Convergence Analysis of a New Distributed Algorithm for Networked Estimation

Zhaorong Zhang[1], Qianqian Cai[2] and Minyue Fu[1]

***Abstract*— This paper presents the convergence analysis of a new distributed algorithm, which is inspired by the celebrated BP (Belief Propagation) algorithm, for networked estimation in large-scale sparse systems. The proposed algorithm owns fast convergence rate and other advanced properties of the BP algorithm. We reveal that the distributed algorithm is guaranteed to converge correctly under the assumption that the system is generalized diagonally dominant. The convergence analysis for both acyclic graphs and loopy graphs have been studied. Specifically, the distributed algorithm will converge after finite number of iterations, which is equal to the diameter of the network graph, if the graph is acyclic. For a loopy network, the distributed algorithm is guaranteed to converge to the optimal estimates asymptotically. It can be shown from simulation results that the proposed distributed algorithm outperforms some existing distributed estimation algorithms.**

I. Introduction

Distributed algorithms, which play an increasingly significant role in modern industrial systems and advanced technologies, have been studied extensively by researchers. It is undeniable that distributed estimation algorithms are robust to communication failures and with low computational complexity. More importantly, these algorithms are scalable to large networked systems. In the early years, centralized estimation, which requires a central controller to collect the measurements from all the nodes in the network in order to carry out estimate, was widely used. Distributed estimation algorithms are preferred nowadays in dealing with large-scale systems. The most critical feature of distributed estimation algorithm is that every node acts as a local estimator and only cooperates with its neighboring nodes. Distributed algorithms convert estimates in each node associated to the system using the combination of local measurement and messages received from neighbors. Hence, the communication burden will be dramatically reduced since each local estimator only needs to compute simple low-dimensional additions and multiplications.

Generally, weighted least squares estimation, weighted average and a series of static estimation problems can be converted into the problem of solving a system of linear equations, which is regarded as a very fundamental problem in numerical computation. In other words, the development in solving linear systems also makes important contributions to the field of state estimation. With this motivation, a distributed algorithm for solving linear systems, which is inspired by the celebrated Belief Propagation algorithm, will be introduced in this paper, and the main properties and convergence analysis of the algorithm will be presented as well. The well-known Belief Propagation algorithm, also called as Pearl's BP [1], was intended to calculate marginal probabilities in a fully distributed way. Gaussian Belief Propagation algorithm [2], which is applied to Gaussian graphical models, is a special case of the BP algorithm. A number of comprehensive studies have been done on the Gaussian BP algorithm. For instance, it has been shown in [2] that Gaussian BP generates correct marginal means asymptotically provided that the associated information matrix is diagonally dominant. After that, [3] has relaxed the convergence condition to generalized diagonally dominance. Recently, necessary and sufficient conditions for convergence of Gaussian BP has been put forward by [4]. In the proposed algorithm, each node repeatedly calculates the estimate of an associated element of the solution to the linear system at every iteration until the correct solution is acquired. To be specific, the proposed algorithm will converge to the correct solution of the linear system in acyclic graphs after d iterations, where d is the diameter of acyclic graph.

The linear systems we focus on here is the so-called generalized diagonally dominant systems. It has been proved that generalized diagonally dominance is equivalent to walk-summability [3]. With the assumption of generalized diagonally dominance, the proposed algorithm is guaranteed to converge asymptotically in loopy networks. Overall, the convergence rate is dramatically faster than other laplacian matrix based methods that is supported by simulation results by comparisons. This is due to the low complexity of proposed algorithm in information exchange between neighboring nodes, local computation and local storage. In order to have a more intuitive understanding of the exact convergence rate , we have put forward a relaxed exponential convergence rate of proposed algorithm.

II. Problem Formulation and Preliminaries

Consider an undirected Gaussian graphical model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \cdots, n\}$ is the node set and $\mathcal{E} = \{(i, j)\}$ denotes the edge set. With the graph G, there is an associated linear system $Ax = b$ where $A = col\{a_1^T, a_2^T, \cdots, a_{n_T}\}$ and $b = col\{b_1, b_2, \cdots, b_n\}$. $A = \{a_{ij}\}$ is a sparse information matrix with $a_{ij} \neq 0$ if and

[1]School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, 2308, NSW, Australia.

[2]School of Automation, Guangdong University of Technology and Guangdong Key Laboratory of Intelligent Decision and Cooperative Control, Guangzhou 510006, China.

