Generalized belief propagation for approximate inference in hybrid Bayesian networks

Tom Heskes **Onno Zoeter**
SNN, University of Nijmegen
Geert Grooteplein 21, 6252 EZ, Nijmegen, The Netherlands
{*tom, orzoeter*}*@snn.kun.nl*

Abstract

We apply generalized belief propagation to approximate inference in hybrid Bayesian networks. In essence, in the algorithms developed for discrete networks we only have to change "strong marginalization" (exact) into "weak marginalization" (same moments) or, equivalently, the "sum" operation in the (generalized) sum-product algorithm into a "collapse" operation. We describe both a message-free single-loop algorithm based on fixed-point iteration and a more tedious double-loop algorithm guaranteed to converge to a minimum of the Kikuchi free energy. With the cluster variation method we can interpolate between the minimal Kikuchi approximation and the (strong) junction tree algorithm. Simulations on the emission network of [7], extended in [13], indicate that the Kikuchi approximation in practice often works really well, even in the difficult case of discrete children of continuous parents.

1 INTRODUCTION

Bayesian networks provide a natural framework for describing multivariate probability distributions and reasoning with uncertainty. The usual application is for domains with only discrete variables, but recently there has been growing interest in hybrid domains with both discrete and continuous variables [13, 16, 9]. Typical applications are in target tracking [1] and fault diagnosis [12]. Exact inference in hybrid Bayesian networks is even more complicated than in "standard" Bayesian networks with only discrete variables: exact inference in hybrid networks with a simple chain-like structure is NP-hard [11].

In discrete networks, exact inference is tractable in singly-connected structures. A popular algorithm is Pearl's belief propagation [18]. When "naively" applied in structures containing cycles, loopy belief propagation often leads to surprisingly accurate performance [17]. The notion that fixed points of loopy belief propagation correspond to extrema of the Bethe free energy [22] has been an important step in the theoretical understanding of this success. The cluster variation method [22, 24] leads to an inference algorithm referred to as "generalized belief propagation" and makes it possible to interpolate between standard belief propagation and exact inference as in the junction tree algorithm [10].

In this article, we will investigate whether it is possible to apply these ideas to approximate inference in hybrid Bayesian networks. The example that will guide us is the emission network from [7], displayed in Figure 1. First we will in Section 2 review (generalized) belief propagation in networks with discrete variables. We will present generalized belief propagation in message-free notation as a variational method that tries to minimize the Kikuchi free energy. Furthermore, we will describe a double-loop algorithm guaranteed to converge to such a minimum, similar to (but arguably simpler than) the CCCP algorithm of [24]. In Section 3 we will then consider the changes needed for approximate inference in hybrid networks, which happen to be really few. We will illustrate the performance of the resulting algorithm on the emission network in Section 4. In the network of Figure 1 all discrete variables are parents of continuous children. The more general case, including discrete children of continuous parents, is (even) more complicated [15]. Here we follow up on [13] to show that the same algorithm can be used here as well.

2 LOOPY BELIEF PROPAGATION

2.1 KIKUCHI FREE ENERGY

In this section we will review (generalized) belief propagation in networks with discrete variables and present it as a variational method that tries to minimize the

Kikuchi free energy. We will use the emission network in Figure 1 for illustration, for the moment acting as if all variables are discrete.

We will use the language of factor graphs [6]. We assume that the distribution over latent variables X can be written in the factorized form

$$P_{\text{exact}}(X) = \frac{1}{Z} \prod_{\alpha} \Psi_\alpha(X_\alpha) \,, \qquad (1)$$

with α numbering the factors or potentials Ψ_α and Z an overall normalization constant. In a Bayesian network, the factors consist of each child and its parents (i.e., the cliques in the moralized graph). Without evidence, the normalization constant Z equals 1. Evidence, both hard and soft, can be incorporated in the definition of the potentials [16]. With X denoting all latent variables and Y the observed ones, $P(X|Y)$ can be written in the form (1) with $Z = P(Y)$ the normalization constant.

The probability $P(X)$ in (1) is the solution of

$$P_{\text{exact}}(X) = \operatorname*{argmin}_{P} F(P) \,,$$

with the free energy

$$F(P) = \sum_{X} P(X) \log \left[\frac{P(X)}{\prod_\alpha \Psi_\alpha(X_\alpha)} \right] \,,$$

and where the minimization is under the constraint that P is a probability, i.e., nonnegative and normalized to 1. In the following these constraints on probability distributions are always implicitly assumed. In the traditional "mean-field" variational methods for approximate inference in graphical models (see e.g. the introduction in [5]), the approach is to restrict $P(X)$ to a tractable distribution. By construction the approximate mean-field free energy $F(P)$ is an upper bound of the exact free energy $F(P_{\text{exact}})$: $F(P) \geq F(P_{\text{exact}})$.

