

Fixing Convergence of Gaussian Belief Propagation

Jason K. Johnson

Center for Nonlinear Studies/T-4
Los Alamos National Laboratory
Los Alamos, NM 87545
Email: jasonj@lanl.gov

Danny Bickson

IBM Haifa Research Lab
Mount Carmel, Haifa 31905, Israel
Email: dannybi@il.ibm.com

Danny Dolev

School of Computer Science and Engineering
Hebrew University of Jerusalem
Jerusalem 91904, Israel
Email: dolev@cs.huji.ac.il

Abstract—Gaussian belief propagation (GaBP) is an iterative message-passing algorithm for inference in Gaussian graphical models. It is known that when GaBP converges it converges to the correct MAP estimate of the Gaussian random vector and simple sufficient conditions for its convergence have been established.

In this paper we develop a double-loop algorithm for forcing convergence of GaBP. Our method computes the correct MAP estimate even in cases where standard GaBP would not have converged. We further extend this construction to compute least-squares solutions of over-constrained linear systems. We believe that our construction has numerous applications, since the GaBP algorithm is linked to solution of linear systems of equations, which is a fundamental problem in computer science and engineering. As a case study, we discuss the linear detection problem. We show that using our new construction, we are able to force convergence of Montanari’s linear detection algorithm, in cases where it would originally fail. As a consequence, we are able to increase significantly the number of users that can transmit concurrently.

I. INTRODUCTION

The Gaussian belief propagation algorithm is an efficient distributed message-passing algorithm for inference over a Gaussian graphical model. GaBP is also linked to the canonical problem of solving systems of linear equations [1]–[3], one of the fundamental problems in computer science and engineering, which explains the large number of algorithm variants and applications. For example, the GaBP algorithm is applied for signal processing [3]–[7], multiuser detection [8], [9], linear programming [10], ranking in social networks [11], support vector machines [12] *etc.* Furthermore, it was recently shown that some existing algorithms are specific instances of the GaBP algorithm, including Consensus propagation [13], local probability propagation [14], multiuser detection [8], Quadratic Min-Sum algorithm [1], Turbo decoding with Gaussian densities [15] and others. Two general sufficient conditions for convergence of GaBP in loopy graphs are known: diagonal-dominance [16] and walk-summability [17]. See also numerous studies in specific settings [1], [8], [13]–[18].

In this work, we propose a novel construction that fixes the convergence of the GaBP algorithm, for any Gaussian model with positive-definite information matrix (inverse covariance matrix), even when the currently known sufficient convergence conditions do not hold. We prove that our construction converges to the correct solution. Furthermore, we consider how this method may be used to solve for the least-squares solution of general linear systems. As a specific application,

we discuss Montanari’s multiuser detection algorithm [8]. By using our construction we are able to show convergence in practical CDMA settings, where the original algorithm did not converge, supporting a significantly higher number of users on each cell.

This paper is organized as follows. Section II outlines the problem model. Section III gives a brief introduction to the GaBP algorithm. Section IV describes our novel double-loop construction for positive definite matrices. Section V extends the construction for computing least-squares solution of general linear systems. We provide experimental results of deploying our construction in the linear detection context in Section VI. We conclude in Section VII.

II. PROBLEM SETTING

We wish to compute the *maximum a posteriori* (MAP) estimate of a random vector x with Gaussian distribution (after conditioning on measurements):

$$p(x) \propto \exp\left\{-\frac{1}{2}x^T J x + h^T x\right\}, \quad (1)$$

where $J \succ 0$ is a symmetric positive definite matrix (the information matrix) and h is the potential vector. This problem is equivalent to solving $Jx = h$ for x given (h, J) or to solve the convex quadratic optimization problem:

$$\text{minimize } f(x) \triangleq \frac{1}{2}x^T J x - h^T x. \quad (2)$$

We may assume without loss of generality (by rescaling variables) that J is normalized to have unit-diagonal, that is, $J \triangleq I - R$ with R having zeros along its diagonal. The off-diagonal entries of R then correspond to *partial correlation coefficients* [19]. Thus, the fill pattern of R (and J) reflects the Markov structure of the Gaussian distribution. That is, $p(x)$ is Markov with respect to the graph with edges $\mathcal{G} = \{(i, j) | r_{i,j} \neq 0\}$.