This work was supported by the National Natural Science Foundation of China (Grant numbers 61633014 and U1701264). E-mails: zhaorong.zhang@uon.edu.au; qianqian@outlook.com; minyue.fu@newcastle.edu.au.

only if $(i,j) \in \mathcal{E}$. We assume that matrix A always satisfies (1) A is invertible and (2) the diagonal part of A is an n-dimensional unit matrix. Hence the unique solution to the linear system is

$$x^* = A^{-1}b \tag{1}$$

where $x^* = \text{col}\{x_1^*, \cdots, x_n^*\}$. Each node $i \in \mathcal{V}$ only acquaints with the information of a_i^T and b_i where a_{ii}, b_i are related to node i and the nonzero a_{ij} is related to the edge (i,j). An estimate of x_i^* which is denoted as $x_i(k)$ is carried out at each node $i \in V$ in the k-th iteration. The distributed algorithm we intend to design is to obtain at each node i the solution x_i^* by usingh the local information and shared information from its neighboring nodes iteratively. The distributed algorithm is inspired by Gaussian BP algorithm [2], which is an elegant distributed algorithm for computing the marginal probability density functions in the Gaussian graphical model $p(x) = \exp\{-\frac{1}{2}x^TAx + b^Tx\}$ where the information matrix A is symmetric and positive definite. Each node i calculates the marginal probabilities associated with unknown variable x_i with local measurement and potential function of pairwise neighbors. It can be easily inferred that the Gaussian means of this model is the solution to a symmetric linear system. For the seek of a wider range of applications, we have proposed the distributed algorithm to adapt to linear systems with non-symmetric matrix A and still converge to the correct solution.

For clarity, some notations and definitions will be given in this paragraph. A graph is undirected when $(i,j) \in \mathcal{E}$ if and only if $(j,i) \in \mathcal{E}$. An acyclic graph contains no loops. A loopy graph contains at least one loop. Note that a loop is a path which starts and ends at the same node i but also passes through at least a node $j \neq i$. The neighbors or neighboring nodes of node i is the node set $\mathcal{N}_i = \{j | (i,j) \in \mathcal{E}\}$ that denotes all of the nodes j connected to node i by some edge (i,j). The cardinality of node i is written as $|\mathcal{N}_i|$. The graphs we discuss here are all connected which means there exists a path $(i,k_1),(k_1,k_2),\cdots,(k_n,j)$ that starts from a node i to another arbitrary node j. The distance between an arbitrary pair of nodes (i,j) is the minimum number of nodes that a path from node i to j passes through. The diameter of a graph is the maximum distance between two nodes.

Definition 1. *A matrix $A = \{a_{ij}\} \in \mathbb{R}^{n\times n}$ is said to be strictly diagonally dominant (or simply, diagonally dominant) if for each row i, $a_{ii} > 0$ and $a_{ii} > \sum_{j\neq i}|a_{ij}|$. The matrix A is said to be generalized diagonally dominant if there exists a diagonal matrix $D = diag\{d_i\}$ with all $d_i > 0$ such that $D^{-1}AD$ is diagonally dominant.*

For a diagonally dominant matrix A, we associate the parameter

$$\rho_i = a_{ii}^{-1}\sum_{j\neq i}|a_{ij}| < 1 \tag{2}$$

with each node $i \in \mathcal{V}$. It has been proved in [3] that generalized diagonal dominance is equivalent to the notion of walk summability.

A brief introduction of walk-summable models will be given in this part. First of all, the diagonal elements $A_{ii}, i \in \mathcal{V}$ of information matrix A is scaled to be 1 after matrix nomalization. It is directed to write A as $A = I - R$ where $R_{ij} \neq 0, (i,j) \in \mathcal{E}$ and $R_{ii} = 0, i \in \mathcal{V}$. A walk of length l in the Gaussian model is a sequence $w = (w_0, w_1, \cdots, w_l)$ where $(w_{i-1}, w_i) \in \mathcal{E}, i \leq l$ is the walk's single step which corresponds to the weight $R_{w_{i-1}w_i}$. The so-called weight of a walk, denoted as $\phi(w)$, is the product of the weights related with every single step, which can be expressed as:

$$\phi(w) = \prod_{k=1}^{l(w)} R_{w_{k-1}w_k}, \tag{3}$$

where $l(w)$ is the length of the walk w. The definition of walk-summability is given as followed.

