Gaussian Message Propagation in *d*-order Neighborhood for Gaussian Graphical Model

Yarui Chen[1], Congcong Xiong[1], and Hailin Xie[2]

[1] School of Computer Science and Information Engineering,
Tianjin University of Science and Technology, Tianjin 300222, P.R. China
[2] School of Information Technical Science
Nankai University

Abstract. Gaussian graphical models are important undirected graphical models with multivariate Gaussian distribution. A key probabilistic inference problem for the model is to compute the marginals. Exact inference algorithms have cubic computational complexity, which is intolerable for large-scale models. Most of approximate inference algorithms have a form of message iterations, and their computational complexity is closely related to the convergence and convergence rate, which causes the uncertain computational efficiency. In this paper, we design a fixed parameter linear time approximate algorithm — the Gaussian message propagation in d-order neighborhood. First, we define the d-order neighborhood concept to describe the propagation scope of exact Gaussian messages. Then we design the algorithm of Gaussian message propagation in d-order neighborhood, which propagates Gaussian messages in variable's d-order neighborhood exactly, and in the $(d+1)$th-order neighborhood partly to preserve the spread of the Gaussian messages, and computes the approximate marginals in linear time $O(n \cdot d^2)$ with the fixed parameter d. Finally, we present verification experiments and comparison experiments, and analyze the experiment results.

Keywords: Gaussian graphical model, Probabilistic inference, Message propagation, d-order neighborhood.

1 Introduction

Gaussian graphical models are basic undirected graphical models with multivariate Gaussian distribution and conditional independence assumptions [1,2], and have wide application in image analysis, natural language processing, time-series analysis etc [3,4]. The key problem of probabilistic inference for the Gaussian graphical model is to compute the marginals of variables [5]. For tree-like models, exact inference algorithms, such as Gaussian elimination, belief propagation or junction tree algorithms, can present the marginals in linear time [6]. For general graphical models, these exact inference algorithms have cubic $O(n^3)$ computational complexity [6]. For large-scale models with more complex graphs, arising in oceanography, 3D-tomography, and seismology, the cubic computational complexity becomes computationally prohibitive [7]. Then various approximate

C. Guo, Z.-G. Hou, and Z. Zeng (Eds.): ISNN 2013, Part I, LNCS 7951, pp. 539–546, 2013.

inference algorithms have been developed, such as loopy belief propagation [8], mean field method [9,10].

The loopy belief propagation algorithm propagates the belief messages in the models with cycles directly, and provides successful approximation in some applications [8]. But the algorithm may converges to local optimum, may even fail to converge for general models, and its computational complexity is closely related to the convergence and convergence rate [11]. The mean field algorithm propagates the variational messages to approximate the marginals, and research shows that if the variational message converge, the algorithm can compute the correct mean parameter [10]. This series of message iteration algorithms have the computational complexity $O(m \cdot n \cdot N)$, but the iteration number N is closed related to the convergence of the algorithms. So these algorithms have uncertain computational efficiency.

In this paper, we design a fixed parameter linear time $O(n \cdot d^2)$ approximate inference algorithm— Gaussian message propagation in d-order neighborhood (GaussianMP-d). First, we define the d-order neighborhood concept to describe the propagation scope of Gaussian message, and show the message propagation process in the d-order neighborhood of univariable (GaussianVariableMP-d), which propagates the Gaussian messages in variable's d-order neighborhood exactly, and in the $(d+1)$th-order neighborhood partly to preserve the computational complexity increasing. Then we design the GaussianMP-d algorithm based on the GaussianVariableMP-d unit, which executes the GaussianVariableMP-d for variables in the elimination order I and the reverse order I' respectively to compute all the Gaussian messages, and calculates the approximate marginals with these Gaussian messages. Finally, we present verification experiments and comparison experiments to demonstrate the efficiency and flexibility of the GaussianMP-d algorithm.