If the model $J = I - R$ is *walk-summable* [17], [18], such that the spectral radius of $|R| = (|r_{ij}|)$ is less than one ($\rho(|R|) < 1$), then the method of GaBP may be used to solve this problem. We note that the walk-summable condition implies $I - R$ is positive definite. An equivalent characterization of the walk-summable condition is that $I - |R|$ is positive definite.

#	Stage	Operation
1.	<i>Initialize</i>	Set $\alpha_{ij} = 0$ and $\beta_{ij} = 0, \forall (i, j) \in \mathcal{G}$
2.	<i>Iterate</i>	For all $(i, j) \in \mathcal{G}$ $\alpha_{i \setminus j} = J_{ii} + \sum_{k \in \mathbb{N}(i) \setminus j} \alpha_{ki}$ $\beta_{i \setminus j} = h_i + \sum_{k \in \mathbb{N}(i) \setminus j} \beta_{ki}$ $\alpha_{ij} = -J_{ij}^2 \alpha_{i \setminus j}^{-1}$ $\beta_{ij} = -J_{ij} \alpha_{i \setminus j}^{-1} \beta_{i \setminus j}$ end
3.	<i>Check</i>	If α 's and β 's have converged, continue to #4. Else, return to #2.
4.	<i>Infer</i>	$\hat{K}_i = (J_{ii} + \sum_{k \in \mathbb{N}(i)} \alpha_{ki})^{-1}$ $\hat{\mu}_i = \hat{K}_i (h_i + \sum_{k \in \mathbb{N}(i)} \beta_{ki})$.
5.	<i>Output</i>	$x_i^* = \hat{\mu}_i, \forall i$.

TABLE I

COMPUTING $x^* = \arg \max_x \exp(-\frac{1}{2}x^T J x + h^T x)$ VIA GaBP.

III. GAUSSIAN BELIEF PROPAGATION

The Gaussian belief propagation algorithm is an efficient distributed message-passing algorithm for inference over a Gaussian graphical model. Given the Gaussian density function (1) or objective function (2), we are interested in computing the MAP assignment:

$$x^* = \arg \max_x p(x) = \arg \min_x f(x)$$

The density $p(x)$ specifies a graphical model with respect to the graph G of the inverse covariance matrix J , with edge potentials ('compatibility functions') ψ_{ij} and self-potentials ('evidence') ψ_i . These graph potentials provide a pairwise factorization of the Gaussian distribution $p(x) \propto \prod_{i=1}^n \psi_i(x_i) \prod_{\{i,j\} \in G} \psi_{ij}(x_i, x_j)$, with $\psi_{ij}(x_i, x_j) \triangleq \exp(-x_i J_{ij} x_j)$, and $\psi_i(x_i) \triangleq \exp(-\frac{1}{2} J_{ii} x_i^2 + h_i x_i)$. Then, we would like to calculate the marginal densities, which must also be Gaussian,

$$p(x_i) \sim \mathcal{N}(\mu_i = (J^{-1}h)_i, K_i \triangleq (J^{-1})_{ii}),$$

where μ_i and K_i are the marginal mean and variance, respectively. The GaBP update rules are summarized in Table I. We write $\mathbb{N}(i)$ to denote the set of neighbors of node i in G .

It is known that if GaBP converges, it results in the exact MAP estimate x^* , although the variance estimates \hat{K}_i computed by GaBP are only approximations to the correct variances [16]. The walk-summable condition guarantees that GaBP converges [17], generalizing the stricter condition [16] that J is diagonally dominant (i.e., $|J_{ii}| > \sum_{j \neq i} |J_{ij}|, \forall i$). An upper bound on convergence speed is given in [10].

