STABLE BELIEF PROPAGATION IN GAUSSIAN DAGS

David Barber

IDIAP Research Institute, Martigny, Switzerland
http://www.idiap.ch/∼barber

Peter Sollich

King's College, London WC2R 2LS, U.K
http://www.mth.kcl.ac.uk/∼psollich

ABSTRACT

We consider approximate inference in the important class of Gaussian distributions corresponding to multiply-connected directed acylic networks (DAGs). We show how Directed Belief Propagation can be implemented in a numerically stable manner by associating backward (λ) messages with an auxiliary variable, enabling intermediate computations to be carried out in moment form. We apply our method to the Fast Fourier Transform network with missing data, and show that the results are more accurate than those obtained using Undirected Belief Propagation on the equivalent Markov network.

Index Terms— State space methods, Discrete Fourier transforms, Bayes procedures, Graph theory

1. GAUSSIAN DIRECTED ACYCLIC GRAPHS

A Gaussian DAG is a distribution over a set of vectors,

$$p(x_1, \ldots, x_N) = \prod_i p(x_i | x_{\mathrm{pa}(i)})$$

in which each local conditional distribution $p(x_i|x_{\mathrm{pa}(i)})$ is a Gaussian. Here $\mathrm{pa}\,(i)$ denotes the variables in the parental set of i. Each conditional Gaussian may be defined by the stochastic linear relation

$$x_i = \sum_{l \in \mathrm{pa}(i)} W_{i \leftarrow l} x_l + \eta_i, \qquad \eta_i \sim \mathcal{N}\left(\mu_i^0, \Sigma_i^0\right) \tag{1}$$

where μ_i^0, Σ_i^0 are the mean and covariance of the Gaussian noise. Perhaps the most celebrated instance of a *singly connected* GaussDAG is the Kalman Filter [1]. More recently GaussDAGs have appeared as interesting models in spatio-temporal filtering [2] and the Fast Fourier Transform [3]. Our interest in this paper is a stable version of approximate inference in *multiply-connected* networks where exact inference is impractical. In a typical application, the variables x are split into a set of hidden and visible variables $x = (h, v)$ and inference corresponds to computing the conditional $p(h|v)$. Formally, the posterior means and covariances are given by

$$\mu_{h|v} = \mu_h + \Sigma_{hv}\Sigma_{vv}^{-1}\left(v - \mu_v\right), \; \Sigma_{h|v} = \Sigma_{hh} - \Sigma_{hv}\Sigma_{vv}^{-1}\Sigma_{vh} \tag{2}$$

Here Σ_{hv} represents the h, v block of the joint covariance of $p(h, v)$. Similarly, μ_h and μ_v are the corresponding mean components. These means and covariances are easily found by forward propagation of the recursions equation (1). Hence exact inference is bounded by $O(|v|^3)$ since this is the complexity of matrix inversion. However, in many applications this is unacceptably large and one seeks to exploit the structure to reduce this computation. Our interest is approximate inference when exact inference even using the Junction Tree algorithm is deemed impractical. Belief Propagation (BP) is an attractive approximation since, if it converges (in either its directed or undirected manifestation) then the inferred means $\mu_{h|v}$ will be exact, although the covariance $\Sigma_{h|v}$ will typically be overconfident [4]. Exactness of the means is an important advantage of BP over other techniques such as approximate block factorisations (see e.g. [5]). Conjugate gradient methods are also interesting for computing the means, although they are less straightforward to apply to covariances.

One approach is to first convert the GaussDAG to an undirected network, $p(x) \propto \prod_{ij} \psi_{ij}(x_i, x_j)$, where $\psi_{ij}(x_i, x_j)$ are suitably defined exponentiated quadratic forms[4]. In general one obtains additional links ψ_{ij} between common parents of any node. *Undirected* BP (UBP) can then be run on the resulting network by using the message recursions $\gamma_{i\to j}(x_j) \propto \int_{x_i} \psi_{ij}(x_i, x_j) \prod_{k \in N(i)\setminus j} \gamma_{k\to i}(x_i)$, where $N(i)$ are the neighbours of node i on the Markov network; the posterior means are given by $p(x_i) \propto \psi_i(x_i) \prod_{k\in N(i)} \gamma_{k\to i}(x_i)$. However, if covariances Σ_i^0 are small (or even zero), then the conversion will introduce accumulating numerical errors [1, 3].