2 Gaussian Elimination Process

The Gaussian graphical model is a undirected graphical model based on graph $G = (V, E)$, where the vertex set V denotes the Gaussian random variable set $\boldsymbol{x} = \{x_1, \cdots, x_n\}$, and the edge set E reflects the conditional independences among variables. The probability distribution of Gaussian graphical model is

$$p(\boldsymbol{x}) = \exp\left\{\langle \boldsymbol{h}, \boldsymbol{x}\rangle + \frac{1}{2}\langle \boldsymbol{J}, \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\rangle - A(\boldsymbol{h}, \boldsymbol{J})\right\},$$
$$A(\boldsymbol{h}, \boldsymbol{J}) = \log \int_{\Xi} \exp\left\{\langle \boldsymbol{h}, \boldsymbol{x}\rangle + \frac{1}{2}\langle \boldsymbol{J}, \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\rangle\right\} \mathrm{d}\boldsymbol{x}.$$

Where the $\boldsymbol{h} = [h_1, \cdots, h_n]^{\mathrm{T}}, \boldsymbol{J} = [J_{ij}]_{n \times n}$ are the model parameters, $\langle , \rangle$ denotes dot product operation, $A(\boldsymbol{h}, \boldsymbol{J})$ is the log partition function, $\Xi = \{(\boldsymbol{h}, \boldsymbol{J}) \in \mathbb{R}^n \times \mathbb{R}^{n \times n} | \boldsymbol{J} \prec 0,\ \boldsymbol{J} = \boldsymbol{J}^{\mathrm{T}}\}$ is the constraint set of parameter $(\boldsymbol{h}, \boldsymbol{J})$.

An important inference problem for the Gaussian graphical model is to compute the marginals $p(x_i)$. Gaussian elimination algorithm is an exact inference method with variable elimination/marginalization. The distribution of

$\boldsymbol{x}_U = \boldsymbol{x} \setminus x_s$ can be computed by eliminating x_s, and the corresponding model parameters $\boldsymbol{J}_U^*, \boldsymbol{h}_U^*$ are

$$\boldsymbol{J}_U^* = \boldsymbol{J}_{U,U} - \boldsymbol{J}_{U,s}\boldsymbol{J}_{s,s}^{-1}\boldsymbol{J}_{s,U},$$
$$\boldsymbol{h}_U^* = \boldsymbol{h}_U - \boldsymbol{J}_{U,s}\boldsymbol{J}_{s,s}^{-1}\boldsymbol{h}_s.$$

During the elimination process of the single variable x_s, the parameters of the neighbors $\{t \mid t \in N(s)\}$ have only been changed. The parameter update formulas of the neighbor variable $\{t \mid t \in N(s)\}$ are

$$J_{tt} \leftarrow J_{tt} + \left(-\frac{J_{st}^2}{J_{ss}}\right), \qquad h_t \leftarrow h_t + \left(-\frac{J_{st}}{J_{ss}}h_s\right).$$

The update formula of the edges $\{(t,u) \mid t,u \in N(s)\}$ is

$$J_{tu} \leftarrow J_{tu} + \left(-\frac{J_{st}J_{su}}{J_{ss}}\right).$$

If there is no edge between node t and u, an edge (t,u) would be added with parameter $J_{tu} \leftarrow \big(-\frac{J_{st}J_{su}}{J_{ss}}\big)$. Obviously, the neighborhoods $\{t \mid t \in N(s)\}$ form a complete graphs with $m = |N(s)|$ nodes, and the computational complexity of elimination of the single variable x_s is $O(m^2)$. The scale of the complete graph becomes larger with the elimination of variables, and the computational complexity of elimination of single variable is trend to $O(n^2)$. Then the computational complexity of the Gaussian elimination algorithm is $O(n^3)$.

3 Gaussian Message Propagation in *d*-order Neighborhood

In this section, we define the concept of d-order Gaussian elimination neighborhood, and design the algorithm of Gaussian message propagation in d-order neighborhood (GaussianMP-d Algorithm).