To arrive at (generalized) belief propagation, we confine our search to "tree-like probability distributions"

$$P_{\text{exact}}(X) \approx \frac{\prod_\alpha P_\alpha(X_\alpha)}{\prod_\beta P_\beta(x_\beta)^{-c_\beta}} \equiv \tilde{P}(X) \,, \qquad (2)$$

with c_β overcounting or Moebius numbers. Here $\tilde{P}(X)$ is some function, not necessarily normalizable. We will refer to x_β as variable subsets and write them in lower case to distinguish them from the "outer clusters" X_α. Typically, the variable subsets x_β are intersections of the outer clusters X_α. We write $\beta \subset \alpha$ to indicate that x_β is a subset of X_α. The overcounting numbers follow from the recursive Moebius formula $c_\beta = 1 - \sum_{\alpha \supset \beta} c_\alpha$, with $c_\alpha = 1$ for all outer clusters. The intuition is that, after canceling terms in the numerator and denominator, we should end up with a single term x_β in the numerator. The overcounting numbers c_β of the variable subsets are usually negative, i.e., terms $P_\beta(x_\beta)$ appear in the denominator of (2). $P_\alpha(X_\alpha)$ and $P_\beta(x_\beta)$ are interpreted as (approximate) local marginals that should normalize to 1, but should also be consistent:

$$\forall_\beta \forall_{\alpha \supset \beta} \quad P_\alpha(x_\beta) = P_\beta(x_\beta) \,. \qquad (3)$$

For singly-connected structures, it can be shown that the exact solution $P_{\text{exact}}(X)$ is of the form (2), with proportionality constant equal to 1 and where $P_\alpha(X_\alpha) = P_{\text{exact}}(X_\alpha)$ and $P_\beta(x_\beta) = P_{\text{exact}}(X_\beta)$. In structures containing cycles this need not be the case, but we can still assume it to be true approximately. Substituting (2) into the free energy and implementing the above assumptions ($P_{\text{exact}}(X_\alpha) = P_\alpha(X_\alpha)$, $P_{\text{exact}}(x_\beta) = P_\beta(x_\beta)$, and proportionally constant equal to 1), we obtain the Kikuchi free energy

$$\begin{aligned} F(\tilde{P}) \;=\; & \sum_\alpha \sum_{X_\alpha} P_\alpha(X_\alpha) \log \left[\frac{P_\alpha(X_\alpha)}{\Psi_\alpha(X_\alpha)} \right] \\ & + \sum_\beta c_\beta \sum_{x_\beta} P_\beta(x_\beta) \log P_\beta(x_\beta) \,. \quad (4) \end{aligned}$$

Here $\tilde{P}$ is now represented by the set of local marginals $P_\alpha(X_\alpha)$ and $P_\beta(x_\beta)$. Unlike the mean-field free energy, the Kikuchi free energy is "just" an approximation and *not* a bound of the exact free energy. Another important difference is that the mean-field methods fit a global probability distribution that is globally and locally consistent by construction, where here we do not care about global consistency and enforce local consistency through the constraints. The hope is that the Kikuchi free energy, which takes into account more of the original structure of the network, is a better approximation of the exact free energy and thus yields more accurate local marginals.

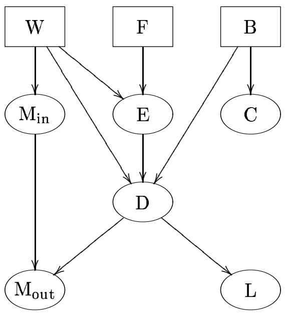
The tree-like approximation (2) corresponding to the example of Figure 1 is, in obvious notation,

$$\tilde{P}(X) = \frac{P(1{,}4)P(1{,}2{,}5)P(1{,}3{,}5{,}7)P(3{,}6)P(4{,}7{,}8)P(7{,}9)}{P(1)P(1{,}5)P(3)P(4)P(7)^2} \,. \qquad (5)$$

The factors in the numerator correspond to the potentials. The variable subsets in the denominator follow from the intersections between the outer clusters (see Figure 1(c)[1]). The overcounting numbers ensure that the numerator and denominator are balanced.

With the cluster variation method, we can choose the outer clusters larger than the subsets of variables as

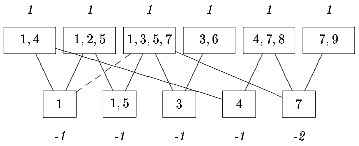
[1] This graphical visualization is similar to the "region graphs" in [23] that are used to compute the overcounting numbers. Here we focus on the communication lines between outer clusters and variable subsets.



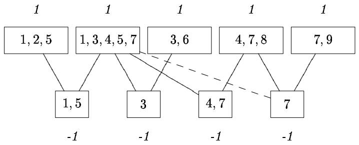
(a) Bayesian network.

