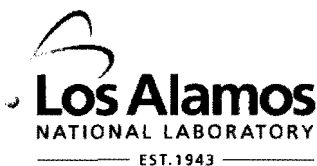


LA-UR- 09-00745

Approved for public release;  
distribution is unlimited.

Title:	Orbit-Product Representation and Correction of Gaussian Belief Propagation
Author(s):	Jason K. Johnson/227983/LANL/T-4 Vladimir Chernyak/Wayne State University Michael Chertkov/149600/LANL/T-4
Intended for:	International Conference on Machine Learning June 14, 2009



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

---

# Orbit-Product Representation and Correction of Gaussian Belief Propagation

---

Jason K. Johnson<sup>a</sup>

Vladimir Y. Chernyak<sup>b,a</sup>

Michael Chertkov<sup>a</sup>

JASONJ@LANL.GOV

CHERNYAK@CHEM.WAYNE.EDU

CHERTKOV@LANL.GOV

<sup>a</sup>Center for Nonlinear Studies and Theoretical Division T-4, LANL, Los Alamos, NM 87544

<sup>b</sup>Department of Chemistry, Wayne State University, Detroit, MI 48202

**Keywords:** Gaussian belief propagation, determinant, walk-sum, orbit-product, zeta function of graph.

## Abstract

We present a new interpretation of Gaussian belief propagation (GaBP) based on the “zeta function” representation of the determinant as a product over orbits of a graph. We show that GaBP captures backtracking orbits of the graph and consider how to correct this estimate by accounting for non-backtracking orbits. We show that the product over non-backtracking orbits may be interpreted as the determinant of the non-backtracking adjacency matrix of the graph with edge weights based on the solution of GaBP. An efficient method is proposed to compute a truncated correction factor including all non-backtracking orbits up to a specified length.

## 1. Introduction

Belief Propagation is a widely used method for inference in graphical models. We study this algorithm in the context of a Gaussian graphical models. There have been several studies of the convergence of Gaussian belief propagation (GaBP) (Weiss & Freeman, 2001; Plarre & Kumar, 2004; Moallemi & Van Roy, 2006) as well as numerous applications (Bickson et al., 2008a, 2008b, 2009). The best known sufficient condition for its convergence is the walk-summable condition (Malioutov et al., 2006; Cseke & Heskes, 2008), which also provides new insights into the algorithm by interpreting it as computing weighted sums of walks (walk-sums) within the graph. Our

aim in this present paper is to extend this graphical/combinatorial view of GaBP to include estimation of the determinant (partition function) of the Gaussian graphical model. This work is also inspired by the loop-series correction method for belief propagation (Chertkov & Chernyak, 2006) that was recently extended to Gaussian graphical models (Chernyak & Chertkov, 2008).

This study leads to a new perspective on GaBP having close ties to graphical zeta functions (Stark & Terras, 1996). We find that in walk-summable models the determinant may be represented as a product over all orbits (periodic walks) of the graph. The estimate of the determinant provided by GaBP is equivalent to taking the product over only the *backtracking* orbits, that is, orbits that can be embedded as orbits in the universal cover (computation tree) of the graph. The remaining orbits may then be grouped into topological equivalence classes corresponding to (strictly) non-backtracking orbits (orbits which *never* backtrack). The orbit-product over each such equivalence class may be simply computed from the solution of GaBP. Also, the product over all non-backtracking orbits may be interpreted as the determinant of a non-backtracking adjacency matrix of the graph (with appropriately defined edge weights based on the GaBP solution). Finally, we propose a simple, efficient method to compute truncated orbit-products including all orbits up to some specified length and provide an error-bound on the resulting estimates. In certain classes of graphs (e.g., grids), this leads to an efficient method with complexity linear in the number of nodes and the required precision of the determinant estimate. This truncation method may also be extended to compute variance corrections and to design an efficient preconditioner for iterative solution of linear systems.

## 2. Gaussian Belief Propagation

A *Gaussian graphical model* is a probability distribution

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

of random variables  $x \in \mathbb{R}^n$  where  $J$  is a sparse, symmetric, positive-definite matrix. The fill-pattern of  $J$  defines a graph  $G$  with vertices  $V = \{1, \dots, n\}$  and edges  $(ij)$  for all  $J_{ij} \neq 0$ . This defines a Markov model over the graph  $G$ . The partition function is defined by  $\mathcal{Z}(h, J) \triangleq \int dx \exp\{-\frac{1}{2} x^T J x + h^T x\}$  so as to normalize the distribution. Given such a model, we may compute the mean vector  $\mu \triangleq \int dp(x) x = J^{-1} h$  and certain elements of the covariance matrix  $K \triangleq \int dp(x) (x - \mu)(x - \mu)^T = J^{-1}$ . This generally requires  $\mathcal{O}(n^3)$  computation using the standard Gaussian elimination method. If only a subset of elements of  $K$  are required, the complexity of sparse Gaussian elimination may be reduced to  $\mathcal{O}(nw^3)$  where  $w$  is the *tree-width* of the graph. In planar graphs, the complexity is bounded by  $\mathcal{O}(n^{3/2})$ .

*Gaussian belief propagation* (GaBP) is a simple, distributed, iterative message-passing algorithm to estimate the marginal distribution of each variable  $x_i \sim \mathcal{N}(\mu_i, K_{ii})$ . It is parameterized by a set of messages  $m_{ij}(x_j) = e^{-\frac{1}{2} x_j^2 \alpha_{ij} + \beta_{ij} x_j}$  defined on each directed arc  $(i, j)$  of the graph. The Gaussian belief propagation equations are:

$$m_{ij}(x_j) \propto \int dx_i \underbrace{\left( \psi_i(x_i) \prod_{k \neq j} m_{ki}(x_i) \right)}_{\psi_{i \setminus j}(x_i)} \psi_{ij}(x_i, x_j)$$

where  $\psi_i = e^{-\frac{1}{2} J_{ii} x_i^2 + h_i x_i}$  and  $\psi_{ij} = e^{-J_{ij} x_i x_j}$ . In terms of  $(\alpha, \beta)$  this is equivalent to:

$$\begin{aligned} \alpha_{ij} &= J_{ij}^2 (J_{ii} - \alpha_{i \setminus j})^{-1} \\ \beta_{ij} &= J_{ij} (J_{ii} - \alpha_{i \setminus j})^{-1} (h_i - \beta_{i \setminus j}) \end{aligned}$$

where  $\alpha_{i \setminus j} \equiv \sum_{k \neq j} \alpha_{ki}$  and  $\beta_{i \setminus j} \equiv \sum_{k \neq j} \beta_{ki}$  ( $\psi_{i \setminus j} = e^{-\frac{1}{2} \alpha_{i \setminus j} x_i^2 + \beta_{i \setminus j} x_i}$ ). These equations are solved by iteratively recomputing each message from the other messages until convergence. The marginal distribution is then estimated as  $p^{\text{bp}}(x_i) \propto \psi_i(x_i) \prod_k m_{ki}(x_i)$ , which gives variance estimates  $K_i^{\text{bp}} = (J_{ii} + \sum_k \alpha_{ki})^{-1}$  and mean estimates  $\mu_i^{\text{bp}} = K_i^{\text{bp}} (h_i + \sum_k \beta_{ki})$ . In trees, this method is equivalent to Gaussian elimination, terminates after a finite number of steps and then provides the correct marginals. In loopy graphs, it may be viewed as

performing Gaussian elimination in the computation tree (universal cover) of the graph (see (Plarre & Kumar, 2004; Malioutov et al., 2006) for details) and may therefore fail to converge due to the infinite extent of the computation tree. If it does converge, the mean estimates are still correct but the variances are only approximate. We also obtain an estimate of the pairwise covariance matrix on edges  $\{i, j\} \in G$  as

$$K_{(ij)}^{\text{bp}} = \begin{pmatrix} J_{ii} - \alpha_{i \setminus j} & J_{ij} \\ J_{ij} & J_{jj} - \alpha_{j \setminus i} \end{pmatrix}^{-1}.$$

In this paper, we are concerned with the BP estimate of the determinant  $Z = \det K = \det J^{-1}$  (which is closely linked to computation of the partition function  $\mathcal{Z}$ ). We obtain an estimate of  $Z$  from the BP solution as:

$$Z^{\text{bp}} = \prod_{i \in V} Z_i^{\text{bp}} \prod_{\{i, j\} \in G} \frac{Z_{ij}^{\text{bp}}}{Z_i^{\text{bp}} Z_j^{\text{bp}}} \quad (2)$$

where  $Z_i^{\text{bp}} = K_i^{\text{bp}}$  and  $Z_{ij}^{\text{bp}} = \det K_{(ij)}^{\text{bp}}$ . Note that this only depends on the parameters  $\alpha$  (for the remainder of the paper we focus of the “zero-mean” version of GaBP, with  $h = 0$ ,  $\beta = 0$  and  $\mu = 0$ ). In fact, the GaBP equations for  $\alpha$  may be derived as solving for a *saddle point* of the quantity  $Z^{\text{bp}}$  with respect to  $\alpha$ . The motivation for this form of estimate is that if  $G$  is a tree then there is a unique saddle-point and it then holds that  $Z^{\text{bp}} = Z$ . In loopy graphs, there may generally be no saddle point (over the real-valued messages) or multiple saddle-points. In walk-summable models, there exists at least one saddle-point based on the walk-sum fixed point of GaBP (to the best of our knowledge this is the only stable fixed point of GaBP in walk-summable models). The main objective of this paper is to interpret this estimate and to suggest a method to correct it.

**Walk-Sum Interpretation** Our approach in this paper may be considered as an extension of the walk-sum interpretation of GaBP (Malioutov et al., 2006). Let  $J$  be normalized to have unit-diagonal, such that  $J = I - R$  with  $R$  having zeros along its diagonal. The walk-sum idea is based on the series  $K = \sum_k R^k$ , which converges if  $\rho(R) < 1$  where  $\rho(R)$  denotes the *spectral radius* of the matrix  $R$  (the maximum modulus of the eigenvalues of  $R$ ). This allows us to interpret  $K_{ij}$  as a sum over all walks in the graph  $G$  which begin at node  $i$  and end at node  $j$  where the weight of a walk is defined as  $R^w = \prod_{(ij) \in w} r_{ij}^{n_{ij}(w)}$  and  $n_{ij}(w)$  is a count of how many times the step  $(ij)$  occurs in the walk. We

write this *walk-sum* as  $K_{ij} = \sum_{w:i \rightarrow j} R^w$ . However, in order for the walk-sum to be well-defined, it must converge to the same value regardless of the order in which we add the walks. This is equivalent to requiring that it converges absolutely. Thus, we say that  $R$  is *walk-summable* if  $\sum_{i \rightarrow j} |R^w|$  converges for all  $i, j \in V$ . This is equivalent to the spectral condition that  $\rho(|R|) < 1$ . A number of other equivalent or sufficient conditions are given in (Malioutov et al., 2006).

In walk-summable models it then holds that variances correspond to closed walk-sums  $K_{ii} = \sum_{w:i \rightarrow i} R^w$  and means correspond to a (reweighted) walk-sum over all walks which end at a specific node  $\mu_i = \sum_{w:* \rightarrow i} h_* R^w$  (here  $*$  denotes the arbitrary starting point of the walk). Moreover, we may then interpret the GaBP message parameters  $(\alpha, \beta)$  as recursively computing these walk-sums within the *computation tree* (see (Malioutov et al., 2006) for details)<sup>1</sup>. This then implies that GaBP converges in walk-summable models and always converges to the same “walk-sum” solution independent of the order in which we update messages. This interpretation also shows that GaBP computes the correct walk-sums for the means but only computes a subset of the closed walks needed for the variances. Specifically,  $K_i^{\text{bp}}$  only includes the *backtracking* walks of the graph, which are closed walks which may be embedded as closed walks in the computation tree. The “missing” walks (e.g., walks which go around a cycle of the graph once and then stop) may be grouped into equivalence classes corresponding to paths of the computation tree connecting replicas of a node of the original graph. These paths of the computation tree correspond to strictly non-backtracking closed walks of the graph.