IV. OUR CONSTRUCTION

This current paper presents a method to solve non-walksummable models, where $J = I - R$ is positive definite but $\rho(|R|) \geq 1$, using GaBP. There are two key ideas: (1) using diagonal loading to create a perturbed model $J' = J + \Gamma$ which is walk-summable (such that the GaBP may be used to solve $J'x = h$ for any h) and (2) using this perturbed model J'

and convergent GaBP algorithm as a *preconditioner* in a simple iterative method to solve the original non-walksummable model.

A. Diagonal Loading

We may always obtain a walk-summable model by *diagonal loading*. This is useful as we can then solve a related system of equations efficiently using Gaussian belief propagation. For example, given a non-walk-summable model $J = I - R$ we obtain a related walk-summable model $J_\gamma = J + \gamma I$ that is walk-summable for large enough values of γ :

Lemma 1: Let $J = I - R$ and $J' \triangleq J + \gamma I = (1 + \gamma)I - R$. Let $\gamma > \gamma^*$ where

$$\gamma^* = \rho(|R|) - 1. \quad (3)$$

Then, J' is walk-summable and GaBP based on J' converges.

Proof. We normalize $J' = (1 + \gamma)I - R$ to obtain $J'_{\text{norm}} = I - R'$ with $R' = (1 + \gamma)^{-1}R$, which is walk-summable if and only if $\rho(|R'|) < 1$. Using $\rho(|R'|) = (1 + \gamma)^{-1}\rho(|R|)$ we obtain the condition $(1 + \gamma)^{-1}\rho(|R|) < 1$, which is equivalent to $\gamma > \rho(|R|) - 1$. \diamond

It is also possible to achieve the same effect by adding a general diagonal matrix Γ to obtain a walk-summable model. For example, for all $\Gamma > \Gamma^*$ where $\gamma_{ii}^* = J_{ii} - \sum_{j \neq i} |J_{ij}|$ it holds that $J + \Gamma$ is diagonally-dominant and hence walk-summable (see [17]). More generally, we could allow Γ to be any symmetric positive-definite matrix satisfying the condition $I + \Gamma \succ |R|$. However, only the case of diagonal matrices is explored in this present paper.

B. Iterative Correction Method

Now we may use the diagonally-loaded model $J' = J + \Gamma$ to solve $Jx = h$ for any value of $\Gamma \geq 0$. The basic idea here is to use the diagonally-loaded matrix $J' = J + \Gamma$ as a *preconditioner* for solving the $Jx = h$ using the iterative method:

$$\hat{x}^{(t+1)} = (J + \Gamma)^{-1}(h + \Gamma \hat{x}^{(t)}) \quad (4)$$

Note that the effect of adding positive Γ is to reduce the size of the scaling factor $(J + \Gamma)^{-1}$ but we compensate for this damping effect by adding a feedback term $\Gamma \hat{x}$ to the input h . Each step of this iterative method may also be interpreted as solving the following convex quadratic optimization problem based on the objective $f(x)$ from (2):

$$\hat{x}^{(t+1)} = \arg \min_x \left\{ f(x) + \frac{1}{2}(x - x^{(t)})^T \Gamma (x - x^{(t)}) \right\} \quad (5)$$

This is basically a regularized version of Newton's method to minimize $f(x)$ where we regularize the step-size at each iteration. Typically, this regularization is used to ensure positive-definiteness of the Hessian matrix when Newton's method is used to optimize a non-convex function. We instead use it to ensure that $J + \Gamma$ is walk-summable, so that the update step can be computed via Gaussian belief propagation. Intuitively, this will always move us closer to the correct solution, but slowly if Γ is large. It is simple to demonstrate the following:

Lemma 2: Let $J \succ 0$ and $\Gamma \succeq 0$. Then, $\hat{x}^{(t)}$ defined by (4) converges to $x^* = J^{-1}h$ for all initializations $\hat{x}^{(0)}$.

Comment. The proof is given for a general (non-diagonal) $\Gamma \succeq 0$. For diagonal matrices, this is equivalent to requiring $\Gamma_{ii} \geq 0$ for $i = 1, \dots, n$.