Directed Belief Propagation

Directed Belief Propagation (DBP) sends messages from node i to its children $j \in \mathrm{ch}\,(i)$ and parents $l \in \mathrm{pa}\,(i)$ [6]

$$\rho_{i\to j}(x_i) = \int p(x_i|x_{\mathrm{pa}(i)}) \prod_{l\in \mathrm{pa}(i)} \rho_{l\to i}(x_l) \prod_{k\in \mathrm{ch}(i)\setminus j} \lambda_{k\to i}(x_i)$$

$$\lambda_{i\to l}(x_l) = \int p(x_i|x_{\mathrm{pa}(i)}) \prod_{l'\in \mathrm{pa}(i)\setminus l} \rho_{l'\to i}(x_{l'}) \prod_{k\in \mathrm{ch}(i)} \lambda_{k\to i}(x_i)$$

where the integrals are over all variables except the one on which the message depends. We choose to parameterise the ρ messages using the moment representation

$$\rho_{i\to j}(x_i) \propto \exp -\frac{1}{2}\left(x_i - f_{i\to j}\right)^T F_{i\to j}^{-1}\left(x_i - f_{i\to j}\right)$$

and the λ messages using the canonical representation

$$\lambda_{i\to l}(x_l) \propto \exp -\frac{1}{2}\left(x_l^T G_{i\to l} x_l - 2x_l^T g_{i\to l}\right) \qquad (3)$$

Whilst there is a choice about whether to parameterise ρ in either moment or canonical form, no such choice exists for the λ messages since G is generally not invertible. DBP then corresponds to updating the parameters F, f, G, g. If one carries out the integrals naively, then inverse noise covariances appear explicitly. This is numerically problematic in the case of very small (or even zero) noise covariances. Fortunately, there is a simple trick which automatically produces recursions in the correct form.

2. THE AUXILIARY VARIABLE TRICK

A naive integration implementation of the DBP recursions above effectively converts ρ messages to a canonical representation, and performs the integral in this representation. However, we would prefer to carry out the integrals in a moment representation for ρ since this is appropriate for small covariances. To do this we express the λ messages in a form where the canonical variables G, g appear in a moment form, by introducing an *auxiliary variable* $x_{k\to i}$ for each message $\lambda_{k\to i}$. This is defined by

$$x_{k\to i} = G_{k\to i} x_i + \eta_{k\to i} \qquad (4)$$

where $\eta_{k\to i} \sim \mathcal{N}(0, G_{k\to i})$. Then equation (3) can be written

$$\lambda_{i\to l}(x_i) \propto p(x_{i\to l}|x_l)|_{x_{i\to l}=g_{i\to l}}$$

Note that this representation of the λ message as a clamped distribution is not unique[1].

Normally in DBP, each node sends a λ message to each parent. However, in the case of an evidential node e with only a single parent i and in the absence of noise, $\lambda_{e\to i}(x_i) \propto p(x_e|x_i)$ has an infinite $G_{e\to h}$. To deal with this, we do not parameterise λ messages from evidential children, but rather simply include the relevant factors $p(x_e|x_i)$ directly in the recursions, as we will now describe. For a fuller explanation, please refer to [7].

The ρ messages

Using the auxiliary variable method, and separating out evidential variables $e \in E$, we have

$$\rho_{i\to j}(x_i) \propto \int p(x_i|x_{\mathrm{pa}(i)}) \prod_{l\in\mathrm{pa}(i)} \rho_{l\to i}(x_l) \prod_{k\in\mathrm{ch}(i)\setminus j, k\notin E} p(x_{k\to i}|x_i) \prod_{e\in\mathrm{ch}(i)\setminus j, e\in E} p(x_e|x_i)$$

where $x_{i\to l}$ is clamped to the value $g_{i\to l}$ and x_e is clamped to value v_e. The usefulness of the auxiliary variable is that we can identify the r.h.s. of the above equation as