### 3. Determinant as Orbit-Product

Let  $Z(R) \triangleq \det(I - R)^{-1}$ . In walk-summable models, we may give this determinant another graphical interpretation as a product over orbits of a graph, one closely related to the so-called *zeta function* of a graph (Stark & Terras, 1996). Let  $G$  be a graph of  $n$  vertices and edges  $\{i, j\}$  for all  $r_{ij} \neq 0$ . A *walk* is a sequence of adjacent nodes of  $G$ , or (equivalently) a sequence of directed edges such that the endpoint of each edge equals the starting point of the next edge in the sequence. Walks may visit the same node or cross the same edge multiple times and are also allowed to *backtrack* (that is, it may step back to the

preceding vertex). A walk is *closed* if it begins and ends at the same vertex. A closed walk is *primitive* if it is not a multiple of some shorter walk. Two closed walks may be considered as equivalent if one is a periodic shift of the other. We define the *orbits* of a graph to be the set of equivalence classes of primitive closed walks. Thus, there is a one-to-one correspondence between orbits and (infinitely long) periodic walks.

**Theorem 1** *If  $\rho(|R|) < 1$  then it holds that  $Z(R) = \prod_{\ell} (1 - R^{\ell})^{-1} \triangleq \prod_{\ell} Z_{\ell}$  where the product is taken over all orbits of  $G$  and  $R^{\ell} = \prod_{(i,j) \in \ell} r_{ij}^{n_{ij}(\ell)}$  where  $n_{ij}(\ell)$  is the number of times orbit  $\ell$  steps across edge  $(i, j)$  in one circuit of the orbit.*

*Proof.*  $\log \det(I - R)^{-1} = \text{tr} \log(I - R)^{-1} = \text{tr} \sum_k \frac{R^k}{k} = \sum_{\text{closed } w} \frac{|R^w|}{|w|} = \sum_{\text{primitive } w} \sum_{m=1}^{\infty} \frac{(R^w)^m}{m|w|} = \sum_{\text{primitive } w} \frac{1}{|w|} \log(1 - R^w)^{-1} = \sum_{\text{orbits } \ell} \log(1 - R^{\ell})^{-1} = \log \prod_{\ell} (1 - R^{\ell})^{-1}$ . We have used the identity  $\log \det A = \text{tr} \log A$  and the series expansion  $\log(I - A)^{-1} = \sum_{k=1}^{\infty} \frac{A^k}{k}$ . Each closed walk is expressed as a multiple of a primitive walk. Every primitive walk  $w$  has exactly  $|w|$  distinct periodic shifts.  $\diamond$

### 4. $Z^{\text{bp}}$ as Backtracking Orbit-Product

Backtracking walks play an important role in the walk-sum interpretation of Gaussian belief propagation. We now derive an analogous interpretation of  $Z^{\text{bp}}$ . We say that an orbit is *backtracking* if it may be embedded as an orbit of the computation tree.

**Theorem 2** *If  $\rho(|R|) < 1$  then Gaussian belief propagation converges and  $Z^{\text{bp}} = \prod_{\ell \in \mathcal{L}_0} Z_{\ell}$  where  $\mathcal{L}_0$  is the set of backtracking orbits of  $G$ .*

Before proving the theorem, we derive several useful lemmas. Let  $A = (A_{11}A_{12}; A_{21}A_{22})$ . The *Schur complement* of block  $A_{11}$  is  $A_{22}^* = A_{22} - A_{21}A_{11}^{-1}A_{12}$ . The determinant is  $\det A = \det A_{11} \det A_{22}^*$ . Also,  $(A_{22}^*)^{-1} = (A^{-1})_{22}$ . Using these facts, it is simple to show:

**Lemma 1** *Let  $R = (R_{11}R_{12}; R_{21}R_{22})$ ,  $K = (I - R)^{-1} = (K_{11}K_{12}; K_{21}K_{22})$ . Then  $\det K_{11} = \frac{Z(R)}{Z(R_{22})}$ .*

Then, for  $\rho(|R|) < 1$ , we have

$$\det K_{11} = \frac{\prod_{\ell} Z_{\ell}}{\prod_{\ell \in G_1} Z_{\ell}} = \prod_{\ell \text{ intersects } G_1} Z_{\ell}$$

where the resulting orbit-product is over all orbits of  $G$  which intersect the subgraph  $G_1$  corresponding

<sup>1</sup>This analysis assumes that the messages are initially set to zero.

to submatrix  $R_{11}$ . We may then apply this principle to interpret  $Z_i^{\text{bp}}$  and  $Z_{ij}^{\text{bp}}$  as orbit-products in the computation tree. Let  $T_i$  denote the computation tree of the graph  $G$  with one particular copy of node  $i$  marked. Let  $T_{ij}$  denote the computation tree with one copy of edge  $\{i, j\} \in G$  marked. Then,

**Lemma 2**  $Z_i^{\text{bp}} = \prod_{\ell \in T_i, i \in \ell} Z_\ell$  where the product is over all orbits of  $T_i$  which intersect the marked node  $i$ . Also,  $Z_{ij}^{\text{bp}} = \prod_{\ell \in T_{ij}, i \in \ell \vee j \in \ell} Z_\ell$  where the product is over all orbits of  $T_{ij}$  which intersect either endpoint of the marked edge  $\{i, j\}$ .