Proof. First, we note that there is only one possible fixed-point of the algorithm and this is $x^* = J^{-1}h$. Suppose \bar{x} is a fixed point: $\bar{x} = (J + \Gamma)^{-1}(h + \Gamma\bar{x})$. Hence, $(J + \Gamma)\bar{x} = h + \Gamma\bar{x}$ and $J\bar{x} = h$. For non-singular J , we must then have $\bar{x} = J^{-1}h$. Next, we show that the method converges. Let $e^{(t)} = \hat{x}^{(t)} - x^*$ denote the error of the k -th estimate. The error dynamics are then $e^{(t+1)} = (J + \Gamma)^{-1}\Gamma e^{(t)}$. Thus, $e^{(t)} = ((J + \Gamma)^{-1}\Gamma)^k e^{(0)}$ and the error converges to zero if and only if $\rho((J + \Gamma)^{-1}\Gamma) < 1$, or equivalently $\rho(H) < 1$ where $H = (J + \Gamma)^{-1/2}\Gamma(J + \Gamma)^{-1/2} \succeq 0$ is a symmetric positive semi-definite matrix. Thus, the eigenvalues of H are non-negative and we must show that they are less than one. It is simple to check that if λ is an eigenvalue of H then $\frac{\lambda}{1-\lambda}$ is an eigenvalue of $\Gamma^{1/2}J^{-1}\Gamma^{1/2} \succeq 0$. This is seen as follows: $Hx = \lambda x$, $(J + \Gamma)^{-1}\Gamma y = \lambda y$ ($y = (J + \Gamma)^{-1/2}x$), $\Gamma y = \lambda(J + \Gamma)y$, $(1 - \lambda)\Gamma y = \lambda Jy$, $J^{-1}\Gamma y = \frac{\lambda}{1-\lambda}y$ and $\Gamma^{1/2}J^{-1}\Gamma^{1/2}z = \frac{\lambda}{1-\lambda}z$ ($z = \Gamma^{1/2}y$) [note that $\lambda \neq 1$, otherwise $Jy = 0$ contradicting $J \succ 0$]. Therefore $\frac{\lambda}{1-\lambda} \geq 0$ and $0 \leq \lambda < 1$. Then $\rho(H) < 1$, $e^{(t)} \rightarrow 0$ and $\hat{x}^{(t)} \rightarrow x^*$ completing the proof. \diamond

Now, provided we also require that $J' = J + \Gamma$ is walk-summable, we may compute $x^{(t+1)} = (J + \Gamma)^{-1}h^{(t+1)}$, where $h^{(t+1)} = h + \Gamma\hat{x}^{(t)}$, by performing Gaussian belief propagation to solve $J'x^{(t+1)} = h^{(t+1)}$. Thus, we obtain a double-loop method to solve $Jx = h$. The inner-loop performs GaBP and the outer-loop computes the next $h^{(t)}$. The overall procedure converges provided the number of iterations of GaBP in the inner-loop is made large enough to ensure a good solution to $J'x^{(t+1)} = h^{(t+1)}$. Alternatively, we may compress this double-loop procedure into a single-loop procedure by performing just one iteration of GaBP message-passing per iteration of the outer loop. Then it may become necessary to use the following damped update of $h^{(t)}$ with step size parameter $s \in (0, 1)$:

$$\begin{aligned} h^{(t+1)} &= (1-s)h^{(t)} + s(h + \Gamma\hat{x}^{(t)}) \\ &= h + \Gamma((1-s)\hat{x}^{(t-1)} + s\hat{x}^{(t)}) \end{aligned} \quad (6)$$

This single-loop method converges for sufficiently small values of s . In practice, we have found good convergence with $s = \frac{1}{2}$. This single-loop method can be more efficient than the double-loop method.

V. EXTENSION TO GENERAL LINEAR SYSTEMS

In this section, we efficiently extend the applicability of the proposed double-loop construction for a general linear system of equations (possibly over-constrained.) Given a full column rank matrix $\tilde{J} \in \mathbb{R}^{n \times k}$, $n \geq k$, and a shift vector \tilde{h} , we are interested in solving the least squares problem $\min_x \|\tilde{J}x - \tilde{h}\|_2^2$. The naive approach for using GaBP would be to take the information matrix $\tilde{J} \triangleq (\tilde{J}^T \tilde{J})$, and the shift vector $\tilde{h} \triangleq \tilde{J}^T \tilde{h}$.

