Extended Version of "Expectation propagation for approximate inference in dynamic Bayesian networks"

Tom Heskes & Onno Zoeter
SNN, University of Nijmegen
Geert Grooteplein 21, 6525 EZ, Nijmegen, The Netherlands

April 8, 2003

Abstract

We consider inference in general dynamic Bayesian networks. We are especially interested in models in which exact inference becomes intractable, as, for example, in switching linear dynamical systems. We introduce expectation propagation as a straightforward extension of Pearl's exact belief propagation. Expectation propagation, is a greedy algorithm, converges in many practical cases, but not always. Our goal is therefore to derive a message propagation scheme that can be guaranteed to converge.

Following Minka, we therefore first derive a Bethe-free energy like functional, the fixed points of which correspond to fixed points of expectation propagation. We turn this primal objective into a dual objective in terms of messages or Lagrange multipliers. Approximate inference boils down to a saddle-point problem: maximization with respect to the difference between forward and backward messages, minimization with respect to the sum of the forward and backward messages. We derive several variants, ranging from a double-loop algorithm that guarantees convergence to the saddle point to a damped version of "standard" expectation propagation. We discuss implications for approximate inference in general, tree-structured or even loopy, Bayesian networks.

This technical report formed the basis of [12]:

```
@inproceedings{Heskes_Zoeter_UAI2002,
AUTHOR = "Heskes, T. and Zoeter, O.",
YEAR = 2002,
TITLE = "Expectation propagation for approximate inference in
         dynamic {B}ayesian networks",
BOOKTITLE = "Proceedings UAI-2002",
EDITOR = "Darwiche, A. and Friedman, N.",
PAGES = "216--233",
URL = "ftp://ftp.snn.kun.nl/pub/snn/pub/reports/Heskes.uai2002.ps.gz"}
```

It is, however, not fully compatible. I tried to update it backwardly, but may not have completely succeeded. If possible, please refer to the published proceedings paper.

1 Introduction

Pearl's belief propagation [29] is a popular algorithm for inference in Bayesian networks. It is known to be exact in special cases, e.g., for tree-structured (singly connected) networks with just Gaussian or just discrete nodes. However, in many cases it fails, for the following reasons.

Structural. When loops are present, the network is no longer singly connected and belief propagation does not necessarily yield the correct marginals. However, loopy belief propagation, that is, Pearl's belief propagation applied to networks containing cycles, empirically leads to good performance (approximate marginals close to exact marginals) in many cases [28, 22].

Non-structural. Even although the network is singly connected, belief propagation itself is intractable or infeasible. A well-known example is inference in hybrid networks consisting of combinations of discrete and continuous (Gaussian) nodes [18]. Another example is inference in factorial hidden Markov models [9], which becomes infeasible when the dimension of the state space gets too large.

In this article, we focus on the "non-structural" failure of Pearl's belief propagation. To this end, we restrict ourselves to dynamic Bayesian networks: these have the simplest singly-connected graphical structure, namely a chain. We will review belief propagation in dynamic Bayesian networks for exact inference in Section 2. Examples of dynamic Bayesian networks in which exact inference can become intractable or highly infeasible are the factorial hidden Markov models mentioned before, switching linear dynamical systems (the dynamic variant of a hybrid network), nonlinear dynamical systems, and variants of dynamic hierarchical models.

Algorithms for approximate inference in dynamic Bayesian networks can be roughly divided into two categories: sampling approaches and parametric approaches. Popular sampling approaches in the context of dynamic Bayesian networks are so-called particle filters (see [6] for a collection of recent advances). With regard to the parametric approaches we can make a further subdivision into variational approaches and greedy projection algorithms. In the variational approaches (see e.g. [14] for an overview) an approximate tractable distribution is fitted against the exact intractable one. The Kullback-Leibler divergence between the exact and approximate distribution plays the role of a well-defined global error criterion. Examples are variational approaches for switching linear dynamical systems [8] and factorial hidden Markov models [9]. The greedy projection approaches are more local: they are similar to standard belief propagation, but include a projection step to a simpler approximate belief. Examples are the extended Kalman filter [13], generalized pseudo-Bayes for switching linear dynamical systems [2, 16], and the Boyen-Koller algorithm for hidden Markov models [4]. In this article, we will focus on these greedy projection algorithms.

In his PhD thesis [25] (see also [24]), Minka introduces expectation propagation, a family of approximate inference algorithms that includes loopy belief propagation and many (improved and iterative versions of) greedy projection algorithms as special cases. In Section 3 we will derive expectation propagation as a straightforward extension of exact belief propagation, the only difference being an additional projection in the procedure for updating messages. We illustrate expectation propagation in Section 3.3 on switching linear dynamical systems and variants of dynamic hierarchical models. Although arguably easier to understand, our formulation of expectation propagation applied to dynamic Bayesian networks is just a specific case of expectation propagation for chain-like graphical structures (see Section 3.2).

In exact belief propagation the updates for forward and backward messages do not interfere, which can be used to show that a single forward and backward pass are sufficient to converge to the correct beliefs. With the additional projection, the messages do interfere and a single forward and backward pass may not be sufficient, suggesting an iterative procedure to try and get better estimates, closer to the exact beliefs. A problem, however, with expectation propagation in

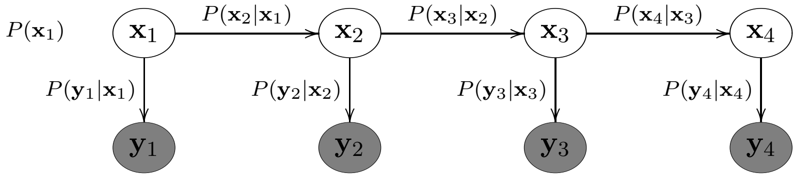
$P(\mathbf{x}_1)$
$\mathbf{x}_1$
$P(\mathbf{x}_2|\mathbf{x}_1)$
$\mathbf{x}_2$
$P(\mathbf{x}_3|\mathbf{x}_2)$
$\mathbf{x}_3$
$P(\mathbf{x}_4|\mathbf{x}_3)$
$\mathbf{x}_4$
$P(\mathbf{y}_1|\mathbf{x}_1)$
$P(\mathbf{y}_2|\mathbf{x}_2)$
$P(\mathbf{y}_3|\mathbf{x}_3)$
$P(\mathbf{y}_4|\mathbf{x}_4)$
$\mathbf{y}_1$
$\mathbf{y}_2$
$\mathbf{y}_3$
$\mathbf{y}_4$

Figure 1: Graphical representation of a dynamic Bayesian network.

general is that it does not always converge [25]. An important and open question is whether this is due to the graphical structure (e.g. messages running in cycles in networks with loops), to projections in the message update (causing interference between forward and backward passes even in chains), or both. By restricting ourselves to approximate inference on chains, we focus on the impact of the projections, unconcealed by structural effects.

In an attempt to derive convergent algorithms, we first have to define what we would like to optimize. Inspired by the Bethe free energy recently proposed for loopy belief propagation [38] and its generalization to expectation propagation [24, 23], we derive in Section 4.1 a similar functional for expectation propagation in dynamic Bayesian networks. It is easy to show that fixed points of expectation propagation correspond to extrema of this free energy and vice versa. The primal objective boils down to a non-convex minimization problem with linear constraints. In Section 4.3 we turn this into a constrained convex minimization problem at the price of an extra minimization over canonical parameters. This formulation suggests a double-loop algorithm, which is worked out in Section 5.

This double-loop algorithm is guaranteed to converge, but requires full completion of each inner loop. The equivalent description in terms of a saddle-point problem in Section 6 suggests faster short-cuts. The first one can be loosely interpreted as a combination of (natural) gradient descent and ascent. The second one is based on a damped version of “standard” expectation propagation. Both algorithms can be shown to be locally stable close to the saddle point.

Simulation results regarding expectation propagation applied to switching linear dynamical systems are presented in Section 7. It is shown that it makes perfect sense to try and find the minimum of the free energy even when standard undamped expectation propagation does not converge. In Section 8 we end with conclusions and a discussion of implications for approximate inference in general Bayesian networks.

2 Dynamic Bayesian networks

2.1 Probabilistic description

We consider general dynamic Bayesian networks with latent variables $\mathbf{x}_t$ and observations $\mathbf{y}_t$. The graphical model is visualized in Figure 1 for $T = 4$ time slices. The joint distribution of latent variables $\mathbf{x}_{1:T}$ and observables $\mathbf{y}_{1:T}$ can be written in the form

$$P(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = \prod_{t=1}^{T} \psi_t(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{y}_t) ,$$

where

$$\psi_t(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{y}_t) = P(\mathbf{x}_t|\mathbf{x}_{t-1}) P(\mathbf{y}_t|\mathbf{x}_t) ,$$

and with the convention $\psi_1(\mathbf{x}_0, \mathbf{x}_1, \mathbf{y}_1) \equiv \psi_1(\mathbf{x}_1, \mathbf{y}_1)$, i.e., $P(\mathbf{x}_1|\mathbf{x}_0) = P(\mathbf{x}_1)$, the prior. Our definition of the potentials $\psi_t(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{y}_t)$ is sketched in Figure 2. In the following we will use the shorthand $\psi_t(\mathbf{x}_{t-1,t}) \equiv \psi_t(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{y}_t)$. That is, we will assume that all evidence $\mathbf{y}_{1:T}$ is fixed and given and include the observations in the definition of the potentials. In principle, $\mathbf{x}_t$ can be any combination of discrete and continuous variables (as for example in a switching linear dynamical systems). However, for notational convenience we will stick to integral notation.

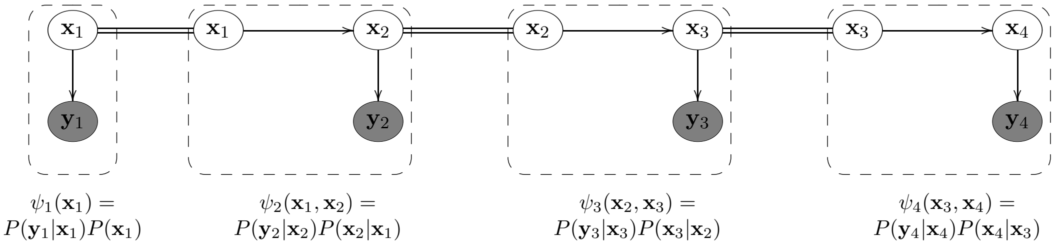
$\mathbf{x}_1$
$\mathbf{y}_1$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{y}_2$
$\mathbf{x}_2$
$\mathbf{x}_3$
$\mathbf{y}_3$
$\mathbf{x}_3$
$\mathbf{x}_4$
$\mathbf{y}_4$
$\psi_1(\mathbf{x}_1) = P(\mathbf{y}_1|\mathbf{x}_1)P(\mathbf{x}_1)$
$\psi_2(\mathbf{x}_1, \mathbf{x}_2) = P(\mathbf{y}_2|\mathbf{x}_2)P(\mathbf{x}_2|\mathbf{x}_1)$
$\psi_3(\mathbf{x}_2, \mathbf{x}_3) = P(\mathbf{y}_3|\mathbf{x}_3)P(\mathbf{x}_3|\mathbf{x}_2)$
$\psi_4(\mathbf{x}_3, \mathbf{x}_4) = P(\mathbf{y}_4|\mathbf{x}_4)P(\mathbf{x}_4|\mathbf{x}_3)$

Figure 2: Definition of potentials.

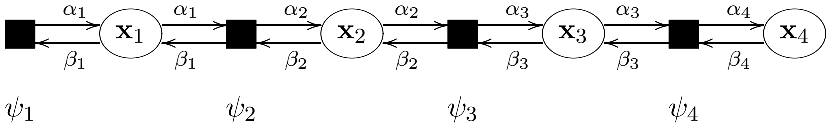

Figure 3: Message propagation.

2.2 Belief propagation

Our goal is to compute one-slice marginals or "beliefs" of the form $P(\mathbf{x}_t|\mathbf{y}_{1:T})$: the probability of the latent variables in a given time slice given all evidence. This marginal is implicitly required in many EM-type learning procedures (e.g., the Baum-Welch procedure for hidden Markov models [3, 36] or parameter estimation in linear dynamical systems [34, 7]), but can also be of interest by itself, especially when the latent variables have a direct interpretation. Often we also need the two-slice marginals $P(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{y}_{1:T})$, but as we will see, belief propagation gives these more or less for free.

A well-known procedure for computing beliefs in general Bayesian networks is Pearl's belief propagation [29]. Specific examples of belief propagation applied to Bayesian networks are the forward-backward algorithm for hidden Markov models [31, 36] and the Kalman filtering and smoothing equations for linear dynamical systems [32, 7]. Here we will follow a description of belief propagation as a specific case of the sum-product rule in factor graphs [17]. This description is symmetric with respect to the forward and backward messages, contrary to the perhaps more standard filtering and smoothing operations. We distinguish variable nodes $\mathbf{x}_t$ and local function nodes ψ_t in between variable nodes $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$. The message from ψ_t forward to $\mathbf{x}_t$ is called $\alpha_t(\mathbf{x}_t)$ and the message from ψ_t back to $\mathbf{x}_{t-1}$ is referred to as $\beta_{t-1}(\mathbf{x}_{t-1})$ (see Figure 3).

The belief at variable node $\mathbf{x}_t$ is the product of all messages sent from neighboring local function nodes:

$$P(\mathbf{x}_t|\mathbf{y}_{1:T}) \propto \alpha_t(\mathbf{x}_t)\beta_t(\mathbf{x}_t) .$$

Following the sum-product rule for factor graphs, the message sent from the variable node $\mathbf{x}_t$ to the function node ψ_t is the product of all messages that $\mathbf{x}_t$ receives, except the one from ψ_t itself. Or, to put it differently, the belief at $\mathbf{x}_t$ divided by the message $\beta_t(\mathbf{x}_t)$, which is (up to irrelevant normalization constants) $\alpha_t(\mathbf{x}_t)$. Note that this a peculiar property of chains: variable nodes simply pass the message they receive on to the next local function node; in a general tree-like structure these operations are slightly more complicated.

Information about the potentials is incorporated at the corresponding local function nodes. The recipe for computing the message from the local function node ψ_t to a neighboring variable node $\mathbf{x}_{t'}$, where t' can be either t (forward message) or $t-1$ (backward message), is as follows.[1]

[1] The standard description is slightly different. In the first step the potential is multiplied with all messages *excluding* the message from $\mathbf{x}_{t'}$ to ψ_t and there is no division afterwards. For exact belief propagation, marginalization over all variables except $\mathbf{x}_{t'}$ commutes with the multiplication by the message from $\mathbf{x}_{t'}$, which therefore

1. Multiply the potential corresponding to the local function node ψ_t with all messages from neighboring variable nodes to ψ_t, yielding

$$\hat{P}(\mathbf{x}_{t-1}, \mathbf{x}_t) \propto \alpha_{t-1}(\mathbf{x}_{t-1})\psi_t(\mathbf{x}_{t-1,t})\beta_t(\mathbf{x}_t) \,, \quad {\scriptstyle (t\,=\,1\,:\,T)} \tag{1}$$

our current estimate of the distribution at the local function node given the incoming messages $\alpha_t(\mathbf{x}_{t-1})$ and $\beta_t(\mathbf{x}_t)$.