Definition 2. *A system is said to be walk-summable (or generalized diagonally dominant) if for all of the $i,j \in \mathcal{V}$,*

$$\sum_{w:i\to j}\phi(w), \tag{4}$$

which is the sum of all of the possible walks from node i to j, is well-defined.

Here, well-definedness indicates that the sum converges to the same value regardless of the order of summation. It is well known that this is equivalent to the absolute convergence condition, i.e., the sum $\sum_{w:i\to j}|\phi(w)|$ converges; see [3] for more details.

Besides, several equivalent conditions for walk-summability are given as followed:

- $\sum_{w:i\to j}|\phi(w)|$ converges for all $i,j \in \mathcal{V}$;
- $\sum_l \bar{R}^l$ converges;
- $\rho(\bar{R}) < 1$;
- $I - \bar{R} \succ 0$,

where $\bar{R} = \{|r_{ij}|\}$.

Some brief notations of walk-sums are necessary for the rest of of the paper. A walk from node i to j is denoted as $w : i \to j$ and denoted as $w : i \xrightarrow{l} j$ with the length of l. Especially, a single node i is treated as a zero-length walk. We write the set of walks from i to j as $\{i \to j\}$ and $\{i \xrightarrow{l} j\}$ with the length of l. The walk-sum interpretation of Gaussian means μ and variances P in Gaussian graphical models are expressed as:

$$P = A^{-1} = (I-R)^{-1} = \sum_{l=0}^{\infty} R^l \tag{5}$$

$$\mu = A^{-1}b = \sum_{l=0}^{\infty} R^l b \tag{6}$$

where $\mu = x^*$ which is the solution to linear system.

Denoting $P = \{p_{ij}\}$ and $x^* = \{x_i^*\}$, it follows that

$$p_{ij} = \sum_{l=0}^{\infty} (R^l)_{ij} = \sum_{w:i\to j} \phi(w) = \phi(i \to j) \tag{7}$$

$$x_i^* = \sum_s P_{is} b_s = \sum_s \phi(s \to i) b_s. \tag{8}$$

III. Main Result

In this section, we give a distributed algorithm for solving the linear system (1). This will be followed by analysis results, studying the convergence properties of the algorithm. The detailed proofs are omitted for this conference version.

A. Distributed Algorithm

The distributed algorithm runs on each node $i \in \mathcal{V}$, using its local information and information received from its neighboring set $\mathcal{N}_i$. The algorithm is given in Algorithm 1.

Algorithm 1 (Distributed Algorithm for Average Consensus)

Initialization: For each node i, do: For each $j \in \mathcal{N}_i$, set $a_{i\to j}(0) = a_{ii}$, $b_{i\to j}(0) = b_i$, and transmit them to node j.

Main loop: At iteration $k = 1, 2, \cdots$, f or each node i, compute

$$\tilde{a}_i(k) = a_{ii} - \sum_{v\in\mathcal{N}_i} \frac{a_{ij}a_{ji}}{a_{v\to i}(k-1)} \tag{9}$$

$$\tilde{b}_i(k) = b_i - \sum_{v\in\mathcal{N}_i} \frac{a_{iv}b_{v\to i}(k-1)}{a_{v\to i}(k-1)} \tag{10}$$

$$\hat{x}_i(k) = \frac{\tilde{b}_i(k)}{\tilde{a}_i(k)}, \tag{11}$$

then for each $j \in \mathcal{N}_i$, compute

$$a_{i\to j}(k) = \tilde{a}_i(k) + \frac{a_{ji}a_{ij}}{a_{j\to i}(k-1)} \tag{12}$$

$$b_{i\to j}(k) = \tilde{b}_i(k) + \frac{a_{ij}b_{j\to i}(k-1)}{a_{j\to i}(k-1)} \tag{13}$$

and transmit them to node j.

B. Convergence Analysis for Acyclic Graphs

For an acyclic graph $\mathcal{G}$, Algorithm 1 has the following excellent finite-time convergence property. In the following, the sub-graph $\mathcal{G}_i(k)$ of $\mathcal{G}$, for $k > 0$, is formed from $\mathcal{G}$ by removing all the nodes and edges beyond k hops away from node i, and $\mathcal{G}_{i\setminus j}(k)$ is formed by further removing all the paths going through node j.