Note that \tilde{J} is positive definite and we can use GaBP to solve it. The MAP solution is

$$x = \tilde{J}^{-1}\tilde{h} = (\tilde{J}^T \tilde{J})^{-1} \tilde{J}^T \tilde{h}, \quad (7)$$

which is the pseudo-inverse solution.

Note, that the above construction has two drawbacks: first, we need to explicitly compute \tilde{J} and \tilde{h} , and second, \tilde{J} may not be sparse in case the original matrix \tilde{J} is sparse. To overcome this problem, following [9], we construct a new symmetric data matrix $\bar{\tilde{J}}$ based on the arbitrary rectangular matrix $\tilde{J} \in \mathbb{R}^{n \times k}$

$$\bar{\tilde{J}} \triangleq \begin{pmatrix} I_{k \times k} & \tilde{J}^T \\ \tilde{J} & \mathbf{0}_{n \times n} \end{pmatrix} \in \mathbb{R}^{(k+n) \times (k+n)}.$$

Additionally, we define a new hidden variable vector $\tilde{x} \triangleq \{x^T, z^T\}^T \in \mathbb{R}^{(k+n)}$, where $x \in \mathbb{R}^k$ is the solution vector and $z \in \mathbb{R}^n$ is an auxiliary hidden vector, and a new shift vector $\bar{\tilde{h}} \triangleq \{0_{k \times 1}^T, h^T\}^T \in \mathbb{R}^{(k+n)}$.

Lemma 3: Solving $\bar{\tilde{x}} = \bar{\tilde{J}}^{-1}\bar{\tilde{h}}$ and taking the first k entries is identical to solving Eq. 7.

Proof. Is given in [9].

For applying our double-loop construction on the new system $(\bar{\tilde{h}}, \bar{\tilde{J}})$ to obtain the solution to Eq. (7), we need to confirm that the matrix $\bar{\tilde{J}}$ is positive definite. (See lemma 2). To this end, we add a diagonal weighting $-\gamma I$ to the lower right block:

$$\hat{\tilde{J}} \triangleq \begin{pmatrix} I_{k \times k} & \tilde{J}^T \\ \tilde{J} & -\gamma I_{n \times n} \end{pmatrix} \in \mathbb{R}^{(k+n) \times (k+n)}.$$

Then we rescale $\hat{\tilde{J}}$ to make it unit diagonal (to deal with the negative sign of the lower right block we use a complex Gaussian notation as done in [8]). It is clear for a large enough γ we are left with a walk-summable model, where the rescaled $\hat{\tilde{J}}$ is a hermitian positive definite matrix and $\rho(|\hat{\tilde{J}} - I|) < 1$. Now it is possible to use the double-loop technique to compute Eq. 7. Note that adding $-\gamma I$ to the lower right block of $\hat{\tilde{J}}$ is equivalent to adding γI into Eq. 7:

$$x = (\tilde{J}^T \tilde{J} + \gamma I)^{-1} \tilde{J}^T \tilde{h} \quad (8)$$

where γ can be interpreted as a regularization parameter.

VI. EXPERIMENTAL RESULTS

A. Linear detection in linear channels

Consider a discrete-time channel with a real input vector $x = \{x_1, \dots, x_K\}^T$ governed by an arbitrary prior distribution, P_x , and a corresponding real output vector $y = \{y_1, \dots, y_K\}^T = f\{x^T\} \in \mathbb{R}^K$. Here, the function $f\{\cdot\}$ denotes the channel transformation. By definition, linear detection compels the decision rule to be

$$\hat{x} = \Delta\{x^*\} = \Delta\{A^{-1}b\}, \quad (9)$$

where $b = y$ is the $K \times 1$ observation vector and the $K \times K$ matrix A is a positive-definite symmetric matrix approximating the channel transformation. The vector x^* is the solution (over \mathbb{R}) to $Ax = b$. Estimation is completed by adjusting the (inverse) matrix-vector product to the input

alphabet, dictated by P_x , accomplished by using a proper clipping function $\Delta\{\cdot\}$ (e.g., for binary signaling $\Delta\{\cdot\}$ is the sign function).