1.	W	Waste Type
2.	F	Filter State
3.	B	Burning Regime
4.	M_{in}	Metal in Waste
5.	E	Efficiency
6.	C	CO2 Emission
7.	D	Dust Emission
8.	M_{out}	Metal Emission
9.	L	Light Penetrability

(b) Variables.



(c) Minimal approximation.



(d) A weak junction tree.

Figure 1: The emission network.

defined by the potentials. So, for example, instead of the six outer clusters (5), we can take five clusters:

$$\tilde{P}(X) = \frac{P(1{,}2{,}5)P(1{,}3{,}4{,}5{,}7)P(3{,}6)P(4{,}7{,}8)P(7{,}9)}{P(1{,}5)P(3)P(4{,}7)P(7)} \,. \tag{6}$$

This choice of outer clusters in fact corresponds to a particular weak junction tree, visualized in Figure 1(d). The restriction in the cluster variation method is that each of the potential subsets should be fully contained in at least one of the outer clusters. If we just redefine the potentials, e.g., through

$$\Psi_\alpha(X_\alpha) = \prod_{\gamma \subset \alpha} \Psi_\gamma(X_\gamma)^{\frac{1}{n_\gamma}} \,,$$

with n_γ the number of outer clusters that contain X_γ, we are back to our original definition (1). Alternatively, we can assign each potential subset to a single outer cluster, as in the junction tree algorithm.

If we choose the outer clusters such that they correspond to the cliques of a junction tree, the Kikuchi free energy is no longer an approximation but exact. With the cluster variation method, we can so interpolate between the minimal approximation (outer clusters as defined by the potentials) and the exact solution. Obviously, the price one has to pay is in the computational complexity that grows with the size of the outer clusters. In the case of non-overlapping subsets, the Kikuchi free energy reduces to the Bethe free energy.

2.2 GENERALIZED BELIEF PROPAGATION

In [22] it is shown that fixed points of loopy belief propagation correspond to extrema of the Bethe free energy. Even better, both empirical and theoretical results [3] show that *stable* fixed points of loopy belief propagation must correspond to (local) *minima* of the Bethe free energy. Both results were derived for the case of non-overlapping subsets x_β. Here we consider the slightly more general case of overlapping subsets. Rather than writing belief propagation as a sum-product rule on factor graphs [6], we describe here the message-free interpretation (see e.g. [20]), which is closer in spirit to the junction tree algorithm. We start by initializing all factor and variable beliefs

$$P_\alpha(X_\alpha) \propto \Psi_\alpha(X_\alpha) \quad \text{and} \quad P_\beta(x_\beta) \propto 1 \,. \tag{7}$$

Our approximation (2) equals, up to irrelevant normalization, the exact distribution (1), but the factor and variable beliefs are obviously incorrect. As in the junction tree algorithm, we are now going to change the factor and variable beliefs to make them approximate the exact marginals as closely as possible. We do this under the constraint that the approximation (2) stays the same.

Given a new estimate of the variable belief $P_\beta^{\text{new}}(x_\beta)$, we want to update the factor belief $P_\alpha(X_\alpha)$ such that $P_\alpha^{\text{new}}(x_\beta) = P_\beta^{\text{new}}(x_\beta)$. This leads to

$$P_\alpha^{\text{new}}(X_\alpha) \propto P_\alpha(X_\alpha) \frac{P_\beta^{\text{new}}(x_\beta)}{P_\alpha(x_\beta)} \,. \tag{8}$$

Now suppose that we update all factors α with $\alpha \supset \beta$ at once (other schedules are possible as well and can be treated similarly; see footnote 2 for the connection with the junction tree algorithm). The update in the variable belief $P_\beta(x_\beta)$ follows from the constraint $\tilde{P}^{\text{new}}(X) \propto \tilde{P}(X)$ and thus, when only β and $\alpha \supset \beta$ are updated,

$$\frac{\prod\limits_{\alpha \supset \beta} P_\alpha^{\text{new}}(X_\alpha)}{P_\beta^{\text{new}}(x_\beta)^{-c_\beta}} \propto \frac{\prod\limits_{\alpha \supset \beta} P_\alpha(X_\alpha)}{P_\beta(x_\beta)^{-c_\beta}} \,.$$

Algorithm 1 Message-free GBP.

```
initialize $P_\alpha(X_\alpha) = \Psi_\alpha(X_\alpha)$ and $P_\beta(x_\beta) = 1$
repeat
  for all variable subsets $\beta$ do
    update $P_\beta(x_\beta) \leftarrow P_\beta^{\text{new}}(x_\beta)$ from (9)
    for all neighboring factors $\alpha \supset \beta$ do
      update $P_\alpha(X_\alpha) \leftarrow P_\alpha^{\text{new}}(X_\alpha)$ from (8)
    end for
  end for
until convergence
return $P_\alpha(X_\alpha)$ and $P_\beta(x_\beta)$
```