*Proof of Theorem 2.* Now,  $Z^{\text{bp}} = \prod_i Z_i^{\text{bp}} \prod_{ij} \frac{Z_{ij}^{\text{bp}}}{Z_i^{\text{bp}} Z_j^{\text{bp}}}$  can be expressed entirely in terms of orbits of the computation tree which correspond to backtracking orbits of the graph:

$$Z^{\text{bp}} = \prod_{\ell \in \mathcal{L}_0} Z_\ell^{N_\ell}$$

where  $N_\ell$  is an integer that gives the number of times  $\ell$  appears in the orbit-product. It remains to show that  $N_\ell = 1$  for each backtracking orbit. Each backtracking orbit traces out a subtree  $T_\ell$  within the computation tree  $T$ . Let  $v$  denote the number of vertices in this subtree and let  $e$  denote the number of edges. Also, let  $c$  denote the number of cut edges along the boundary of the subtree. First, we count how many powers of  $Z_\ell$  appear in the orbit product  $\prod_i Z_i^{\text{bp}}$ . For each vertex of  $T_\ell$  we may pick this as the marked node in the computation tree and this shows one way that  $\ell$  can be embedded in the computation tree  $T_i$  of the corresponding node  $i$  of the original graph. Thus, the total number of multiples of  $Z_\ell$  in  $\prod_i Z_i^{\text{bp}}$  is precisely  $v$ . Similarly, we could mark any edge of  $T_\ell$  or of its cut edges and this gives one way to embed  $\ell$  into the corresponding computation tree  $T_{ij}$ . Thus, the product  $\prod_{ij} Z_{ij}^{\text{bp}}$  contributes  $e + c$  powers of  $Z_\ell$ . Lastly, the product  $\prod_{ij} Z_i^{\text{bp}} Z_j^{\text{bp}}$  contains  $2e + c$  powers of  $Z_\ell$ . This represents the number of ways we may pick an ordered edge  $(i, j)$  in the computation tree such that the first endpoint  $i$  is a node of  $T_\ell$ . The total count is then

$$N_\ell = v + (e + c) - (2e + c) = v - e = 1$$

because  $e = v - 1$  for the tree  $T_\ell$ .  $\diamond$

Combining Theorems 1 and 2, we obtain:

**Corollary 1**  $Z = Z^{\text{bp}} \times \prod_{\ell \notin \mathcal{L}_0} Z_\ell$ .

This implies that  $Z = Z^{\text{bp}}$  for trees since all orbits of trees are backtracking.

Let  $g$  denote the *girth* of the graph  $G$ , defined as the length of the shortest cycle of  $G$ . Then, we derive the GaBP error bound:

**Corollary 2**  $\frac{1}{n} \left| \log \frac{Z^{\text{bp}}}{Z} \right| \leq \frac{\rho(|R|)^g}{g(1-\rho(|R|))}.$

*Proof.* We use the inequality:  $|\log Z_\ell| = \left| \sum_{k=0}^{\infty} \frac{(R^\ell)^k}{k} \right| \leq \sum_{k=0}^{\infty} \frac{(|R|^\ell)^k}{k} \triangleq \log \hat{Z}_\ell$ . Then,  $|\log \frac{Z}{Z^{\text{bp}}}| = \left| \sum_{\ell \notin \mathcal{L}_0} \log Z_\ell \right| \leq \sum_{|\ell| \geq g} |\log Z_\ell| \leq \sum_{|\ell| \geq g} \log \hat{Z}_\ell = \text{tr} \sum_{k=g}^{\infty} \frac{|R|^k}{k} \leq \frac{n\rho^g}{g(1-\rho)}.$   $\diamond$

## 5. Non-Backtracking Orbit-Product

In this section we show that the set of orbits omitted in the BP estimate can be grouped into topological equivalence classes corresponding to non-backtracking orbits (orbits which *never* backtrack) and that the orbit-product over each such equivalence class is simply computed with the aid of the BP solution. Our ultimate goal in this section is to demonstrate:

**Theorem 3**  $Z = Z^{\text{bp}} \times \prod_\gamma Z'_\gamma$  where the product is over all non-backtracking orbits of  $G$  and we define

$$Z'_\gamma = (1 - \prod_{(ij) \in \gamma} (r'_{ij})^{\alpha_{ij}(\gamma)})^{-1}$$

where  $r'_{ij} \triangleq \frac{r_{ij}}{1-\alpha_{i \setminus j}}$  and  $\alpha_{i \setminus j} = \sum_{k \neq j} \alpha_{ki}$  is computed from the solution of GaBP.

First, we describe the partitioning of orbits into equivalence classes based on non-backtracking orbits. We define a map  $\phi$  from the set of all orbits to the subset of non-backtracking orbits. This map is called the *backtrack-deletion* operator. Given an orbit  $\ell$ , the backtrack-deleted orbit  $\phi(\ell)$  is obtained by iteratively deleting backtracking sub-orbits within the orbit  $\ell$ . Let  $\ell$  be expressed in terms of its steps:  $[(\ell_1 \ell_2)(\ell_2 \ell_e) \cdots (\ell_L \ell_1)]$ . The backtrack-deletion operator removes pairs of steps of the form  $(ij)(ji)$  until there are no more such backtracking steps. The resulting irreducible orbit is then non-backtracking. It can be seen that the result of the procedure is invariant to the order in which we delete backtracking sub-orbits. Thus, each non-backtracking orbit  $\gamma$  defines an equivalence class of orbits  $\mathcal{L}_\gamma \triangleq \{\ell | \phi(\ell) = \gamma\}$  which are topologically equivalent in that any orbit of this class may be transformed to any other orbit of the same class through a series of insertions/deletions of backtracking orbits. Also note that the set of backtracking orbits  $\mathcal{L}_0$  correspond to one such equivalence class, the set of all orbits

which reduce to the empty orbit  $\emptyset$  under the back-track deletion procedure. The remaining orbits may then be naturally grouped according to (non-trivial) non-backtracking orbits.