Theorem 1. *For any node $i \in \mathcal{V}$, $j \in \mathcal{N}_i$ and $k \geq 0$, running Algorithm 1 will yield*

$$a_{i\to j}(k) = \frac{1}{\phi(\{i \to i\}|\mathcal{G}_{i\setminus j}(k))} > 0, \tag{14}$$

$$b_{i\to j}(k) = \sum_s \phi(\{s \xrightarrow{i} i\}|\mathcal{G}_{i\setminus j}(k)) b_s, \tag{15}$$

where s ranges over all nodes in $\mathcal{G}_{i\setminus j}(k)$. In addition, $a_{i\to j}(k) = a_{i\to j}(d-1)$, $b_{i\to j}(k) = b_{i\to j}$ for all $k \geq d$, where d is the diameter of graph $\mathcal{G}$.

C. Convergence Analysis for Loopy Graphs

For a general cyclic (loopy) graph, the unwrapped tree is an effective tool to analyze the convergence properties of Algorithm 1. This is constructed as follows.

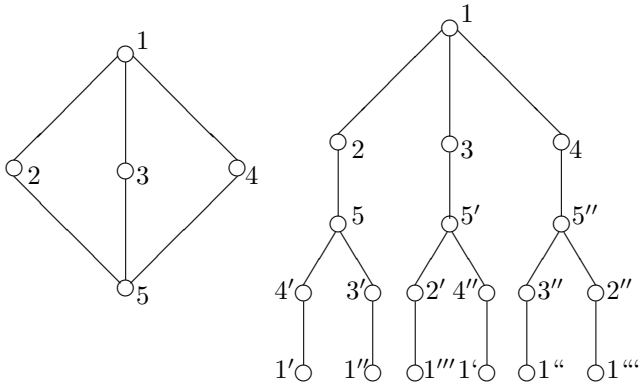

Fig. 1. Left: A loopy graph. Right: The unwrapped tree around node 1 with 4 layers ($t = 4$)

- Find all leaves of the tree (start with the root);
- For each leaf, find all the nodes in the loopy graph that neighbor this leaf node, except its parent node in the tree, and add all these node as the children to this leaf node.

The variables and weights for each node in the unwrapped tree are copied from the corresponding nodes in the loopy graph. It is clear that taking each node as root node will generate a different unwrapped tree. Fig. 1 shows the unwrapped tree around node 1 for the loopy graph. Note, for example, that nodes $1', 1'', 1''', 1‘, 1“, 1‘‘$ all carry the same values y_1 and $A_1 1$. Based on the result of convergence analysis, we have given a further and deeper insight into the analysis of the convergence rate of proposed algorithm.

The unwrapped tree is expanded in breadth first order which starts from the root node followed by the first layer, then the second layer and so on until to the t-th layer. Denote the unwrapped tree with t layers as $\mathcal{G}_t$. When the t increases to be large enough, there must be a one-to-one correspondence between finite-length walks in $\mathcal{G}$ and finite-length walks in $\mathcal{G}_t$ for any $i \in \mathcal{V}$ and $k \geq 0$.

Lemma 1. *With the assumption of generalized diagonal dominance, we have*

$$\hat{x}_i(t) = \sum_{j=1}^{n} \sum_{w:j\to i|\mathcal{G}_t} \phi(w) b_j, \tag{16}$$

where $i \in \mathcal{V}$ and $w : j \to i|\mathcal{G}_t$ denotes a walk in unwrapped tree $\mathcal{G}_t$ from node i to j.

Using the unwrapped tree model, we can obtain the following asymptotic convergence property.

Theorem 2. *Suppose the linear system 1 is generalized diagonally dominant. Then, running Algorithm 1 yields $a_{i\to j}(k) > 0$ for all $i \in \mathcal{V}$, $j \in \mathcal{N}_i$ and $k = 0, 1, \cdots$, and that*