2. Integrate out all variables except variable $\mathbf{x}_{t'}$ to obtain the current estimate of the one-slice marginal $\hat{P}(\mathbf{x}_{t'})$.

3. Conditionalize, i.e., divide by the message from $\mathbf{x}_{t'}$ to ψ_t.

Applying this recipe to the forward and backward messages we obtain, respectively,

$$\begin{aligned}
\alpha_t(\mathbf{x}_t) &\propto \frac{\displaystyle\int d\mathbf{x}_{t-1}\; \alpha_{t-1}(\mathbf{x}_{t-1})\psi_t(\mathbf{x}_{t-1}, \mathbf{x}_t)\beta_t(\mathbf{x}_t)}{\beta_t(\mathbf{x}_t)} \\
&= \int d\mathbf{x}_{t-1}\; \alpha_{t-1}(\mathbf{x}_{t-1})\psi_t(\mathbf{x}_{t-1}, \mathbf{x}_t) \quad {\scriptstyle (t\,=\,1\,:\,T)} \\
\beta_{t-1}(x_{t-1}) &\propto \frac{\displaystyle\int d\mathbf{x}_t\; \alpha_{t-1}(\mathbf{x}_{t-1})\psi_t(\mathbf{x}_{t-1}, \mathbf{x}_t)\beta_t(\mathbf{x}_t)}{\alpha_{t-1}(\mathbf{x}_{t-1})} \\
&= \int d\mathbf{x}_t\; \psi_t(\mathbf{x}_{t-1}, \mathbf{x}_t)\beta_t(\mathbf{x}_t) \,, \quad {\scriptstyle (t\,=\,2\,:\,T)}
\end{aligned} \tag{2}$$

with convention $\alpha_0(\mathbf{x}_0) \equiv \beta_T(\mathbf{x}_T) \equiv 1$. It is easy to see that the forward and backward messages do not interfere: they can be computed in parallel and a single forward and backward pass is sufficient to compute the exact beliefs. Furthermore note that the two-slice marginals follow directly from (1) after all $\alpha_{1:T}$ and $\beta_{1:T}$ have been calculated.

Despite the apparent simplicity of the above expressions, even in chains exact inference can become intractable or computationally too expensive. We can think of the following, often related, causes for trouble.

- The integrals at the function nodes are not analytically doable. This happens when, for example, the dynamics in the (continuous) latent variables is nonlinear. Popular approximate approaches are the extended Kalman filter [13] and recent improvements [15].

- The operations, i.e., the integrals or summations, at the function nodes become computationally too expensive because, for example, computing the integral involves operations that scale with a polynomial of the dimension of $\mathbf{x}_t$. This is the motivation behind variational approximations for factorial hidden Markov models [9], among others. Our own specific interest is in dynamic hierarchical models, the graphical structure of which is visualized in Figure 4. All nodes are Gaussian and all transitions linear, which makes this a specific case of a linear dynamical system. Exact inference is of order N^3, with N the number of nodes within each time slice. The goal is to find an approximate inference algorithm that is linear in N.

- A description of the belief states $P(\mathbf{x}_t)$ themselves is already exponentially large. For example, with N binary states at each time, we need on the order of 2^N parameters to specify the probability $P(\mathbf{x}_t)$. The Boyen-Koller algorithm [4] projects the belief onto an approximate factored belief state that requires only N variables. The factored frontier algorithm [27] is a further extension that executes the forward and backward operations in a computationally efficient manner.

cancels with the division afterwards [see Equation (2)]. This makes the standard procedure more efficient. The procedure outlined in the text is, as for as the results are concerned, equivalent and generalizes directly to the expectation propagation algorithm described in the next section.

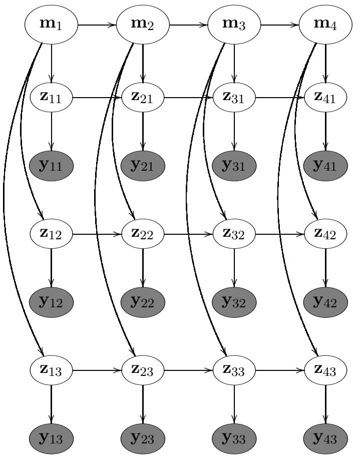

Figure 4: A dynamic hierarchical model.

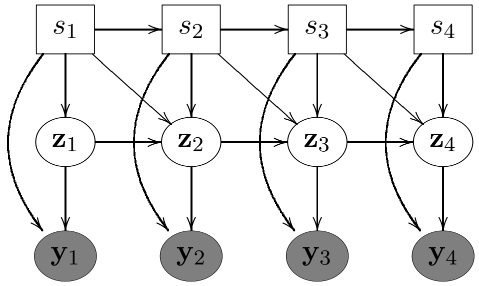

Figure 5: Switching linear dynamical system.

- Over time, the number of components needed to describe the beliefs becomes exponential. This is the case in switching linear dynamical systems [35, 26] (see Figure 5). Suppose that we have M different switch states. Then, at each forward step the number of mixture components multiplies by M and we need M^T mixture components to express the exact posterior. This is an example of the well-known fact that inference in hybrid Bayesian networks is much harder than in networks with just discrete or just Gaussian nodes [20].

Most of these problems can be tackled by the approach described below, which we will refer to as expectation propagation.

3 Expectation propagation as approximate belief propagation

3.1 Projection to the exponential family

Expectation propagation is a straightforward generalization of exact belief propagation. As explained above, problems with exact inference may arise in the marginalization step: exact calculation may not be doable and/or might not keep the beliefs within a family of distributions that is easy to describe and keep track of. The suggestion is therefore to work with approximate belief states, rather than with the exact ones. That is, we extend the marginalization step as follows.

2. Integrate out all variables except variable $\mathbf{x}_{t'}$ to obtain the current estimate of the belief state $\hat{P}(\mathbf{x}_{t'})$ *and project this belief state on to a chosen family of distribution, yielding the approximate belief* $q_{t'}(\mathbf{x}_{t'})$.

The remaining questions are then how to choose the family of distributions and how to project.

For the family of distributions we take a particular member of the exponential family, i.e.,

$$q_t(\mathbf{x}_t) \propto \mathrm{e}^{\boldsymbol{\gamma}_t^T \mathbf{f}(\mathbf{x}_t)} \,, \quad {\scriptstyle (t\,=\,1\,:\,T)} \tag{3}$$

with $\boldsymbol{\gamma}_t$ the canonical parameters and $\mathbf{f}(x_t)$ the sufficient statistics.[2] Typically, $\boldsymbol{\gamma}$ and $\mathbf{f}(\mathbf{x})$ are vectors with many components. For example, the sufficient statistics for an N-dimensional Gaussian are

$$\mathbf{f}(\mathbf{x}) = \{x_i \quad (i = 1:N), x_i x_j \quad (i,j = 1:N; j \leq i)\} \ .$$

Furthermore, note that a distribution over discrete variables can be written in exponential form, simply by defining $\mathbf{f}(\mathbf{x})$ as a long vector of Kronecker delta-functions, one for each possible state, and taking the corresponding component of $\boldsymbol{\gamma}$ equal to the logarithm of the probability of this state.

Our choice for the exponential family is motivated by the fact that multiplication and division of two exponential forms yields another exponential form. That is, if we initialize the forward and backward messages as

$$\alpha_t(\mathbf{x}_t) \propto \mathrm{e}^{\boldsymbol{\alpha}_t^T \mathbf{f}(\mathbf{x}_t)} \ \text{ and } \ \beta_t(\mathbf{x}_t) \propto \mathrm{e}^{\boldsymbol{\beta}_t^T \mathbf{f}(\mathbf{x}_t)} \,,$$

for example choosing $\boldsymbol{\alpha}_t = \boldsymbol{\beta}_t = \mathbf{0}$, they will stay of this form: $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ fully specify the messages and are all that we have to keep track of. As in exact belief propagation, the belief $q_t(\mathbf{x}_t)$ is a product of incoming messages:

$$q_t(\mathbf{x}_t) \propto \mathrm{e}^{(\boldsymbol{\alpha}_t + \boldsymbol{\beta}_t)^T \mathbf{f}(\mathbf{x}_t)} \,. \quad {\scriptstyle (t\,=\,1\,:\,T)} \tag{4}$$

Typically, there are two kinds of reasons for making a particular choice within the exponential family.

- The main problem is that the exact belief is not in the exponential family and therefore difficult to handle. The approximating distribution is of a particular exponential form, but usually further completely free. Examples are a Gaussian for the nonlinear Kalman filter or a conditional Gaussian for the switching Kalman filter of Figure 5.
- The exact belief is in the exponential family, but requires too many variables to fully specify it. The approximate belief is part of the same exponential family but with additional constraints for simplifications, e.g., factorized over (groups of) variables. This is the case for the Boyen-Koller algorithm [5] and would also be the way to go for the linear dynamical system of Figure 4.

Although the motivation is somewhat different, both can be treated within the same framework.

In the projection step, we replace the current estimate $\hat{P}(\mathbf{x})$ by the approximate $q(\mathbf{x})$ of the form (3) that is closest to $\hat{P}(\mathbf{x})$ in terms of the Kullback-Leibler divergence

$$\mathrm{KL}(\hat{P}|q) = \int d\mathbf{x}\, \hat{P}(\mathbf{x}) \log \left[\frac{\hat{P}(\mathbf{x})}{q(\mathbf{x})} \right] \,. \tag{5}$$

With $q(\mathbf{x})$ in the exponential family, it is easy to show that the solution follows from moment matching: we have to find the canonical parameters $\boldsymbol{\gamma}$ such that

$$\mathbf{g}(\boldsymbol{\gamma}) \equiv \langle \mathbf{f}(\mathbf{x}) \rangle_q \equiv \int d\mathbf{x}\, q(\mathbf{x}) \mathbf{f}(\mathbf{x}) = \int d\mathbf{x}\, \hat{P}(\mathbf{x}) \mathbf{f}(\mathbf{x}) \,.$$

[2] A perhaps more standard definition of a distribution in the exponential family reads

$$q(\mathbf{x}) = \exp \left[\sum_i \gamma_i f_i(\mathbf{x}) + D(\mathbf{x}) + S(\boldsymbol{\gamma}) \right] \,,$$

which can be turned into the form (3) by defining $f_0(\mathbf{x}) \equiv D(\mathbf{x})$ and $\gamma_0 \equiv 1$. The only thing to keep in mind then is that γ_0 is never to be treated as a parameter, but always kept fixed.

For members of the exponential family the so-called link function $\mathbf{g}(\boldsymbol{\gamma})$ is unique and invertible, i.e., there is a one-to-one mapping from canonical parameters to moments.

In fact, to compute new messages $\alpha_t(\mathbf{x}_t)$ and $\beta_{t-1}(\mathbf{x}_{t-1})$, we do not have to compute the approximate marginals $\hat{P}(\mathbf{x}_{t'})$. We only need the expectation of $\mathbf{f}(\mathbf{x}_{t'})$ over the two-slice marginal

$$\hat{p}_t(\mathbf{x}_{t-1,t}) \equiv \hat{P}(\mathbf{x}_{t-1}, \mathbf{x}_t) \propto \mathrm{e}^{\boldsymbol{\alpha}_{t-1}^T \mathbf{f}(\mathbf{x}_{t-1})} \psi_t(\mathbf{x}_{t-1,t}) \mathrm{e}^{\boldsymbol{\beta}_t^T \mathbf{f}(\mathbf{x}_t)} \,. \quad {\scriptstyle (t\,=\,1\,:\,T)} \tag{6}$$

In terms of the canonical parameters $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$, the forward and backward passes require the following operations.

Forward pass. Compute $\boldsymbol{\alpha}_t$ such that

$$\langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_t} = \langle \mathbf{f}(\mathbf{x}_t) \rangle_{q_t} = \mathbf{g}(\boldsymbol{\alpha}_t + \boldsymbol{\beta}_t) \,. \quad {\scriptstyle (t\,=\,1\,:\,T)}$$

Note that $\langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_t}$ only depends on the messages $\boldsymbol{\alpha}_{t-1}$ and $\boldsymbol{\beta}_t$. With $\boldsymbol{\beta}_t$ kept fixed, the solution $\boldsymbol{\alpha}_t = \tilde{\boldsymbol{\alpha}}_t(\boldsymbol{\alpha}_{t-1}, \boldsymbol{\beta}_t)$ can be computed by inverting $\mathbf{g}(\cdot)$, i.e., translating from a moment form to a canonical form.

Backward pass. Compute $\boldsymbol{\beta}_{t-1}$ such that

$$\langle \mathbf{f}(\mathbf{x}_{t-1}) \rangle_{\hat{p}_t} = \langle \mathbf{f}(\mathbf{x}_{t-1}) \rangle_{q_{t-1}} = \mathbf{g}(\boldsymbol{\alpha}_{t-1} + \boldsymbol{\beta}_{t-1}) \,. \quad {\scriptstyle (t\,=\,2\,:\,T)}$$

Similar to the forward pass, the solution can be written $\boldsymbol{\beta}_{t-1} = \tilde{\boldsymbol{\beta}}_{t-1}(\boldsymbol{\alpha}_{t-1}, \boldsymbol{\beta}_t)$. By definition we can set $\boldsymbol{\beta}_T \equiv \mathbf{0}$ (no information propagating back from beyond T).

The order in which the messages are updated is free to choose. However, iterating the standard forward-message passes seems to be most logical.

1. Start with all $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ equal to zero.
2. Go forward by updating $\boldsymbol{\alpha}_1$ to $\boldsymbol{\alpha}_T$ leaving all $\boldsymbol{\beta}_t$ intact.
3. Go backward by updating $\boldsymbol{\beta}_{T-1}$ to $\boldsymbol{\beta}_1$ leaving all $\boldsymbol{\alpha}_t$ intact.
4. Iterate the forward and backward passes 2. and 3. until convergence.

Note that we can set $\boldsymbol{\beta}_T = \mathbf{0}$ and can evaluate $\boldsymbol{\alpha}_T$ once after convergence: it does not affect the other $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$.

Without projection, i.e., if the exponential distribution is not an approximation but exact, we have a standard forward-backward algorithm, guaranteed to converge in a single forward and backward pass. In these cases, one can easily show $\tilde{\boldsymbol{\alpha}}_t(\boldsymbol{\alpha}_{t-1}, \boldsymbol{\beta}_t) = \tilde{\boldsymbol{\alpha}}_t(\boldsymbol{\alpha}_{t-1})$, independent of $\boldsymbol{\beta}_t$ and similarly $\tilde{\boldsymbol{\beta}}_{t-1}(\boldsymbol{\alpha}_{t-1}, \boldsymbol{\beta}_t) = \tilde{\boldsymbol{\beta}}_{t-1}(\boldsymbol{\beta}_t)$: the forward and backward messages do not interfere and there is no need to iterate. This is the case for standard Kalman smoothing [32] and for the forward-backward algorithm for hidden Markov models [31].