$$\rho_{i\to j}(x_i) \propto p(x_i, \{x_{k\to i}\}, \{x_e\})|_{\{x_{k\to i}=g_{k\to i}\},\{x_e=v_e\}}$$

where the joint distribution is found using equation (4) and

$$x_i = \sum_{l\in\mathrm{pa}(i)} W_{i\leftarrow l} x_l + \eta_i,\ x_e = W_{e\leftarrow i} x_i + \sum_{i'\in\mathrm{pa}(e)\setminus i} W_{e\leftarrow i'} x_{i'} + \eta_e \qquad (5)$$

From these equations, $\mu_i = \sum_{l\in\mathrm{pa}(i)} W_{i\leftarrow l} f_{l\to i} + \mu_i^0, \mu_{k\to i} = G_{k\to i}\mu_i, \Sigma_{ii} = \sum_{l\in\mathrm{pa}(i)} W_{i\leftarrow l}\Sigma_{l\to i} W_{i\leftarrow l}^T + \Sigma_i^0, \mu_e = \mu_e^0 + W_{e\leftarrow i}\mu_i + \sum_{i'\in\mathrm{pa}(e)\setminus i} W_{e\leftarrow i'} f_{i'\to e}, \Sigma_{ki} = G_{k\to i}\Sigma_{ii}, \Sigma_{kk} = G_{k\to i}\Sigma_{ii}G_{k\to i} + G_{k\to i}, \Sigma_{ek} = W_{e\leftarrow i}\Sigma_{ii}G_{k\to i}, \Sigma_{ei} = W_{e\leftarrow i}\Sigma_{ii}$, (see[7]) and

$$\Sigma_{ee} = \mu_e^0 + W_{e\leftarrow i}\Sigma_{ii} W_{e\leftarrow i}^T + \sum_{i'\in\mathrm{pa}(e)\setminus i} W_{e\leftarrow l} F_{i'\to e} W_{e\leftarrow i'}^T \qquad (6)$$

Now that we have the joint distribution $p(x_i, \{x_{k\to i}\}, \{x_e\})$, we need to clamp the $x_{k\to i}$ and x_e into their appropriate states. The structure of this is that we have defined the joint $p(y, z)$ where $y \equiv x_i$ and $z \equiv \{x_{k\to i}\}, \{x_e\}$, and wish to find $p(y|z)$ with z clamped to specific values. In both the mean and covariance, a slight difficulty is that Σ_{zz} may not be invertible; but $\Sigma_{yz}\Sigma_{zz}^{-1}$ remains well defined. In general, the form of Σ_{yz} and Σ_{zz} is (here shown a non-evidential child k other than j and one evidential child e):

$$\Sigma_{yz} = \begin{pmatrix} \Sigma_{ii}G_{k\to i} & \Sigma_{ii}W_{e\leftarrow i}^T \end{pmatrix}$$

We can write the covariance Σ_{zz} as

$$\begin{pmatrix} G_{k\to i}\Sigma_{ii} + I & G_{k\to i}\Sigma_{ii}W_{e\leftarrow i}^T \\ W_{e\leftarrow i}\Sigma_{ii} & \Sigma_{ee} \end{pmatrix} \begin{pmatrix} G_{k\to i} & 0 \\ 0 & I \end{pmatrix}$$

so that $\Sigma_{yz}\Sigma_{zz}^{-1}$ becomes

$$\begin{pmatrix} \Sigma_{ii} & \Sigma_{ii}W_{e\leftarrow i}^T \end{pmatrix} \begin{pmatrix} G_{k\to i}\Sigma_{ii} + I & G_{k\to i}\Sigma_{ii}W_{e\leftarrow i}^T \\ W_{e\leftarrow i}\Sigma_{ii} & \Sigma_{ee} \end{pmatrix}^{-1}$$

This gives $f_{i\to j} = \mu_y + \Sigma_{yz}\Sigma_{zz}^{-1}(z - \mu_z)$ and covariance $F_{i\to j} = \Sigma_{yy} - \Sigma_{yz}\Sigma_{zz}^{-1}\Sigma_{zy}$ where explicitly $\mu_y = \mu_i, \Sigma_{yy} = \Sigma_{ii}, z^T = (g_{k\to i}^T, v_e^T)$ and $\mu_z^T = (\mu_k^T, \mu_e^T)$.

