Gaussian belief with dynamic data and in dynamic network

E. Aurell\(^1,2\) and R. Pfitzner\(^1,3(a)\)

\(^1\)KTH Linnaeus Centre ACCESS, KTH-Royal Institute of Technology - Stockholm, Sweden, EU
\(^2\)Department of Information and Computer Science, TKK-Helsinki University of Technology - Espoo, Finland, EU
\(^3\)Friedrich-Schiller-University Jena - Germany, EU

received 1 July 2009; accepted in final form 14 September 2009
published online 9 October 2009

PACS 89.90.+n – Other topics in areas of applied and interdisciplinary physics
PACS 89.20.Ff – Computer science and technology

Abstract – In this paper we analyze Belief Propagation over a Gaussian model in a dynamic environment. Recently, this has been proposed as a method to average local measurement values by a distributed protocol (Consensus Propagation, Moallemi C. C. and Van Roy B., IEEE Trans. Inf. Theory, 52 (2006) 4753) where the average is available for read-out at every single node. In the case that the underlying network is constant but the values to be averaged fluctuate (“dynamic data”), convergence and accuracy are determined by the spectral properties of an associated Ruelle-Perron-Frobenius operator. For Gaussian models on Erdős-Rényi graphs, numerical computation points to a spectral gap remaining in the large-size limit, implying exceptionally good scalability. In a model where the underlying network also fluctuates (“dynamic network”), averaging is more effective than in the dynamic data case. Altogether, this implies very good performance of these methods in very large systems, and opens a new field of statistical physics of large (and dynamic) information systems.

Copyright © EPLA, 2009

Message-passing algorithms have over the last two decades turned out to be an important paradigm in fields as distant as iterative decoding, image processing and AI, see [1] for the intuition behind Belief Propagation (BP) in AI, and [2–4] for more recent reviews. It has been realized that systems where the message-passing algorithms are effective can often be assimilated to disordered systems in statistical physics, and that the message-passing algorithms themselves are closely related to the Bethe approximation in statistical physics [5]. Most applications pursued concern inference in static models; how to do this effectively (if approximately), and when these methods work. In another direction, Consensus Propagation (CP) has been proposed as a message-passing scheme to average measurement values in a network of connected nodes [6]. This is a naturally dynamic setting, where, in large networks, and in many scenarios of interest, one must allow the measurement values, and perchance the network itself, to change on the same time scale as the averaging process. The two strands of inquiry are connected by the fact that CP is equivalent to BP on a class of Gaussian-Markov random fields, as we will briefly review in the next section [6,7].

In this contribution we study CP both in a static network with changing measurement values (dynamic data), and in a network where the strengths of the interconnections also change (dynamic network). We will show that the method has very good scalability, i.e. that its performance degrades very slowly as the systems grow. In a sense, to be made precise below, performance does not degrade with size at all. This should make CP a very interesting method for aggregation tasks in large and dynamic networks, possibly competitive to alternative schemes such as gossiping [8]. From a physics perspective the salient points are the following: i) CP with dynamic data is (after a transient) a linear averaging process; ii) the kernel of this averaging process, being the linearization of Gaussian BP, is related to the second variation of the Bethe Free Energy of the Gauss-Markov random field; iii) the leading eigenvalue of the kernel is a self-averaging quantity in Erdős-Rényi networks, which in addition does not depend on the network size; iv) CP with dynamic network and dynamic data functions as well (or better) as CP with dynamic data only. Points ii) and iii) imply that we identify a new random matrix construction with unexpected properties, and possibly important practical consequences. Point iv) means concretely that dynamic data is the slow stable (and also flat) manifold of the kernel, while dynamic network spans the fast stable manifold. Perturbations in the dynamic network directions hence relax faster than dynamic data, which explains the good properties.