For example, linear channels, which appear extensively in many applications in communication and data storage systems, are characterized by the linear relation

$$y = f\{x\} = Cx + n,$$

where n is a $K \times 1$ additive noise vector and $C = S^T S$ is a positive-definite symmetric matrix, often known as the correlation matrix. The $N \times K$ matrix S describes the physical channel medium while the vector y corresponds to the output of a bank of filters matched to the physical channel S .

Assuming linear channels with AWGN with variance σ^2 as the ambient noise, the linear minimum mean-square error (MMSE) detector can be described by using $A = C + \sigma^2 I_K$, known to be optimal when the input distribution P_x is Gaussian. In general, linear detection is suboptimal because of its deterministic underlying mechanism (i.e., solving a given set of linear equations), in contrast to other estimation schemes, such as MAP or maximum likelihood, that emerge from an optimization criteria.

B. Montanari's iterative algorithm for computing the MMSE detector

Recent work by Montanari *et al.* [8] introduces an efficient iterative algorithm for computing the MMSE detector. Following this work, Bickson *et al.* showed that this algorithm is an instance of the GaBP algorithm [9].

In the current work, we apply our novel technique for forcing the convergence of Montanari's algorithm. To remind, Montanari's algorithm computes the MMSE solution

$$x = (C + \sigma^2 I_K)^{-1} y.$$

We use the following setting: given a random-spreading CDMA code with chip sequence length $n = 256$, and $k = 64$ users. We assume a diagonal AWGN with $\sigma^2 = 1$. Matlab code of our implementation is available on [20].

Using the above settings, we have drawn at random random-spreading CDMA matrix. Typically, the sufficient convergence conditions for the GaBP algorithm do not hold. For example, we have drawn at random a randomly-spread CDMA matrix with $\rho(|I_K - C^N|) = 4.24$, where C^N is a diagonally-normalized version of $(C + \sigma^2 I_K)$. Since $\rho(|I_K - C^N|) > 1$, the GaBP algorithm for multiuser detection is not guaranteed to converge.

Figure 1 shows that under the above settings, the GaBP algorithm indeed diverged. The x -axis represent iteration number, while the values of different x_i are plotted using different colors. This figure depicts well the fluctuating divergence behavior.

Next, we deployed our proposed construction and used a diagonal loading to force convergence. Figure 2 shows two different possible diagonal loadings. The x -axis shows the Newton step number, while the y -axis shows the residual. We experimented with two options of diagonal loading. In the

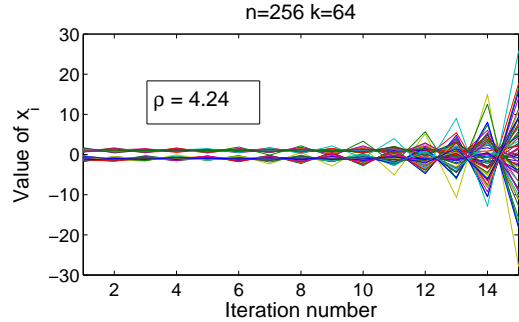


Fig. 1. Divergence of the GaBP algorithm for the multiuser detection problem, when $n = 256, k = 64$.

first, we forced the matrix to be diagonally-dominant (DD). In this case, the spectral radius $\rho = 0.188$. In the second case, the matrix was not DD, but the spectral radius was $\rho = 0.388$. Clearly, the Newton method converges faster when the spectral radius is larger. In both cases the inner iterations converged in five steps to an accuracy of 10^{-6} .