The λ messages

Using the same auxiliary variable trick as before, the λ messages may be written as

$$\lambda_{i\to l}(x_l) = \int p(x_i|x_{\mathrm{pa}(i)}) \prod_{l'\in\mathrm{pa}(i)\setminus l} \rho_{l'\to i}(x_{l'}) \prod_{k\in\mathrm{ch}(i)\setminus j, k\notin E} p(x_{k\to i}|x_i) \prod_{e\in\mathrm{ch}(i)\setminus j, e\in E} p(x_e|x_i)$$

subject to clamping the auxiliary and evidential variables to their appropriate states. The λ message is then given by the conditional

$$p(\{x_{k\to i}\}, \{x_e\}|x_l) \qquad (7)$$

[1] E.g. let $x_{i\to l} = Bx_i + \eta_{i\to l}$ be clamped to a, with $\eta_{i\to l} \sim \mathcal{N}(0, A)$. We then require $B^T A^{-1} B = G_{i\to l}$ and $B^T A^{-1} a = g_{i\to l}$. We choose simply $A \equiv B \equiv B^T = G_{i\to l}$.

This is a little different from the ρ case since we are now interested in the functional dependence on the conditioning variable x_l. We therefore directly isolate the x_l dependent terms in the conditional distribution; from the quadratic form in x_l in the exponent we can then read off the λ messages. The joint distribution of $\{x_{k\to i}\}, \{x_e\}$ (conditional on x_l) is obtained from the relations (4) and (5), except that in x_i we now need to separate off the part dependent on x_l:

$$x_i = \sum_{l' \neq l} W_{i\leftarrow l'} x_{l'} + W_{i\leftarrow l} x_l + \eta_i$$

Explicit expressions for the means and covariances involve the mean of x_i, which is $\tilde{\mu} + W_{i\leftarrow l} x_l$ with

$$\tilde{\mu} \equiv \sum_{l' \neq l} W_{i\leftarrow l'} f_{l'\to i} + \mu_i^0$$

as well as the covariance of x_i,

$$\tilde{\Sigma}_{ii} = \sum_{l' \neq l} W_{l'\to i} F_{l'\to i} W_{i\leftarrow l'}^T + \Sigma_i^0$$

For a non-evidential child k, and evidential child e, equation (7) is then proportional to $\exp -\frac{1}{2} m^T (MD)^{-1} m$, where

$$m \equiv \begin{pmatrix} g_{k\to i} - G_{k\to i}\left(\tilde{\mu} + W_{i\leftarrow l} x_l\right) \\ v_e - \mu_e^0 - W_{e\leftarrow i}\left(\tilde{\mu} + W_{i\leftarrow l} x_l\right) \end{pmatrix}$$

$$M = \begin{pmatrix} G_{k\to i}\tilde{\Sigma}_{ii} + I & G_{k\to i}\tilde{\Sigma}_{ii} W_{e\leftarrow i}^T \\ W_{e\leftarrow i}\tilde{\Sigma}_{ii} & \tilde{\Sigma}_{ee} \end{pmatrix}, D = \begin{pmatrix} G_{k\to i} & 0 \\ 0 & I \end{pmatrix}$$

where $\tilde{\Sigma}_{ee}$ is obtained by replacing $\Sigma_{ii} \to \tilde{\Sigma}_{ii}$ in (6). To isolate the x_l dependence we define

$$c_k = g_{k\to i} - G_{k\to i}\tilde{\mu}, \quad c_e = e - \mu_e - W_{e\leftarrow i}\tilde{\mu},$$

$$U_k = W_{i\leftarrow l}, \quad U_e = W_{e\leftarrow i} W_{i\leftarrow l}$$

Then the above λ message is proportional to

$$\exp -\frac{1}{2}\left(c - DUx_l\right)^T (MD)^{-1}\left(c - DUx_l\right)$$

where $c^T = (c_k^T, c_e^T)$ and $U^T = (U_k^T, U_e^T)$. This gives the message update