It remains to understand how to efficiently compute the orbit-product over an equivalence class. As a first step toward this goal, we map all orbits of an equivalence class  $\mathcal{L}_\gamma$  into an associated cover  $G_\gamma$  of the graph  $G$ , defined to be the unique cover of the graph  $G$  which contains exactly one directed cycle based on the non-backtracking orbit  $\gamma$ . This is equivalent to the following construction. First, we start with a single directed graph comprised of the cycle  $\gamma = [\gamma_1 \gamma_2 \dots \gamma_L]$  (any duplicated nodes of the orbit map to distinct nodes in this directed cycle). Then, for each node  $\gamma_k$  of this graph, we attach a copy to the computation tree  $T_{\gamma_k \setminus \gamma_{k+1}}$ , obtained by taking the full computation tree  $T_{\gamma_k}$  rooted at node  $\gamma_k$  and deleting the subtree beneath the branch  $(\gamma_k, \gamma_{k+1})$  incident to the root. This construction is illustrated in Figure 1(a,b) for the graph  $G = K_4$  and orbit  $\gamma = [(12)(23)(31)]$ . Note that edges of the cycle are “one-way” and those within each subtree are “two-way”. This defines a cover of  $G$  in the sense that the set of outward edges at each node of  $G_\gamma$  is isomorphic to the outward edges at the corresponding node of  $G$ . The importance of this graph is based on the following topological lemma:

**Lemma 3** *Let  $\gamma$  be a non-backtracking orbit of  $G$ . Then, there is a one-to-one correspondence between the class of orbits  $\mathcal{L}_\gamma = \{\ell | \phi(\ell) = \gamma\}$  of  $G$  and the non-trivial orbits of the graph  $G_\gamma$ .*

*Proof.* We aim to show that each orbit  $\ell \in \mathcal{L}_\gamma$  may be embedded in the graph  $G_\gamma$  in exactly one way. We use the equivalence between orbits and infinitely long periodic walks. We trace the periodic walk  $w$  described by  $\ell$  in the computation tree (universal cover)  $T$  of  $G$ . This walk reduces to an infinitely long path  $w' = \phi(w)$  of the computation tree. This path  $w'$  is also the periodic walk corresponding to the non-backtracking orbit  $\gamma = \phi(\ell)$ . Now, we break up the walk  $w$  into backtracking segments as follows. For each edge of the path  $w'$  in the computation tree, the walk  $w$  crosses this edge some number of times (once more in the forward direction than in the reverse direction). We mark the step of  $w$  corresponding to the first time that it crosses this edge. After doing this for every edge of  $w'$ , we will have broken up the walk  $w$  into segments as  $w = \dots (\gamma_1 \gamma_2) w_2 (\gamma_2 \gamma_3) w_3 \dots (\gamma_L \gamma_1) w_1 \dots$  (where this pattern is repeating). It is simple to observe that each walk  $w_k$  may be embedded as a closed-walk

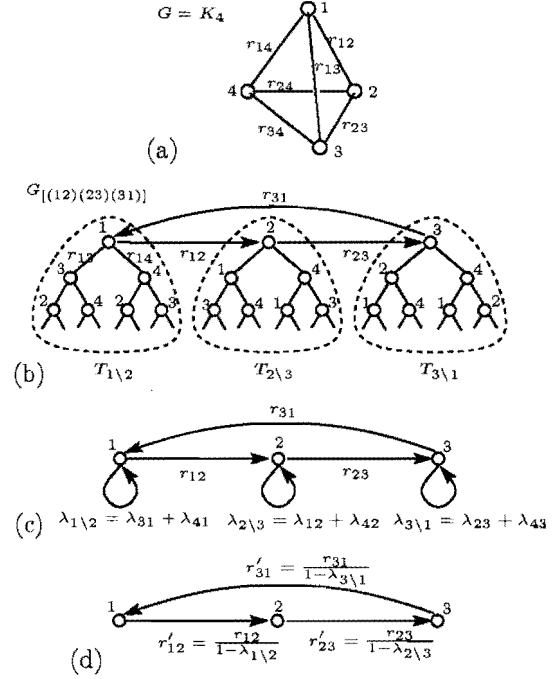


Figure 1. Illustration of construction to combine topologically equivalent orbits. (a) The graph  $G = K_4$ . (b) The computation graph  $G_\gamma$  for  $\gamma = [(12)(23)(31)]$ . (c) Finite graph with self-loops at each node to capture backtracking walks. (d) Equivalent graph with modified edge weights to capture backtracking walks.

at the root node of  $T_{\gamma_k \setminus \gamma_{k+1}}$ . Thus, this decomposition of the periodic walk  $w$  also determines how  $\ell$  may be embedded into  $G_\gamma$ . This defines a map  $\tau$  from  $\mathcal{L}_\gamma$  to the non-trivial orbits of  $G_\gamma$ . One can verify that this mapping is bijective. The inverse map is defined by  $\tau^{-1}(\ell') = [\nu(\ell'_1) \nu(\ell'_2) \dots \nu(\ell'_L)]$  where  $\nu$  is map from (distinct) vertices of  $G_\gamma$  back to the corresponding vertices of  $G$ .  $\diamond$

Next, we demonstrate how to compute all of the orbits within an equivalence class as a simple determinant calculation based on the non-backtracking orbit  $\gamma$  and the BP solution. Let  $R'_\gamma$  be defined as the edge-weight matrix a simple single-loop graph based on  $\gamma$  with edge-weights defined by  $r'_{\gamma_k, \gamma_{k+1}} = \frac{r_{\gamma_k, \gamma_{k+1}}}{1 - \alpha_{\gamma_k \setminus \gamma_{k+1}}}$ . This construction is illustrated in Figure 1(d). Then,

**Lemma 4**  $Z'_\gamma \triangleq \det(I - R'_\gamma)^{-1} = \prod_{\ell \in \mathcal{L}_\gamma} Z_\ell$ .

*Proof.* Using Lemma 3, we see that the orbit-product  $\prod_{\ell \in \mathcal{L}_\gamma} Z_\ell$  is equal to the product over all non-trivial orbits of the graph  $G_\gamma$ , that is, the product over all orbits in  $G_\gamma$  which intersect the subgraph corresponding to  $\gamma$ . Using Lemma 1, this is equivalent to computing the determinant of the cor-