3.1 *d*-order Gaussian Elimination Neighborhood

Definition 1 (dth-order Neighborhood). *For the Gaussian graphical model G, let $N_d(i)$ denote the dth-order Gaussian elimination neighborhood (abbreviated to dth-order neighborhood) of node i, which is defined recursively as following:*

1. *The 1-st order neighborhood $N_1(i)$ is the set of neighbors of the node i, that is $N_1(i) = \{j \mid (i,j) \in E\}$.*
2. *Let s, $t \in N_1(i)$. If the edge (s,t) is added during Gaussian elimination of variable x_i, we label*

$$s \in N_2(t), \;\; t \in N_2(s).$$

3. *Let $u \in N_a(i)$, $v \in N_b(i)$. If the edge (u, v) is added during Gaussian elimination of variable i, we label*

$$u \in N_{a+b}(v), \ \ v \in N_{a+b}(u).$$

Definition 2 (d-order Neighborhood). *For the Gaussian graphical model G, the d-order neighborhood is the union set of $N_d(i), d = 1, \cdots, d$, that is*

$$\cup N_d(i) = N_1(i) \cup N_2(i) \cup \cdots \cup N_d(i).$$

The Gaussian message propagation in the d-order neighborhood of the variable x_i (GaussianVariableMP-d) is to propagate the Gaussian messages in the d-order neighborhood exactly, and in the $(d+1)$th-order neighborhood partly through avoiding adding new edges to decrease the computational complexity. Let I denote the node set in elimination order, I_{elim} the set of nodes eliminated in elimination order, I_{left} the set of nodes left in elimination order, the GaussianVariableMP-d algorithm contains the following four steps, and he formal description is shown in Algorithm 1.

1. Update the parameters $\{\hat{J}_{ii}, \hat{h}_i\}$ of the variable x_i with the Gaussian messages propagated from the variables in I_{elim}, that is

$$\begin{aligned} \hat{J}_{ii} &= J_{ii} + \sum_{j \in (\cup N_{d+1}(i)) \cap I_{\text{elim}}} \triangle J_{j \to i}, \\ \hat{h}_i &= h_i + \sum_{j \in (\cup N_{d+1}(i)) \cap I_{\text{elim}}} \triangle h_{j \to i}, \end{aligned} \tag{1}$$

   where $\triangle J_{j \to i}, \triangle h_{j \to i}$ are the Gaussian messages from node j to i.
   Then update the parameters $\{\hat{J}_{ik}(k) \mid k \in (\cup N_{d+1}(i)) \cap I_{\text{left}}\}$ corresponding to the edges $\{(i,k) \mid k \in (\cup N_{d+1}(i)) \cap I_{\text{left}}\}$, that is

$$\hat{J}_{ik} = J_{ik} + \sum_{j \in (\cup N_{d+1}(i)) \cap I_{\text{elim}}} \triangle J_{j \to ik}, \tag{2}$$

   where $\triangle h_{j \to ik}$ denote the message from the j to the edge (i, k).
2. Compute the Gaussian messages $\{\triangle J_{i \to s}, \triangle h_{i \to s} \mid s \in (\cup N_d(i)) \cap I_{\text{left}}\}$ in the d-order neighborhood exactly, that is

$$\triangle J_{i \to s} = -\hat{J}_{is} \ \hat{J}_{ii}^{-1} \ \hat{J}_{is}, \qquad \triangle h_{i \to s} = -\hat{J}_{is} \ \hat{J}_{ii}^{-1} \ \hat{h}_i. \tag{3}$$

   For $\forall \ \ s, \ t \in (\cup N_d(i)) \cap I_{\text{left}}$, if the edge $(s,t) \notin E$, we add the edge (s,t). Then compute the message from i to the edge (s,t), that is

$$\triangle J_{i \to st} = -\hat{J}_{is} \hat{J}_{ii}^{-1} \hat{J}_{it}. \tag{4}$$