$$g_{i\to l} = U^T M^{-1} c, \qquad G_{i\to l} = U^T M^{-1} DU$$

The posterior marginals, finally, are obtained in the standard way by just a slight modification of the ρ message:

$$p(x_i|v) = \int p(x_i|x_{\mathrm{pa}(i)}) \prod_{l\in \mathrm{pa}(i)} \rho_{l\to i}(x_l) \prod_{k\in \mathrm{ch}(i)} \lambda_{k\to i}(x_i)$$

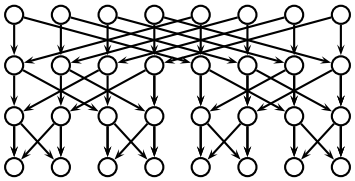

Fig. 1. The FFT network for computing the Fourier Transform of an 8 component data vector. The data vector v is clamped in reverse bit order in the bottom layer of the network. The Fourier components (in standard order) are, after inference in this network, given by the bits in the top layer. All nodes below the top layer have zero covariance so that with no missing data the top layer produces the exact FFT. In the case of missing data, the prior in the top layer is used to make the inference well-posed.

3. EXPERIMENTS ON AN FFT NETWORK

An example of a GaussDAG which contains zero covariances was discussed in [3], and provides an interesting and practical example for comparison of directed versus undirected BP methods. The Fast Fourier Transform FFT(x) normally deals with only the case of complete observations x. An elegant and potentially extremely useful method for dealing with missing data was proposed in [3], in which the one dimensional FFT was considered as a generative GaussDAG with the structure of a butterfly network, fig. 1. This is based on the well-known recursion for computing the FFT of an $n = 2^d$ dimensional vector x in $O(n \log n)$ time. For $W = \exp(-2\pi i/n)$, the k^{th} Fourier coefficient F_k is given by

$$F_k = \sum_{j=0}^{n/2-1} W^{2kj} x_{2j} + \sum_{j=0}^{n/2-1} W^{(2j+1)k} x_{2j+1} = F_k^e + W^k F_k^o$$

where F_k^e and F_k^o denote the k^{th} component of the length-$n/2$ Fourier transform of the even and odd components of x_j, respectively. This means that we can generate, from the Fourier coefficients at the top layer, the data points themselves (in reverse bit order) in the bottom layer. Using a prior on the Fourier coefficients in the top layer, computing the Fourier coefficients in the case of missing data is an inference problem in a GaussDAG in which some of the bottom layer nodes are clamped to their evidential states, and we wish to infer the top layer Fourier coefficients (as well as possibly the missing data in the bottom layer). In our implementation, we represent complex arithmetic using an equivalent two component vector arithmetic. We use a zero mean prior on the Fourier coefficients, and independent isotropic covariances. All noise covariances below the top layer are set to zero. In [3] inference in this network was performed using undirected BP (UBP) by first transforming the network into a pairwise Markov network. To deal with the problem of zero covariances, a small

"jitter" was substituted instead. The jitter can have a large effect on the numerical accuracy of UBP, and we used 10^{-11} which was based on experiments with small networks.