responding submatrix of  $K_{G_\ell} = (I - R_{G_\ell})^{-1}$  where  $R_{G_\ell}$  is the edge-weight matrix of the infinite computation graph. This is equivalent to first eliminating the subtrees (by Gaussian elimination/belief propagation) beneath each root node along the loop  $\gamma$  of  $G_\gamma$ , to obtain the reduced graphical model  $I - R_\gamma$  and then computing  $\det(I - R_\gamma)^{-1}$ . Using the GaBP solution, the effect of eliminating each subtree is to add a “self-loop” (diagonal element) to  $R_\gamma$  with edge-weight  $\alpha_{\gamma_k \setminus \gamma_{k+1}} = \sum_{v \neq \gamma_{k+1}} \alpha_{v, \gamma_k}$ , obtained by summing the incoming messages to node  $\gamma_k$  from each of its neighbors in the subtree  $T_{\gamma_k \setminus \gamma_{k+1}}$ . This elimination step is illustrated in Figure 1(b,c). We may use the orbit-product formula to compute the determinant. However, there are infinitely many orbits in this graph due to the presence of a self-loop at each of the remaining nodes. At each node, an orbit may execute any number of steps  $m$  around this self-loop each with edge-weight  $\alpha_{\gamma_k \setminus \gamma_{k+1}}$ . Summing these up, we obtain  $\sum_{m=0}^{\infty} \alpha_{\gamma_k \setminus \gamma_{k+1}}^m = (1 - \alpha_{\gamma_k \setminus \gamma_{k+1}})^{-1}$ . The convergence of this series ( $|\alpha_{\gamma_k \setminus \gamma_{k+1}}| < 1$ ) is assured by the walk-summable condition. Thus, it is equivalent (preserves the determinant) to deleting each self-loop and multiply the weight of the following edge by  $(1 - \alpha_{\gamma_k \setminus \gamma_{k+1}})^{-1}$ . This final reduction step is illustrated in Figure 1(c,d). Then, the orbit-product  $\prod_{\ell \in \mathcal{L}_\gamma} Z_\ell$  is equal to the determinant  $\det(I - R'_\gamma)^{-1}$  (e.g., based on the graph seen in Figure 1(d)).  $\diamond$

It is straight-forward to compute this determinant.

**Lemma 5**  $Z'_\gamma = (1 - (R')^\gamma)^{-1}$  where  $(R')^\gamma = \prod_{(ij) \in \gamma} (r'_{ij})^{n_{ij}(\gamma)}$  and  $r'_{ij} = r_{ij}(1 - \alpha_{i \setminus j})^{-1}$ .

*Proof of Theorem 3.* Using these lemmas, it is now simple to show  $\frac{Z}{Z^{\text{bp}}} = \prod_{\ell \in \mathcal{L}_\emptyset} Z_\ell = \prod_{\gamma \neq \emptyset} \prod_{\ell \in \mathcal{L}_\gamma} Z_\ell = \prod_{\gamma \neq \emptyset} \det(I - R'_\gamma)^{-1} = \prod_{\gamma \neq \emptyset} (1 - (R')^\gamma)^{-1}$ , which proves the theorem.  $\diamond$

## 6. Non-Backtracking Determinant

Next, we show that the correction factor

$$\frac{Z}{Z^{\text{bp}}} = \prod_{\gamma \neq \emptyset} Z'_\gamma = \prod_{\gamma \neq \emptyset} \det(I - R'_\gamma)^{-1}$$

may also be computed as a *single determinant calculation* based on the following *non-backtracking matrix* of the graph. We define  $R' \in \mathbb{R}^{2|G| \times 2|G|}$  as follows. Let the rows and columns of  $R'$  be indexed by directed arcs  $(ij)$  of the graph  $G$  (each symmetric edge  $\{i, j\} \in G$  is split into two directed edges  $(ij)$

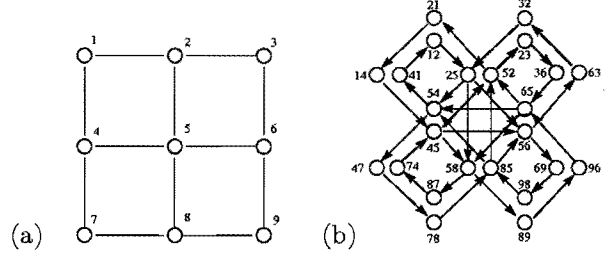


Figure 2. (a)  $3 \times 3$  grid  $G$ . (b) Graph  $G'$  representing the non-backtracking adjacency matrix  $R'$ . Each node  $ij$  represents a directed edge of  $G$ , arcs are drawn between nodes  $ij$  and  $ji$  which are non-backtracking ( $k \neq i$ ).

and  $(ji)$ ). Then, the elements of  $R'$  are defined

$$R'_{(ij),(kl)} = \begin{cases} r'_{kl}, & j = k \text{ and } i \neq l \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

This construction is illustrated in Figure 2. Note that the walks generated by taking powers  $R'$  correspond to non-backtracking walks of the graph  $G$ . The weight of an arc  $((ij)(jk))$  in  $R'$  is defined as the (modified) edge-weight  $r'_{jk}$  of the endpoint  $(jk)$ . The weight of an orbit in  $R'$  may then be equivalently defined as the product of *node* weights  $r'_{ij}$  taken over the orbit in  $R'$ , which is equal to the weight of the corresponding non-backtracking orbit of  $G$  (using the modified edge weights  $r'_{ij}$ ).

Let  $Z' = \det(I - R')^{-1}$ . We will prove the following:

**Theorem 4**  $Z = Z^{\text{bp}} \times Z'$ , that is,  $\det(I - R)^{-1} = Z^{\text{bp}} \times \det(I - R')^{-1}$ .

Before providing the proof, we present two useful lemmas. First, we establish that walk-summability with respect to  $R$  implies walk-summability with respect to  $R'$ .