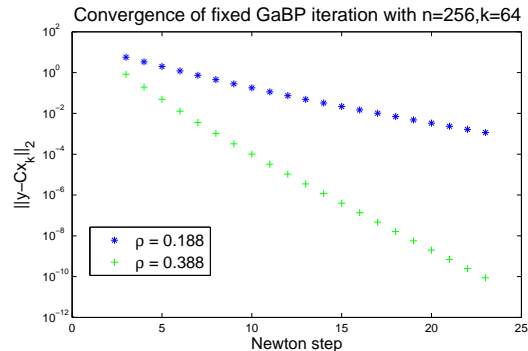


Fig. 2. Convergence of the fixed GaBP iteration under the same settings ($n = 256, k = 64$)

The tradeoff between the amount of diagonal weighting to the total convergence speed is shown in Figures 3,4. A CDMA multiuser detection problem is shown ($k = 128, n = 256$). Convergence threshold for the inner and outer loops where 10^{-6} and 10^{-3} . The x -axis present the amount of diagonal weighting normalized such that 1 is a diagonally-dominant matrix. y -axis represent the number of iterations. As expected, the outer loop number of iterations until convergence grows with the increase of γ . In contrast, the average number of inner loop iterations per Newton step (Figure 4) tends to decrease as γ increases. The total number of iterations (inner \times outer) represents the tradeoff between the inner and outer iterations and has a clear global minima.

VII. CONCLUSIONS AND FUTURE WORK

We have presented an iterative method based on Gaussian belief propagation which always converges to the correct

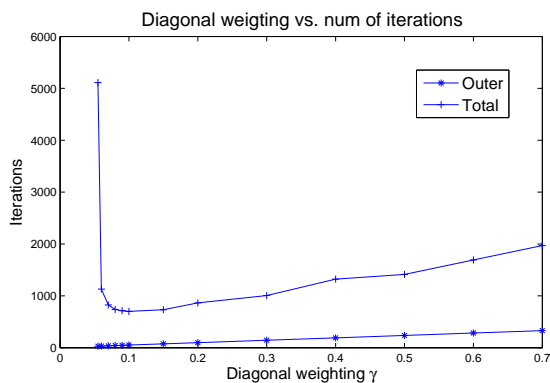


Fig. 3. Effect of diagonal weighting on outer loop convergence speed.

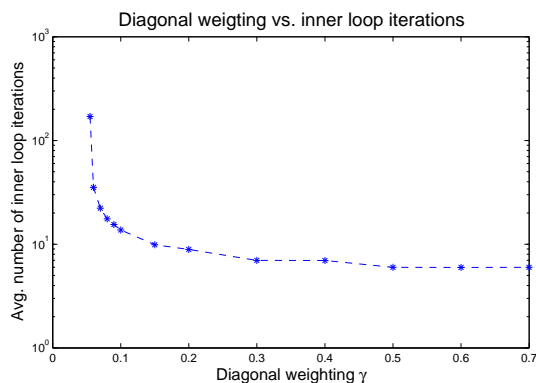


Fig. 4. Effect of diagonal weighting on inner loop convergence speed.

global solution, even in models where Gaussian belief propagation alone does not converge. Essentially, this involves adding a diagonal-loading term to force the model to become walk-summable such that GaBP converges in this modified model and adding a feedback mechanism that corrects the damping caused by the diagonal-loading term.

We believe that there are numerous applications for our construction in many domains, since GaBP is related to the solution of linear systems of equations. As an example, we discuss the case of multiuser detection. We gave a concrete example, where a state-of-the-art linear iterative algorithm for detection fails to converge. Using our construction we are able to force convergence for computing the correct MMSE detector.

There are a number of directions for further development. Most importantly, it would be very useful to develop a simple method to select Γ so as to optimize the rate of convergence of the overall method. In the double-loop method, it is seen that there is a trade-off in deciding how large Γ should be. For larger Γ (beyond the threshold of walk-summability) GaBP converges faster by accelerating the inner-loop of our algorithm. However, larger Γ will also make the outer-loop converge more slowly. Hence, we must somehow balance these competing objectives in choosing Γ . In the single-loop method, it would be useful to develop an adaptive method to optimize the step-size parameter s . Lastly, it may also prove useful

to exploit a more general class of perturbations beyond the diagonal-loading method used in this paper.

