1+\lambda-\delta_t+|Q_{it}^{(n)}|} \leq \delta_i$, hence 4 holds for $n+1$.

□

### Proof of Theorem 2

Let $\mathbf{S}$ be a symmetric, positive definite matrix with diagonal entries equal to 1, and let its entries be denoted by $S_{ij}$. Values $Q_{ij}(\lambda)$ are characterized by the system

$$Q_{ij} = Q_{ij}(\lambda) = -\frac{S_{ij}^2}{1+\lambda+\sum_{t\in\mathcal{N}_i/j} Q_{ti}(\lambda)}, \qquad 1 \leq i, j \in \mathcal{N}_i.$$

We are particularly interested in the behaviour as $\lambda \to \infty$. A consequence of Theorem 1 is that $\lim_{\lambda\to\infty} Q_{ij}(\lambda) = 0$. For convenience, set $\delta = \lambda^{-1}$, so that $\delta \to 0$. The system can be rewritten as

$$Q_{ij}\Big(-\delta\sum_{t\in\mathcal{N}_i/j} Q_{ti} - 1 - \delta\Big) - \delta S_{ij}^2 = 0, \qquad 1 \leq i, j \in \mathcal{N}_i.$$

Note that

$$\frac{\partial}{\partial Q_{kl}}\left(Q_{ij}\Big(-\delta\sum_{t\in\mathcal{N}_i/j} Q_{ti} - 1 - \delta\Big) - \delta S_{ij}^2\right) = \begin{cases} -\delta\sum_{t\neq i,j} Q_{ti} - 1 - \delta & (k,l) = (i,j), \\ -\delta Q_{ij} & l = i, k \in \mathcal{N}_i/j, \\ 0 & \text{otherwise.} \end{cases}$$

As $\delta \to 0$, we see that the Jacobian of the system tends to a negative identity matrix, so in particular it is invertible. This means that the $Q_{ij}$ are analytic functions of $\delta$ if $\delta$ is in a suitable neighbourhood of 0. Consequently, the $Q_{ij}$ have power series expansions in $\delta$:

$$Q_{ij} = a_{ij}\delta + b_{ij}\delta^2 + \cdots$$

Plugging this back into the system, we see that in fact $a_{ij} = -S_{ij}^2$, so we have

$$C_{ij} = \frac{Q_{ij}}{S_{ij}} = -S_{ij}\delta + \mathcal{O}(\delta^2).$$

Consider again the matrix $\mathbf{L}$ given in (20). Let $l = p^2 - p$, we now define a $l \times p$ matrix $\mathbf{G} = \begin{bmatrix}\mathbf{g}_1 & \mathbf{g}_2 & \ldots & \mathbf{g}_p\end{bmatrix}$. The vector $\mathbf{g}_i$ has entries 1 in positions $(p-1)(i-1)+1, \ldots, (p-1)i$. It can be shown that

$$\mathbf{L} : p^2 \times p^2 = \begin{bmatrix} \mathbf{L}_{11} : l \times l & \frac{\lambda}{p-2}\mathbf{L}_{11}\mathbf{G} \\ \frac{1}{\lambda}\mathbf{G}'\mathbf{L}_{22} & \mathbf{L}_{22} : p \times p \end{bmatrix}, \tag{26}$$

with the understanding that $\lambda > 0$. Let,

$$\mathbf{D} = \begin{bmatrix} \mathbf{I} : l \times l & \mathbf{0} : l \times p \\ \mathbf{0} : p \times l & \lambda[\mathbf{I} : p \times p] \end{bmatrix}, \tag{27}$$

and set $\tilde{\mathbf{L}} = \mathbf{DLD}^{-1}$. It is easy to see that,

$$\tilde{\mathbf{L}} : p^2 \times p^2 = \begin{bmatrix} \mathbf{L}_{11} : l \times l & \frac{1}{p-2}\mathbf{L}_{11}\mathbf{G} \\ \mathbf{L}_{22}\mathbf{G}' & \mathbf{L}_{22} : p \times p \end{bmatrix}, \tag{28}$$