**Lemma 6** If  $\rho(|R|) < 1$  then  $\rho(|R'|) \leq \rho(|R|)$ .

*Proof Sketch.* We note that once the quadratic parameters  $\alpha$  converge in GaBP, the remaining linear parameters  $\beta$  follow a linear systems  $\beta_{k+1} = R'\beta_k + b$  (here, we consider the “parallel” update scheme, where all messages at step  $k+1$  are independently computed from the messages at step  $k$ ). Thus, the asymptotic convergence rate of GaBP is  $\rho(R')$ . We compare this to the simpler Gauss-Jacobi (GJ) iteration  $\mu_{k+1} = \mu_k + (h - J\mu_k)$ , which has convergence rate  $\rho(R)$ . From the walk-sum interpretation of both algorithms, it is clear that the GaBP iteration captures a superset of those walks computed by GJ (at each iteration). Hence, for non-negative models ( $R \geq 0$  and  $h \geq 0$ ) it must hold that the error in the GaBP estimate of  $\mu$  is less than or equal



to the error of GJ (at every iteration). This implies  $\rho(R') \leq \rho(R)$  if  $R \geq 0$ . Even if  $R$  is not non-negative, it still holds  $\rho(|R'|) \leq \rho(|R|)$ . This is seen as we could run GaBP/GJ on the model  $J' = I - |R|$  instead and then apply the same error bounds.  $\diamond$

We also note:

**Lemma 7** *There is a one-to-one correspondence between the set of all orbits of  $G'$  and the non-backtracking orbits of  $G$ .*

*Proof of Theorem 4.* Then, we may again apply the orbit-product representation of  $Z'$  as a product over orbits of  $G'$ , which is equivalent to a product over non-backtracking orbits of  $G$ :

$$Z' = \prod_{\gamma \neq \emptyset} \left( 1 - \prod_{(ij) \in \gamma} (r'_{ij})^{n_{ij}(\gamma)} \right)^{-1} \quad (4)$$

Lastly, we note that the weight of each of the orbits above is exactly equal to the determinant  $Z'_\gamma = \det(I - R'_\gamma)$  corresponding to the previously defined single-orbit matrices  $R'_\gamma$ . Thus,  $Z' = \prod_{\gamma \neq \emptyset} Z'_\gamma$ .  $\diamond$

Note that the error bound of Corollary 2 can now be improved to  $\frac{1}{n} \left| \log \frac{Z^{\text{bp}}}{Z} \right| = \frac{1}{n} |\log Z'| \leq \frac{\rho(|R'|)^g}{g(1-\rho(|R'|))}$ . Of course it is impractical to actually compute the complete correction factor  $Z' = \det(I - R')^{-1}$ , as this is not easier than directly computing  $Z = \det(I - R)^{-1}$ . However, this representation does suggest a useful approximation based on the method of the next section.

## 7. Efficient Truncation Method

Next, we consider an efficient method to approximate  $Z(A) = \det(I - A)^{-1}$  for walk-summable matrices  $\rho(|A|) < 1$ . This method be used to either directly approximate  $Z(R)$  or to approximate the GaBP correction  $Z(R')$ .

Given a graph  $G$  based on vertices  $V$ , we specify a set of blocks  $\mathcal{B} = (B_k \subset V, k = 1, \dots, |\mathcal{B}|)$  chosen such that: (1) Every short orbit  $|\ell| < L$  is covered by some block  $B \in \mathcal{B}$ , and (2) If  $B, B' \in \mathcal{B}$  then  $B \cap B' \in \mathcal{B}$ . We also define block weights  $w_B$  as follows: Let  $w_B = 1$  for maximal blocks (not contained by any other block) and define  $w_B = 1 - \sum_{B' \supset B} w_{B'}$  for non-maximal blocks (these weights may be negative). This definition insures that  $\sum_{B' \supset B} w_{B'} = 1$  for each  $B \in \mathcal{B}$ . Then, we define our estimate

$$Z_B \triangleq \prod_B Z_B^{w_B} \triangleq \prod_B (\det(I - A_B)^{-1})^{w_B} \quad (5)$$

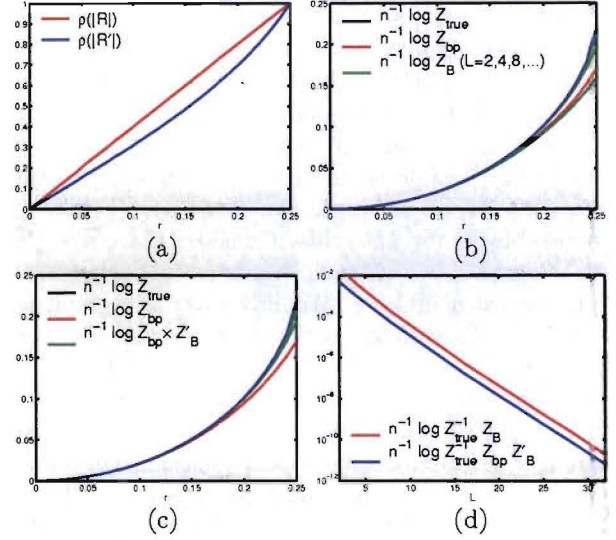


Figure 3. Demonstration of determinant approximation method for  $256 \times 256$  periodic grid with uniform edge weights  $r \in [0, .25]$ . Plots of (a)  $\rho(|R|)$  and  $\rho(|R'|)$  vs  $r$ ; (b)  $\frac{1}{n} \log$  of  $Z$ ,  $Z^{\text{bp}}$  and  $Z_B$  (with  $L = 2, 4, 8, 16, 32$ ) vs  $r$ ; (c)  $\frac{1}{n} \log(Z^{\text{bp}} Z'_B)$  vs  $r$ ; and (d)  $\frac{1}{n} \log(Z^{-1} Z_B)$  and  $\frac{1}{n} \log(Z^{-1} Z^{\text{bp}} Z'_B)$  vs  $L = 2, 4, 8, 16$  for  $r = .2$ .

where  $A_B$  denotes the  $|B| \times |B|$  principle submatrix of  $A$  corresponding to  $B$ . This estimate computes a truncated orbit-product:

**Theorem 5** *Let  $\rho(|R|) < 1$ . Then,  $Z_B = \prod_{\ell \in \mathcal{L}_B} Z_\ell$  where  $\mathcal{L}_B = \cup_{B \in \mathcal{B}} \mathcal{L}_B$  and  $\mathcal{L}_B$  is the set of all orbits covered by  $B$ .*

*Proof.*  $Z_B = \prod_B \prod_{\ell \in B} Z_\ell^{w_B} = \prod_{\ell \in \mathcal{L}_B} Z_\ell^{\sum_{B \supset \ell} w_B} = \prod_{\ell \in \mathcal{L}_B} Z_\ell$  where  $\sum_{B \supset \ell} w_B = 1$  follows from the definition of the block weights.  $\diamond$

Next, we use the fact that  $\mathcal{L}_B$  includes all short orbits to bound the weight of the missing orbits:

**Corollary 3**  $\frac{1}{n} \left| \log \frac{Z_B}{Z} \right| \leq \frac{\rho(|A|)^L}{L(1-\rho(|A|))}$ .

Thus, for the class of models with  $\rho < 1$ , we obtain an approximation scheme which converges to the correct determinant as the parameter  $L$  is made large with error decaying exponentially in  $L$ . The estimate  $Z_B(R)$  includes all orbits that are covered by some block. The improved BP-based estimate  $Z^{\text{bp}} Z_B(R')$  includes all orbits which are topologically equivalent to any orbit covered by a block. Thus, the BP-based correction includes a much larger subset of orbits. We also note that the error bound  $\rho^L$  generally decays faster using the BP-based estimate as it holds that  $\rho(|R'|) \leq \rho(|R|)$ .