\(^{\text{(a)}}\)E-mail: rene.pfitzner@uni-jena.de
Belief Propagation (BP) and Consensus Propagation (CP). – BP is an algorithm to infer marginal probability distributions of a joint probability function using its representation as a graphical model [4]. It works via passing messages \( m_{ij} \) on the underlying model from each node \( i \) to every neighboring node \( j \). The \( m_{ij} \)'s are locally computed at every node \( i \) in each iteration step and can depend on messages previously gathered at \( i \). (fig. 1). BP is correct on tree-like graph topologies and has been shown to often yield good results in topologies including loops [5,9]. The messages in BP can be seen as 1-node marginal conditional probability distributions, which implies that BP works best computationally when the size of local state space is limited, e.g. for Ising spins. BP on Gauss-Markov random fields is a special case, since Gaussianity is preserved under convolution, and the BP messages can be parametrized by two real values corresponding to (conditional) mean and (conditional) variance. A further very special property of BP on Gaussian models is that it is exact for the modes of the marginal distributions, in a very wide class of models [7,10].

A special instance of Gaussian Belief Propagation is Consensus Propagation [6]. This algorithm aims to solve the problem of calculating the average \( \bar{y} \) of some local data \( y_i \), gathered by nodes \( i \) in a network \( G \), in a distributed way. The \( y_i \)'s can, in a potential application, be any kind of real valued local data. The graph \( G = (V,E) \) is a model of the underlying network with \( V \), the set of all nodes \( i = 1, \ldots, N \), and \( E \), the set of all edges in the graph. The Gaussian model associated to CP is [6]:

\[
p(x, \beta) = \frac{1}{Z} \exp \left( - \| x - y \|^2 - \beta \sum_{(i,j) \in E} Q_{ij}(x_i - x_j)^2 \right).
\]  

(1)

In (1), \( Z \) is a normalization, \( \beta \) is a global and \( Q_{ij} \) are local coupling parameters, \( y \) and \( x \) are \( N \)-dimensional vectors consisting of the variables \( y_i \) and \( x_i \) respectively, where \( y_i \) are local data and \( x_i \) are local internal variables. The sum in (1) goes over all undirected edges in the graph. BP on (1) is guaranteed to converge for any finite \( \beta \) (since (1) defines a walksummable model), and the modes of any one-node marginals computed by BP converge to the average \( \bar{y} \) as \( \beta \) tends to infinity (as follows from [7]). In this way estimates of \( \bar{y} \) can be obtained locally at every node, where a trade-off must be made between convergence time and accuracy. The following message update rules define Consensus Propagation, as follows from minimizing the Bethe Free Energy of the model (see appendix):

\[
K_{ij}^{(t+1)} = \frac{1 + \sum_{k \in N(i) \setminus j} K_{ki}^{(t)}}{1 + \frac{1}{|\mathcal{Q}_{ij}|} \sum_{k \in N(i) \setminus j} K_{ki}^{(t)}},
\]

(2)

\[
\mu_{ij}^{(t+1)} = \frac{y_i + \sum_{k \in N(i) \setminus j} K_{ki}^{(t)} \mu_{ki}^{(t)}}{1 + \sum_{k \in N(i) \setminus j} K_{ki}^{(t)}}.
\]

(3)

This parametrization of BP yields two-dimensional real-valued messages \( m_{ij} \) consisting of a local state update \( \mu \) and a topology message \( K \). The notation \( X_i^{(t)} \) means that message \( X \) is sent from originating node \( i \) to target node \( j \) at iteration step \( t \): \( N(i) \) is the set of all neighbours of node \( i \), and \( N(i) \setminus j \) is the set of all neighbours of \( i \) except \( j \). The algorithm is said to have attained consensus, if the messages are fixed points of (2) and (3). A belief for the average \( \bar{y} \) at time \( t \) and node \( i \) is obtained via the CP output rule:

\[
\bar{y}_i = \frac{y_i + \sum_{k \in N(i)} K_{ki}^{(t)} \mu_{ki}^{(t)}}{1 + \sum_{k \in N(i)} K_{ki}^{(t)}}.
\]

(4)

The consensus beliefs (4), with \( K \) and \( \mu \) at a fixed point of (2) and (3), are the Belief Propagation predictions of the modes of the one-node marginals of the probability distribution (1).