Expectation propagation generalizes several known algorithms. For example, GPB2 for generalized pseudo-Bayes [2, 16], arguably the most popular inference algorithm for a switching linear dynamical system, is nothing but a single forward pass of the above algorithm. The Boyen-Koller algorithm [5] corresponds to one forward and one backward pass for a dynamical Bayesian network with discrete nodes and independency assumptions over nodes within each time slice. In both cases, expectation propagation can be interpreted as an attempt to iteratively improve the existing estimates.

Some of the problems mentioned in the previous section may not have been solved. For example, we may still have to compute complicated nonlinear or highly dimensional integrals or summations. However, in any case we "only" have to compute moments of the two-slice distribution, not the one-slice marginals themselves. A more important concern is that the exact beliefs might not be accurately approximated with a distribution in the exponential family.

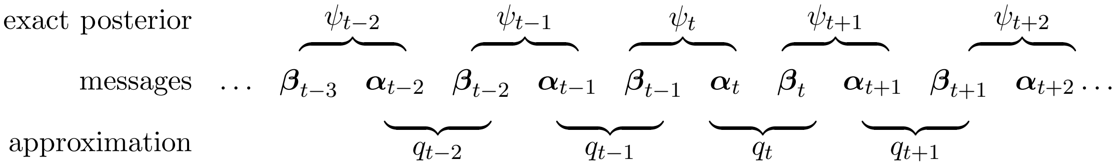
exact posterior
messages
approximation
ψ_{t-2}
ψ_{t-1}
ψ_t
ψ_{t+1}
ψ_{t+2}
$\ldots\ \boldsymbol{\beta}_{t-3}\ \boldsymbol{\alpha}_{t-2}\ \boldsymbol{\beta}_{t-2}\ \boldsymbol{\alpha}_{t-1}\ \boldsymbol{\beta}_{t-1}\ \boldsymbol{\alpha}_t\ \boldsymbol{\beta}_t\ \boldsymbol{\alpha}_{t+1}\ \boldsymbol{\beta}_{t+1}\ \boldsymbol{\alpha}_{t+2}\ldots$
q_{t-2}
q_{t-1}
q_t
q_{t+1}

Figure 6: Expectation propagation for dynamic Bayesian networks. The approximate distribution is a product of one-slice beliefs q_t. Each belief q_t follows from the product of the forward message $\boldsymbol{\alpha}_t$ and backward message $\boldsymbol{\beta}_t$. The product of the messages corresponding to $\boldsymbol{\beta}_{t-1}$ and $\boldsymbol{\alpha}_t$ represents the "effect" of potential ψ_t. That is, to recompute the effect of ψ_t, we take out the terms $\boldsymbol{\beta}_{t-1}$ and $\boldsymbol{\alpha}_t$ from the approximate distribution, substitute ψ_t instead, and project back to an uncoupled approximate distribution, yielding updates for q_{t-1} and q_t and thus for $\boldsymbol{\beta}_{t-1}$ and $\boldsymbol{\alpha}_t$.

3.2 Link with expectation propagation

Here we will illustrate that the algorithm described in the previous section is indeed a special case of expectation propagation as described in [25, 24].

We approximate the full posterior, which is a product of potentials or "terms" $\psi_t(\mathbf{x}_{t-1,t})$, with an uncoupled distribution of exponential form:

$$P(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) \approx \prod_{t=1}^{T} q_t(\mathbf{x}_t) \propto \prod_{t=1}^{T} \mathrm{e}^{(\boldsymbol{\alpha}_t+\boldsymbol{\beta}_t)^T\mathbf{f}(\mathbf{x}_t)} \,.$$

Here $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ are supposed to quantify how adding a potential affects the approximate belief $q_t(\mathbf{x}_t)$: $\boldsymbol{\alpha}_t$ stands for the effect of adding $\psi_t(\mathbf{x}_{t-1,t})$, $\boldsymbol{\beta}_t$ for the effect of adding $\psi_{t+1}(\mathbf{x}_{t,t+1})$.

Now suppose that we would like to recompute the effect of the potential $\psi_t(\mathbf{x}_{t-1,t})$. We remove the corresponding terms $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_{t-1}$ and replace them with $\psi_t(\mathbf{x}_{t-1,t})$ to arrive at the new approximation

$$\hat{P}(\mathbf{x}_{1:T}) \propto \prod_{t'<t-1} q_{t'}(\mathbf{x}_{t'}) \quad \left[\mathrm{e}^{\boldsymbol{\alpha}_{t-1}^T\mathbf{f}(\mathbf{x}_{t-1})}\psi_t(\mathbf{x}_{t-1,t})\mathrm{e}^{\boldsymbol{\beta}_t^T\mathbf{f}(\mathbf{x}_t)}\right] \quad \prod_{t'>t} q_{t'}(\mathbf{x}_{t'}) \,, \qquad {\scriptstyle (t\,=\,1\,:\,T)}$$

and thus at the approximate marginal over two time slices

$$\hat{P}(\mathbf{x}_{t-1},\mathbf{x}_t) \equiv \hat{p}_t(\mathbf{x}_{t-1,t}) = \frac{1}{c_t}\mathrm{e}^{\boldsymbol{\alpha}_{t-1}^T\mathbf{f}(\mathbf{x}_{t-1})}\psi_t(\mathbf{x}_{t-1,t})\mathrm{e}^{\boldsymbol{\beta}_t^T\mathbf{f}(\mathbf{x}_t)} \,, \qquad {\scriptstyle (t\,=\,1\,:\,T)}$$

with c_t a normalization constant and as above with convention $\hat{P}(\mathbf{x}_0,\mathbf{x}_1) \equiv \hat{P}(\mathbf{x}_1)$. These normalization constants can be used to estimate the data likelihood:

$$P(\mathbf{y}_{1:T}) \approx \prod_{t=1}^{T} c_t \,.$$

3.3 Examples

3.3.1 Switching linear dynamical system

Here we will illustrate the operations required for application of expectation propagation to switching linear dynamical systems. The potential corresponding to the switching linear dynamical system graphically visualized in Figure 5 can be written

$$\psi_t(s_{t-1}=i, s_t=j, \mathbf{z}_{t-1}, \mathbf{z}_t) = p_\psi(s_t=j|s_{t-1}=i)\Phi(\mathbf{z}_t; C_{ij}\mathbf{z}_{t-1}, R_{ij})\Phi(\mathbf{y}_t; A_j\mathbf{z}_t, Q_j) \,,$$

where $\Phi(\mathbf{z};\mathbf{m},V)$ stands for a Gaussian with mean $\mathbf{m}$ and covariance matrix V. The messages are taken to be conditional Gaussian potential of the form

$$\begin{aligned}\alpha_{t-1}(s_{t-1}=i,\mathbf{z}_{t-1}) &\propto p_\alpha(s_{t-1}=i)\Psi(\mathbf{z}_{t-1};\mathbf{m}^\alpha_{i,t-1},V^\alpha_{i,t-1})\\ \beta_t(s_t=j,\mathbf{z}_t) &\propto p_\beta(s_t=j)\Psi(\mathbf{z}_t;\mathbf{m}^\beta_{j,t},V^\beta_{j,t})\ ,\end{aligned}$$

where the potential $\Psi(\mathbf{z};\mathbf{m},V)$ is of the same form as $\Phi(\mathbf{z};\mathbf{m},V)$, but without the normalization and in fact need not be normalizable, i.e., can have a negative covariance. Note that a message $\alpha(s_{t-1},\mathbf{z}_{t-1})$ is in fact a combination of M Gaussian potentials, one for each switch state i. It can be written in exponential form with sufficient statistics

$$\mathbf{f}(s,\mathbf{z})=\{\delta_{s,i}\quad(i=1:M),\delta_{s,i}z_k\quad(i=1:M;k=1:N),\delta_{s,i}z_kz_l\quad(i=1:M;k,l=1:N;l\le k)\}\ .$$

The two-slice marginal $\hat{P}(s_{t-1},s_t,\mathbf{z}_{t-1},\mathbf{z}_t)$, with elements

$$\hat{P}_t(s_{t-1}=i,s_t=j,\mathbf{z}_{t-1},\mathbf{z}_t)\propto\alpha_{t-1}(s_{t-1}=i,\mathbf{z}_{t-1})\Phi_t(s_{t-1}=i,s_t=j,\mathbf{z}_{t-1},\mathbf{z}_t)\beta_{t-1}(s_t=j,\mathbf{z}_t)\ ,$$

consists of M^2 Gaussians: one for each combination $\{i,j\}$. With some bookkeeping, which involves the translation from canonical parameters to moments, we can compute the moments of these M^2 Gaussians, i.e., we can rewrite

$$\hat{P}(s_{t-1}=i,s_t=j,\mathbf{z}_{t-1},\mathbf{z}_t)\propto\hat{p}_{ij}\Phi(z_{t-1},z_t;\hat{\mathbf{m}}_{ij},\hat{V}_{ij})\ ,\tag{7}$$

where $\hat{\mathbf{m}}_{ij}$ is a $2N$-dimensional vector and $\hat{V}_{ij}$ a $2N\times 2N$ covariance matrix.

To obtain the forward pass in expectation propagation, we have to integrate out $\mathbf{z}_{t-1}$ and sum over s_{t-1}. Integrating out $\mathbf{z}_{t-1}$ is trivial:

$$\hat{P}(s_{t-1}=i,s_t=j,\mathbf{z}_t)\propto\hat{p}_{ij}\Phi(z_t;\hat{\mathbf{m}}_{ij},\hat{V}_{ij})\ ,$$

where now $\hat{\mathbf{m}}_{ij}$ and $\hat{V}_{ij}$ are supposed to be restricted to the components corresponding to $\mathbf{z}_t$, i.e., the components $N+1$ to $2N$ in the means and covariances of (7). Summation over s_{t-1} yields a mixture of M Gaussians for each switch state j, which is *not* a member of the exponential family. The conditional Gaussian of the form

$$q_t(s_t=j,\mathbf{z}_t)=\hat{p}_j\Phi(\mathbf{z}_t|\hat{\mathbf{m}}_j,\hat{V}_j)$$

closest in KL-divergence to this mixture of Gaussians follows from moment matching:

$$\begin{aligned}\hat{p}_j=\sum_i\hat{p}_{ij}\ ,\quad\hat{\mathbf{m}}_j=\sum_i\hat{p}_{ij}\hat{\mathbf{m}}_{ij}\ ,\\ \text{and}\ \ \hat{V}_j=\sum_i\hat{p}_{ij}\hat{V}_{ij}+\sum_i\hat{p}_{ij}(\hat{\mathbf{m}}_{ij}-\hat{\mathbf{m}}_i)(\hat{\mathbf{m}}_{ij}-\hat{\mathbf{m}}_i)^T\ .\end{aligned}$$

To find the new forward message $\alpha_t(s_t,\mathbf{z}_t)$ we have to divide the approximate belief $q_t(s_t,\mathbf{z}_t)$ by the backward message $\beta_t(s_t,\mathbf{z}_t)$. This is most easily done by translating $q_t(s_t,\mathbf{z}_t)$ from the moment form above to a canonical form and subtracting the canonical parameters corresponding to $\beta_t(s_t,\mathbf{z}_t)$ to yield the new $\alpha_t(s_t,\mathbf{z}_t)$ in canonical form.

The procedure for the backward pass follows in exactly the same manner by integrating out $\mathbf{z}_t$ and summing over s_t. All together, it is just a matter of bookkeeping, with frequent transformations from canonical to moment form and vice versa. Efficient implementations can be made if, for example, the covariance matrices Q_i and R_i are restricted to be diagonal. The forward filtering pass is equivalent to a method called GPB2 [2, 16]. An attempt has been made to obtain a similar smoothing procedure, but this required quite some additional approximations [26]. In the above description however, forward and backward passes are completely symmetric and smoothing does not require any additional approximations beyond the ones already made for filtering. Furthermore, the forward and backward passes can be iterated until convergence in the hope to find a more consistent and better approximation.

3.3.2 Dynamic hierarchical model

As a second example, we consider the dynamic hierarchical model of Figure 4. For ease of notation, we assume all nodes to correspond to one-dimensional variables $z_{i,t}$ and use $z_{0,t}$ to denote the highest-level node (referred to as m_t in the figure). $\mathbf{z}_t$ refers to the state of all hidden variables in time slice t. In this notation, the potentials are of the form

$$\psi_t(\mathbf{z}_{t-1,t}) = \prod_{i=1}^{N} P(y_{i,t}|z_{i,t}) P(z_{i,t}|z_{i,t-1}, z_{0,t}) P(z_{0,t}|z_{0,t-1}) \, .$$

We take the messages to be independent Gaussians, i.e.,

$$\alpha_t(\mathbf{z}_t) = \prod_{i=0}^{N} \alpha_{i,t}(z_{i,t}) \, ,$$

with $\alpha_{i,t}(x_{i,t})$ one-dimensional Gaussians, and similarly for $\beta_t(\mathbf{z}_t)$. All messages being independent, it is easy to see that the approximate two-slice marginal obeys

$$\hat{P}(\mathbf{z}_{t-1,t}) = \hat{P}(z_{0,t-1}, z_{0,t}) \prod_{i=1}^{N} \hat{P}(z_{i,t-1}, z_{i,t}|z_{0,t}) \, ,$$

i.e., the distributions over the lower-level nodes are independent of each other given the higher level node. Straightforwardly collecting terms, we have

$$\hat{P}(z_{i,t-1}, z_{i,t}|z_{0,t}) = \frac{1}{c_{i,t}(z_{0,t})} \alpha_{i,t-1}(z_{i,t-1}) P(y_{i,t}|z_{i,t}) P(z_{i,t}|z_{i,t-1}, z_{0,t}) \beta_{i,t}(z_{i,t}) \, , \tag{8}$$

with $c_{i,t}(z_{0,t})$ the appropriate normalization constant and thus

$$\hat{P}(z_{0,t-1}, z_{0,t}) \propto \alpha_{0,t-1}(z_{0,t-1}) P(z_{0,t}|z_{0,t-1}) \beta_{0,t}(z_{0,t}) \prod_{i} c_{i,t}(z_{0,t}) \, . \tag{9}$$

Note that these normalizations can be written in the form of a Gaussian potential and can be computed independently of each other.

In the forward pass, we have to integrate out $z_{0,t-1}$ in (9) yielding $\hat{P}(z_{0,t})$. Similarly, we can integrate over $z_{i,t-1}$ in (8) to obtain $\hat{P}(z_{i,t}|z_{0,t})$ and from that the approximate belief

$$\hat{P}(z_{i,t}) = \int dz_{0,t} \, \hat{P}(z_{i,t}|z_{0,t}) \hat{P}(z_{0,t}) \, .$$

Again, the backward pass proceeds in a completely symmetric manner by integrating out $z_{0,t}$ and $z_{i,t}$.