**Construction of  $\mathcal{B}$  for Grids** To achieve an error bound  $\frac{1}{n} |\log \frac{Z_{\mathcal{B}}}{Z}| \leq \varepsilon$  we must choose  $L \sim \log \varepsilon^{-1}$ . Then, the computation needed to achieve this precision will depend on both the number of blocks and the block size needed to cover all orbits up to this size. In certain classes of sparse graphs, it should be possible to control the complexity of the method. As an example, we demonstrate how to choose blocks for 2D grids. Consider the  $\sqrt{n} \times \sqrt{n}$  square grid in which each vertex is connected to its four nearest neighbors. We may cover this graph by  $L \times L$  blocks shifted (both vertically and horizontally) in increments of  $\frac{L}{2}$  (let  $L$  be even). It can be seen that this set of blocks covers all loops shorter than  $L$ . To include all intersections of blocks, we add  $L \times \frac{L}{2}$ ,  $\frac{L}{2} \times L$  and  $\frac{L}{2} \times \frac{L}{2}$  blocks. The block weights are  $w_{L \times L} = 1$ ,  $w_{L \times L/2} = w_{L/2 \times L} = -1$  and  $w_{L/2 \times L/2} = 1$ . The complexity of computing the determinant of an  $L \times L$  block is  $\mathcal{O}(L^3)$  and the total number of blocks is  $\mathcal{O}(n/L^2)$ . Hence, the total complexity is  $\mathcal{O}(nL) = \mathcal{O}(n \log \varepsilon^{-1})$ .

We test our approach numerically on an  $256 \times 256$  square grid (with periodic boundary conditions). We set all edge weights to  $r$  and test the quality of approximation using both estimates  $Z_{\mathcal{B}}(R)$  and  $Z^{\text{bp}} Z_{\mathcal{B}}(R')$  for  $r \in [0, .25]$  ( $J = I - R$  becomes singular/indefinite for larger values of  $r$ ) and block sizes  $L = 2, 4, 8, 16, 32$ . The results are shown in Figure 3. As expected, accuracy rapidly improves with increasing  $L$  in both methods and the BP-correction is the more accurate method.

## 8. Conclusion and Future Work

We have demonstrated an orbit-product (zeta function) representation of the determinant (the partition function of the Gaussian model) and interpreted the estimate obtained by Gaussian belief propagation as corresponding to backtracking orbits. Further, we have shown how to compute the remaining non-backtracking orbits as a determinant calculation of the non-backtracking walk matrix and demonstrated an efficient approach to compute truncated orbit-products in sparse graphs (demonstrated for grids). This method also extends to approximate the matrix inverse (the covariance matrix of the Gaussian model), which may in turn be used as an efficient preconditioner for iterative solution of linear systems. Due to space limitations, we leave these extensions for a longer report.

In future work, we plan to extend the method of constructing an efficient set of blocks to other classes of sparse graphs besides grids (e.g., planar graphs,

graphs of locally-bounded treewidth and graphs with low doubling-dimension). In another direction, we plan to explore multiscale extensions of the truncation scheme. The basic idea is to estimate the contribution of longer orbits using a coarse-grained version of the Gaussian graphical model. A related idea is to explore methods to “bootstrap” GaBP to capture longer orbits, based on the factorization  $Z(R) = (\prod_{k=0}^{K-1} Z(-R^{2^k}))^{-1} Z(R^{2^K})$ . Thus, by applying GaBP to powers of  $R$  we may capture some of the non-backtracking orbits of the graph. Another research direction is to investigate integral representations of the formula  $Z = Z^{\text{bp}} Z'$  for *any* fixed point of GaBP (using methods more along the lines of (Chernyak & Chertkov, 2008)), which could prove useful beyond the class of walk-summable models.

## References

- Bickson, Shental, Wolf, & Dolev (2008a). Gaussian belief propagation based multiuser detection. *IEEE Int. Symp. on Inform. Theory*.
- Bickson, Shental, Wolf, & Dolev (2008b). A Gaussian belief propagation solver for support vector machines. *5th European Complex Systems Conf.*
- Bickson, Tock, Zymnis, Boyd, & Dolev (2009). Distributed large scale network utility maximization. *Int. Sym. on Information Theory (in review)*.
- Chernyak, & Chertkov (2008). Fermions and loops on graphs I: Loop calculus for determinant. *J. Statistical Mechanics: Theory and Experiments*.
- Chertkov, & Chernyak (2006). Loop series for discrete statistical models on graphs. *J. Statistical Mechanics: Theory and Experiments*.
- Cseke, & Heskes (2008). Bounds on the Bethe free energy for Gaussian networks. *Uncertainty in Artificial Intelligence*.
- Malioutov, Johnson, & Willsky (2006). Walk-sums and belief propagation in Gaussian graphical models. *J. of Machine Learning Research*, 7.
- Moallemi, & Van Roy (2006). Consensus propagation. *IEEE Trans. on Information Theory*, 52.
- Plarre, & Kumar (2004). Extended message passing algorithm for inference in loopy Gaussian graphical models. *Ad Hoc Networks*.
- Stark, & Terras (1996). Zeta functions of finite graphs and coverings. *Adv. in Math.*, 121.
- Weiss, & Freeman (2001). Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13.