Convergence for different initializations. – Figure 2 shows performance of Consensus Propagation after initialising all messages to zero: the algorithm shows first an oscillating behaviour with resulting fast convergence to a good approximation of the correct mean \( \bar{y} \) followed by a slow power law decay. After changing every node value and NOT re-initialising the messages, the algorithm exhibits only the slow damping as governing convergence process. As will become clear later the slow damping in both areas of convergence follows the same law and so yielding identical convergence ratios. The second behaviour corresponds to the case of dynamic data, where the topology message (K-message) and local state update (\( \mu \)-message) start at their converged values before the perturbation. Once a fixed point \( K^* \) is reached, the topology messages will not change if only the local measurement values \( y_i \) change, since (2) is an equation only involving topology messages. Except for an initial transient, the dynamic data case can hence be completely understood by the linear operator expressed by the right-hand side of (3) (see below). Generally, it seems that the topology messages converge much faster than the local state messages, and that therefore
the linear theory (explained below) also bounds the behaviour of dynamic data, where both local values $y_i$ and local couplings $Q_{ij}$ change. Before we turn to the linear analysis, let us however point out the observation that different initialisations of the messages yield different performance, and, perhaps surprisingly, that initialising with $K = \mu = 0$ seems to be the superior choice. The observations of fig. 3 contradict a conjecture put forward in [6] that convergence times for $K^{(0)} = 0$ and $K^{(0)} = K^*$ are equivalent. In fact, initialising with $K = 0$ improves convergence dramatically. Let us note that if $K$ be re-initialised to zero, then the re-initialisation of $\mu$ is arbitrary, since by (3), $\mu^{(1)}$ will then be equal to $y_i$, i.e. independent of $\mu^{(0)}$. In a scenario where many measurement values (and/or also the underlying network) change simultaneously, re-starting Consensus Propagation using $K = 0$ may therefore by a valid option to improve convergence time. We stress that this is not obvious, but follows if the dynamic behaviour is as illustrated in fig. 4, and may not be true in all underlying topologies. However, in the case that the underlying topology is locally tree-like, as is the case for the random graphs in fig. 3, a heuristic explanation for the faster convergence of Consensus Propagation, when initialising with $K = 0$, is the following: as was shown by [6] Consensus Propagation yields the exact node average on tree-like graphs with the global coupling constant $\beta = \infty$ and $K^{(0)} = 0$. Initialising CP on a random graph with $K = 0$ and a large value of $\beta$ will yield nearly the same messages, after a finite number of iterations, as initialising CP with $K = 0$ on a computational tree associated with the graph (using the construction of [7]) at $\beta = \infty$. This explains the improved behaviour of starting with $K = 0$ qualitatively, but does not explain it quantitatively, i.e. the apparent complete absence of the slow process in fig. 3.

The numerical experiments above indicate that the $\mu$-message subspace spans a slow stable and the $K$-message subspace a fast stable manifold (see fig. 4).

\[
\begin{pmatrix}
\mu_{ij}^{(n+1)} \\
\cdots \\
K_{ij}^{(n+1)} \\
\cdots
\end{pmatrix} = R
\begin{pmatrix}
\mu_{ij}^{(n)} \\
\cdots \\
K_{ij}^{(n)} \\
\cdots
\end{pmatrix}
\]

(5)

Fig. 2: (Colour on-line) Convergence of the $\bar{y}$ belief at one node in a random network with 500 nodes. The solid line is the CP performance, the dashed line indicates the correct $y$-average. This insert shows the behaviour in 1–10 rounds; the main figure shows the behaviour in $10^3$–$10^4$ rounds. Node values were generated randomly, and then scaled by 90% in round $5 \times 10^5$. At iteration step $n = 0$ all CP-messages were initialised uniformly to zero. Fast convergence followed by an overshoot (damped oscillation) is observed. After the reset at $n = 5 \times 10^3$ the CP-messages were left unchanged, and a markedly slower convergence, but without an overshoot, is observed.

Fig. 3: (Colour on-line) Convergence behaviour of Consensus Propagation on a random graph for different initial messages. As model we used a random graph with 80 nodes, all edges present with probability 0.1, $\beta = 100$ and $Q_{ij}$ chosen i.i.d. random variables uniform between 0.5 and 2. The plot shows the time evolution of the deviation of two messages from their converged values ($K^* - K$ and $\mu^* - \mu$), sent from node 15 to node 10 during 500 iterations. The messages were initialised proportional to their fixed point values, and start at the top right corner of each trajectory in the figure. The trajectories exhibit an initial fast decay of the error in topology messages (abscissa) followed by a slower decay of the local state message (ordinate). The only exception is when the messages are initialised as $K = \mu = 0$ in which case the trajectory seems to fall into the (more) stable manifold of the fixed point (a “direct hit”), with the second slow process along the ordinate absent. For a graphical illustration of the conjectured behaviour, see fig. 4. The fixed point in this example have $K_{15,10}^{(0)} = 62.61$ and $\mu_{15,10}^{(0)} = 3.95$. The lower plot shows a zoom for the two extreme cases.