The above scheme can be easily generalized to more than two levels. An update starts at the lowest level, conditioned on the parameters of the next level. Normalizing terms for the lower level then appear in the update for the next level. So we can go up to the highest level, always taken into account the normalization terms of the next lower level and conditioned on the parameters of the next higher level. The highest level is unconditioned and thus directly yields the required marginal distribution. The projection to a factorized form then goes in the opposite direction, subsequently integrating out over the distribution at the next higher level. The number of computations required in each forward and backward is proportional to the number of nodes ($N + 1$ in the above two-level case), to be contrasted with the N^3 complexity for exact inference. Obviously, exactly the same procedure can be used if all nodes are discrete rather than Gaussian.

4 A free energy function

4.1 The free energy

In order to derive a convergent algorithm, we first have to define what we would like to optimize. The primal objective that we will derive is inspired by the recently proposed Bethe free energy functional for loopy belief propagation in [38] and follows the reasoning in [23].

The exact posterior $P(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ is the solution of

$$\begin{aligned} P(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) &= \underset{\hat{P}(\mathbf{x}_{1:T})}{\operatorname{argmin}} \mathrm{KL}(\hat{P}(\mathbf{x}_{1:T})|P(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})) \\ &= \underset{\hat{P}(\mathbf{x}_{1:T})}{\operatorname{argmin}} \int d\mathbf{x}_{1:T}\, \hat{P}(\mathbf{x}_{1:T}) \log \left[\frac{\hat{P}(\mathbf{x}_{1:T}) P(\mathbf{y}_{1:T})}{P(\mathbf{x}_{1:T}, \mathbf{y}_{1:T})} \right] \\ &= \underset{\hat{P}(\mathbf{x}_{1:T})}{\operatorname{argmin}} \left\{ -\sum_{t=1}^{T} \int d\mathbf{x}_{1:T}\, \hat{P}(\mathbf{x}_{1:T}) \log \psi_t(\mathbf{x}_{t-1,t}) + \int d\mathbf{x}_{1:T}\, \hat{P}(\mathbf{x}_{1:T}) \log \hat{P}(\mathbf{x}_{1:T}) \right\} , \end{aligned} \tag{10}$$

under the constraint that $\hat{P}(\mathbf{x}_{1:T})$ is a probability distribution, i.e., is nonnegative and marginalizes to 1. Here and in the following this constraint is always implicitly assumed when we consider probability distributions and marginals. We will use notation $\min'$ to indicate that there are other constraints as well (which ones should be clear from the text).

The above expression is nothing but a definition of the conditional $P(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$. In the following we will simplify and approximate it. The first observation is that the exact solution can be written as a product of two-slice marginals divided by a product of one-slice marginals:

$$\hat{P}(\mathbf{x}_{1:T}) = \frac{\prod_{t=1}^{T} \hat{P}(\mathbf{x}_{t-1,t})}{\prod_{t=1}^{T-1} \hat{P}(\mathbf{x}_t)} , \tag{11}$$

again with convention $\hat{P}(\mathbf{x}_0, \mathbf{x}_1) \equiv \hat{P}(\mathbf{x}_1)$. Plugging this into (10), our objective is to minimize

$$\min_{\hat{P}(\mathbf{x}_{1:T})} \left\{ \sum_{t=1}^{T} \int d\mathbf{x}_{t-1,t}\, \hat{P}(\mathbf{x}_{t-1,t}) \log \left[\frac{\hat{P}(\mathbf{x}_{t-1,t})}{\psi_t(\mathbf{x}_{t-1,t})} \right] - \sum_{t=1}^{T-1} \int d\mathbf{x}_t\, \hat{P}(\mathbf{x}_t) \log \hat{P}(\mathbf{x}_t) \right\} ,$$

for a distribution $\hat{P}(\mathbf{x}_{1:T})$ of the form (11). This can be rewritten as a minimization over two-slice marginals under consistency constraints

$$\int d\mathbf{x}_{t-1}\, \hat{P}(\mathbf{x}_{t-1,t}) = \hat{P}(\mathbf{x}_t) \quad {\scriptstyle (t\,=\,1\,:\,T)} \quad \text{and} \quad \int d\mathbf{x}_t\, \hat{P}(\mathbf{x}_{t-1,t}) = \hat{P}(\mathbf{x}_{t-1}) \quad {\scriptstyle (t\,=\,2\,:\,T)} . \tag{12}$$

We will refer to these as the forward and backward constraints, respectively. Note that there is no optimization with respect to beliefs $P(\mathbf{x}_t)$: these are fully determined given the two-slice marginals. For chains, and similarly for trees, this is all still exact. In networks containing loops, a factorization of the form (11) serves as an approximation (see e.g. [38]). The connection with variational approaches [14] is discussed in Section 8.

4.2 An approximate free energy

Now, in order to arrive at a free energy functional for expectation propagation we make two assumptions or approximations. First, we assume that the beliefs are of the exponential form $P(\mathbf{x}_t) \approx q_t(\mathbf{x}_t)$ in (3). Next we replace the marginalization constraints (12) by weaker expectation constraints on $\hat{p}_t(\mathbf{x}_{t-1,t}) \approx \hat{P}(\mathbf{x}_{t-1,t})$:

$$\langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_t} = \langle \mathbf{f}(\mathbf{x}_t) \rangle_{q_t} \quad {\scriptstyle (t\,=\,1\,:\,T)} \quad \text{and} \quad \langle \mathbf{f}(\mathbf{x}_{t-1}) \rangle_{\hat{p}_t} = \langle \mathbf{f}(\mathbf{x}_{t-1}) \rangle_{q_{t-1}} \quad {\scriptstyle (t\,=\,2\,:\,T)} . \tag{13}$$

Our objective remains the same, i.e.,

$$\min_{\hat{p}}{}' \left\{ \sum_{t=1}^{T} \int d\mathbf{x}_{t-1,t}\, \hat{p}_t(\mathbf{x}_{t-1,t}) \log \left[\frac{\hat{p}_t(\mathbf{x}_{t-1,t})}{\psi_t(\mathbf{x}_{t-1,t})} \right] - \sum_{t=1}^{T-1} \int d\mathbf{x}_t\, q_t(\mathbf{x}_t) \log q_t(\mathbf{x}_t) \right\} , \tag{14}$$

but now under the constraints (13) and with $q_t(\mathbf{x}_t)$ of the exponential form (3). Note that, as before, these constraints make the one-slice marginals $q_t(\mathbf{x}_t)$ an implicit function of the two-slice marginals $\hat{p}_t(\mathbf{x}_{t-1,t})$: there is no minimization or maximization with respect to $q_t(\mathbf{x}_t)$. Furthermore, $q_T(\mathbf{x}_T)$ does not appear in the objective: it can be computed afterwards from the forward constraint (13) for $t = T$.

Adding Lagrange multipliers $\boldsymbol{\beta}_t$ and $\boldsymbol{\alpha}_t$ for the forward and backward constraints and taking derivatives we find that at a fixed point of (14) $q_t(\mathbf{x}_t)$ and $\hat{p}_t(\mathbf{x}_{t-1,t})$ are of the form (4) and (6). The other way around, at a fixed point of expectation propagation, the expectation constraints are automatically satisfied. Combination of these two observations proves that the fixed points of expectation propagation indeed correspond to fixed points of the "free energy" (14).

4.3 Bounding the free energy

Finding the minimum of the "primal" energy function (14) is a constrained optimization problem over functions. Due to the negative $q \log q$ term, this objective is not necessarily convex in $\hat{p}_{1:T}$. To get rid of the concave $q \log q$ term, we make use of the Legendre transformation

$$-\int d\mathbf{x}_t\, q_t(\mathbf{x}_t) \log q_t(\mathbf{x}_t) = \min_{\boldsymbol{\gamma}_t} \left\{ -\boldsymbol{\gamma}_t^T \langle \mathbf{f}(\mathbf{x}_t) \rangle_{q_t} + \log \int d\mathbf{x}_t\, \mathrm{e}^{\boldsymbol{\gamma}_t^T \mathbf{f}(\mathbf{x}_t)} \right\} . \tag{15}$$

Substitution into (14) yields[3]

$$\min_{\hat{p}}{}' \min_{\boldsymbol{\gamma}} \left\{ \sum_{t=1}^{T} \int d\mathbf{x}_{t-1,t}\, \hat{p}_t(\mathbf{x}_{t-1,t}) \log \left[\frac{\hat{p}_t(\mathbf{x}_{t-1,t})}{\psi_t(\mathbf{x}_{t-1,t})} \right] + \sum_{t=1}^{T-1} \left[-\boldsymbol{\gamma}_t^T \langle \mathbf{f}(\mathbf{x}_t) \rangle_{q_t} + \log \int d\mathbf{x}_t\, \mathrm{e}^{\boldsymbol{\gamma}_t^T \mathbf{f}(\mathbf{x}_t)} \right] \right\} . \tag{16}$$

Now that we have eliminated the concave term, the remaining functional in (16) is convex in $\hat{p}$. The price that we had to pay is an extra minimization with respect to $\boldsymbol{\gamma}$.

An alternative and in fact equivalent interpretation of the Legendre transformation is the linear bound

$$\int d\mathbf{x}_t\, q_t(\mathbf{x}_t) \log q_t(\mathbf{x}_t) \leq \int d\mathbf{x}_t\, q_t(\mathbf{x}_t) \log q_t^{\text{old}}(\mathbf{x}_t) ,$$

where $q_t^{\text{old}}(\mathbf{x}_t)$ can be any distribution and is typically the solution given the old parameter settings (here the two-slice marginals $\hat{p}_t(\mathbf{x}_{t-1,t})$ and $\hat{p}_{t+1}(\mathbf{x}_{t,t+1})$). In the following we will stick to the explicit minimization over $\boldsymbol{\gamma}$ to make the connection with expectation propagation below.

Both formulations suggest a double-loop algorithm: in the inner loop we keep $\boldsymbol{\gamma}$ fixed and minimize with respect to $\hat{p}$ under the appropriate constraints; in the outer loop we keep $\hat{p}$ fixed and minimize with respect to $\boldsymbol{\gamma}$.

5 A convergent algorithm

5.1 The inner loop

Let us first focus on the inner loop for fixed $\boldsymbol{\gamma}$. We are left with a convex minimization problem with linear constraints, which can be turned into a dual unconstrained concave maximization problem in terms of Lagrange multipliers (see e.g. [21]).

[3] For ease of notation, $\boldsymbol{\gamma}$ and $\hat{p}$ without subscript refer to all relevant $\boldsymbol{\gamma}_t$ and $\hat{p}_t$ (here $\boldsymbol{\gamma}_{1:T-1}$, often $\boldsymbol{\gamma}_{1:T}$).

5.1.1 The dual objective

To get rid of any dependency on $q_t(\mathbf{x}_t)$, we substitute the constraint[4]

$$\langle \mathbf{f}(\mathbf{x}_t) \rangle_{q_t} = \frac{1}{2} \left[\langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_t} + \langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_{t+1}} \right] \quad {\scriptstyle (t\,=\,1\,:\,T\,-\,1)}$$

in (16) and neglect terms independent of $\hat{p}_t$ to arrive at the objective

$$\min_{\hat{p}}{}' \left\{ \sum_{t=1}^{T} \int d\mathbf{x}_{t-1,t}\, \hat{p}_t(\mathbf{x}_{t-1,t}) \log \left[\frac{\hat{p}_t(\mathbf{x}_{t-1,t})}{\psi_t(\mathbf{x}_{t-1,t})} \right] - \frac{1}{2} \sum_{t=1}^{T-1} \boldsymbol{\gamma}_t^T \left[\langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_t} + \langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_{t+1}} \right] \right\} . \tag{17}$$

The only remaining constraints are now "forward equals backward", i.e.,

$$\langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_t} = \langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_{t+1}} . \quad {\scriptstyle (t\,=\,1\,:\,T\,-\,1)}$$

Introducing Lagrange multipliers $\boldsymbol{\delta}_t/2$ and collecting all terms in the Lagrangian that depend on $\hat{p}_t(\mathbf{x}_{t-1,t})$, we obtain

$$\int d\mathbf{x}_{t-1,t}\, \hat{p}_t(\mathbf{x}_{t-1,t}) \left\{ \log \left[\frac{\hat{p}_t(\mathbf{x}_{t-1,t})}{\psi_t(\mathbf{x}_{t-1,t})} \right] - \frac{1}{2}(\boldsymbol{\gamma}_t - \boldsymbol{\delta}_t)^T \mathbf{f}(\mathbf{x}_t) - \frac{1}{2}(\boldsymbol{\gamma}_{t-1} + \boldsymbol{\delta}_{t-1})^T \mathbf{f}(\mathbf{x}_{t-1}) \right\} , \quad {\scriptstyle (t\,=\,1\,:\,T)} \tag{18}$$

where we have the convention that $\boldsymbol{\gamma}_0 \equiv \boldsymbol{\delta}_0 \equiv \boldsymbol{\gamma}_T \equiv \boldsymbol{\delta}_T \equiv \mathbf{0}$. Taking the functional derivative with respect to $\hat{p}_t(\mathbf{x}_{t-1,t})$ we regain the form (6) if we substitute

$$\boldsymbol{\alpha}_{t-1} = \frac{1}{2}(\boldsymbol{\gamma}_{t-1} + \boldsymbol{\delta}_{t-1}) \ \text{ and } \ \boldsymbol{\beta}_t = \frac{1}{2}(\boldsymbol{\gamma}_t - \boldsymbol{\delta}_t) . \quad {\scriptstyle (t\,=\,1\,:\,T)} \tag{19}$$

Substitution into the Lagrangian (18) and back into (17) yields the dual objective

$$\max_{\boldsymbol{\delta}} F_1(\boldsymbol{\delta}) \ \text{ where } \ F_1(\boldsymbol{\delta}) = -\sum_{t=1}^{T} \log \int d\mathbf{x}_{t-1,t}\, \mathrm{e}^{\frac{1}{2}(\boldsymbol{\gamma}_{t-1}+\boldsymbol{\delta}_{t-1})^T \mathbf{f}(\mathbf{x}_{t-1})} \psi_t(\mathbf{x}_{t-1,t}) \mathrm{e}^{\frac{1}{2}(\boldsymbol{\gamma}_t - \boldsymbol{\delta}_t)^T \mathbf{f}(\mathbf{x}_t)} . \tag{20}$$

Again recall that $q_T(\mathbf{x}_T)$ does not appear in the primal objective, and there is no optimization with respect to $\boldsymbol{\delta}_T$ in the dual objective. We can always compute the approximate belief $q_T(\mathbf{x}_T)$ afterwards from $\hat{p}_T(\mathbf{x}_{T-1,T})$.

5.1.2 Unconstrained maximization

With the primal (17) being convex in $\hat{p}$, the dual $F_1(\boldsymbol{\delta})$ is concave in $\boldsymbol{\delta}$, and thus has a unique solution. In principle, any optimization algorithm will do, but here we will propose a specific one, which can be interpreted as a damped version of fixed-point iteration.