and that $\mathbf{L}$ and $\tilde{\mathbf{L}}$ will have the same eigenvalues. As a first step, we show that the eigenvalues of $\tilde{\mathbf{L}}$ are all clustered around 0 and 1 as $\delta \to 0$. We have already discussed the construction of $\mathbf{L}_{11}$ from the elements $C_{ij}$. Using the fact that $C_{ij} = -\delta S_{ij} + \mathcal{O}(\delta^2)$ we see that $\mathbf{L}_{11} = \delta\mathbf{A} + \mathcal{O}(\delta^2)$ where $\mathbf{A}$ does not depend on $\lambda$ and $\mathcal{O}(\delta^2)$ is of a suitable dimension each entry being $\mathcal{O}(\delta^2)$. The matrix $\mathbf{A}$ is constructed exactly as $\mathbf{L}_{11}$; however, $-S_{ij}$'s are used instead of $-C_{ij}$'s. As discussed, the matrix $\mathbf{L}_{22}$ is diagonal with entries $\frac{\lambda}{1+\lambda+\sum_{t\neq i} Q_{ti}} = \frac{\lambda}{1+\lambda+\mathcal{O}(\delta)} = 1 - \delta + \mathcal{O}(\delta^2)$ and therefore $\mathbf{L}_{22} = \mathbf{I} - \delta\mathbf{I} + \mathcal{O}(\delta^2)$. We now consider the matrix,

$$\tilde{\mathbf{L}} = \begin{bmatrix} \delta\mathbf{A} & \frac{\delta}{p-2}\mathbf{AG} \\ (1-\delta)\mathbf{G}' & (1-\delta)\mathbf{I} \end{bmatrix} + \mathcal{O}(\delta^2), \tag{29}$$

and the following Lemma.

**Lemma 1** *Let* $\mathbf{M}$ *be a square matrix, and let* $c$ *be a positive constant that satisfies* $c > \|\mathbf{M}\|_\infty$ *(*$\|\mathbf{M}\|_\infty$ *is the* $\infty$*-norm of* $\mathbf{M}$*, which can be obtained by calculating the row sums of the absolute values of entries in* $\mathbf{M}$ *and taking the maximum of these sums). For every* $x$ *with* $|x| \geq c$*, the matrices* $x\mathbf{I} - \mathbf{M}$ *and* $\mathbf{I} - \frac{1}{x}\mathbf{M}$ *are invertible, and the entries of* $(\mathbf{I} - \frac{1}{x}\mathbf{M})^{-1}$ *are bounded by constants that only depend on* $c$ *and* $\mathbf{M}$*.*

*Proof* The invertibility follows directly from the fact that the matrix $x\mathbf{I} - \mathbf{M}$ is strictly diagonally dominant by our assumptions. For the second statement, let $|\mathbf{M}|$ be obtained from $\mathbf{M}$ by replacing all entries by their absolute values. Note that $|\mathbf{M}|$ has the same $\infty$-norm as $\mathbf{M}$. Clearly, the entries of

$$\left(\mathbf{I} - \frac{1}{x}\mathbf{M}\right)^{-1} = \sum_{j=0}^{\infty} x^{-j}\mathbf{M}^j$$

are bounded by the entries of

$$\left(\mathbf{I} - \frac{1}{c}|\mathbf{M}|\right)^{-1} = \sum_{j=0}^{\infty} c^{-j}|\mathbf{M}|^j,$$

which readily proves the desired statement. □

**Lemma 2** *There exists a constant* $K > 0$ *such that for sufficiently small* $\delta$*, 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 first $||\mathbf{L}_{11}||_\infty = ||\delta\mathbf{A} + \mathcal{O}(\delta^2)||_\infty \leq \delta||\mathbf{A}||_\infty + \mathcal{O}(\delta^2)$. If we choose $K$ large enough (e.g. $K \geq \|\mathbf{A}\|_\infty + 1$), then the matrix $x\mathbf{I} - \mathbf{L}_{11} = x\mathbf{I} - \delta\mathbf{A} + \mathcal{O}(\delta^2)$ is invertible by the previous lemma for sufficiently small $\delta$, and the entries of $(\mathbf{I} - \frac{1}{x}\mathbf{L}_{11})^{-1}$ are bounded by absolute constants. Now, we use the Schur complement on 29:

$$\begin{aligned} &\det(x\mathbf{I} - \tilde{\mathbf{L}}) = \det(x\mathbf{I} - \mathbf{L}_{11}) \\ &\quad \times \det\left(x\mathbf{I} - \mathbf{L}_{22} - \frac{1}{p-2}\mathbf{L}_{22}\mathbf{G}'(x\mathbf{I} - \mathbf{L}_{11})^{-1}\mathbf{L}_{11}\mathbf{G}\right). \end{aligned} \tag{30}$$