The numerical experiments above indicate that the $\mu$-message subspace spans a slow stable and the $K$-message subspace a fast stable manifold (see fig. 4).
Fig. 5: (Colour on-line) Length of projections of (normlength) eigenvectors on the $\mu$-subspace for each eigenvalue $\lambda$ of the linearised Ruelle-Perron-Frobenius-Operator $\mathbb{R}'$ for a $G(N = 20, c = 8)$ Erdős-Rényi model. $\beta = 100$, and $Q_{ij}$ randomly generated as in fig. 3.

We will use the eigenvalues of a linearised version of $\mathbb{R}$ to verify this. Following [7], we refer to the non-linear iterated map transfer operator $\mathbb{R}$ as a Ruelle-Perron-Frobenius Operator. The linear part of $\mathbb{R}$ has the matrix representation:

$$
\begin{pmatrix}
K^{(n+1)}_1 \\
\vdots \\
K^{(n+1)}_m
\end{pmatrix} =
\begin{pmatrix}
\mu_1^{(n)} + \Delta \mu_1^{(n+1)} \\
\vdots \\
\mu_m^{(n)} + \Delta \mu_m^{(n+1)}
\end{pmatrix} = \begin{pmatrix}
\mu_1^{(n)} \\
\vdots \\
\mu_m^{(n)}
\end{pmatrix} + \mathbb{R}'
$$

(6)

$\mathbb{R}'$ is the linearised transfer operator. The matrix representation of this operator can be decomposed into four quadratic submatrices:

$$
\mathbb{R}' = \begin{pmatrix}
A & C \\
0 & B
\end{pmatrix}.
$$

(8)

Submatrix $A$ is the transfer matrix in the dynamic data case, when the topology messages have converged, submatrix $B$ is the linear part of the transfer matrix acting in the dynamic network on the topology messages alone, and submatrix $C$ is the linear action of the topology messages on the local state variables. Around the fixed point, we can verify that topology messages converge faster than local state updates, by comparing the size of the eigenvalues of $\mathbb{R}'$ to the projection of the corresponding eigenvectors on the subspace spanned by the $\mu$-messages. As shown in fig. 5, the (isolated) largest eigenvalue lies in the subspace of local state updates. In addition, most of the other eigenvectors in the subspace of local updates also have eigenvalues larger than all the eigenvalues projecting on the topology messages. Table 1 compares the leading eigenvalues of submatrices $A$ and $B$ for four different Erdős-Rényi graphs, reinforcing the observation from fig. 5. In linear theory, the limiting factor on convergence is therefore the dynamics of the local state updates.

Table 1: Comparison of leading eigenvalues of linearised matrices $A$ and $B$ in four Erdős-Rényi graphs $G(N, p = c/N)$. The much smaller eigenvalues of matrix $B$ imply much faster convergence of the topology messages.

| $N$ | $c$ | $\lambda_{max}(A)$ | $\lambda_{max}(B)$ |
|-----|-----|---------------------|---------------------|
| 20  | 18  | 0.99949152          | 0.00054415          |
| 30  | 14  | 0.99924356          | 0.00083034          |
| 40  | 10  | 0.99895415          | 0.00119833          |
| 50  | 8   | 0.99851962          | 0.00186674          |

Fig. 6: (Colour on-line) Experimental convergence ratios of the linear averaging process (9) compared to the leading eigenvalue $\lambda_{max}$ of the operator $A$ in four examples of Erdős-Rényi models $G(N, p = c/N)$. $q$: convergence ratio, $\lambda_{max}$: largest eigenvalue. The solid line represents $q = \lambda_{max}$. Cases: 1: $c = 8$, $N = 50$, 2: $c = 10$, $N = 40$, 3: $c = 14$, $N = 30$, 4: $c = 18$, $N = 20$. $\beta = 100$, and $Q_{ij}$ randomly generated as in fig. 3 for all cases.