In terms of the standard forward and backward updates $\tilde{\boldsymbol{\alpha}}_t \equiv \tilde{\boldsymbol{\alpha}}_t(\boldsymbol{\alpha}_{t-1}, \boldsymbol{\beta}_t)$ and $\tilde{\boldsymbol{\beta}}_t \equiv \tilde{\boldsymbol{\beta}}_t(\boldsymbol{\alpha}_t, \boldsymbol{\beta}_{t+1})$, with $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ related to $\boldsymbol{\delta}_t$ and $\boldsymbol{\gamma}_t$ as in (19), the gradient with respect to $\boldsymbol{\delta}_t$ reads

$$\frac{\partial F(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}_t} = \frac{1}{2} \left[\mathbf{g}(\tilde{\boldsymbol{\alpha}}_t + \boldsymbol{\beta}_t) - \mathbf{g}(\boldsymbol{\alpha}_t + \tilde{\boldsymbol{\beta}}_t) \right] . \tag{21}$$

Setting the gradient to zero suggests the update

$$\boldsymbol{\delta}_t^{\text{new}} = \tilde{\boldsymbol{\delta}}_t \equiv \tilde{\boldsymbol{\alpha}}_t - \tilde{\boldsymbol{\beta}}_t . \tag{22}$$

Without any projections, i.e., with $\tilde{\boldsymbol{\alpha}}_t = \tilde{\boldsymbol{\alpha}}_t(\alpha_{t-1})$ and $\tilde{\boldsymbol{\beta}}_t = \tilde{\boldsymbol{\beta}}_t(\boldsymbol{\beta}_{t+1})$, this fixed point iteration can be interpreted as finding the maximum of $F(\boldsymbol{\delta})$ in the direction of $\boldsymbol{\delta}_t$, keeping the other

[4] Any other convex linear combination of forward and backward expectations will do as well, but this symmetric choice appears to be most natural.

$\boldsymbol{\delta}_{t'}$ fixed. Each update (22) would guarantee an increase of $F(\boldsymbol{\delta})$, unless the corresponding gradient (21) vanishes.

However, in expectation propagation both $\tilde{\boldsymbol{\alpha}}_t$ and $\tilde{\boldsymbol{\beta}}_t$ do depend on $\boldsymbol{\delta}_t$ (through $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$, respectively) and the full update (22) does not necessarily go uphill. A simple suggestion is to consider a damped version of the form

$$\boldsymbol{\delta}_t^{\text{new}} = \boldsymbol{\delta}_t + \epsilon_\delta(\tilde{\boldsymbol{\delta}}_t - \boldsymbol{\delta}_t) \,. \tag{23}$$

It is easy to show[5] that this update is "aligned" with the gradient (21) and thus, with sufficiently small ϵ_δ, will always lead to an increase in $F_1(\boldsymbol{\delta})$. Expecting the dependency of $\tilde{\boldsymbol{\delta}}_t$ on $\boldsymbol{\delta}_t$ to be rather small, the hope is that we can take a rather large ϵ_δ.

In terms of the moments, the update (23) reads

$$\boldsymbol{\delta}_t^{\text{new}} = \boldsymbol{\delta}_t + \epsilon_\delta \left[\mathbf{g}^{-1}\left(\langle \mathbf{f}(\mathbf{x}_t)\rangle_{\hat{p}_t}\right) - \mathbf{g}^{-1}\left(\langle \mathbf{f}(\mathbf{x}_t)\rangle_{\hat{p}_{t+1}}\right)\right] \,, \tag{24}$$

i.e., to update a single $\boldsymbol{\delta}_t$ we have to compute one forward and one backward moment.

5.2 The outer loop

As could have been expected, especially in the interpretation of the Legendre transformation as a bounding technique, the outer loop is really straightforward. Setting the gradient of (16) with respect to $\gamma_t(\mathbf{x}_t)$ to zero yields the update

$$\mathbf{g}(\boldsymbol{\gamma}_t^{\text{new}}) = \langle \mathbf{f}(\mathbf{x}_t)\rangle_{q_t} = \frac{1}{2}\left[\langle \mathbf{f}(\mathbf{x}_t)\rangle_{\hat{p}_t} + \langle \mathbf{f}(\mathbf{x}_t)\rangle_{\hat{p}_{t+1}}\right] \,, \quad {\scriptstyle (t\,=\,1\,:\,T\,-\,1)} \tag{25}$$

with $\hat{p}$ the solution of the inner loop. In terms of the messages $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ and their standard expectation propagation updates $\tilde{\boldsymbol{\alpha}}_t$ and $\tilde{\boldsymbol{\beta}}_t$ this is equivalent to

$$\boldsymbol{\gamma}_t^{\text{new}} = \mathbf{g}^{-1}\left(\frac{1}{2}\left[\mathbf{g}(\boldsymbol{\alpha}_t + \tilde{\boldsymbol{\beta}}_t) + \mathbf{g}(\tilde{\boldsymbol{\alpha}}_t + \boldsymbol{\beta}_t)\right]\right) \,. \tag{26}$$

With $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ the result of the maximization in the inner loop, we have $\boldsymbol{\delta}_t = \tilde{\boldsymbol{\delta}}_t$ and thus $\boldsymbol{\alpha}_t + \tilde{\boldsymbol{\beta}}_t = \tilde{\boldsymbol{\alpha}}_t + \boldsymbol{\beta}_t$. This can be used to simplify the update (26) to

$$\boldsymbol{\gamma}_t^{\text{new}} = \boldsymbol{\gamma}_t + \frac{1}{2}\left[\tilde{\boldsymbol{\alpha}}_t + \tilde{\boldsymbol{\beta}}_t - \boldsymbol{\gamma}_t\right] \,. \tag{27}$$

5.3 Link with Yuille's CCCP

In [39], Yuille proposes a double-loop algorithm for (loopy) belief propatation and Kikuchi approximations. We will show the link with the above double-loop algorithm by describing what Yuille's CCCP amounts to when applied to approximate inference in dynamic Bayesian networks.

We rewrite the primal objective (14) as

$$\begin{aligned}\min_{\hat{p}}{}' \quad & \left\{\sum_{t=1}^{T}\int d\mathbf{x}_{t-1,t}\, \hat{p}_t(\mathbf{x}_{t-1,t}) \log\left[\frac{\hat{p}_t(\mathbf{x}_{t-1,t})}{\psi_t(\mathbf{x}_{t-1,t})}\right] + \kappa\sum_{t=1}^{T-1}\int d\mathbf{x}_t\, q_t(\mathbf{x}_t)\log q_t(\mathbf{x}_t)\right. \\ & \left. -(1+\kappa)\sum_{t=1}^{T-1}\int d\mathbf{x}_t\, q_t(\mathbf{x}_t)\log q_t(\mathbf{x}_t)\right\} \,,\end{aligned}$$

[5] In terms of $\boldsymbol{\gamma}_1 \equiv \tilde{\boldsymbol{\alpha}}_t + \boldsymbol{\beta}_t$ and $\boldsymbol{\gamma}_2 \equiv \boldsymbol{\alpha}_t + \tilde{\boldsymbol{\beta}}_t$, the update (23) is proportional to $\tilde{\boldsymbol{\delta}}_t - \boldsymbol{\delta}_t = \boldsymbol{\gamma}_2 - \boldsymbol{\gamma}_1$ and we can write

$$2(\tilde{\boldsymbol{\delta}}_t - \boldsymbol{\delta}_t)^T \frac{\partial F(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}_t} = (\boldsymbol{\gamma}_2 - \boldsymbol{\gamma}_1)^T\left[\mathbf{g}(\boldsymbol{\gamma}_2) - \mathbf{g}(\boldsymbol{\gamma}_1)\right] = \mathrm{KL}(q_1|q_2) + \mathrm{KL}(q_2|q_1) \geq 0 \,,$$

with q_1 and q_2 the exponential distributions with canonical parameters $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$ and link function $\mathbf{g}(\cdot)$.

where κ can be any positive number (Yuille takes $\kappa = 1$). Now we apply the Legendre transformation (15) only to the second (negative) term to arrive at [compare with (16)]

$$\min_{\boldsymbol{\gamma}} \min_{\hat{p},q}{}' \quad \left\{ \sum_{t=1}^{T} \int d\mathbf{x}_{t-1,t}\, \hat{p}_t(\mathbf{x}_{t-1,t}) \log \left[\frac{\hat{p}_t(\mathbf{x}_{t-1,t})}{\psi_t(\mathbf{x}_{t-1,t})} \right] + \kappa \sum_{t=1}^{T-1} \int d\mathbf{x}_t\, q_t(\mathbf{x}_t) \log q_t(\mathbf{x}_t) - \right.$$
$$\left. + (1+\kappa) \sum_{t=1}^{T-1} \left[-\boldsymbol{\gamma}_t^T \left\langle \mathbf{f}(\mathbf{x}_t) \right\rangle_{q_t} + \log \int d\mathbf{x}_t\, \mathrm{e}^{\boldsymbol{\gamma}_t^T \mathbf{f}(\mathbf{x}_t)} \right] \right\} . \tag{28}$$

For convenience, we treat the one-slice marginals q as free parameters. Yuille's CCCP algorithm is now nothing but an iterative minimization procedure on the objective (28).

In the inner loop, we keep $\boldsymbol{\gamma}$ fixed and try and find the minimum with respect to $\hat{p}$ and q under the constraints (13). Since the objective is convex in $\hat{p}$ and q for fixed $\boldsymbol{\gamma}$, we can turn this into an unconstrained maximization problem over Lagrange multipliers. Introducing Lagrange multipliers $\boldsymbol{\gamma} - \boldsymbol{\beta}$ and $\boldsymbol{\gamma} - \boldsymbol{\alpha}$ for the forward and backward constraints, respectively, and minimizing with respect to $\hat{p}_t$ and q_t, we obtain

$$\begin{aligned} q_t(\mathbf{x}_t) &\propto \mathrm{e}^{\hat{\boldsymbol{\gamma}}_t \mathbf{f}_t(\mathbf{x}_t)} \\ \hat{p}_t(\mathbf{x}_{t-1,t}) &\propto \mathrm{e}^{(\boldsymbol{\gamma}_{t-1} - \boldsymbol{\beta}_{t-1})^T \mathbf{f}(\mathbf{x}_{t-1})} \psi_t(\mathbf{x}_{t-1,t}) \mathrm{e}^{(\boldsymbol{\gamma}_t - \boldsymbol{\alpha}_t)^T \mathbf{f}(\mathbf{x}_t)} \, . \quad {\scriptstyle (t\, =\, 1\, :\, T)} \, , \end{aligned}$$

with

$$\hat{\boldsymbol{\gamma}}_t \equiv \boldsymbol{\gamma}_t - \frac{1}{\kappa} \left[\boldsymbol{\gamma}_t - (\boldsymbol{\alpha}_t + \boldsymbol{\beta}_t) \right] \, .$$

Substitution into (28) yields for the inner loop

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \left\{ -\kappa \sum_{t=1}^{T-1} \log \int d\mathbf{x}_t\, \mathrm{e}^{\hat{\boldsymbol{\gamma}}_t^T \mathbf{f}(\mathbf{x}_t)} - \sum_{t=1}^{T} \log \int d\mathbf{x}_{t-1,t}\, \mathrm{e}^{(\boldsymbol{\gamma}_{t-1} - \boldsymbol{\beta}_{t-1})^T \mathbf{f}(\mathbf{x}_{t-1})} \psi_t(\mathbf{x}_{t-1,t}) \mathrm{e}^{(\boldsymbol{\gamma}_t - \boldsymbol{\alpha}_t)^T \mathbf{f}(\mathbf{x}_t)} \right\} . \tag{29}$$

In the limit $\kappa \to 0$, we regain the inner loop described in Section 5.1: the first term induces the constraint $\boldsymbol{\alpha}_t + \boldsymbol{\beta}_t = \boldsymbol{\gamma}_t$, which we could get rid of by introducing $\boldsymbol{\delta}_t \equiv \boldsymbol{\alpha}_t - \boldsymbol{\beta}_t$. In Yuille's CCCP inner loop the maximization is unconstrained. The extra $q \log q$ term complicates the analysis, but with the introduction of extra Lagrange multipliers for the normalization of $\hat{p}$ and q, similar fixed point iterations can be found, at least in the situation without projections (as for loopy belief propagation). In any case, the unique solution of (29) obeys

$$\boldsymbol{\alpha}_t = \tilde{\boldsymbol{\alpha}}_t + \frac{1}{2+\kappa} \left[\boldsymbol{\gamma}_t - (\tilde{\boldsymbol{\alpha}}_t + \tilde{\boldsymbol{\beta}}_t) \right] \quad \text{and} \quad \boldsymbol{\beta}_t = \tilde{\boldsymbol{\beta}}_t + \frac{1}{2+\kappa} \left[\boldsymbol{\gamma}_t - (\tilde{\boldsymbol{\alpha}}_t + \tilde{\boldsymbol{\beta}}_t) \right] \, , \tag{30}$$

which corresponds to our $\boldsymbol{\delta}_t = \tilde{\boldsymbol{\delta}}_t$ with an additional change in $\boldsymbol{\alpha}_t + \boldsymbol{\beta}_t$. This latter change vanishes in the limit $\kappa \to 0$.

The outer loop in Yuille's CCCP algorithm minimizes (28) with respect to $\boldsymbol{\gamma}$, keeping q fixed, and yields, after some manipulations using (30),

$$\boldsymbol{\gamma}_t^{\mathrm{new}} = \hat{\boldsymbol{\gamma}}_t = \boldsymbol{\gamma}_t + \frac{1}{2+\kappa} \left[\tilde{\boldsymbol{\alpha}}_t + \tilde{\boldsymbol{\beta}}_t - \boldsymbol{\gamma}_t \right] \, , \tag{31}$$

to be compared with (27).

Summarizing, the crucial trick to turn the constrained minimization of the primal, which consists of a convex and concave part, into something doable, is to get rid of the concave part through a Legendre transformation (or an equivalent bounding technique). The remaining functional in the two-slice marginals $\hat{p}$ is then convex and can be "dualized", yielding a maximization with respect to the Lagrange multipliers. Taking over part of the concave term to the convex side is unnecessary and makes the bound introduced through the Legendre transformation less accurate, yielding a (slightly and unnecessarily) less efficient algorithm.

6 Faster alternatives

6.1 Saddle-point problem

Explicitly writing out the constrained minimization in the inner loop into maximization over Lagrange multipliers $\boldsymbol{\delta}$, we can turn the minimization in (16) into the equivalent saddle-point problem

$$\min_{\boldsymbol{\gamma}} \max_{\boldsymbol{\delta}} F(\boldsymbol{\gamma}, \boldsymbol{\delta}) \quad \text{with} \quad F(\boldsymbol{\gamma}, \boldsymbol{\delta}) \equiv F_0(\boldsymbol{\gamma}) + F_1(\boldsymbol{\gamma}, \boldsymbol{\delta}) \, ,$$

where

$$F_0(\boldsymbol{\gamma}) = \sum_{t=1}^{T-1} \log \int d\mathbf{x}_t \, \mathrm{e}^{\boldsymbol{\gamma}_t^T \mathbf{f}(\mathbf{x}_t)} \, ,$$

and where $F_1(\boldsymbol{\gamma}, \boldsymbol{\delta})$ follows from (20) if we take into account its dependency on $\boldsymbol{\gamma}$ as well.