Substitution of (8) and a little rewriting yields

$$P_\beta^{\text{new}}(x_\beta) \propto P_\beta(x_\beta)^{\frac{c_\beta}{n_\beta + c_\beta}} \bar{P}_\beta(x_\beta)^{\frac{n_\beta}{n_\beta + c_\beta}} \tag{9}$$

with

$$\bar{P}_\beta(x_\beta) \propto \left[\prod_{\alpha \supset \beta} P_\alpha(x_\beta)\right]^{\frac{1}{n_\beta}} \tag{10}$$

the logarithmic average of all $P_\alpha(x_\beta)$. The final algorithm is summarized in Algorithm 1. Although in principle normalization can be delayed until the (unnormalized) beliefs have converged, for numerical stability it helps to normalize the beliefs once in a while. More importantly, the full update (8) is often too greedy and one has to resort to the damped version

$$P_\alpha^{\text{new}}(X_\alpha) \propto P_\alpha(X_\alpha) \left[\frac{P_\beta^{\text{new}}(x_\beta)}{P_\alpha(x_\beta)}\right]^\epsilon , \tag{11}$$

with $0 < \epsilon \leq 1$ an appropriate step size.

With straightforward manipulations, it can be shown that fixed points of Algorithm 1 correspond to extrema of the Kikuchi free energy under the constraints (3). The proof introduces Lagrange multipliers for each of the constraints, constructs the Lagrangian, takes the derivatives w.r.t. $P_\alpha(X_\alpha)$ and $P_\beta(x_\beta)$ and sets them to zero, yielding

$$\begin{aligned} \log \Psi_\alpha(X_\alpha) - \log P_\alpha(X_\alpha) + \sum_{\beta \subset \alpha} \lambda_{\alpha\beta}(x_\beta) &= \text{constant} \\ c_\beta \log P_\beta(X_\beta) + \sum_{\alpha \supset \beta} \lambda_{\alpha\beta}(x_\beta) &= \text{constant} \end{aligned}$$

Taking in the first line a sum over all factors α and in the second line over all variable subsets β, we can get rid of the Lagrange multipliers and find that at an extremum of the Kikuchi free energy the factor and variable beliefs should satisfy

$$\frac{\prod_\alpha P_\alpha(X_\alpha)}{\prod_\beta P_\beta(x_\beta)^{-c_\beta}} \propto \prod_\alpha \Psi_\alpha(X_\alpha) , \tag{12}$$

which is how we initialize in (7). Furthermore, since we derived the update of the variable beliefs in (9) such that $\tilde{P}(X)$ remains invariant, the equality (12) also holds after convergence. At convergence, i.e., when $P_\alpha^{\text{new}} \propto P_\alpha$ in (8), the constraints (3) are satisfied.

In Algorithm 1, we have taken the convention that we update the factor beliefs of all outer clusters α that subsume a particular variable subset β. The equivalent statement is that we take into account and introduce Lagrange multipliers for all possible constraints between variable subsets β and neighboring factors $\alpha \supset \beta$. This, however, is not always necessary. Consider for example the distribution in (5). The above convention implies that we "send messages" (update factors based on variable beliefs) between variable subset 1 and all its neighboring outer clusters, i.e., $(1,4)$, $(1,2,5)$, and $(1,3,5,7)$. However, sending messages between 1 and $(1,3,5,7)$ is unnecessary: the constraint $P_{(1,3,5,7)}(1) = P_1(1)$ is already implied by the constraint $P_{(1,3,5,7)}(1,5) = P_{(1,2,5)}(1,5)$. In the construction of the Lagrangian we can leave out the constraint and corresponding Lagrange multiplier. An alternative interpretation is that we redefine the notion of "neighbors" and no longer refer to $(1,3,5,7)$ as a neighboring outer cluster of variable subset 1. This then brings us closer to the standard junction tree algorithm[2]. Unnecessary links are in Figure 1(c) and (d) indicated by the dashed lines. Another option, as described in [22], is to only pass messages between variable subsets and their direct sub- and superclusters (which may be other variable subsets). For notational convenience we stick to the full set of all constraints between outer clusters and variable subsets and to the ordering of updates as outlined in Algorithm 1. In practice, there may be a lot to gain with clever choices, especially with regard to speed of convergence (see e.g. [20] where the scheduling of updates follows the structure of a spanning tree).