It remains to show that the second determinant is not equal to 0. We rewrite the matrix as follows:

$$\begin{aligned} &x\mathbf{I} - \mathbf{L}_{22} - \frac{1}{p-2}\mathbf{L}_{22}\mathbf{G}'(x\mathbf{I} - \mathbf{L}_{11})^{-1}\mathbf{L}_{11}\mathbf{G} \\ &\quad = (x-1)\mathbf{I} - (\mathbf{L}_{22} - \mathbf{I}) \\ &\qquad - \frac{1}{x(p-2)}\mathbf{L}_{22}\mathbf{G}'\left(\mathbf{I} - \frac{1}{x}\mathbf{L}_{11}\right)^{-1}\mathbf{L}_{11}\mathbf{G} \\ &\quad = (x-1)\mathbf{I} + \mathbf{H}_1 + \frac{1}{x}\mathbf{H}_2. \end{aligned} \tag{31}$$

Consider $(\mathbf{L}_{22} - \mathbf{I}) = -\delta\mathbf{I} + \mathcal{O}(\delta^2)$ and $||\mathbf{L}_{22} - \mathbf{I}||_\infty = || - \delta\mathbf{I} + \mathcal{O}(\delta^2)||_\infty \leq \delta||\mathbf{I}||_\infty + ||\mathcal{O}(\delta^2)||_\infty = \delta + \mathcal{O}(\delta^2)$. Therefore, $||\mathbf{H}_1||_\infty \leq \kappa_1\delta$ for a constant $\kappa_1$ and sufficiently small $\delta$. The entries of $(\mathbf{I} - \frac{1}{x}\mathbf{L}_{11})^{-1}$ are bounded by $(\mathbf{I} - \frac{1}{K\delta}|\mathbf{L}_{11}|)^{-1} = \mathbf{I} + \frac{|\mathbf{L}_{11}|}{K\delta} + \sum_{j=2}^{\infty}\frac{|\mathbf{L}_{11}|^j}{K\delta}$ for sufficiently small $\delta$ by Lemma 1. Since $\mathbf{L}_{11} = \delta\mathbf{A} + \mathcal{O}(\delta^2)$ we have that $(\mathbf{I} - \frac{1}{x}\mathbf{L}_{11})^{-1} = \mathbf{I} + \frac{|\mathbf{A}|}{K} + \mathcal{O}(\delta) = \mathcal{O}(1)$. Furthermore, $\mathbf{L}_{22} = \mathcal{O}(1)$ and $\mathbf{L}_{11} = \mathcal{O}(\delta)$ from which we have that $\mathbf{H}_2 = \mathcal{O}(\delta) + \mathcal{O}(\delta^2)$ and $||\mathbf{H}_2||_\infty \leq \kappa_2\delta$ for a constant $\kappa_2$ and sufficiently small $\delta$. If $|x| \geq \frac{1}{2}$, we find that

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

if $K$ is chosen large enough (greater than $\kappa_1 + 2\kappa_2$). If $|x| \leq \frac{1}{2}$, we get

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

if $K$ is chosen large enough and $\delta$ is sufficiently small. In either case, we can apply the previous lemma to see that the matrix in (31) is in fact invertible. □

Now, we focus on the eigenvalues that are close to 1, setting $x = 1 - \delta t$ for some $t$ with $|t| \leq K$. Returning to (30), we observe that $x\mathbf{I} - \mathbf{L}_{11}$ is invertible for sufficiently small $\delta$, again by Lemma 1. Hence, we consider the second matrix:

$$x\mathbf{I} - \mathbf{L}_{22} - \frac{1}{p-2}\mathbf{L}_{22}\mathbf{G}'(x\mathbf{I} - \mathbf{L}_{11})^{-1}\mathbf{L}_{11}\mathbf{G} = (1-t)\delta\mathbf{I} - \frac{\delta}{p-2}\mathbf{G}'\mathbf{A}\mathbf{G} + \mathcal{O}(\delta^2).$$

The entries of the matrix hidden by the $\mathcal{O}(\delta^2)$ term are in fact analytic in $\delta$ and $t$, since we proved earlier that the entries of $\tilde{\mathbf{L}}$ are analytic functions. We can take out a factor $\delta$ to be left with the equation

$$\det\left((1-t)\mathbf{I} - \frac{1}{p-2}\mathbf{G}'\mathbf{A}\mathbf{G} + \mathbf{M}\right) = 0, \tag{32}$$

where the matrix $\mathbf{M}$ has entries that are analytic functions of $\delta$ and $t$ (if $\delta$ is restricted to a sufficiently small neighbourhood of 0 and $|t| \leq K$). Moreover, $\mathbf{M} = \mathcal{O}(\delta)$. As $\delta \to 0$, we obtain (up to a change of variable $1-t = u$) the characteristic equation of the matrix $\frac{1}{p-2}\mathbf{G}'\mathbf{A}\mathbf{G}$ (we will show later that this matrix is in fact equal to $\mathbf{I} - \mathbf{S}$). Its $p$ solutions (counted with multiplicity) give rise to $p$ branches $t_1(\delta), t_2(\delta), \ldots, t_p(\delta)$ that solve the implicit equation (32). In the same way, we can treat the "small" eigenvalues that are close to 0. We set $x = \delta t$ for some $t$ with $|t| \leq K$, and use the Schur complement with respect to the other diagonal block:

$$\det(x\mathbf{I} - \tilde{\mathbf{L}}) = \det(x\mathbf{I} - \mathbf{L}_{22}) \times \det\left(x\mathbf{I} - \mathbf{L}_{11} - \frac{1}{p-2}\mathbf{L}_{11}\mathbf{G}(x\mathbf{I} - \mathbf{L}_{22})^{-1}\mathbf{L}_{22}\mathbf{G}'\right). \tag{33}$$

Since

$$x\mathbf{I} - \mathbf{L}_{22} = (x - 1 + \delta)\mathbf{I} + \mathcal{O}(\delta^2) = -\mathbf{I} + \mathcal{O}(\delta),$$

this matrix is invertible for sufficiently small $\delta$, again by Lemma 1. Moreover, we have

$$x\mathbf{I} - \mathbf{L}_{11} - \mathbf{L}_{11}\mathbf{G}(x\mathbf{I} - \mathbf{L}_{22})^{-1}\mathbf{L}_{22}\mathbf{G}' = \delta(t\mathbf{I} - \mathbf{A} + \frac{1}{p-2}\mathbf{A}\mathbf{G}\mathbf{G}') + \mathcal{O}(\delta^2),$$

so we can repeat the argument for the "large" eigenvalues. We obtain $p^2 - p$ branches $\bar{t}_1(\delta), \bar{t}_2(\delta), \ldots, \bar{t}_{p^2-p}(\delta)$ that correspond to the eigenvalues of $\mathbf{A} - \frac{1}{p-2}\mathbf{A}\mathbf{G}\mathbf{G}'$.

Returning to the large eigenvalues, we consider the product $\mathbf{G}'\mathbf{A}\mathbf{G}$. The matrix $\mathbf{A}$ is constructed by taking the first $l$ rows and columns of $\mathbf{L}$ and replacing the $C_{ij}$ elements with $-S_{ij}$. The rows, $(p-1)(j-1)+1, \ldots, (p-1)j$, correspond to messages received by node $j$ (in order) and hence $\mathbf{g}_j$ contains ones at the rows corresponding to messages received by node $j$ and zeros otherwise. Consider a row corresponding to a message from node $i$ to node $j$ which requires communication from other nodes (excluding $j$) to node $i$, this row will therefore contain $-S_{ij}$ where $\mathbf{g}_i$ is equal to 1, except the element corresponding to the message from $j$ to $i$. Now, $\mathbf{A}\mathbf{g}_i$ will be equal to $-(p-2)S_{ij}$ in the rows corresponding to the message from $i$ to $j$ and zero otherwise. The vector $\mathbf{g}_j$ contains references to rows corresponding to messages received by node $j$ and since there is only one message from $i$ to $j$ the nonzero elements of $\mathbf{g}_j$ will overlap with the nonzero elements of $\mathbf{A}\mathbf{g}_i$ at one element and hence $\mathbf{g}_j'\mathbf{A}\mathbf{g}_i = -(p-2)S_{ij}$ for $j \neq i$. Furthermore, since there is no message from node $i$ to node $i$ we have that $\mathbf{g}_i'\mathbf{A}\mathbf{g}_i = 0$. We see that $\frac{1}{p-2}\mathbf{G}'\mathbf{A}\mathbf{G} = \mathbf{I} - \mathbf{S}$. Equation (32) becomes

$$\det\left(\mathbf{S} - t\mathbf{I} + \mathbf{H}\right) = 0.$$

Since $\mathbf{S}$ is symmetric, it is diagonalizable. There exists an orthogonal matrix $\mathbf{U}$ such that $\mathbf{U}^{-1}\mathbf{S}\mathbf{U} = \mathbf{D}$ is a diagonal matrix. We have

$$\det\left(\mathbf{S} - t\mathbf{I} + \mathbf{H}\right) = \det\left(\mathbf{U}^{-1}(\mathbf{S} - t\mathbf{I} + \mathbf{H})\mathbf{U}\right) = \det\left(\mathbf{D} - t\mathbf{I} + \mathbf{U}^{-1}\mathbf{H}\mathbf{U}\right).$$