Note that $F(\boldsymbol{\gamma}, \boldsymbol{\delta})$ is concave in $\boldsymbol{\delta}$, but, through the dependency of $F_1(\boldsymbol{\gamma}, \boldsymbol{\delta})$ on $\boldsymbol{\gamma}$, non-convex in $\boldsymbol{\gamma}$. We can guarantee convergence to a correct saddle-point with a double-loop algorithm that completes the concave maximization procedure in the inner loop before making the next step going downhill in the outer loop. This is, of course, exactly the procedure described in Section 5. The full completion of the inner loop seems quite inconvenient and a straightforward simplification is then to intermix updates in the inner and the outer loop.

The first alternative that we will consider can be interpreted as a kind of natural gradient descent in $\boldsymbol{\gamma}$ and ascent in $\boldsymbol{\delta}$: decrease with respect to $\boldsymbol{\gamma}$ and increase with respect to $\boldsymbol{\delta}$ can be guaranteed at each step, but this can, alas, not guarantee convergence to the correct saddle point. The second alternative is a damped version of “standard” expectation propagation as outlined in Section 3. In a first-order expansion around a saddle-point, both algorithms will be shown to be equivalent.

6.2 Combined gradient descent and ascent

Our first proposal is to apply the inner loop updates (24) in $\boldsymbol{\delta}$ and outer loop updates (25) in $\boldsymbol{\gamma}$ sequentially. With each update, we can guarantee

$$F(\boldsymbol{\gamma}^{\text{new}}, \boldsymbol{\delta}) \leq F(\boldsymbol{\gamma}, \boldsymbol{\delta}) \leq F(\boldsymbol{\gamma}, \boldsymbol{\delta}^{\text{new}}) \, , \tag{32}$$

where to satisfy the second inequality, we may have to resort to a damped update in $\boldsymbol{\delta}$ with $\epsilon_\delta < 1$. Note that the update (25) minimizes the primal objective with respect to $\boldsymbol{\gamma}_t$ for any $q_t(\mathbf{x}_t)$, not necessarily one that results from the optimization in the inner loop. However, the perhaps simpler update (27) is specific to $\boldsymbol{\delta}$ being at a maximum of $F_1(\boldsymbol{\gamma}, \boldsymbol{\delta})$ and *cannot* guarantee a decrease in $F(\boldsymbol{\gamma}, \boldsymbol{\delta})$ for general $\boldsymbol{\delta}$.[6]

Alternatively, we can update $\boldsymbol{\delta}_t$ and $\boldsymbol{\gamma}_t$ at the same time, for example damping the update in $\boldsymbol{\gamma}_t$ as well, e.g. taking

$$\boldsymbol{\gamma}_t^{\text{new}} \;\; = \;\; \boldsymbol{\gamma}_t + \epsilon_\gamma \left[\mathbf{g}^{-1} \left(\frac{1}{2} \left[\langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_t} + \langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_{t+1}} \right] \right) - \boldsymbol{\gamma}_t \right]$$

[6] In fact, even a damped version does not work. If it did, we should be able to show that the update in $\boldsymbol{\gamma}_t$ is always in the direction of the gradient $\partial_{\boldsymbol{\gamma}_t} F(\boldsymbol{\gamma}, \boldsymbol{\delta})$, which boils down to

$$(2\boldsymbol{\gamma}_0 - \boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2)^T \left[2\mathbf{g}(\boldsymbol{\gamma}_0) - \mathbf{g}(\boldsymbol{\gamma}_1) - \mathbf{g}(\boldsymbol{\gamma}_2) \right] \geq 0$$

for all $\boldsymbol{\gamma}_0$, $\boldsymbol{\gamma}_1$, and $\boldsymbol{\gamma}_2$. This inequality is valid everywhere if and only if $\mathbf{g}(\boldsymbol{\gamma})$ is linear in $\boldsymbol{\gamma}$. Otherwise we can construct cases that violate the inequality. For example, take $\gamma_1 = \gamma_0 + \epsilon_1$ and $\gamma_2 = \gamma_0 - \epsilon_2$ with ϵ_1 and ϵ_2 small and expand up to second order to get (in one-dimensional notation)

$$(2\gamma_0 - \gamma_1 - \gamma_2) \left[2g(\gamma_0) - g(\gamma_1) - g(\gamma_2) \right] = 4g'(\gamma_0)(\epsilon_1 - \epsilon_2)^2 + g''(\gamma_0)(\epsilon_1^2 + \epsilon_2^2)(\epsilon_1 - \epsilon_2) \, .$$

Now we can always construct a situation in which the second term dominates the first ($\epsilon_1^2 + \epsilon_2^2 \gg \epsilon_1 - \epsilon_2 \approx 0$) and choose $\epsilon_1 - \epsilon_2$ such that this dominating term is negative. Such a situation is perhaps unlikely to occur in practice, but it shows that a downhill step cannot be guaranteed, not even for ϵ_γ small.

$$\boldsymbol{\delta}_t^{\text{new}} = \boldsymbol{\delta}_t + \epsilon_\delta \left[\mathbf{g}^{-1}\left(\langle \mathbf{f}(\mathbf{x}_t)\rangle_{\hat{p}_t}\right) - \mathbf{g}^{-1}\left(\langle \mathbf{f}(\mathbf{x}_t)\rangle_{\hat{p}_{t+1}}\right)\right] . \tag{33}$$

We can either update the canonical parameters for all t at the same time or one after the other, e.g., by going forward and backward as one would do in standard message passing.

Both update schemes can be loosely interpreted as doing combined gradient descent in $\boldsymbol{\gamma}$ and gradient ascent in $\boldsymbol{\delta}$. Gradient descent-ascent is a standard approach in the field of optimization for finding saddle points of an objective function. Convergence to an, in fact unique, saddle point can be guaranteed if $F(\boldsymbol{\gamma}, \boldsymbol{\delta})$ is convex in $\boldsymbol{\gamma}$ for $\boldsymbol{\delta}$ and concave in $\boldsymbol{\delta}$ for all $\boldsymbol{\gamma}$, provided that the step sizes ϵ_δ and ϵ_γ are sufficiently small [30]. In our more general case, where $F(\boldsymbol{\gamma}, \boldsymbol{\delta})$ need not be convex in $\boldsymbol{\gamma}$ for all $\boldsymbol{\delta}$, gradient descent-ascent may not be a reliable way of finding a saddle point. For example, it is possible to construct situations in which gradient descent-ascent leads to a limit cycle [33].

The most we can therefore say is that the above update schemes with sufficient damping are locally stable, i.e., will converge back to the saddle point if slightly perturbed away from it, if the functional $F(\boldsymbol{\gamma}, \boldsymbol{\delta})$ is locally convex-concave.[7] A more detailed proof for the updates (33) is given in the Appendix.

6.3 Damped expectation propagation

The joint updates (33) have the flavor of a damped update in the direction of "standard" expectation propagation, but are not exactly the same. Straightforwardly damping the full updates $\boldsymbol{\alpha}_t^{\text{new}} = \tilde{\boldsymbol{\alpha}}_t = \tilde{\boldsymbol{\alpha}}_t(\boldsymbol{\alpha}_{t-1}, \boldsymbol{\beta}_t)$ and $\boldsymbol{\beta}_t^{\text{new}} = \tilde{\boldsymbol{\beta}}_t = \tilde{\boldsymbol{\beta}}_t(\boldsymbol{\alpha}_t, \boldsymbol{\beta}_{t+1})$, we get

$$\begin{aligned} \boldsymbol{\alpha}_t^{\text{new}} &= \boldsymbol{\alpha}_t + \epsilon(\tilde{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}_t) \\ \boldsymbol{\beta}_t^{\text{new}} &= \boldsymbol{\beta}_t + \epsilon(\tilde{\boldsymbol{\beta}}_t - \boldsymbol{\beta}_t) , \end{aligned} \tag{34}$$

or, in terms of $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$,

$$\begin{aligned} \boldsymbol{\gamma}_t^{\text{new}} &= \boldsymbol{\gamma}_t + \epsilon(\tilde{\boldsymbol{\gamma}}_t - \boldsymbol{\gamma}_t) \\ \boldsymbol{\delta}_t^{\text{new}} &= \boldsymbol{\delta}_t + \epsilon(\tilde{\boldsymbol{\delta}}_t - \boldsymbol{\delta}_t) , \end{aligned} \tag{35}$$

with $\tilde{\boldsymbol{\gamma}}_t \equiv \tilde{\boldsymbol{\alpha}}_t + \tilde{\boldsymbol{\beta}}_t$.

This update for $\boldsymbol{\delta}_t$ is the same as in (33) if we take $\epsilon_\delta = \epsilon$. The updates for $\boldsymbol{\gamma}_t$ are slightly different. The connection between them becomes clearer when we take $\epsilon_\gamma = 2\epsilon$ and rewrite (33) as

$$\boldsymbol{\gamma}_t^{\text{new}} = \boldsymbol{\gamma}_t + \epsilon(\boldsymbol{\Delta}_t + \tilde{\boldsymbol{\gamma}}_t - \boldsymbol{\gamma}_t) ,$$

with

$$\boldsymbol{\Delta}_t \equiv 2\mathbf{g}^{-1}\left(\frac{1}{2}\left[\mathbf{g}(\boldsymbol{\alpha}_t + \tilde{\boldsymbol{\beta}}_t) + \mathbf{g}(\tilde{\boldsymbol{\alpha}}_t + \boldsymbol{\beta}_t)\right]\right) - \left([\boldsymbol{\alpha}_t + \tilde{\boldsymbol{\beta}}_t] + [\tilde{\boldsymbol{\alpha}}_t + \boldsymbol{\beta}_t]\right) .$$

This term $\boldsymbol{\Delta}_t$ is then the only difference between the gradient descent-ascent updates (33) and the damped expectation propagation updates (35). Recall that it is zero at a particular maximum $\boldsymbol{\delta}^*(\boldsymbol{\gamma})$, but *not* in general for all $\boldsymbol{\delta}$.

However, local stability only depends on properties of the linearized versions of the updated close to the saddle point. In the Appendix it is shown that these coincide for the gradient descent-ascent and damped expectation propagation updates. Basically, both $\boldsymbol{\Delta}_t$ and its derivatives with respect to $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ vanish at a fixed point. Therefore, we can conclude that the damped expectation propagation updates (35) and (34) have the same local stability properties. Furthermore, since the ordering of updates does not affect these notions of stability in the limit of small ϵ, this also applies to a standard sequential message passing scheme.

[7] While turning this report into a submission, we extended this argumentation and managed to prove the stronger statement that a stable fixed point of damped expectation propagation must correspond to a *minimum* of the Bethe free energy (but not necessarily the other way around). See the conference paper [12].

7 Simulation results

We have tested our algorithms on randomly generated switching linear dynamical systems. The structure of a switching linear dynamical system is visualized in Figure 5 and Section 3.3.1 describes "standard" expectation propagation propagation.

We generated many random instances. Each instance is a combination of a switching linear dynamical system, consisting of all parameters specifying the model (observation and transition matrices, covariance matrices, and priors) and evidence $\mathbf{y}_{1:T}$. Each instance then corresponds to a particular setting of the potentials $\psi_t(\mathbf{x}_{t-1}, \mathbf{x}_t)$. We generated all model parameters randomly and independently from "normal" distributions (e.g., elements of the observation matrices from a Gaussian with mean zero and unit variance, covariance matrices of order one from Wishart distributions, rows of the transition matrices between switches and continuous latent variables independently, and so on). We varied the length of the time sequence between 3 and 5, the number of switches between 2 and 4, and the dimension of the continuous latent variables and the observations between 2 and 4. The evidence that we used was generated by a switching linear dynamical system with the same "structure" (length of time sequence, number of switches, and dimensions), but different model parameters.

In the following we will focus on the quality of the approximated beliefs $\hat{P}(\mathbf{x}_t|\mathbf{y}_{1:T})$ and compare them with the exact beliefs that result from the algorithm of [18] based on strong marginalization. As a quality measure we consider the Kullback-Leibler divergence $\mathrm{KL} \equiv \sum_{t=1}^{T} \mathrm{KL}(P_t|\hat{P}_t)$ between the exact beliefs $P_t(\mathbf{x}_t)$ and the approximate beliefs $\hat{P}_t(\mathbf{x}_t)$, both of which are conditional Gaussians. Although any particular choice is somewhat arbitrary and depends on the application at hand, for the problems considered here (relatively small time sequences) we tend to make the following crude characterization.

		KL	$>$	10^1	useless
10^1	$>$	KL	$>$	10^0	doubtful
10^0	$>$	KL	$>$	10^{-2}	useful
10^{-2}	$>$	KL			excellent

In most cases (more than 95% of all instances generated following the procedure described above), "standard" (undamped) expectation propagation works fine and converges within a couple of iterations. For a typical instance (see Figure 7 on the left), the Kullback-Leibler divergence drops after a single forward pass (equivalent to GPB2, the standard procedure for inference in switching linear dynamical systems) to an acceptably low value, decreases a little more in the smoothing step, and perhaps a little further in one or two more sweeps until no more significant changes can be seen. Damped expectation propagation and the double-loop algorithm converge to the same fixed point, but are less efficient.

The instances for which undamped expectation propagation does not converge can be roughly subdivided into two categories. The first category consists of instances that run into numerical problems, in some cases already in the forward pass. Damping as well as the double-loop algorithm seem to help in some cases, but in the end often also run into trouble when getting close to the fixed point. Instability is a well-known problem for inference in hybrid networks. It can occur especially when covariance matrices corresponding to the conditional probabilities of the continuous latent variables become singular. Solving this is an important subject, see e.g. [19], but beyond the scope of the current study.

We therefore focus on the second category, where undamped expectation propagation gets stuck in a limit cycle. A typical instance is shown in Figure 7 on the right. Here the period of the limit cycle is 8 (eight sweeps, each consisting of a forward pass and a backward pass); smaller and even larger periods can be found as well. The approximate beliefs jump between 16 different solutions, ranging from "useful" to "useless".