3. Compute the messages $\{\triangle J_{iu}, \triangle h_{iu} \mid u \in N_{n+1}(i) \cap I_{\text{left}}\}$ in the $(d+1)$th-order neighborhood partly, that is

$$\triangle J_{i \to u} = -\hat{J}_{iu} \ \hat{J}_{ii}^{-1} \ \hat{J}_{iu}, \quad \triangle h_{i \to u} = -\hat{J}_{iu} \ \hat{J}_{ii}^{-1} \ \hat{h}_i. \tag{5}$$

Let $v \in (\cup N_d(i)) \cap I_{\text{left}}$, if there is an edge $(u, v) \in E$, we compute the message $\triangle J_{i\to uv}$ from node i to the edge (u, v), that is

$$\triangle J_{i\to uv} = -\hat{J}_{iu}\hat{J}_{ii}^{-1}\hat{J}_{iv}. \tag{6}$$

Obviously, we didn't add new edge in this step, which decreases the computational complexity.

4. Add the node i to I_{elim}, and delete the node i from I_{left}.

Algorithm 1. GaussianVariableMP-d of variable x_i

```
Data: Gaussian graphical model $G$, variable $x_i$, $I$, $I_{\text{elim}}$, $I_{\text{left}}$
Result: $\{\triangle J_{i\to j},\ \triangle h_{i\to j},\ \triangle J_{i\to jk}\}$
begin
    Label the neighborhoods of variable $x_i$ from $N_1(i)$ to $N_{d+1}(i)$;
    Update the parameters $\hat{J}_{ii}, \hat{h}_i$ with (1);
    for $k \in (\cup N_{d+1}(i)) \cap I_{\text{left}}$ do
        Update the parameters $\hat{J}_{ik}$ with (2);
    end
    for $s \in (\cup N_d(i)) \cap I_{left}$ do
        Update the messages $\triangle J_{i\to s}, \triangle h_{i\to s}$ with (3);
    end
    for $\forall\ s, t \in (\cup N_d(i)) \cap I_{left}$ do
        if $(s, t) \notin E$ then
            add $(s, t) \in E$;
        end
        Update the messages $\triangle J_{i\to st}$ with (4);
    end
    for $u \in N_{n+1}(i) \cap I_{left}$ do
        Update the messages $\triangle J_{i\to u}, \triangle h_{i\to u}$ with (5);
        for $v \in (\cup N_d(i)) \cap I_{left}$ do
            if $(u, v) \in E$ then
                Update the messages $\triangle J_{i\to uv}$ with (6);
            end
        end
    end
    Add the node $i$ to $I_{\text{elim}}$,   delete the node $i$ from $I_{\text{left}}$.
end
```

3.2 GaussianMP-*d* Algorithm

The GaussianMP-d Algorithm is to execute the GaussianVariableMP-d for variables in the elimination order I and its inverse elimination order I' respectively, and compute the approximate marginal distributions with the Gaussian messages in the d-order neighborhood. Here, we select a cutset $P = \{\boldsymbol{x}_{A_1}, \boldsymbol{x}_{A_2}, \cdots\}$ of the Gaussian graphical model $G = (V, E)$, which is also the cutset of the elimination order I. Then, we can compute the approximate marginal distribution of $\boldsymbol{x}_A$, that is

$$\tilde{p}(\boldsymbol{x}_A) \propto \exp\left\{\langle \tilde{h}_A, \boldsymbol{x}_A\rangle + \frac{1}{2}\langle \tilde{J}_A, \boldsymbol{x}_A\boldsymbol{x}_A^{\mathrm{T}}\rangle\right\}. \tag{7}$$