$$\lim_{k\to\infty} \hat{x}_i(k) = x_i^*, \forall i \in \mathcal{V}. \tag{17}$$

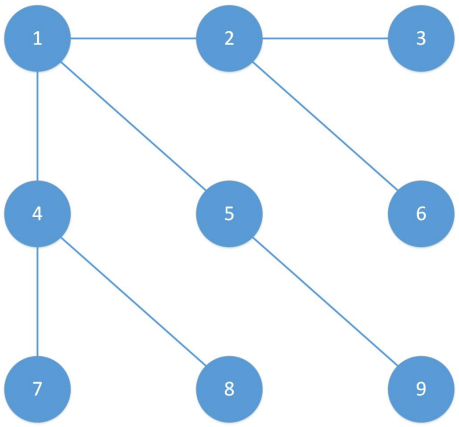

Fig. 2. A 9-node Acyclic Graph

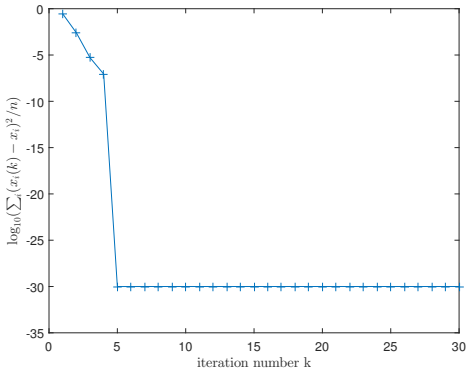

Fig. 3. Convergence of Algorithm 1 in an Acyclic Graph

D. Convergence Rate Analysis

The definition of simple cycle is critical to the derivation of convergence rate of the proposed algorithm. A simple cycle is a cycle or loop with a sequence of unique nodes, other than the the starting and ending node are the same. The geometric-mean node gain of a simple cycle $p = (i_0, i_1, \cdots, i_0)$ is define to be

$$\rho(p) = (\rho_{i_0}, \rho_{i_1}, \cdots, \rho_{i_{k-1}})^{\frac{1}{k}} \tag{18}$$

where $\rho_{i_m}, m < k$ has been defined in section 2.

Theorem 3. *Give a matrix $A \in \mathbb{R}^{n\times n}$ with positive diagonals and associated graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, suppose $0 < \rho_i < 1$ for all $i = 1, 2, \cdots, n$. Then, it holds for algorithm 1 that*

$$|\hat{x}_i(k) - x_i| \leq \max_{p\in\mathcal{C}} \rho(p)^k C \tag{19}$$

where $\mathcal{C}$ is the set of all of the simple cycles in a graph and C is a constant.

IV. Simulation

We have illustrated the convergence performance of Algorithm 1 in both acyclic and loopy graphs. Among the loopy graphs, we compare the convergence rate of the proposed algorithm in graphs with different graph degrees. The systems are all generalized diagonally dominant and each element in the information matrix is selected to be random. In Fig. 2, the acyclic graph has the longest path with length of 5 which leads to $d = 5$. In Fig. 3, the error of Algorithm 1 converges to around 10^{-30} and remains stable after 5 iterations which verifies our conclusion. Consequently, we add several edges to Fig. 2 to form a loopy graph in Fig. 4. It turns out that it takes Algorithm 1 more iterations to converge to the same error precision as in the acyclic graph. In this case, despite the number of nodes are the same, the loops can make a big difference in the convergence performance.

Consequently, we have done more simulations in a series of loopy graphs with different degrees. Suppose there are 50 random nodes and connect to a fixed number of neighboring nodes that are selected arbitrarily. As shown in Fig. 6, Fig. 7 and Fig. 8, the degrees of the 50-node graphs, which have the