2.3 DOUBLE-LOOP ALGORITHM

One of the problems with (generalized) loopy belief propagation is that convergence to a minimum of the Kikuchi free energy cannot be guaranteed. In fact, simple examples can be constructed in which minima of the Bethe free energy are unstable under loopy belief propagation, even in the limit of infinitely small

[2] To make the connection with the junction tree algorithm explicit, consider two cliques, called α and α', with their separator β. By construction we have $c_\beta = -1$ and $n_\beta = 2$. Alternately we set, instead of (9), $P_\beta^{\text{new}}(x_\beta) = P_\alpha(x_\beta)$ and $P_\beta^{\text{new}}(x_\beta) = P_{\alpha'}(x_\beta)$. Suppose that at some point we have $P_\beta(x_\beta) = P_\alpha(x_\beta)$ and thus set $P_\beta^{\text{new}}(x_\beta) = P_{\alpha'}(x_\beta)$. We then apply the update (8) for $P_\alpha(X_\alpha)$, without updating $P_{\alpha'}(X_{\alpha'})$. It is easy to check that this update scheme also keeps (2) invariant.

step sizes ϵ [3]. The more direct approach is then to minimize the Kikuchi free energy (4) under the constraints (3). This constrained minimization problem is well-defined, but not necessarily convex, mainly because of the negative $P_\beta \log P_\beta$-terms. The crucial trick, implicit or explicit in recently suggested procedures is to bound [24] or clamp [19] the possibly concave part (outer loop: recompute the bound) and solve the remaining convex problem (inner loop: maximization with respect to Lagrange multipliers). Here we will restrict ourselves to the case in which all overcounting numbers c_β corresponding to the variable subsets β are negative (the terms with positive overcounting numbers are convex anyways and incorporating them only complicates notation; furthermore positive overcounting numbers c_β do not occur in any of the examples below).

Using the linear bound

$$-\sum_{x_\beta} P_\beta(x_\beta) \log P_\beta(x_\beta) \leq -\sum_{x_\beta} P_\beta(x_\beta) \log P_\beta^{\text{old}}(x_\beta) \,, \tag{13}$$

from $\text{KL}(P_\beta, P_\beta^{\text{old}}) \geq 0$ and with $P_\beta^{\text{old}}(x_\beta)$ the previous setting of the variable beliefs, we can construct a convex bound of the Kikuchi free energy:

$$F_{\text{bound}}(P) = \sum_\alpha \sum_{X_\alpha} P_\alpha(X_\alpha) \log \left[\frac{P_\alpha(X_\alpha)}{\hat{\Psi}_\alpha(X_\alpha)} \right] \geq F(P) \,, \tag{14}$$

with

$$\log \hat{\Psi}_\alpha(X_\alpha) \equiv \log \Psi_\alpha(X_\alpha) - \sum_{\beta \subset \alpha} \frac{c_\beta}{n_\beta} \log P_\alpha^{\text{old}}(x_\beta) \,. \tag{15}$$

Each outer-loop step corresponds to a reset of the bound, i.e., at the start of the inner loop we have $F_{\text{bound}}(P) = F(P)$. In the inner loop, we solve the constrained minimization problem implied by the convex bound. This is a convex problem with linear constraints, which thus has a unique solution. At the end of the inner loop, we then have $F(P^{\text{new}}) \leq F_{\text{bound}}(P^{\text{new}}) \leq F_{\text{bound}}(P) = F(P)$.

For the inner loop, we can simply apply the algorithm outlined in the previous section. Comparing the bound (14) with the original Kikuchi free energy, we note that it has exactly the same form: we only have to make the substitutions $\hat{\Psi}_\alpha$ for Ψ_α and $c_\beta = 0$. With these substitutions, the updates described in Algorithm 1 are not just fixed point iterations, but guarantee convergence to the unique minimum of the bound (14) under the appropriate constraints. In fact, with the particular scheduling of Algorithm 1 (run over variable subsets and update all neighboring factor beliefs) we do not need any damping, i.e., can take $\epsilon = 1$. A proof can be found in [3] (construct the concave Lagrangian, derive updates of the Lagrange multipliers from the fixed point equations, show that these guarantee an increase in the Langragian unless the Lagrange multipliers are at the unique maximum, turn the updates in the Lagrange multipliers into updates in variable and factor beliefs).

Algorithm 2 Convergent double-loop algorithm.

```
initialize $P_\beta(x_\beta) = 1$
repeat
    update $\hat{\Psi}_\alpha(X_\alpha)$ as in (15)
    run Algorithm 1 with $\Psi_\alpha = \hat{\Psi}_\alpha$ and $c_\beta = 0$
until convergence
return $P_\alpha(X_\alpha)$ and $P_\beta(x_\beta)$
```

The resulting double-loop algorithm is summarized in Algorithm 2. It is similar in spirit to the CCCP algorithm of [24]. The use of the bound (13) and the observation that there is no need to pass messages between variable subsets and other variable subsets makes Algorithm 2 somewhat easier to implement. Furthermore, it translates directly to the case of hybrid Bayesian networks, to be discussed next.

3 HYBRID BAYESIAN NETS

3.1 WEAK MARGINALIZATION

Now that we have done all the ground work, we can try and apply the above machinery to approximate inference in hybrid Bayesian networks consisting of both discrete and continuous nodes.