Where the model parameters $\tilde{h}_A = [\tilde{h}_a]^{\mathrm{T}}, \tilde{J}_A = [\tilde{J}_{a,b}]_{|A|\times|A|}$ can be computed with the these messages, that is

$$\begin{aligned}\tilde{h}_a &= h_a + \sum_{j\in\cup N_{d+1}(i), j\notin A} \triangle h_{j\to a},\\ \tilde{J}_{aa} &= J_{aa} + \sum_{j\in\cup N_{d+1}(i), j\notin A} \triangle J_{j\to a},\\ \tilde{J}_{ab} &= J_{ab} + \sum_{\substack{k\in\cup N_{d+1}(a),\\ k\in\cup N_{d+1}(b),\ k\notin A}} \triangle J_{k\to ab}.\end{aligned} \tag{8}$$

Generally speaking, the probability inference in subset $\boldsymbol{x}_A$ is trackable. The approximate marginal distributions of variables $x_a \in \boldsymbol{x}_A$ can be computed with the Gaussian elimination algorithm exactly. The formal description of GaussianMP-d algorithm is shown in Algorithm 2. The computational complexity of the GaussianMP-d algorithm is fixed parameter linear time $O(n \cdot d^2)$.

Algorithm 2. GaussianMP-d Algorithm

```
Data: Gaussian graphical model $G$
Result: $\{\tilde{p}(x_i) \mid x_i \in \boldsymbol{x}\}$
begin
    Select a elimination ordering $I$;
    for $x_i \in I$ do
        Run GaussianVariableMP-$d$ algorithm;
    end
    for $x_i \in I'$ do
        Run GaussianVariableMP-$d$ algorithm;
    end
    Select a cutset $P$;
    for variable subset $\boldsymbol{x}_A \in P$ do
        Compute approximate marginal distribution $\widetilde{p}(\boldsymbol{x}_A)$ with (7),(8);
        Run exact belief propagation algorithm in $\boldsymbol{x}_A$;
        Output $\{\tilde{p}(x_{a_1}) \mid x_{a_1} \in \boldsymbol{x}_A\}$;
    end
    Output $\{\tilde{p}(x_i) \mid x_i \in \boldsymbol{x}\}$.
end
```

4 Experiments

In this section, we compare the approximate marginal distributions with some numerical experiments. We experiment with a 8×8 two dimension lattice Gaussian graphical model. Specially, we generate the attractive Gaussian model G_1

with model parameters $J_{ij} \in (0, 20),\ h_i \in (0, 30))$, and the repulsive Gaussian model G_2 with model parameters $J_{ij} \in (-20, 0),\ h_i \in (-30, 0))$, which all satisfy the parameter conditions $\boldsymbol{J} = \boldsymbol{J}^{\mathrm{T}}, \boldsymbol{J} \succ 0$.

For the model G_1, we select the elimination order from left to right, and from bottom up, and execute the GaussianMP-$d(d = 1, 2, 3)$ algorithms respectively. The experiment results are shown in Figure 1, which show that the approximate marginal distributions become tighter as the neighborhood order d increasing, and the GaussianMP-d algorithms present the low bounds of the parameters compared with the exact values.

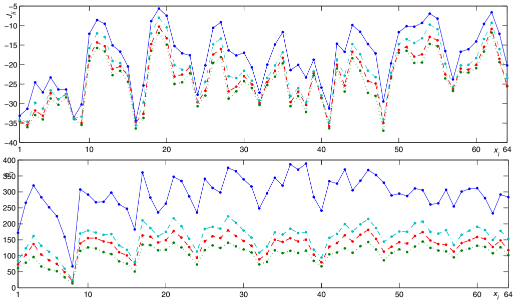
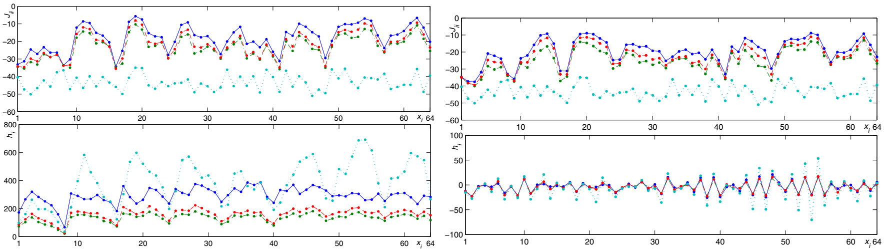


Fig. 1. The comparisons of approximate marginals of the GaussianMP-$d(d = 1, 2, 3)$ algorithms for G_1. The solid lines denote exact model parameters of marginals, the dashed lines, dashdot lines and dotted lines denote the model parameters based on the GaussianMP-d(d=1,2,3) algorithms respectively.