**Dynamic data case.** – The case when topology messages have converged is also of interest when data to be measured keep on changing: in this scenario CP is a linear averaging process. Indeed, the local state update equation (3) is then a linear equation of one free vector $\mu = (\ldots \mu_{ij} \ldots)$ and can be expressed in linear operator form:

$$
\mu^{(n+1)} = b + A \mu^{(n)}.
$$

(9)

The operator $A$ in (9) (acting on $\mu$-messages) is the same as the submatrix $A$ of the operator $\mathbb{R}'$ in (8), and its spectral properties are as described in fig. 5 (rightmost set of eigenvalues, all completely in the subspace of local state updates). The vector $b$ is the $\mu$-independent part of (3), and in particular depends on the set of local measurement values $y$. If these change in time, (9) is obviously a linear averaging process with kernel $A$. If the $y$ do not change, and the $\mu$-messages are initialised in some manner, we expect from fig. 5 that convergence will eventually be dominated by the largest (isolated) eigenvalue of $A$. Figure 6 shows that this is indeed the case, for several different Erdős-Rényi graphs. In these models, we always find an isolated largest eigenvalue (data not shown).

**Scalability of CP in Erdős-Rényi graphs.** – The above discussion leads up to the conclusion that the largest eigenvalue of operator $A$ of (9) is a quantity of major importance to understand the performance of CP in dynamic environments—both dynamic data only, and also
dynamic network. The scaling properties of this largest eigenvalue therefore determines how effective the CP averaging procedure can be in a large network. Following the general principles of random graph theory, we should compare random graphs of increasing size \( N \), but with the same average node degree \( c \). This means that every link is present in the graph with probability \( p = \frac{c}{N} \) (up to corrections decaying with \( N \)). Table 2 shows that in a family of Erdős–Rényi graphs with asymptotic average node degree \( c = 8 \) the largest eigenvalue of \( A \) seems to converge to a finite value less than one. In the experiments, the local couplings \( Q_{ij} \) are generated randomly between 0.5 and 2. For the smaller instances in tables 2 also the standard deviation of the largest eigenvalue computed from 100 experiments (independent realizations of the random graphs, and independent realizations of the local coupling constants \( Q_{ij} \)) is shown. The decay of the standard deviation with \( N \) indicates that the leading eigenvalue is a self-averaging quantity in this ensemble.

Let us note that our results concur with (and extend) a result of [6] for regular graphs, where the authors showed that convergence time is not dependent on the graph size. If this be true, the leading eigenvalue in that ensemble must also be a self-averaging quantity, independent of graph size.

**Summary.** – Statistical physics has contributed very significantly in recent years to the understanding of Belief Propagation approaches to inference, which have very important applications to, e.g., iterative decoding [5]. In this contribution, we have looked at a Belief Propagation-based algorithm for averaging, with potentially numerous applications to network management. We showed that this Consensus Propagation algorithm, in a dynamic environment, is a dynamical system which can be fruitfully analysed by the tools of statistical physics and non-linear dynamics. We showed that CP responds quickly to changes in the network topology, and more slowly to changing data. This can be understood intuitively as a dynamic network improves mixability, which should not be a disadvantage when computing an average (or an estimate of an average). In a real-world application, CP is therefore not limited by a changing network structure but by dynamic data. Secondly, and of interest to statistical physicists, we exhibited an interesting self-averaging property of the leading eigenvalue of the transfer matrix describing the the dynamic data case. Perhaps surprisingly, this leading eigenvalue seems to be asymptotically independent of network size.

***

RP was partially funded by the European “Life Long Learning Program” under project number DE-2008-ERA/MOB-KonsZuV01-CP07. EA acknowledges support from the Swedish Science Council through the KTH Linnæus Centre ACCESS, and from the Academy of Finland.