The crucial operations for the discrete case are the updates (8) and (9), combined with the definition (10). All operations are (weighted) products and divisions of marginals, except for the marginalization

$$P_\alpha(x_\beta) = \sum_{X_{\alpha \setminus \beta}} P_\alpha(X_\alpha) \,, \tag{16}$$

which corresponds to a sum-operation (hence the name "sum-product algorithm"). Discrete potentials and beliefs can be represented as tables. Marginalizing out variables yields another table of smaller size. Changing summation into integration, we can handle Bayesian networks consisting of continuous Gaussian variables in a similar manner. Continuous Gaussian potentials and beliefs are summarized with a mean, covariance matrix, and (if necessary) proportionality constant. Marginalization is again "closed": integrating out variables of a Gaussian yields another (smaller) Gaussian.

The combination of both discrete and continuous Gaussian variables yields a conditional Gaussian. A conditional Gaussian potential on $X = \{S, Z\}$ is a different Gaussian distribution on the continuous variable

Z for each realization of the discrete part S. It can be written in the form

$$\Psi(S=i,Z) = p_i \exp\left[-\frac{1}{2}(Z-\mu_i)'\Sigma_i^{-1}(Z-\mu_i)\right] .$$

The important difference with the pure discrete and pure Gaussian case is that marginalization of conditional Gaussian beliefs is *not* closed: summing out discrete variables yields a mixture of Gaussians, which is not a conditional Gaussian. To see this, consider the conditional Gaussian $P(Z, S_1, S_2)$, with Z the continuous variable and S_1 and S_2 two discrete variables. We have $|S_1| \times |S_2|$ different Gaussians, one for each realization of $\{S_1, S_2\}$. Now, the distribution $P(Z, S_1)$ that follows by summing out S_2 as in (16) boils down to a mixture of $|S_2|$ Gaussians for each realization of S_1, which is *not* a conditional Gaussian. This "non-closure" of conditional Gaussian potentials under marginalization makes exact inference in hybrid Bayesian networks much harder than in networks with just discrete or just Gaussian nodes. In fact, it can be shown that in general inference in hybrid Bayesian networks, even with a singly-connected structure, is NP-hard [11]: the number of mixture components required to describe the exact distribution is exponential in the number of discrete variables.

The standard approximation is to replace the "strong" marginal in (16) with a "weak" marginal [7]. The weak marginal can be defined as the conditional Gaussian with the same moments as the strong marginal. Since the conditional Gaussian is in the exponential family, this is the best approximation in the sense of minimizing the KL divergence [8]. With S_1 running over states i and S_2 over j, the weak marginal $P(S_1, Z)$ of $P(S_1, S_2, Z)$ is the conditional Gaussian with components $p_i = \sum_j p_{ij}$, $\mu_i = \sum_j p_{j|i}\mu_{ij}$, and $\Sigma_i = \sum_j p_{j|i}\left[\Sigma_{ij} + \delta_{ij}\delta'_{ij}\right]$, where $p_{j|i} = p_{ij}/p_i$ and $\delta_{ij} = \mu_{ij} - \mu_i$. Basically, for each different i, we collapse the mixture of $|S_2|$ Gaussians to a single Gaussian with the same mean and covariance.

3.2 IMPLICATIONS FOR GBP

Summarizing, to guarantee closure under marginalization we propose to work with weak rather than strong marginals, i.e., we replace (16) with

$$P_\alpha(x_\beta) = \underset{X_{\alpha\backslash\beta}}{\text{Collapse}}\, P(X_\alpha) . \tag{17}$$

Perhaps surprisingly, this is about the only change that we have to make to apply the algorithms outlined in Section 2 for approximate inference in hybrid Bayesian networks. We can make the following statements.

- Fixed points of Algorithm 1 correspond to extrema of the Kikuchi free energy (4) under (3), which are now to be interpreted as *weak* rather than strong marginalization constraints, i.e., the factor beliefs $P_\alpha(x_\beta)$ and $P_{\alpha'}(x_\beta)$ only have to agree upon their moments.
- The bounds (13) and (14) still apply and thus Algorithm 2 guarantees convergence to a minimum of the Kikuchi free energy under the weak marginalization constraints[3].

Proofs of these statements are a direct generalization of the ones for strong marginalization.

The collapse operation (17) turns the sum-product algorithm into a "collapse-product" algorithm. In the restricted case of non-overlapping variable subsets x_β, this collapse-product algorithm can be mapped onto expectation propagation [14]. Perhaps the case of overlapping subsets and weak marginalization can be interpreted as "generalized" expectation propagation, in much the same sense as "generalized" belief propagation generalizes upon belief propagation.

