LA-UR- 09-00350

Approved for public release; distribution is unlimited.

Title: Fixing Convergence of Gaussian Belief Propagation

Author(s): Jason Johnson/227983/LANL/T-4
Danny Bickson/IBM Research Lab
Danny Dolev/Hebrew University of Jerusalem

Intended for: 2009 International Symposium on Information Theory

Los Alamos
NATIONAL LABORATORY
EST. 1943

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*— The Gaussian belief propagation algorithm (GaBP) is an iterative message-passing algorithm for computing inference in a Gaussian graphical model. It is known that when the GaBP converges it converges to the correct global solution. The exact region of convergence is a challenging open problem. Currently there are two known sufficient conditions for convergence.**

In this paper we develop a double-loop algorithm for forcing the convergence of the GaBP algorithm applied to any positive definite covariance matrix. Our novel method works even when the sufficient conditions for convergence do not hold. We further extend this construction to non-square column dependent matrices.

We believe that our novel construction has numerous applications, since the GaBP algorithm is linked to the solution of linear system 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 Montanri's linear detection algorithm, in cases where it would originally fail. As a consequence, we are able to increase significantly the number of users transmitting concurrently.

I. Introduction

The Gaussian belief propagation algorithm (GaBP) is an efficient distributed message-passing algorithm for computing inference over a Gaussian graphical model. Recently, GaBP was explicitly 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.

Identifying the exact region of convergence of the GaBP algorithm is a challenging open question. Currently, only two sufficient convergence conditions are known, (when the algorithm is applied on a loopy graph) [16], [17]. A vast research effort was invested in characterizing the convergence of the GaBP algorithm in various settings [1], [8], [13]–[18].

In this work, we propose a novel construction that fixes the convergence of the GaBP algorithm, for any positive-definite matrices, even when the known sufficient convergence condition do not hold. We prove that our construction converges to the correct solution and characterize the speed of convergence. Furthermore, we extend our construction for non-square matrices.

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

This paper is organized as follows. Section II outlines the problem model. Section III describes our novel double-loop construction for positive definite matrices. Section IV extends the construction for arbitrary column depended non-square matrices. We provide experimental results of deploying our construction in the linear detection context in Section V. We conclude in Section VI.

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):

$$P(x) \propto \exp\{-\frac{1}{2}x^T Jx + h^T x\} \quad (1)$$

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

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

We may assume without loss of generality (by rescaling variables) that J is normalized to have unit-diagonal, that is, $J = I - R$ 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 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] Malioutov et al], such that the spectral radius of $|R| = (|r_{ij}|)$ is less than one ($\rho(|R|) < 1$), then the method of *Gaussian belief propagation* (GaBP) may be used to solve this problem. We note that walk-summable condition implies $I - R$ is positive definite. An equivalent characterization of the walk-summable condition is that $I - |R|$ is positive definite.

III. 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 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

$$\gamma^* = \rho(|R|) - 1. \tag{3}$$

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:

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

Note that the effect of adding positive Γ is to reduce the size of the scaling factor $(J+\Gamma)^{-1}$ but this effect is compensated for by adding a feedback term $\Gamma\hat{x}$ to the input h. This may also be interpreted as solving the following convex quadratic optimization problem at each step:

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

This is basically a regularized version of Newton's method to minimize $f(x)$ where we use Tychonoff-regularization to control the step-size at each step. Typically, regularization is used to insure positive-definiteness of the Hessian matrix when using Newton's method to optimize a non-convex function. We instead use it to insure that the Hessian $J + \Gamma$ 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, $\hat{x}^{(t)} \to \hat{x} = J^{-1}h$ 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 $i = 1, \ldots, n$.

Proof. First, we note that there is only one possible fixed-point of the algorithm and this is $\hat{x} = J^{-1}h$. 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 $e^{(t)} = \hat{x}^{(t)} - \hat{x}$ 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 $\rho((J+\Gamma)^{-1}\Gamma) < 1$, 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: $Hx = \lambda x$, $(J+\Gamma)^{-1}\Gamma y = \lambda y$ ($y = (J+\Gamma)^{-1/2}x$), $\Gamma y = \lambda(J+\Gamma)y$, $(1-\lambda)\Gamma y = \lambda J y$, $J^{-1}\Gamma y = \frac{\lambda}{1-\lambda}y$ 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 $\rho(H) < 1$, $e^{(t)} \to 0$ and $\hat{x}^{(t)} \to \hat{x}$ 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 $J'x^{(t+1)} = h^{(t)}$. 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 insure a good solution to $J'x = h$. Alternatively, we may compress this double-loop procedure into a single-loop procedure by peforming 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 $s \in (0,1)$:

$$\begin{aligned} h^{(t+1)} &= (1-s)h^{(t)} + s(h + \Gamma\hat{x}^{(t)}) \quad (6) \\ &= h + \Gamma((1-s)\hat{x}^{(t-1)} + s\hat{x}^{(t)}) \quad (7) \end{aligned}$$

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 is usually more efficient then the double-loop method.