**Appendix**

Consider the following general Gaussian model

\[
p(x_1, \ldots, x_N) \propto \exp \left( -\frac{1}{T} \left( \sum_i |x_i - y_i|^2 + \beta \sum_{(i,j) \in E} Q_{ij} (x_i - x_j)^2 \right) \right).
\]

(A.1)

The \( y_i \) are local data, while the \( x_i \) are local internal variables, one for each node \( i \). The \( Q_{ij} \), which are non-negative, are coupling constants between local variables, and \( \beta \) is an overall parameter determining the trade-off between aligning to local data, and aligning to other local variables. The parameter \( T \) in (A.1) will eventually be set to one, and is not equal to one over the parameter \( \beta \) in (A.1). The free energy

\[
F(T) = -T \ln Z
\]

of the model, with \( Z \) the partition function, can be written as

\[
F(T) = \frac{NT}{2} \ln \frac{1}{T} - \frac{T}{2} \ln \det M + \sum_i y_i^2 - \sum_{ij} y_i (M^{-1})_{ij} y_j
\]

(A.3)

where we used the multidimensional Gaussian Integral. Here the diagonal elements of matrix \( M \) are \( M_{ii} = 1 + \beta \sum_j Q_{ij} \), and the off-diagonal elements are \( M_{ij} = -\beta Q_{ij} \). From (A.3) we learn that the free energy is a quadratic function of the local data, and also depends on \( \beta \) and the \( Q_{ij} \)’s through the determinant of \( M \). From now on we set \( T = 1 \). The Bethe Free Energy is the following ansatz built on the beliefs \( b_{ij}(x_i, x_j) \), proxies for the marginal probability distribution over variables \( x_i \) and \( x_j \):

\[
F_B = \sum_{ij} b_{ij} (E_{ij} + \ln b_{ij}) - \sum_i (n_i - 1) \sum_{x_i} b_i (E_i + \ln b_i).
\]

(A.4)

68004-p5
Using the Lagrangian Method we build a Lagrangian $L_B$ using multipliers $\alpha$, $\beta$ and $\lambda$ to allow for the two normalization and the marginalization constraints of the beliefs:

$$L_B = \sum_{ij} \sum_{x_i,x_j} b_{ij}(E_{ij} + \ln b_{ij})$$

$$- \sum_i (n_i - 1) \sum_{x_i} b_i(E_i + \ln b_i)$$

$$+ \sum_i \alpha_i \left( \sum_{x_i} b_i - 1 \right) + \sum_{ij} \beta_{ij} \left( \sum_{x_i,x_j} b_{ij} - 1 \right)$$

$$+ \sum_{ij} \sum_{x_i} \lambda_{ij}(x_i) \left( \sum_{x_j} b_{ij} - b_i \right). \tag{A.5}$$

In the above $b_i$ are proxies for one-node probabilities, $b_{ij}$ are proxies for pair probabilities and $n_i$ is the coordination number of node $i$, i.e. the number of other nodes $j$ such that $Q_{ij}$ is non-zero (number of neighbors). The pair energy $E_{ij}$ is $|x_i - y_i|^2 + |x_j - y_j|^2 + \beta Q_{ij}(x_i - x_j)^2$, and the one-node energy $E_i$ is simply $|x_i - y_i|^2$. We note that the true marginal distribution $p_i(x_i, x_j)$ must be a Gaussian for this Gaussian model. By variation, $L_B$ can be expressed in terms of the Lagrange multipliers $\lambda_{ij}$ as

$$L_B = - \sum_{ij} \ln \sum_{x_i,x_j} \exp(-E_{ij} - \lambda_{ij} - \lambda_{ji})$$

$$+ \sum_i (n_i - 1) \ln \sum_{x_i} \exp(-E_i - \frac{1}{n_i-1} \sum_j \lambda_{ij}). \tag{A.6}$$

In the above equation $\lambda_{ij} = 0$ whenever $n_i = 1$. It is easily seen that one can add a constant to any other Lagrange multiplier i.e. $\lambda_{ij} \rightarrow \lambda_{ij} + c_{ij}$ without changing $L_B$. Varying $L_B$ with respect to $\lambda_{ij}(x_i)$ then gives

$$\frac{\delta L_B}{\delta \lambda_{ij}(x_i)} = \frac{1}{\Omega_{ij}} \sum_{x_j} \exp(-E_{ij} - \lambda_{ij} - \lambda_{ji})$$

$$- \frac{1}{\Omega_i} \exp(-E_i - \frac{1}{n_i-1} \sum_k \lambda_{ik}). \tag{A.7}$$

where $\Omega_{ij} = \sum_{x_i,x_j} \exp(-E_{ij} - \lambda_{ij} - \lambda_{ji})$ and $\Omega_i = \sum_{x_i} \exp(-E_i - \frac{1}{n_i-1} \sum_j \lambda_{ij})$. The vanishing of the variation can then be written as the equation:

$$e^{\lambda_{ij} - \frac{1}{n_i-1} \sum_k \lambda_{ik}} = \frac{\Omega_i}{\Omega_{ij}} \sum_{x_j} \exp(E_i - E_{ij} - \lambda_{ji}). \tag{A.8}$$