Generalized belief propagation with weak marginalization relates to the strong junction tree algorithm of [7] in the same way as generalized belief propagation with strong marginalization relates to the (standard) junction tree algorithm. The strong junction tree results from a procedure called strong triangulation. In terms of an elimination ordering strong triangulation corresponds to eliminating the continuous variables before the discrete ones [2]. In practice, this often boils down to having all discrete variables in one clique (as is the case for all strong junction trees that can be constructed from the example in Figure 1). The strong junction tree algorithm is "exact" in the sense that it yields the correct distribution over the discrete variables and the correct means and covariances for the continuous ones. The claim, to be empirically checked below, is that if we choose as our outer clusters the cliques of the strong junction tree, we arrive at the same "exact" solution as the strong junction tree algorithm.

4 SIMULATIONS

4.1 CONDITIONAL GAUSSIAN MODEL

Our first set of simulations considers the emission network of [7], visualized in Figure 1. We followed the

[3] A slight difference is that in the inner loop we may have to resort to a step size $\epsilon < 1$: the guaranteed increase in the Lagrangian for step size $\epsilon = 1$ is specific to strong marginalization. However, the proposed updates still correspond to gradient ascent on the Lagrangian and standard techniques can be applied to find an appropriate step size. See [4] for the same phenomenom.

exact same experiments as those described in [7]. We ran our algorithms for many different choices of outer clusters. Typical results are summarized in Table 1. In all cases that the outer clusters correspond to the cliques of a strong junction tree, the obtained single-node probabilities, means and covariances are equal to the ones computed with "brute force" (putting all nodes in a single clique). Without evidence, the clusters in (6) that correspond to a weak rather than strong junction tree also give the exact results. Minor differences are found for the minimal approximation (both with and without evidence) and the "weak" approximation (with evidence).

4.2 DISCRETE CHILDREN OF CONTINUOUS PARENTS

The emission network of Figure 1 is a conditional linear Gaussian model. In conditional linear Gaussian models there are no discrete children with continuous parents. As a consequence, the exact distribution is a conditional Gaussian. The more complicated case that includes discrete children of continuous parents is treated in [15, 13]. It fits well within our framework: the only extra complication is that the collapse operation (17) can no longer be computed analytically, but requires numerical integration (see [13] for details). As an example, we tried to reproduce the experiment reported in [13] on the extended emission network, which includes three discrete sensor nodes (dust, CO2, and metal) attached to the corresponding continuous emission nodes. In this experiment, both the metal and CO2 sensor are clamped to "high" and the distribution of the dust sensor (D) is queried.

We ran the algorithms on several junction trees, all "strong" according to the definitions in [13], and compared the resulting marginals with the brute-force marginals obtained by putting all variables into a single cluster. With all strong junction trees, we replicate the brute-force result $D = 3.419 \pm 1.007$ and obtain a summed KL-divergence equal to zero within machine precision. Results obtained with clusters following the cliques of a weak junction tree (as in (6) with the additional softmax potentials added individually) and with the minimal approximation corresponding to the cliques in a moralized graph are of about the same (acceptable) quality: $D = 3.397 \pm 0.838$ with a summed KL-divergence of 0.029 and $D = 3.494 \pm 1.005$ with a summed KL-divergence of 0.014, respectively. Similar performance is obtained with other cluster choices. Both the "weak" and the minimal approximation lead to a considerable speedup. For example, our weak junction tree had at most 3 discrete variables in each cluster and the minimal approximation at most 2, which is to be compared with 5 in the smallest strong junction trees.

5 DISCUSSION

We have shown that by changing strong into weak marginalization, algorithms designed for approximate inference in (loopy) discrete networks can be directly transfered to hybrid networks. The connection with the Bethe and Kikuchi free energies give these methods a strong theoretical basis and the empirical results are very promising. At least, the Kikuchi approximation provide a viable alternative to the (structured) mean-field approaches [5, 21].

There is still a lot of work to do. Sometimes convergence can be really slow, for no obvious reason. This may have to do with numerical stability, but can also be related to "supportiveness": what guarantees that the constructed factor beliefs are normalizable? More generally, both the single-loop and double-loop algorithms can probably be much improved upon, especially with more clever scheduling of updates following ideas in e.g. [20] and choice of necessary constraints (see the discussion in Section 2.2). Important open theoretical questions are under which conditions the Bethe and Kikuchi free are bounded from below (the proof in [14] for the Bethe free energy of expectation propagation has the premise that all potentials are finite, which need not be the case in hybrid networks) and have a unique minimum. Last but not least, the current set of algorithms incorporates all evidence in the definition of the potentials and therefore does not allow for fast retraction, which may be an important issue in practical applications.

Acknowledgements

We would like to thank Wim Wiegerinck for helpful input, the reviewers for their constructive comments, and acknowledge support by the Dutch Technology Foundation STW and the Dutch Centre of Competence Paper and Board.

References

[1] Y. Bar-Shalom and X. Li. *Estimation and Tracking: Principles, Techniques, and Software.* Artech House, 1993.