IV. Extension to General Linear Systems

In this section, we efficiently extend the applicability of the proposed double loop construction for systems of linear equations with positive definite matrices to systems with any square (*i.e.*, also nonsymmetric or non positive definite) or rectangular matrices. For this construction to work, we require that the matrix should be column dependent.

Given a column-dependent matrix $\tilde{J}_{n\times k}$, $n \geq k$, and a shift vector $\tilde{h}$, we are interested in solving the linear systems of equations $\tilde{J}x = \tilde{h}$. The naive approach for using GaBP would be to take the information matrix $\bar{J} \triangleq (\tilde{J}^T\tilde{J})$, and the shift vector $\bar{h} \triangleq \tilde{J}^T\tilde{h}$. Note, that $\bar{J}$ is positive definite and we can use GaBP as before. In this case, the MAP solution is

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

which is the pseudo inverse solution.

Note, that in the above construction has two drawbacks: first, we need to explicitly compute $\bar{J}$ and $\bar{h}$, and second, $\bar{J}$ 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 $\bar{\bar{J}}$ based on the arbitrary (non-rectangular) column dependent matrix $\tilde{J} \in \mathbb{R}^{n\times k}$

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

Additionally, we define a new hidden variable vector $\tilde{x} \triangleq \{x^T, z^T\}^T \in \mathbb{R}^{(k+n)\times 1}$, where $x \in \mathbb{R}^{k\times 1}$ is the solution vector and $z \in \mathbb{R}^{n\times 1}$ is an auxiliary hidden vector, and a new shift vector $\bar{\bar{h}} \triangleq \{\mathbf{0}^T_{k\times 1}, h^T\}^T \in \mathbb{R}^{(k+n)\times 1}$. Now, we run GaBP on the new system $(\bar{\bar{h}}, \bar{\bar{J}})$ to obtain the solution to 8. The proof of the correctness of the above construction is given in [9].

V. Experimental results

A. Linear detection in linear channels

Consider a discrete-time channel with a real input vector $x = \{x_1, \ldots, x_K\}^T$ governed by an arbitrary prior distribution, P_x, and a corresponding real output vector $y = \{y_1, \ldots, y_K\}^T = f\{x^T\} \in \mathbb{R}^K$. 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

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

where n is a $K \times 1$ additive noise vector and $R = S^T S$ 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 $A = R + \sigma^2 I_K$. 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 criterion.

B. Montnanri'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 Montnanri's algorithm. 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 $\sigma^2 = 0.001$. Matlab code of our implementation is available on [20].

We have drawn at random random-spearing CDMA matrix. Typically, the sufficient convergence conditions for the GaBP algorithm do not hold. For example, we have drawn at random a random-spreading CDMA matrix with $\rho(I - |R|) = 1.0906$. Since $\rho(I - R) > 1$, 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.

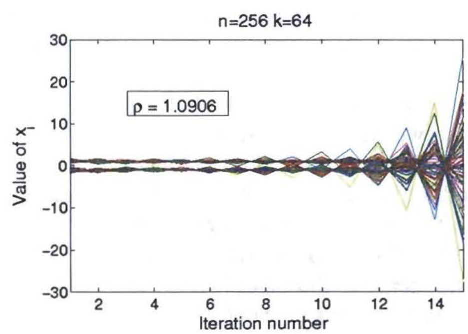

Fig. 1. Divergence of the GaBP algorithm for the multiuser detection problem, when $n = 256, k = 64$.

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 first, we forced the matrix to be diagonally dominant (DD). In this case, the spectral radius $\rho = 0.20355$. In the second case, the matrix was not DD, but the spectral radius was $\rho = 0.33854$. Clearly, the newton method converges faster when the spectral radius is larger.

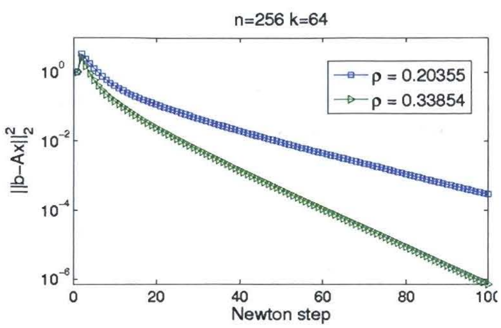

Fig. 2. Convergence of the fixed GaBP iteration under the same settings ($n = 256, k = 64$)

VI. Conclusions and Future Work

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

References

[1] C. Moallemi and B. V. 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 *in Codes, Graphs, and Systems, (festschrift in honour of David Forney*, R. E. Blahut and R. Koetter, Eds. Kluwer, 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*, no. 6, 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, 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. 43th 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. V. 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] Rusmevichientong and 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/`.