For the Gaussian graphical model G_1, G_2, we first execute the GaussianMP-$d(d = 2, 3)$ algorithms respectively, then run the mean field method with full factorial free distribution. The experiment results are shown in Figure 2, which shows that: (1) The GaussianMP-$d(d = 2, 3)$ algorithms have higher computational accuracy on the parameters $\{J_{ii} \mid i \in V\}$ than the mean field method has. (2) The mean field algorithm provides some better values for parameters $\{h_i\}$, also some worse values. Conversely, the GaussianMP-d $(d = 2, 3)$ algorithms provide more stable approximate values for parameters $\{h_i \mid i \in V\}$.

5 Conclusions

For the Gaussian graphical models, we have defined the d-order neighborhood concept, and designed the GaussianMP-d algorithm with fixed parameter linear $O(n \cdot d^2)$ computational complexity. The d-order neighborhood concept describes the propagation scope of Gaussian messages, which reveals that the Gaussian messages become less accurate as the neighborhood order d increases. Based on this, the GaussianMP-d algorithm makes full use of the messages in the d-order neighborhood, and obtains the linear running time at the cost of the accuracy of the left neighborhoods. The order parameter d also provides a trade-off criterion for the computational complexity and the approximate accuracy.



(a) Parameter comparisons for model G_1. (b) Parameter comparisons for model G_2.

Fig. 2. Accuracy comparisons of the approximate marginal distributions for model G_1 and G_2. The solid lines denote the exact values of parameters, the dashed lines and the dashdot lines denote the approximate values based on the GaussianMP-$d(d = 2, 3)$, the dotted lines the approximate values based on the mean field algorithm.

Acknowledgments. This work is supported in part by Tianjin Science and Technology Development of High School (Grant No. 20110806) and Scientific Research Foundation of Tianjin University of Science and Technology (Grant No. 20110404).

References

1. Cozman, F.G.: Graphical models for imprecise probabilities. International Journal of Approximate Reasoning 39(2-3), 167–184 (2005)
2. Rue, H., Held, L.: Gaussian Markov random fields: Theory and applications. Chapman & Hall, London (2005)
3. de Camposa, L.M., Romero, A.E.: Bayesian network models for hierarchical text classification from a thesaurus. International Journal of Approximate Reasoning 50(7), 932–944 (2009)
4. Malioutov, D.: Approximate inference in Gaussian graphical models. PhD thesis, Massachusetts Institute of Technology (2008)
5. Bishop, C.M.: Pattern recognition and machine learning, ch. 10, pp. 461–522. Springer (2006)
6. Jordan, M.I., Weiss, Y.: Graphical models: probabilistic inference. In: Arbib, M.A. (ed.) The Handbook of Brain Theory and Neural Networks, pp. 490–496. The MIT Press (2002)
7. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning 1(1-2), 1–305 (2008)
8. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalization. Technical Report TR2001-22, Mitsubishi Electric Research Labs (January 2002)
9. Winn, J.: Variational message passing and its applications. PhD thesis, University of Cambridge (2003)
10. Winn, J., Bishop, C.M.: Variational message passing. Journal of Machine Learning Research 6, 661–694 (2005)
11. Johnson, J.K.: Convex relaxation methods for graphical models: lagrangian and maximum entropy approaches. PhD thesis, Massachusetts Institute of Technology (2008)