[2] R. Cowell. Advanced inference in Bayesian networks. In M. Jordan, editor, *Learning in Graphical Models*, pages 27–49. Kluwer Academic Publishers, Dordrecht, 1998.

[3] T. Heskes. Stable fixed points of loopy belief propagation are minima of the Bethe free energy. In *Advances in Neural Information Processing Systems 15 (accepted)*, 2003.

	no evidence			with evidence		
	exact	weak	minimal	exact	weak	minimal
W	0.71	0.71	0.71	0	0	0
F	0.95	0.95	0.95	0.9995	*0.9996*	*0.9996*
B	0.85	0.85	0.85	0.01	0.01	0.01
$\mathrm{M_{in}}$	-0.21 ± 0.46	-0.21 ± 0.46	-0.21 ± 0.46	0.50 ± 0.10	0.50 ± 0.10	0.50 ± 0.10
E	-3.25 ± 0.71	-3.25 ± 0.71	-3.25 ± 0.71	-3.90 ± 0.08	-3.90 ± *0.07*	-3.90 ± *0.07*
C	-1.85 ± 0.51	-1.85 ± 0.51	-1.85 ± 0.51	-0.90 ± 0	-0.90 ± 0	-0.90 ± 0
D	3.04 ± 0.77	3.04 ± 0.77	3.04 ± 0.77	3.61 ± 0.33	3.61 ± 0.33	3.61 ± 0.33
$\mathrm{M_{out}}$	2.83 ± 0.86	2.83 ± 0.86	2.83 ± *0.90*	4.11 ± 0.34	4.11 ± 0.34	4.11 ± 0.34
L	1.48 ± 0.63	1.48 ± 0.63	1.48 ± 0.63	1.10 ± 0	1.10 ± 0	1.10 ± 0
KL	0	0	0.002	0	0.003	0.003

Table 1: Results on the emission network. Stated are the probabilities for the discrete variables and means and standard deviations of the continuous ones. “Exact”: outer clusters correspond to the cliques of a strong junction tree. “Weak”: outer clusters are the cliques of the weak junction tree (6). “Minimal”: outer clusters are the factors corresponding to the potentials as in (5). Lower row gives the KL-divergence with the strong junction tree marginals, summed over all variables. The evidence is on nodes W, C, and L as in [7].

[4] T. Heskes and O. Zoeter. Expectation propagation for approximate inference in dynamic Bayesian networks. In *Proceedings UAI-2002*, pages 216–233, 2002.

[5] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. In M. Jordan, editor, *Learning in Graphical Models*, pages 183–233. Kluwer Academic Publishers, Dordrecht, 1998.

[6] F. Kschischang, B. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[7] S. Lauritzen. Propagation of probabilities, means and variances in mixed graphical association models. *Journal of American Statistical Association*, 87:1098–1108, 1992.

[8] S. Lauritzen. *Graphical models.* Oxford University Press, Oxford, 1996.

[9] S. Lauritzen and F. Jensen. Stable local computation with conditional Gaussian distributions. *Statistics and Computing*, 11:191–203, 2001.

[10] S. Lauritzen and D. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistics Society B*, 50:157–224, 1988.

[11] U. Lerner and R. Parr. Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proceedings UAI-2001*, pages 310–318, 2001.

[12] U. Lerner, R. Parr, D. Koller, and G. Biswas. Bayesian fault detection and diagnosis in dynamic systems. In *AAAI/IAAI*, pages 531–537, 2000.

[13] U. Lerner, E. Segal, and D. Koller. Exact inference in networks with discrete children of continuous parents. In *Proceedings UAI-2001*, pages 319–328, 2001.

[14] T. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings UAI-2001*, pages 362–369, 2001.

[15] K. Murphy. Learning switching Kalman-filter models. Technical report, Compaq CRL., 1998.

[16] K. Murphy. A variational approximation for Bayesian networks with discrete and continuous latent variables. In *Proceedings UAI-1999*, pages 457–466, 1999.

[17] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings UAI-1999*, pages 467–475, 1999.

[18] J. Pearl. *Probabilistic Reasoning in Intelligent systems: Networks of Plausible Inference.* Morgan Kaufmann, San Francisco, CA, 1988.

[19] Y. Teh and M. Welling. The unified propagation and scaling algorithm. In *Advances in Neural Information Processing Systems 14*, 2002.

[20] M. Wainwright, T. Jaakola, and A. Willsky. Tree-based reparameterization for approximate estimation on loopy graphs. In *Advances in Neural Information Processing Systems 14*, 2002.

[21] W. Wiegerinck. Variational approximations between mean field theory and the junction tree algorithm. In *Proceedings UAI-2000*, pages 626–633, 2000.

[22] J. Yedidia, W. Freeman, and Y. Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems 13*, pages 689–695, 2001.

[23] J. Yedidia, W. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *IJCAI 2002*, 2002.

[24] A. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14:1691–1722, July 2002.