Recall that $\mathbf{H} = \mathcal{O}(\delta)$, uniformly in $t$ (for $|t| \leq K$), so we also have $\mathbf{U}^{-1}\mathbf{H}\mathbf{U} = \mathcal{O}(\delta)$. Let $\kappa$ be a constant such that $\|\mathbf{U}^{-1}\mathbf{H}\mathbf{U}\|_\infty \leq \kappa\delta$ (for sufficiently small $\delta$ and $|t| \leq K$). If

$$\det\left(\mathbf{D} - t\mathbf{I} + \mathbf{U}^{-1}\mathbf{H}\mathbf{U}\right) = 0,$$

then we must have $|t - d_{ii}| \leq \kappa\delta$ for one of the diagonal entries $d_{ii}$ of $\mathbf{D}$, for otherwise the matrix $\mathbf{D} - t\mathbf{I} + \mathbf{U}^{-1}\mathbf{H}\mathbf{U}$ will be strictly diagonally dominant and thus invertible. The diagonal entries of $\mathbf{D}$ are the eigenvalues $\sigma_1, \sigma_2, \ldots, \sigma_p$ of $\mathbf{S}$, so it follows that $t = \sigma_i + \mathcal{O}(\delta)$.

We can deal with the small eigenvalues in the same way, it only remains to determine the entries of $\mathbf{A} - \frac{1}{p-2}\mathbf{A}\mathbf{G}'\mathbf{G}$ (thereby verifying that this matrix is also symmetric and thus diagonalizable). It is easy to verify that $\mathbf{G}'\mathbf{G}$ is a block diagonal matrix where the blocks are of dimension $(p-1)\times(p-1)$ with all entries equal to one, in fact $\mathbf{G}'\mathbf{G} = [\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_p]$ where $\mathbf{B}_i$ is $\mathbf{g}_i$ appended $p-1$ times as columns. Consider a row in $\mathbf{A}$ corresponding to a message from $i$ to $j$, say $\mathbf{a}_{ij}'$, we have already verified that this row contains $-S_{ij}$ where $\mathbf{g}_i$ is one, except for the 1 corresponding to the message from $j$ to $i$. Now, $\mathbf{a}_{ij}'\mathbf{G}'\mathbf{G}$ will contain nonzero elements in $\mathbf{a}_{ij}'\mathbf{B}_i$ and these will all equal $-(p-2)S_{ij}$. A row, $\mathbf{b}_{ij}$, of $\frac{1}{p-2}\mathbf{A}\mathbf{G}'\mathbf{G}$ corresponding to a message from $i$ to $j$ will contain $-S_{ij}$ where $\mathbf{g}_i$ equals one (even for the message from $j$ to $i$). Hence, $\mathbf{a}_{ij}$ and $\mathbf{b}_{ij}$ will be identical except for the element corresponding to the message from $i$ to $j$, where $\mathbf{a}_{ij}$ is zero and $\mathbf{b}_{ij}$ is

$-S_{ij}$. Hence, a row of $\mathbf{A} - \frac{1}{p-2}\mathbf{AG}'\mathbf{G}$ corresponding to a message from $i$ to $j$ will have one element (at the message from $j$ to $i$) equal to $S_{ij}$ and the rest are zero. Furthermore, the row corresponding to the message from $j$ to $i$ will have $S_{ji} = S_{ij}$ as an element in the position corresponding to the message from $i$ to $j$. Hence, $\mathbf{A} - \frac{1}{p-2}\mathbf{AG}'\mathbf{G}$ is symmetric. In conclusion, the eigenvalues of $\tilde{\mathbf{L}}$ are

- $1-\sigma_i\delta+\mathcal{O}(\delta^2)$, where $\sigma_1, \sigma_2, \ldots, \sigma_p$ are the eigenvalues of $\mathbf{S}$, and
- $\pm S_{ij}\delta + \mathcal{O}(\delta^2), \quad 1 \le i < j \le p.$

In particular, the largest eigenvalue of $\tilde{\mathbf{L}}$ is connected to the least eigenvalue $\sigma_{\min}$ of $\mathbf{S}$ by

$$\max\{\mu \,:\, \mu \text{ is an eigenvalue of } \tilde{\mathbf{L}}\} = 1 - \sigma_{\min}\delta + \mathcal{O}(\delta^2).$$

Since $\mathbf{S}$ is a positive definite matrix, we know that $\sigma_{\min} > 0$. It follows that

$$\max\{\mu \,:\, \mu \text{ is an eigenvalue of } \tilde{\mathbf{L}}\} < 1$$

for sufficiently small $\delta$.

**Proof of Theorem 3**

**Theorem 3** *Under the assumption that sGaBP converges, with precision matrix* $\mathbf{S}$ *and potential vector* $\boldsymbol{b}$ *as inputs, and setting* $\boldsymbol{\mu}$ *equal to the converged posterior means, we have that* $\mathbf{S}\boldsymbol{\mu} = \boldsymbol{b}$.