ACKNOWLEDGMENT

Danny Dolev is Incumbent of the Berthold Badler Chair in Computer Science. Danny Dolev was supported in part by the Israeli Science Foundation (ISF) Grant number 0397373.

REFERENCES

- [1] C. C. Moallemi and B. Van Roy, "Convergence of the min-sum algorithm for convex optimization," in *Proc. of the 45th Allerton Conference on Communication, Control and Computing*, Monticello, IL, September 2007.
- [2] O. Shental, D. Bickson, P. H. Siegel, J. K. Wolf, and D. Dolev, "Gaussian belief propagation solver for systems of linear equations," in *IEEE Int. Symp. on Inform. Theory (ISIT)*, Toronto, Canada, July 2008.
- [3] H. A. Loeliger, "Least squares and Kalman filtering on Forney graphs," in *Codes, Graphs, and Systems, (festschrift in honour of David Forney)*, R. E. Blahut and R. Koetter, Eds. Kluwer, 2002, pp. 113–135.
- [4] H. A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," in *Proceedings of the IEEE*, vol. 95, no. 6, 2007, pp. 1295–1322.
- [5] P. Vontobel and H. Loeliger, "On factor graphs and electrical networks," in *Mathematical Systems Theory in Biology, Communication, Computation*, J. Rosenthal and D. S. Gilliam, Eds. Springer Verlag, 2003, pp. 469–492.
- [6] F. Kschischang, B. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," in *IEEE Transactions on Information Theory*, vol. 47, Feb. 2001, pp. 498–519.
- [7] D. Bickson, O. Shental, and D. Dolev, "Distributed Kalman filter via Gaussian belief propagation," in *the 46th Allerton Conf. on Communications, Control and Computing*, Monticello, IL, USA, 2008.
- [8] A. Montanari, B. Prabhakar, and D. Tse, "Belief propagation based multi-user detection," in *Proc. 43rd Allerton Conf. on Communications, Control and Computing*, Monticello, IL, USA, Sept. 2005.
- [9] D. Bickson, O. Shental, P. H. Siegel, J. K. Wolf, and D. Dolev, "Gaussian belief propagation based multiuser detection," in *IEEE Int. Symp. on Inform. Theory (ISIT)*, Toronto, Canada, July 2008.
- [10] D. Bickson, Y. Tock, D. Dolev, and O. Shental, "Polynomial linear programming with Gaussian belief propagation," in *the 46th Allerton Conf. on Communications, Control and Computing*, Monticello, IL, USA, 2008.
- [11] D. Bickson and D. Malkhi, "A unifying framework for rating users and data items in peer-to-peer and social networks," in *Peer-to-Peer Networking and Applications (PPNA) Journal*, Springer-Verlag, April 2008.
- [12] D. Bickson, D. Dolev, and E. Yom-Tov, "A Gaussian belief propagation solver for large scale support vector machines," in *5th European Conference on Complex Systems*, Sept. 2008.
- [13] C. C. Moallemi and B. Van Roy, "Consensus propagation," in *IEEE Transactions on Information Theory*, vol. 52, no. 11, 2006, pp. 4753–4766.
- [14] B. Frey, "Local probability propagation for factor analysis," in *Neural Information Processing Systems (NIPS)*, 1999.
- [15] P. Rusmevichientong and B. Van Roy, "An analysis of belief propagation on the Turbo decoding graph with Gaussian densities," in *IEEE Transactions on Information Theory*, vol. 47, 2001.
- [16] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation*, vol. 13, no. 10, pp. 2173–2200, 2001.
- [17] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *Journal of Machine Learning Research*, vol. 7, Oct. 2006.
- [18] J. K. Johnson, D. M. Malioutov, and A. S. Willsky, "Walk-sum interpretation and analysis of Gaussian belief propagation," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, 2006, pp. 579–586.
- [19] S. Lauritzen, "Graphical models," in *Oxford Statistical Science Series*, Oxford University Press, 1996.
- [20] Gaussian Belief Propagation implementation in matlab [online] <http://www.cs.huji.ac.il/labs/danss/p2p/gabp/>.