Damping the belief updates a little, say with $\epsilon = 0.5$ as in Figure 7, is for almost all instances sufficient to converge to a stable solution. Occasionally, in less than 1% of all "cyclical" instances

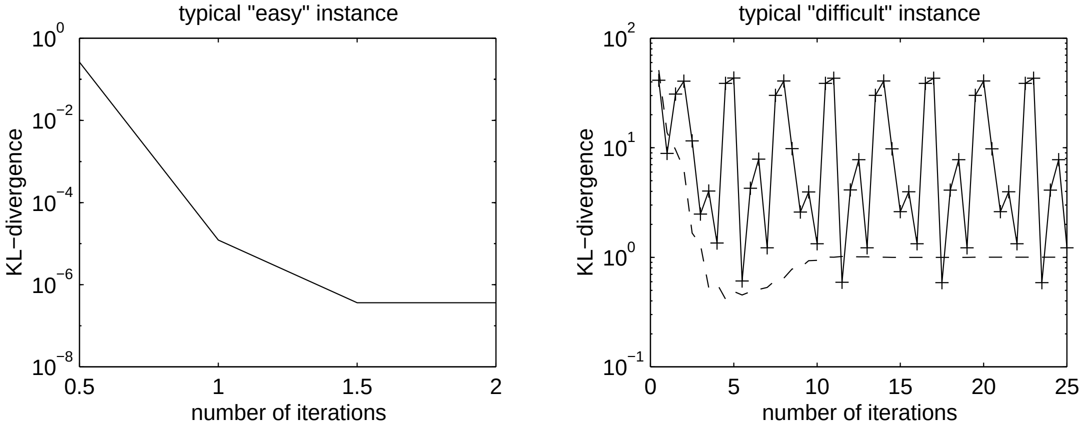

Figure 7: Typical examples of "easy" (left) and "difficult" (right) problems. Both are randomly generated instances of a switching linear dynamical systems with 3 switch states, 3-dimensional continuous latent variables, 4-dimensional observations, and sequence length $T = 5$. The KL-divergence between exact and approximated beliefs is plotted as a function of the number of iterations. Each iteration consists of a forward and a backward pass of expectation propagation. For the "easy" instance, nondamped expectation belief propagation converges in a few iterations. Without damping (solid line), the "difficult" instance gets stuck in a limit cycle of period 8. Damping with step size $\epsilon = 0.5$ (dashed line) is sufficient to convergence to a stable fixed point.

that we encountered, we did not manage to damp expectation propagation to a stable solution with reasonable values of the step size ϵ. An example is given in Figure 8. Lowering the step size ϵ mainly affects the period of the cycle, hardly its amplitude. The double-loop algorithm converges in several iterations (comparable to the number of iterations required for other instances) to a stable fixed point, which, in this case, happens to be rather far from the exact beliefs. A simple check reveals that this fixed point, by construction of the double-loop algorithm a local minimum of the free energy, is *unstable* under damped expectation propagation with step size as low as $\epsilon = 0.01$[8].

In the recent literature, it has been suggested at several places (see e.g. [37, 23]) that when undamped (loopy) belief propagation does not converge, it makes no sense to search for the minimum of the Bethe free energy with a more complicated algorithm: the failure of undamped belief propagation to converge indicates that the solution is inaccurate anyways. To check this hypothesis, we did the following experiment. For each of the 138 nonconvergent cyclic instances that we found, we generated another converging instance with the same "structure" (length of time sequence, number of switch states, and dimensions). As before, we refer to the former set of nonconvergent instances as "difficult" and the latter set of convergent instances as "easy". In Figure 9 we have plotted the KL-divergences after a single forward pass (corresponding to GPB2) and after convergence (with damped expectation propagation or the double-loop algorithm for the difficult instances). Two important conclusions can be drawn.

- *It makes sense to search for the minimum of the free energy.* For almost all instances, the beliefs corresponding to the minimum of the free energy are closer to the exact beliefs than the ones obtained after a single forward pass. This is not only the case for all 138 "easy" instances that we have seen, but also for almost all (132 out of 138) "difficult" ones. Given our crude subdivision between "useful" and "useless" or "doubtful" approximations based on the value of the KL-divergence (KL <> 1), the improvement can be considered relevant for many instances: for 64 out of 138 "easy" instances and for even 92 out of 138

[8] See [12] for a possible explanation of this effect.

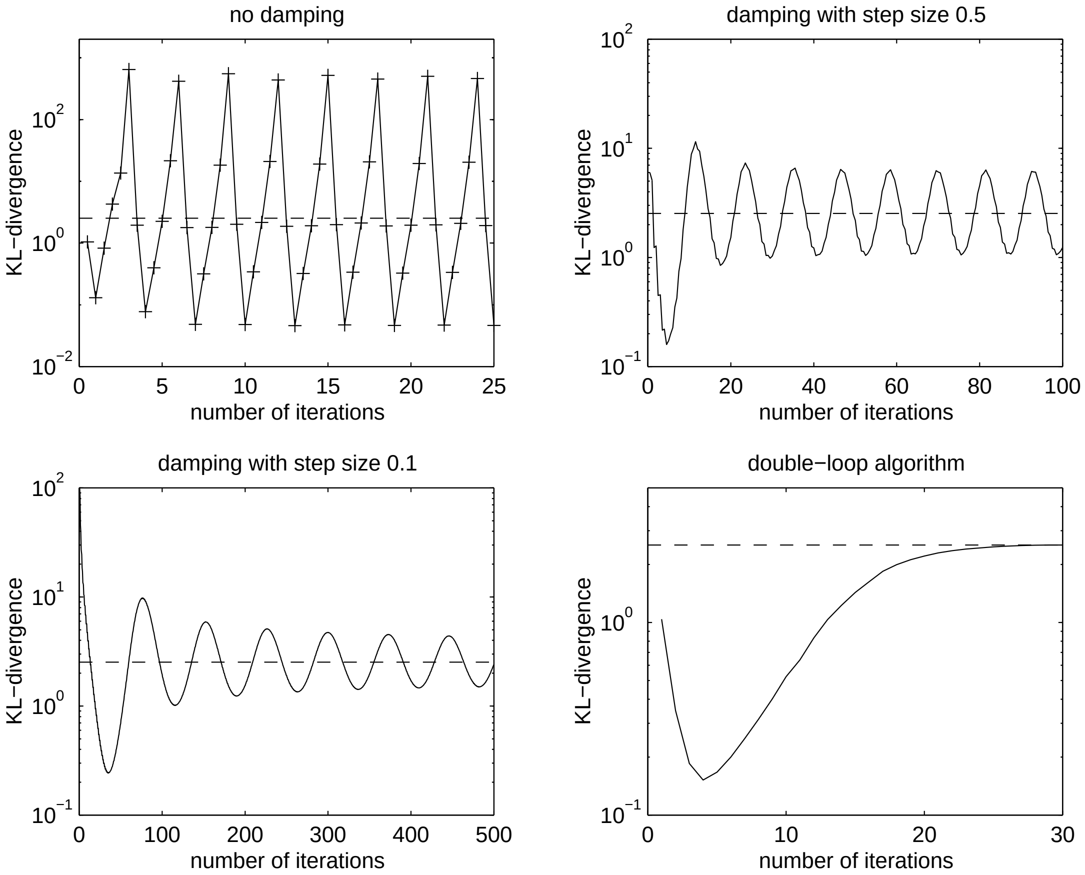

Figure 8: A rare instance in which damping with reasonable values of the step size ϵ does not lead to convergence to a stable fixed point. Changing the step size ϵ from 1 (upper left: no damping) to 0.5 (upper right) and further to 0.1 (lower left) lengthened the period of the cycle, but (going from 0.5 to 0.1) hardly its amplitude. The double-loop algorithm (lower right) converges within a few iterations (each iteration consists of a full inner-loop maximization and outer-loop step). The KL-divergence corresponding to the minimum of the free energy obtained with the double-loop algorithm is indicated with a dashed line in all plots for reference. This particular instance has 2 switch states, 3-dimensional continuous latent variables and observations, and $T = 3$.

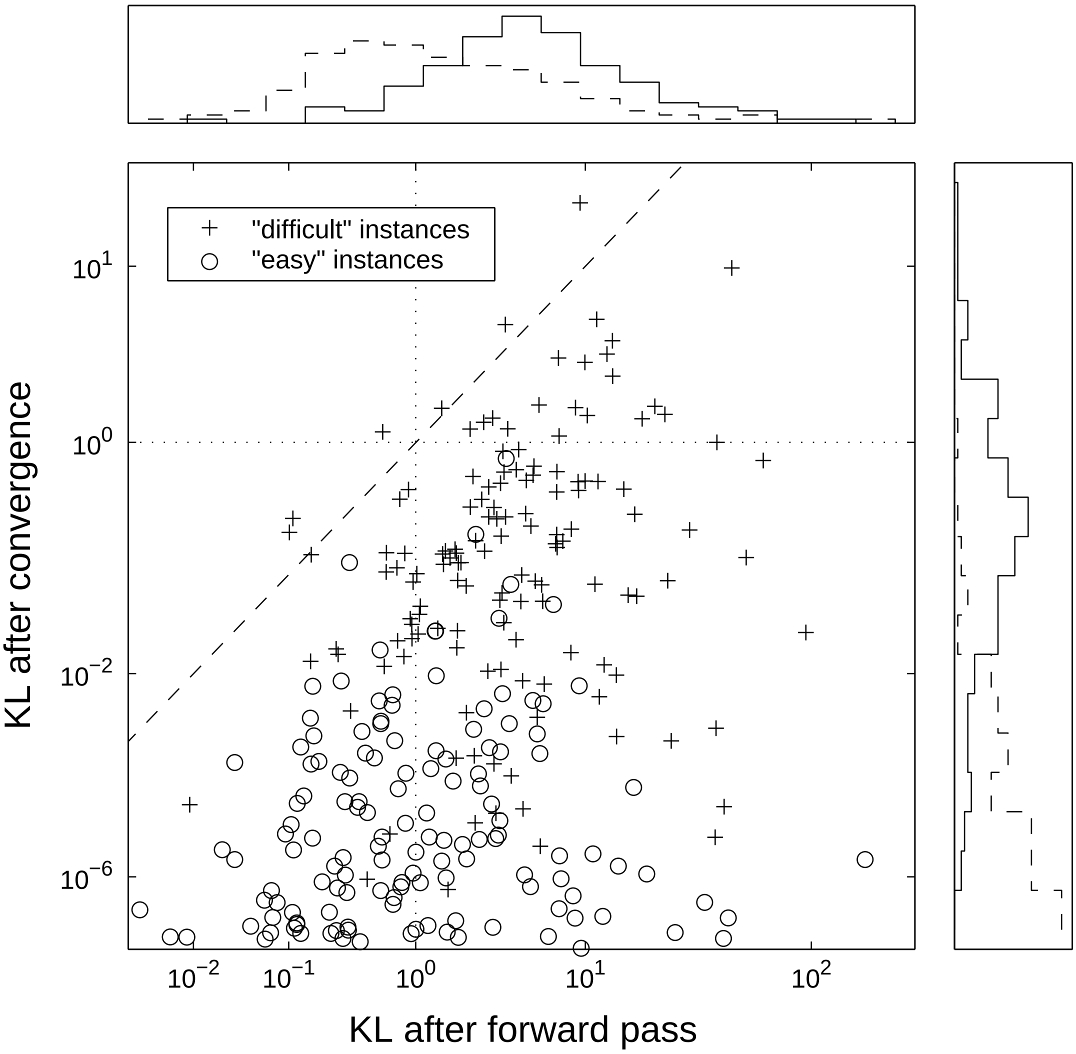

Figure 9: KL-divergences for "easy" ('o', converging without damping) and "difficult" ('+', getting stuck in a limit cycle without damping) instances after a single forward pass of expectation propagation versus after convergence to a minimum of the free energy. The dashed line corresponds to no improvement ($y = x$). The dotted grid lines crudely subdivide the KL-divergences between more and less "useful" (KL smaller than or larger than 1, which is obviously somewhat arbitrary). It can be seen that searching for the minimum in almost all instances leads to an improvement in the accuracy of the approximate beliefs, often making the difference between more or less "useful" (all instances in the lower right corner). The histograms visualize the distributions of the KL-divergences along the corresponding axes (dashed for "easy" instances, solid for "difficult" ones). The overlap between both distributions indicates that (non-)convergence of undamped expectation propagation is not a clear-cut indicator for the applicability of the minimum of the free energy for approximate inference.

"difficult" instances.

- *Convergence of undamped belief propagation is not a clear-cut criterion for the quality of the approximation.* Although the "easy" instances typically have a smaller KL-divergence than the "difficult" ones, there is considerable overlap both for the KL-divergence after a single forward pass and after convergence. For example, in 8 of the 138 cases the "difficult" instance happened to converge to a better (lower KL-divergence) solution than the corresponding "easy" one.

8 Discussion and conclusions

We have derived expectation propagation as a straightforward extension of exact belief propagation. The following ingredients are crucial to this simple interpretation.

1. A description of belief propagation that is symmetric with respect to the forward and backward messages.
2. The notion that we should better project the beliefs and derive the messages from these approximate beliefs, rather than approximating the messages themselves.

The symmetric description dramatically simplifies smoothing in switching linear dynamical systems. In [26] an attempt has been made to derive an approximate smoothing procedure based on the asymmetric description of belief propagation in terms of filtered and smoothed estimates. The derivation of this procedure requires quite some additional assumptions, making it rather complicated and resulting in a highly specific implementation. The symmetric algorithm outlined in Section 3.3.1 is easy to implement and mainly boils down to the computation of the moments of a mixture of Gaussians, transformation of moment to canonical form and vice versa, and simple subtractions and additions of canonical parameters. Generalization of this procedure to message propagation in singly-connected or even loopy networks is straightforward. Applied to these networks, it would improve the "weak marginalization approach" outlined in [18].

Approximating the messages rather than the beliefs is tempting, especially when the messages themselves have a direct interpretation (e.g., $\alpha_t(\mathbf{x}_t)$ being proportional to the filtered estimate $P(\mathbf{x}_t|\mathbf{y}_{1:t})$ and $\beta_t(\mathbf{x}_t)$ to $P(\mathbf{x}_t|\mathbf{y}_{t+1:T})$). Both are equivalent when the messages in the opposite direction are still the initial ones (i.e., set to 1), but can yield quite different approximate beliefs otherwise. Furthermore, when we approximate the beliefs, the forward and backward passes start to interfere, suggesting iterations as an attempt to improve our estimates. An iterative variant of the Boyen-Koller algorithm [5] has been applied in [27] with clear improvements.

An important difference between greedy projection methods and variational approaches as in [14, 9, 8] is that the latter minimize a clearly and globally defined Kullback-Leibler divergence. A disadvantage of the variational approaches is that they minimize the Kullback-Leibler divergence with the approximate distribution Q and exact distribution P in the "wrong order", i.e., they minimize $\text{KL}(Q|P)$, where minimizing $\text{KL}(P|Q)$ seems to make better sense. Expectation propagation minimizes the "right" KL-divergence, as can be seen in (5), but does this in a greedy and local manner. The relationship between the variational approach and expectation propagation can be clarified in two different ways.