*Proof* In Theorems 1 and 2, we proved convergence to the following stationary equations:

$$\begin{aligned} Q_{ij} &= \frac{-S_{ij}^2}{\lambda + q_i - Q_{ji}} \\ V_{ij} &= \frac{Q_{ij}}{S_{ij}}(\lambda\mu_i + z_i - V_{ji}) \\ \mu_i &= \frac{\lambda\mu_i + z_i}{\lambda + q_i}, \end{aligned} \tag{34}$$

for all $i$ and $j \in \mathcal{N}_i$. Furthermore, $q_i = 1 + \sum_{t\in\mathcal{N}_i} Q_{ti}$ and $z_i = b_i + \sum_{t\in\mathcal{N}_i} V_{ti}$. Using (34):

$$z_i + \lambda\mu_i = V_{ji} + \frac{S_{ij}}{Q_{ij}}V_{ij},$$

for all $i$ and $j \in \mathcal{N}_i$. For any $k \in \mathcal{N}_i$ we can write

$$S_{ki}(z_i + \lambda\mu_i) = S_{ki}V_{ji} + S_{ki}\frac{S_{ij}}{Q_{ij}}V_{ij} = S_{ki}V_{ki} + \frac{S_{ik}^2}{Q_{ik}}V_{ik}.$$

Furthermore, since $\frac{S_{ik}^2}{Q_{ik}} = Q_{ki} - (q_i + \lambda_i)$, we have

$$S_{ki}(z_i + \lambda\mu_i) = S_{ki}V_{ki} + (Q_{ki} - (q_i + \lambda_i))V_{ik}. \tag{35}$$

Dividing (35) by $q_i + \lambda$ gives

$$S_{ki}\mu_i = \frac{1}{q_i + \lambda}S_{ki}V_{ki} + \frac{Q_{ki}}{q_i + \lambda}V_{ik} - V_{ik}. \tag{36}$$

Further simplification can be done by noting that $S_{ki}V_{ki} = Q_{ki}(\lambda u_k + z_k - V_{ik})$, and substituting into (36):

$$\begin{aligned} S_{ki}\mu_i &= \frac{Q_{ki}}{q_i + \lambda}(\lambda\mu_k + z_k - V_{ik}) + \frac{Q_{ki}}{q_i + \lambda}V_{ik} - V_{ik} \\ &= \frac{Q_{ki}}{q_i + \lambda}(\lambda\mu_k + z_k) - \frac{Q_{ki}}{q_i + \lambda}V_{ik} + \frac{Q_{ki}}{q_i + \lambda}V_{ik} - V_{ik} \\ &= \frac{Q_{ki}}{q_i + \lambda}(\lambda\mu_k + z_k) - V_{ik}. \end{aligned} \tag{37}$$

Summing $S_{ki}\mu_i$ over $i$, substituting (37) for $i \in \mathcal{N}_k$, gives

$$\sum_{i\in\mathcal{N}_k\cup k} S_{ki}\mu_i = \mu_k + (\lambda\mu_k + z_k)\sum_{i\in\mathcal{N}_k}\frac{Q_{ki}}{q_i + \lambda} - \sum_{i\in\mathcal{N}_k} V_{ik}. \tag{38}$$

Since $Q_{ik}(\lambda + q_i - Q_{ki}) = -S_{ik}^2 = -S_{ki}^2 = Q_{ki}(\lambda + q_k - Q_{ik})$,

$$\frac{Q_{ik}}{\lambda + q_k} = \frac{Q_{ki}}{\lambda + q_i}. \tag{39}$$

Substituting (39) into (38):

$$\begin{aligned} \sum_{i\in\mathcal{N}_k\cup k} S_{ki}\mu_i &= \mu_k + (\lambda\mu_k + z_k)\sum_{i\in\mathcal{N}_k}\frac{Q_{ik}}{\lambda + q_k} - \sum_{i\in\mathcal{N}_k} V_{ik} \\ &= \mu_k + \mu_k\sum_{i\in\mathcal{N}_k} Q_{ik} - \sum_{i\in\mathcal{N}_k} V_{ik} \\ &= \mu_k + \mu_k(q_k - 1) - (z_k - b_k) \\ &= q_k\mu_k - z_k + b_k. \end{aligned} \tag{40}$$