If $n_i = 1$, the left-hand side of above is one (since then $\lambda_{ij} = 0$). For any node such that $n_i > 1$ it is convenient to introduce the Belief Propagation variable $m_{j\rightarrow i}(x_i) = \exp(\lambda_{ij} - \frac{1}{n_i-1} \sum_k \lambda_{ik})$, and in terms of which the variational equation is the Belief Propagation update equation:

$$m_{j\rightarrow i}(x_i) = \frac{\Omega_i}{\Omega_{ij}} \sum_{x_j} \exp(E_i - E_{ij}) \prod_{k \in \partial j \setminus i} m_{k\rightarrow j}(x_j). \tag{A.9}$$

The product in the right-hand side of above goes over all neighbours $k$ of $j$, except $i$. If $j$ is a leaf node ($n_i = 1$) the product is hence empty. The variational solution of the $b_{ij}$’s is proportional to $\exp(-E_{ij} - \lambda_{ij} - \lambda_{ji})$, which if $i$ is a leaf node (but $j$ is not) is $\exp(-E_{ij}) \prod_{k \in \partial i \setminus i} m_{k\rightarrow j}(x_j)$. Similarly, the variational solutions of the $b_{ij}$’s, if $i$ is not a leaf node, is proportional to $\exp(-E_i - \frac{1}{n_i-1} \sum_{k \in \partial i} \lambda_{ik})$, which is $\exp(-E_i) \prod_{k \in \partial i} m_{k\rightarrow i}(x_i)$. For a leaf node $i$ we can hence define a Belief Propagation variable $m_{j\rightarrow i}(x_i)$ through (A.9), from which we would then have $b_i \propto \exp(-E_i) m_{j\rightarrow i}(x_i)$ (in parallel to the expression for all the other $b_i$).

Now, the only way in which the proxies $b_{ij}$ can be Gaussian, as are the true marginal probabilities $p_i$, is if all the Belief propagation variables $m_{j\rightarrow i}$ are also Gaussian. Let us therefore introduce the parametrization

$$m_{j\rightarrow i}(x_i) \propto \exp(-K_{ij}(x_i - \mu_{ij})^2). \tag{A.10}$$

The Belief propagation update equations then translate into

$$K_{ij} = \frac{1 + \sum_{k \in \partial i \setminus j} K_{ki}}{1 + \sum_{k \in \partial i \setminus j} K_{ki}}, \tag{A.11}$$

$$\mu_{ij} = \frac{y_j + \sum_{k \in \partial i \setminus j} K_{ki} \mu_{ki}}{1 + \sum_{k \in \partial i \setminus j} K_{ki}}. \tag{A.12}$$

which are the Consensus Propagation equations.

REFERENCES

[1] PEARL J., Probabilistic Reasoning in Intelligent Systems (Morgan Kaufmann) 1988.
[2] COWELL R., in Learning in Graphical Models, edited by JORDAN M. (Kluwer Academic Publishers) 1998.
[3] Kschischang F. R., Frey B. J. and Loeliger H.-A., IEEE Trans. Inf. Theory, 47 (2001) 498.
[4] YEDidia J. S., FREeman W. T. and WEISS Y., in Exploring Artificial Intelligence in the New Millennium (Science & Technology Books) 2003, pp. 239–269.
[5] MEzARD M. and MONTANARI A., Information, Physics and Computation (Oxford University Press) 2009.
[6] MoAlLEMI C. C. and VAN Roy B., IEEE Trans. Inf. Theory, 52 (2006) 4753.
[7] MALIOUTOV D. M., JOHNSON J. K. and WILLSKY A. S., J. Mach. Learn. Res., 7 (2006) 2031.
[8] Jelasity M., Montresor A. and BaraOGLU O., ACM Trans. Comput. Syst., 23 (2005) 219.
[9] Murphy K. P., WEISS Y. and JORDAN M., Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99) (Morgan Freeman) 1999, p. 467.
[10] WEISS Y. and FREeman W. T., Neural Comput., 13 (2001) 2173.