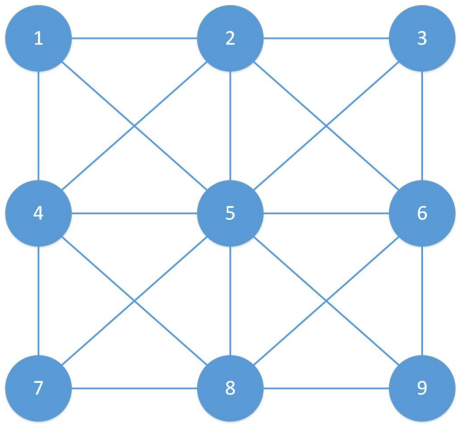

Fig. 4. A 9-node Loopy Graph

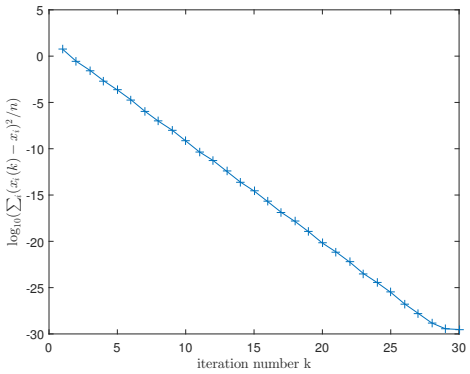

Fig. 5. Convergence of Algorithm 1 in a Loopy Graph

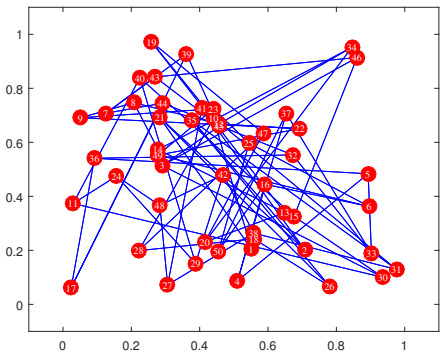

Fig. 6. A 50-node Graph with Degree of 3

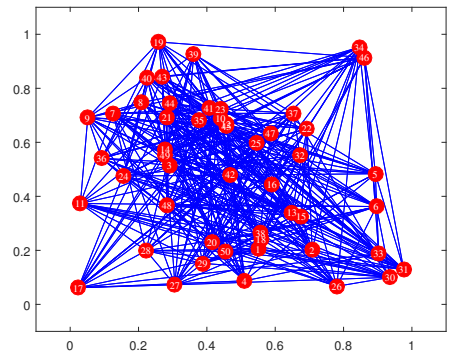

Fig. 8. A 50-node Graph with Degree of 15

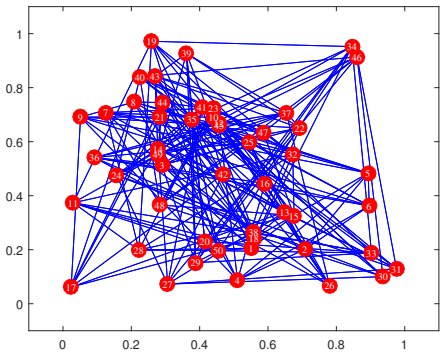

Fig. 7. A 50-node Graph with Degree of 8

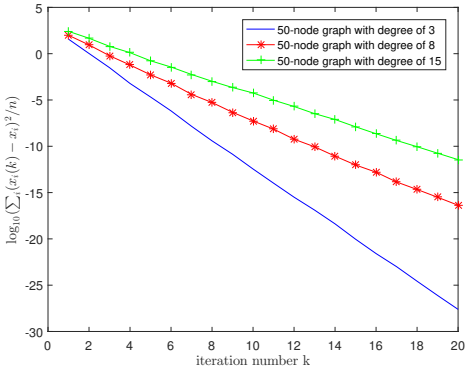

Fig. 9. A 50-node Graph with Degree of 15

same node number and positions, are 3, 8 and 15 respectively. It can be observed that the convergence rate of proposed algorithm is decreasing as the graph degree increases. Since the the number of loops in a graph increases the burden and complexity of information exchange and computation between neighbors which will induce the proposed algorithm to converge after more iterations. Although the connectivity and information fusion performance of graphs with higher degree is enhanced, the number of loops in a graph has a more serious impact on the convergence rate of proposed algorithm. Thus, we can see from Fig. 9 that the convergence rate of proposed algorithm is the fastest with the minimum graph degree among the three examples.

V. Conclusion

In this paper, we have studied a novel distributed algorithm, which is generalized from the Gaussian BP algorithm, to solve the linear matrix equations that is also the essence of solving distributed estimation problems. In a system with an associated acyclic graph, the distributed algorithm converges in a finite number of iterations which is equal to the diameter of the graph. For a loopy graph, the algorithm will converge asymptotically under the assumption of generalized diagonal dominance. We also give a bound on the exponential convergence rate of the distributed algorithm.

References

[1] J. Pearl, Probabilistic Reasoning in Intelligent Systems, Morgan Kaufman, 1988.

[2] Y. Weiss and William T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," Neural Computation, vol. 13, no. 10, pp. 2173-2200, 2001.

[3] D. M. Malioutov, J. Johnson, and A. Wilsky, "Walk-sums and Belief Propagation in Gaussian Graphical Models," J. Mach. Learn. Res., vol.7, no. 1, pp. 2031-2064, 2006.

[4] Q. Su and Y-C. Wu. "On convergence conditions of Gaussian belief propagation algorithm," IEEE Transactions on Signal Processing. vol. 63, no. 5, pp. 1144-1155, 2015.