Finally, since $\mu_k = \frac{\lambda\mu_k + z_k}{\lambda + q_k}$, we have that $\mu_k(\lambda + q_k) = \lambda\mu_k + z_k$ and

$$q_k\mu_k = z_k. \tag{41}$$

Substituting (41) into (40) completes the proof. □

## Appendix 2: Simulation information

### Simulation scheme

We briefly describe the simulation scheme (Bach et al. 2011) used in our empirical work. This simulation scheme also relates GaBP to least squares estimation under the linear model. In order to apply sGaBP, we need to generate a positive definite symmetric precision matrix ($\mathbf{S}$) and potential vector ($\boldsymbol{b}$). One way to do this is to generate a data structure according to the linear model with $n$ observations and $p$ inputs. This yields a design matrix $\mathbf{X} : n \times p$ and a response vector $\mathbf{y} : n \times 1$. We then form the sample correlation matrix $\mathbf{S} = \mathbf{X}'\mathbf{X}$ where we assume the columns of $\mathbf{X}$ are standardized to have zero mean and unity $L_2$ norm. We assume the same for $\mathbf{y}$ and form the sample correlation vector $\boldsymbol{b} = \mathbf{X}'\mathbf{y}$. As long as $n > p$, $\mathbf{S}$ will be positive definite and we can use this as a valid precision matrix for the application of sGaBP. Explanatory variables are generated from $N(\mathbf{0}, \frac{1}{n}\mathbf{I}_p)$, where $n$ is the number of observations and $p$ the number of explanatory variables. The generated explanatory variables are stored in $\mathbf{X}$. Coefficients are generated, $\beta_i \sim$ idd $N(0, 1)$, and sparsity is introduced by randomly selecting half of the $\beta_i$'s and setting these equal to zero. Observations of the response are generated, $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\epsilon_i \sim$ idd $N(0, \sigma^2)$ and $\sigma^2 = 0.01 \times \frac{||\mathbf{X}\boldsymbol{\beta}||^2}{n}$. All variables were standardized to have zero mean and unit $L_2$ norm and we form $\mathbf{S} = \mathbf{X}'\mathbf{X}$ and $\boldsymbol{b} = \mathbf{X}'\mathbf{y}$. We used $n = p$ throughout the empirical section. The matrix, $\mathbf{S}$, was ensured to be positive by regulating its zero-diagonal spectral radius using the method discussed in the next section. We then apply sGaBP on a multivariate Gaussian with precision matrix $\mathbf{S}$ and potential vector $\boldsymbol{b}$.

### Regulating the zero-diagonal spectral radius

Suppose we have a precision matrix $\mathbf{S} : p \times p$ normalized to have ones along its diagonal. Set $\mathbf{R} = \mathbf{I}_p - \mathbf{S}$ and let $\tau_i : i = 1, 2, \ldots, p$ be the eigenvalues of $\mathbf{R}$. Suppose we wish to find a new precision matrix, $\mathbf{S}^*$, with zero-diagonal spectral radius set to a specified value (say $\alpha$). First, we compute the eigen decomposition of $\mathbf{R}$,

$$\mathbf{R} = \mathbf{V}\mathbf{D}\mathbf{V}',$$

where $\mathbf{V}'\mathbf{V} = \mathbf{V}\mathbf{V}' = \mathbf{I}$ and $\mathbf{D} = \text{diag}(\tau_1, \ldots, \tau_p)$. We form a new diagonal matrix, $\mathbf{D}^* = \frac{\alpha}{\tilde{\rho}(\mathbf{S})}\mathbf{D}$, and set $\mathbf{R}^* = \mathbf{V}\mathbf{D}^*\mathbf{V}'$, $\mathbf{S}^* = \mathbf{I} - \mathbf{R}^*$. We now show that $\mathbf{S}^*$ is a valid precision matrix with diagonal entries equal to one if $\alpha < 1$. Since $\mathbf{S}$ is a normalized precision matrix, the diagonal of $\mathbf{R}$ will contain only zeros, the same is true for $\mathbf{R}^*$ (being a scalar multiple of $\mathbf{R}$) and therefore the diagonal of $\mathbf{S}^*$ will contain only ones. Suppose $\lambda_i, i = 1, 2, \ldots, p$ and $\lambda_i^*, i = 1, 2, \ldots, p$ represent the eigenvalues of $\mathbf{S}$ and $\mathbf{S}^*$, respectively. The following holds:

$$\begin{aligned}\lambda_i^* &= 1 - \frac{\alpha}{\tilde{\rho}(\mathbf{S})}(1 - \lambda_i) \\ &= 1 - \alpha\frac{1 - \lambda_i}{\max_j\{|1 - \lambda_j|\}} \\ &= 1 - \alpha \times \text{sign}(1 - \lambda_i) \times \frac{|1 - \lambda_i|}{\max_j\{|1 - \lambda_j|\}}.\end{aligned}$$

Since $\frac{|1-\lambda_i|}{\max_j\{|1-\lambda_j|\}} \leq 1$, we have that $1 - \alpha \leq \lambda_i^* \leq 1 + \alpha$. If $0 \leq \alpha < 1$, then $\mathbf{S}^*$ will be positive definite. In our simulations, when $\alpha > 1$, we used a check to ensure that $\mathbf{S}^*$ is positive definite.