- The Kullback-Leibler divergence corresponding to the variational approach is obtained if we substitute the two-slice marginal in the free energy (14) by the product of corresponding one-slice marginals, i.e.,
$$\hat{p}_t(\mathbf{x}_{t-1,t}) = q_t(\mathbf{x}_{t-1})q_t(\mathbf{x}_t) \, ,$$

and take the minimum with respect to the one-slice marginals q_t of the chosen exponential form:

$$\min_{q_{1:T}}{}' \left\{ \sum_t \int d\mathbf{x}_{t-1,t}\, q_{t-1}(\mathbf{x}_{t-1}) q_t(\mathbf{x}_t) \log \left[\frac{q_{t-1}(\mathbf{x}_{t-1}) q_t(\mathbf{x}_t)}{\psi_t(\mathbf{x}_{t-1,t})} \right] - \sum_t \int d\mathbf{x}_t\, q_t(\mathbf{x}_t) \log q_t(\mathbf{x}_t) \right\}$$

$$= \min_Q \mathrm{KL}(Q|P) \quad \text{with} \quad Q(\mathbf{x}_{1:T}) = \prod_t q_t(\mathbf{x}_t) \quad \text{and} \quad P(\mathbf{x}_{1:T}) \propto \prod_t \psi_t(\mathbf{x}_{t-1,t}) \,. \quad (36)$$

- An equivalent definition of step 2 in expectation propagation (beginning of Section 3) is as follows.

  2. Project the two-slice marginal $\hat{P}(\mathbf{x}_{t-1,t})$ on to the distribution $q_{t-1}(\mathbf{x}_{t-1})q_t(\mathbf{x}_t)$, by minimizing the KL-divergence

  $$\mathrm{KL}(\hat{P}_{t-1,t}|q_{t-1}q_t) = \int d\mathbf{x}_{t-1,t}\, \hat{P}(\mathbf{x}_{t-1,t}) \log \left[\frac{\hat{P}(\mathbf{x}_{t-1,t})}{q_{t-1}(\mathbf{x}_{t-1}) q_t(\mathbf{x}_t)} \right] \,, \qquad (37)$$

  with respect to $q_{t'}(\mathbf{x}_{t'})$ (keeping the other approximate belief fixed).

  This yields exactly the same procedure since integrating out all variables except $\mathbf{x}_{t'}$, as in the original formulation, is then implicitly done while matching the moments. The variational approach can be obtained when we reverse the role of $\hat{P}(\mathbf{x}_{t-1,t})$ and $q_{t-1}(\mathbf{x}_{t-1})q_t(\mathbf{x}_t)$ in the above KL-divergence.[9]

In much the same way as loopy belief propagation is an improvement over mean-field approximations [38], we expect expectation propagation to outperform a variational approach that is based on the same approximate structure (independent time slices in the above case of dynamic Bayesian networks). Some evidence for that can be found in [8], where the variational approach did not lead to better results than generalized pseudo-Bayes, which is just a single forward pass of expectation propagation.

An important question is how the results obtained for chains in this article generalize to tree-like or even loopy structures. Analyzing loopy belief propagation (no projections, but a network structure containing loops) along the same lines, we arrive at a quite similar double-loop algorithm for guaranteed convergence and single-loop short-cuts [11]. Our current interpretation is that, as soon as messages start to interfere, which happens both with approximations and with loops but not with exact inference on trees, we have to take special care that we update the messages in a special way: going uphill relative to each other to satisfy the constraints, going downhill together to minimize the free energy. That damped versions of expectation propagation and loopy belief propagation seem to move in the right uphill/downhill directions might explain why single-loop algorithms converge well in many practical cases.

References

[1] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.

[9] To see this, define

$$q_t(\mathbf{x}_t) = \alpha_t(\mathbf{x}_t)\beta_t(\mathbf{x}_t) \,,$$

and replace the minimization over $q_{1:T}$ in (36) by a joint minimization over $\alpha_{1:T}$ and $\beta_{1:T}$ under the constraint that their product normalizes to 1 (and that both are of the chosen exponential form). Now consider the minimization of KL$(P|Q)$ with respect to α_t keeping all other α's and β's fixed. Collecting relevant and neglecting irrelevant terms, it is easy to see that

$$\min_{\alpha_t}{}' \mathrm{KL}(Q|P) = \min_{\alpha_t}{}' \mathrm{KL}(q_{t-1}q_t|\hat{P}_{t-1,t}) \,,$$

the KL-divergence of (37) in reverse order. In other words, greedy projection based on KL$(q_{t-1}q_t|\hat{P}_{t-1,t})$ corresponds to some kind of iterative procedure for minimizing KL$(Q|P)$.

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

[3] L. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[4] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 33–42, San Francisco, 1998. Morgan Kaufmann.

[5] X. Boyen and D. Koller. Approximate learning of dynamic models. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 396–402. MIT Press, 1999.

[6] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*, New York, 2001. Springer-Verlag.

[7] Z. Ghahramani and G. Hinton. Parameter estimation for linear dynamical systems. Technical report, Department of Computer Science, University of Toronto, 1996.

[8] Z. Ghahramani and G. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12:963–996, 1998.

[9] Z. Ghahramani and M. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–275, 1997.

[10] T. Heskes. On "natural" learning and pruning in multilayered perceptrons. *Neural Computation*, 12:1037–1057, 2000.

[11] T. Heskes. Stable fixed points of loopy belief propagation are minima of the Bethe free energy. In *NIPS 15 (in press)*, 2002.

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

[13] A. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, New York, 1970.

[14] 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.

[15] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In I. Kadar, editor, *The 11th Annual International Symposium on Aerospace/Defence Sensing, Simulation, and Controls*, volume 3068, pages 182–193. SPIE, 1997.

[16] C. Kim. Dynamic linear models with Markov-switching. *Journal of Econometrics*, 60:1–22, 1994.

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

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

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

[20] U. Lerner and R. Parr. Inference in hybrid networks: Theoretical limits and practical algorithms. In *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference (UAI-2001)*, pages 310–318, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[21] D. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Reading, Massachusetts, 1984.

[22] R. McEliece, D. MacKay, and J. Cheng. Turbo decoding as as an instance of Pearl's 'belief propagation' algorithm. *IEEE Journal on Selected Areas in Communication*, 16(2):140–152, 1998.

[23] T. Minka. The EP energy function and minimization schemes. Technical report, MIT Media Lab, 2001.

[24] T. Minka. Expectation propagation for approximate Bayesian inference. In *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference (UAI-2001)*, pages 362–369, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[25] T. Minka. *A family of algorithms for approximate Bayesian inference.* PhD thesis, MIT Media Lab, 2001.

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

[27] K. Murphy and Y. Weiss. The factored frontier algorithm for approximate inference in DBNs. In *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference (UAI-2001)*, pages 378–385, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[28] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Articial Intelligence*, pages 467–475, San Francisco, CA, 1999. Morgan Kaufmann.

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

[30] J. Platt and A. Barr. Constrained differential optimization. In D. Anderson, editor, *Neural Information Processing Systems*, pages 612–621. American Institute of Physics, New York, NY, 1987.

[31] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[32] H. Rauch. Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*, 8:371–372, 1963.

[33] S. Seung, T. Richardson, J. Lagarias, and J. Hopfield. Minimax and Hamiltonian dynamics of excitatory-inhibitory networks. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 329–335. MIT Press, 1998.

[34] R. Shumway and Y. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3:682–687, 1982.

[35] R. Shumway and Y. Stoffer. Dynamic linear models with switching. *Journal of the American Statistical Association*, 86:763–769, 1991.

[36] P. Smyth, D. Heckerman, and M. Jordan. Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, pages 227–269, 1997.

[37] Y. Teh and M. Welling. The unified propagation and scaling algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 953–960. MIT Press, 2002.

[38] J. Yedidia, W. Freeman, and Y. Weiss. Generalized belief propagation. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 689–695. MIT Press, 2001.

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

Appendix: Local stability

We consider the local stability of the updates (33) near a fixed point with canonical parameters $\boldsymbol{\gamma}^*$ and $\boldsymbol{\delta}^*$. First note that, with shorthand notation

$$\mathbf{m}_t^- \equiv \langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_t} \ , \quad \mathbf{m}_t^+ \equiv \langle \mathbf{f}(\mathbf{x}_t) \rangle_{\hat{p}_{t+1}} \ , \ \text{ and } \ \mathbf{m}_t^0 \equiv \langle \mathbf{f}(\mathbf{x}_t) \rangle_{q_t} \ ,$$

the gradients with respect to $\boldsymbol{\gamma}_t$ and $\boldsymbol{\delta}_t$ can be written

$$\frac{\partial F(\boldsymbol{\gamma}, \boldsymbol{\delta})}{\partial \boldsymbol{\gamma}_t} = \mathbf{m}_t^0 - \frac{1}{2}(\mathbf{m}_t^- + \mathbf{m}_t^+) \ \text{ and } \ \frac{\partial F(\boldsymbol{\gamma}, \boldsymbol{\delta})}{\partial \boldsymbol{\delta}_t} = \frac{1}{2}(\mathbf{m}_t^+ - \mathbf{m}_t^-) \, . \tag{38}$$

An infinitesimal change in the parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ has the following effect on the gradient-ascent update (25) in $\boldsymbol{\gamma}_t$,

$$\partial \Delta \boldsymbol{\gamma}_t = \epsilon \left. \frac{\partial \mathbf{g}^{-1}(\mathbf{m}_t)}{\partial \mathbf{m}_t^T} \right|_{\mathbf{m}_t = \frac{1}{2}(\mathbf{m}_t^- + \mathbf{m}_t^+)} \left(\partial \mathbf{m}_t^- + \partial \mathbf{m}_t^+ \right) - 2\epsilon \partial \boldsymbol{\gamma}_t \, .$$

At the fixed point itself we have $\mathbf{m}^- = \mathbf{m}^+ = \mathbf{m}^0 = \mathbf{m}^*$. Defining

$$\left. \frac{\partial \mathbf{g}^{-1}(\mathbf{m}_t)}{\partial \mathbf{m}_t^T} \right|_{\mathbf{m}_t = \frac{1}{2}(\mathbf{m}_t^- + \mathbf{m}_t^+) = \mathbf{m}_t^*} = \left[\left. \frac{\partial \mathbf{g}(\boldsymbol{\gamma}_t)}{\partial \boldsymbol{\gamma}_t^T} \right|_{\boldsymbol{\gamma}_t = \boldsymbol{\gamma}_t^*} \right]^{-1} \equiv H_t^{-1} \, ,$$

with H_t a positive definite Hessian, we have for an infinitesimal perturbation away from the fixed point,

$$\partial \Delta \boldsymbol{\gamma}_t = \epsilon H_t^{-1} \left(\partial \mathbf{m}_t^- + \partial \mathbf{m}_t^+ - 2 \partial \mathbf{m}_t^0 \right) \, . \tag{39}$$

Comparing with the gradient above, it is easy to see that a small change in $\Delta \boldsymbol{\gamma}_t$ is proportional to a small change in the gradient $\partial_{\boldsymbol{\gamma}_t} F(\boldsymbol{\gamma})$, where the "Hessian" H_t can be loosely interpreted as the "Riemannian metric" in Amari's work on natural gradients [1, 10]. Similarly, for a damped version of the update (24) in $\boldsymbol{\delta}$ as in (33) we have, close to the fixed point,

$$\partial \Delta \boldsymbol{\delta}_t = \epsilon H_t^{-1} \left(\partial \mathbf{m}_t^+ - \partial \mathbf{m}_t^- \right) \, . \tag{40}$$

Combination of (39) and (40) with (38) yields, in a first-order expansion around the fixed point $\{\boldsymbol{\gamma}^*, \boldsymbol{\delta}^*\}$,

$$\begin{aligned} \Delta \boldsymbol{\gamma}_t &= -2\epsilon \sum_s H_t^{-1} \left[\frac{\partial^2 F}{\partial \boldsymbol{\gamma}_t \boldsymbol{\gamma}_s^T} (\boldsymbol{\gamma}_s - \boldsymbol{\gamma}_s^*) + \frac{\partial^2 F}{\partial \boldsymbol{\gamma}_t \boldsymbol{\delta}_s} (\boldsymbol{\delta}_s - \boldsymbol{\delta}_s^*) \right] \\ \Delta \boldsymbol{\delta}_t &= 2\epsilon \sum_s H_t^{-1} \left[\frac{\partial^2 F}{\partial \boldsymbol{\delta}_t \boldsymbol{\delta}_s^T} (\boldsymbol{\delta}_s - \boldsymbol{\delta}_s^*) + \frac{\partial^2 F}{\partial \boldsymbol{\delta}_t \boldsymbol{\gamma}_s} (\boldsymbol{\gamma}_s - \boldsymbol{\gamma}_s^*) \right] \, , \end{aligned} \tag{41}$$

with all second derivatives evaluated at the fixed point. Let us consider the distance to the fixed point with "natural" metric H_t:

$$L \equiv \frac{1}{2} \sum_t \left[(\boldsymbol{\gamma}_t - \boldsymbol{\gamma}_t^*)^T H_t (\boldsymbol{\gamma}_t - \boldsymbol{\gamma}_t^*) + (\boldsymbol{\delta}_t - \boldsymbol{\delta}_t^*)^T H_t (\boldsymbol{\delta}_t - \boldsymbol{\delta}_t^*) \right] .$$

In lowest order of ϵ, the change in L due to the changes (41) reads

$$\Delta L = -2\epsilon \sum_{t,s} (\boldsymbol{\gamma}_t - \boldsymbol{\gamma}_t^*) \frac{\partial^2 F}{\partial \boldsymbol{\gamma}_t \boldsymbol{\gamma}_s^T} (\boldsymbol{\gamma}_s - \boldsymbol{\gamma}_s^*) + 2\epsilon \sum_{t,s} (\boldsymbol{\delta}_t - \boldsymbol{\delta}_t^*)^T \frac{\partial^2 F}{\partial \boldsymbol{\delta}_t \boldsymbol{\delta}_s^T} (\boldsymbol{\delta}_s - \boldsymbol{\delta}_s^*) + \mathcal{O}(\epsilon^2) \leq 0 ,$$

where the cross-terms cancelled and where the inequality follows when the Hessian w.r.t. $\boldsymbol{\gamma}$ is negative definite (which it does not have to be, see [12]) and the Hessian w.r.t. $\boldsymbol{\delta}$ is positive definite (which it is by construction). The decrease in L brings the canonical parameters back to the saddle point, which proves that the saddle point is indeed stable under the gradient updates (33).

Taking the derivative of the damped update of $\boldsymbol{\gamma}_t$ in (35) we have

$$\partial \Delta \boldsymbol{\gamma}_t = \epsilon \left[\left. \frac{\partial \mathbf{g}^{-1}(\mathbf{m}_t)}{\partial \mathbf{m}_t^T} \right|_{\mathbf{m}_t = \mathbf{m}_t^-} \partial \mathbf{m}_t^- + \left. \frac{\partial \mathbf{g}^{-1}(\mathbf{m}_t)}{\partial \mathbf{m}_t^T} \right|_{\mathbf{m}_t = \mathbf{m}_t^+} \partial \mathbf{m}_t^+ \right] - 2\epsilon \partial \boldsymbol{\gamma}_t .$$

Considering an infinitesimal perturbation away from the fixed point, where $\mathbf{m}_t^- = \mathbf{m}_t^+$, we obtain (39) above: the same as for the gradient update (25). The update for $\boldsymbol{\delta}$ being equivalent everywhere, we conclude that the local stability of the partial updates (35) and thus (34) is the same is that of the gradient updates (33).