Naive UBP doesn't work well since convergence is hampered by many loops of length four. In our DBP we also found that the tight loops in the network caused difficulties with convergence and, as in [3], we used a clustering scheme to ameliorate this. We merged the two children of any parent node into a cluster variable; each node except for those in the top layer is then contained in one such cluster. In the top layer, we clustered nodes with common children, giving again non-overlapping clusters containing two nodes each. In the UBP implementation of [3], nodes that form a loop of length four were merged into four-node clusters (which can overlap each other). The two algorithms, UBP and DBP, then have roughly the same computational complexity per iteration. For a network of $n = 16$ nodes in each layer, we ran 100 experiments with half of the n data points missing at random and the prior FFT variances selected from a uniform distribution on $(0, 1)$. First we ran both UBP and DBP to convergence, defined as the point where the posterior means change by less than 10^{-15} from one iteration to the next. The number of iterations used by both methods to reach this convergence tolerance differed only by a small number of iterations. However, in about a third of runs UBP struggled to converge to this high tolerance and we then stopped it after a maximum of 50 iterations, which gave similar accuracy to otherwise converged runs. The resulting mean absolute error of the inferred FFT components from the true target values was $(1.5 \pm 1) \times 10^{-8}$ for UBP and $(3.2 \pm 1.4) \times 10^{-14}$ for DBP. Similar experiments give for $n = 32$: $(2.8 \pm 1.1) \times 10^{-8}$ for UBP and $(8.6 \pm 4) \times 10^{-14}$ for DBP. Here UBP usually did not converge within the maximally allowed 100 iterations, whereas DBP always converged within less than 50 iterations. For $n = 64$, both UBP and DBP struggled to converge within 100 iteration, giving errors of $(9.2 \pm 4) \times 10^{-8}$ for UBP and $(2.6 \pm 1.1) \times 10^{-13}$ for DBP. The increase in the errors stresses the importance of numerical accuracy as we increase the network size.

In fig. 2 we also consider the accuracy of the two methods as we increase n at a *fixed* number of iterations, chosen here as 20. The same random prior was used as before, with half the data vector missing at random, and plotted are the mean error and standard deviation over 20 such experiments. Whilst the performance of both DBP and UBP deteriorates with increasing n, DBP remains consistently superior by at least two orders of magnitude; this becomes critically important as the network size increases to practically interesting limits.

4. DISCUSSION

Directed Gaussian networks are a common form of continuous graphical models, for which accurate and stable inference techniques are of considerable interest. In cases where exact inference is impractical, BP is a useful approximation since, when it converges, the posterior means are correct. Whilst the two forms of BP – directed and undirected – may be made mathematically equivalent, their numerical stability is different, and may be dramatically so in many practical scenarios where noise covariances are small or even zero. Our approach was to derive a directed BP implementation without the explicit appearance of inverse noise covariances, for which the numerical performance is superior to undirected approximations on the equivalent pairwise Markov Network. MATLAB code is available from the first author's homepage.

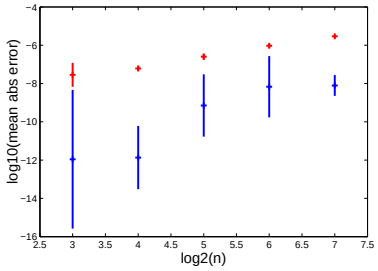

Fig. 2. Comparison of the accuracy of the UBP and DBP methods after 20 iterations of each algorithm. Plotted are the mean of the log of the absolute error, together with $\pm$ one standard deviation over the 20 random experiments (see text) for each network size. The upper results are for UBP and the lower ones for DBP.

5. REFERENCES

[1] M. Verhaegen and P. Van Dooren, "Numerical Aspects of Different Kalman Filter Implementations," *IEEE Transactions of Automatic Control*, vol. 31, no. 10, pp. 907–917, 1986.

[2] A. Galka, O. Yamashita, T. Ozaki, R. Biscay, and P. Valdes-Sosa, "A solution to the dynamical inverse problem of EEG generation using spatiotemporal Kalman filtering," *NeuroImage*, pp. 435 – 453, 2004.

[3] A. J. Storkey, "Generalised Propagation for Fast Fourier Transforms with Partial or Missing Data," in *Neural Information Processing Systems 16*, 2004.

[4] Y. Weiss and W.T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation*, vol. 13, no. 10, 2001.

[5] M. Verlaan and A. W. Heemink, "Reduced Rank Square Root Filters for Large Scale Data Assimilation Problems," *Second International Symposium on Assimilation of Observations in Meteorology and Oceanography, World Meteorological Organization*, pp. 247–252, 1995.

[6] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufman, San Mateo, 1988.

[7] D. Barber and P. Sollich, "Stable Directed Belief Propagation in Gaussian DAGs using the auxiliary variable trick," Tech. Rep. IDIAP-RR 05-72, IDIAP Research Institute, 2005.