## References


Aji, S., McEliece, R.: The generalized distributive law. IEEE Trans. Inform. Theory **46**, 325–343 (2000)

Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Convex optimization with sparsity-inducing norms. In: Sra, S., Nowozin, S., Wright, J. (eds.) Optimization for Machine Learning. MIT Press, Cambridge (2011)

Bickson, D.: Gaussian Belief Propagation: Theory and Application, PhD thesis. The Hebrew University of Jerusalem (2008)

Chandrasekaran, V., Johnson, J.K., Willsky, A.S.: Estimation in Gaussian graphical models using tractable subgraphs: a walk-sum analysis. IEEE Trans. Signal Process. **56**, 1916–1930 (2008)

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Annal. Stat. **32**(2), 407–499 (2004)

El-Kurdi, Y., Giannacopoulos, D., Gross, W.J.: Relaxed Gaussian belief propagation. In: Proceedings of the 2012 IEEE International Symposium on Information Theory (2012a)

El-Kurdi, Y., Gross, W.J., Giannacopoulos, D.: Efficient implementation of Gaussian belief propagation solver for large sparse diagonally dominant linear systems. IEEE Trans. Magn. **48**, 471–474 (2012b)

Frey, B., Kschischang, F.: Probability propagation and iterative decoding. In: Proceedings of the 34th Annual Allerton Conference on Communication, Control, and Computing, Allerton House, Monticello (1996)

Gallager, R.G.: Low-Density Parity-Check Codes. MIT Press, Cambridge (1963)

Guo, Q., Huang, D.: EM-based joint channel estimation and detection for frequency selective channels using Gaussian message passing. IEEE Trans. Signal Process. **59**, 4030–4035 (2011)

Guo, Q., Li, P.: LMMSE turbo equalization based on factor graphs. IEEE J. Sel. Areas Commun. **26**, 311–319 (2008)

Johnson, J.K., Bickson, D., Dolev, D.: Fixing convergence of Gaussian belief propagation. In: International Symposium on Information Theory (ISIT), Seoul (2009)

Koller, D., Friedman, N.: Probabilistic Graphical Models Principles and Techniques. MIT Press, Cambridge (2009)

Lauritzen, S,, Spiegelhalter, D. Local computations with probabilities on graphical structures and their application to expert systems. J. R. Stat. Soc. B **50**,157–224 (1988)

Liu, Y., Chandrasekaran, V., Anandkumar, A., Willsky, A.S.: Feedback message passing for inference in Gaussian graphical models. IEEE Trans. Signal Process. **60**(8), 4135–4150 (2012)

Malioutov, D.M., Johnson, J.K., Willsky, A.S.: Walk-sums and belief propagation in gaussian graphical models. J. Mach. Learn. Res. **7**, 2031–2064 (2006)

Montanari, A., Prabhakar, B., Tse, D.: Belief propagation based multi-user detection. In: IEEE Information Theory Workshop, Punta del Este, Uruguay (2006)

Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)

Seeger, M.W., Wipf, D.P.: Variational Bayesian inference techniques. IEEE Signal Process. Mag. **27**, 81–91 (2010)

Shachter, R.: Probabilistic inference and influence diagrams. Oper. Res. **36**, 589–605 (1988)

Shafer, G., Shenoy, P.: Probability propagation. Ann. Mat. Art. Intell. **2**, 327–352 (1990)

Shental, O., Siegel, P.H., Wolf. J.K., Bickson, D., Dolev, D.: Gaussian belief propagation solver for systems of linear equations. In: IEEE International Symposium on Informational Theory (ISIT), pp 1863–1867 (2008)

Shewchuk, J.R.: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. School of Computer Science. Carnegie Mellon University, Pittsburgh, pp. 15213 (1994)

Su, Q., Wu, Y.: On convergence conditions of Gaussian belief propagation. IEEE Int. Trans. Signal Process. **63**, 1144–1155 (2015)

Weiss, Y.: Correctness of local probability in graphical models with loops. Neural Comput. **12**, 1–41 (2000)

Weiss, Y., Freeman, W.T.: Correctness of belief propagation in Gaussian graphical models of arbitrary topology. Neural Comput. **13**(10), 2173–2200 (2001)