Steepest descent on factor graphs

Justin Dauwels, Sascha Korl, and Hans-Andrea Loeliger

Abstract—We show how steepest descent can be used as a tool for estimation on factor graphs. From our exposition, it should also become clear how steepest descent can be elegantly combined with other summary propagation algorithms such as the sum/max-product algorithm, expectation maximization, Kalman filters/smoothers, and particle filters.

1 Introduction

Suppose we wish to find

$$\theta_{\max} \triangleq \operatorname*{argmax}_{\theta} f(\theta), \tag{1}$$

where θ takes values in $\mathbb{R}$ or $\mathbb{R}^n$. We assume that $f(\theta)$ is a "marginal" of a real valued function $f(x, \theta)$:

$$f(\theta) \triangleq \int_x f(x, \theta), \tag{2}$$

where $\int_x g(x)$ stands for either summation or integration of $g(x)$ over the range of x. The function $f(x, \theta)$ is assumed to be nonnegative, i.e., $f(x, \theta) \geq 0$ for all x and θ. We also assume that the integral (or sum) $\int_x f(x, \theta) \log f(x, \theta')$ exists for all θ and θ'.
In principle, one can apply the sum-product algorithm in order to find (1), which involves the following two steps [2]:

1. Determine $f(\theta)$ by sum-product message passing.
2. Maximization step: compute $\theta_{\max} \triangleq \operatorname{argmax}_\theta f(\theta)$.

This procedure is often not feasible, since

- When the variable x is continuous, the sum-product rule may lead to intractable integrals, i.e., the (exact) computation of $f(\theta)$ is in this case intractable.
- The maximization step may be infeasible.

Expectation maximization is one approach to deal with both issues; the expectation maximization algorithm attempts to find (1) as follows [1] (see also [5] and [7]):

1. Make some initial guess $\theta^{(0)}$.
2. Expectation step: evaluate
$$f^{(k)}(\theta) \triangleq \int_x f(x, \hat{\theta}^{(k)}) \log f(x, \theta). \tag{3}$$
3. Maximization step: compute
$$\theta^{(k+1)} \triangleq \operatorname*{argmax}_{\theta} f^{(k)}(\theta). \tag{4}$$

The authors are with the Dept. of Information Technology and Electrical Engineering, ETH, CH-8092 Zürich, Switzerland. Email: {dauwels, korl, loeliger}@isi.ee.ethz.ch.

4. Repeat 2–3 until convergence or until the available time is over.

However, also this approach has its drawbacks:

- If the variable x is continuous, the expectation step (3) may lead to intractable integrals.
- Also the maximization steps can often not be carried out analytically.

In the present paper, we explain how such issues can be handled by steepest descent. Steepest descent (a.k.a. "gradient descent") is a simple, yet powerful optimization algorithm and consequently a widely used tool in non-linear optimization [4]. The algorithm works as follows: suppose one wishes to maximize the differentiable non-negative real function $g(\theta)$; one starts with an initial estimate $\hat{\theta}^{(0)}$ and iterates the update rule

$$\hat{\theta}^{(k+1)} = \hat{\theta}^{(k)} + \lambda_k \, \nabla_\theta g(\theta)|_{\hat{\theta}^{(k)}} \,, \tag{5}$$

where λ_k is a positive number called the "step size" or "learning rate".
An alternative rule is

$$\hat{\theta}^{(k+1)} = \hat{\theta}^{(k)} + \lambda_k \, \nabla_\theta \log g(\theta)|_{\hat{\theta}^{(k)}} \,. \tag{6}$$

The update rule (5) or (6) is iterated until a fixed point is reached or until the available time is over. The step size λ_k may be constant, i.e., $\lambda_k \triangleq \lambda$, or a monotonic decreasing function. Note that rule (6) is preferable to (5) when $g(\theta)$ strongly varies, since the logarithm in (6) compresses the range of $g(\theta)$.
The maximization step in the EM and sum-product algorithm may be carried out by iterating the rules (5) or (6) with $g(\theta) \triangleq f^{(k)}(\theta)$ and $g(\theta) \triangleq f(\theta)$ respectively. In the following sections, we will show how this may be viewed as summary propagation on a factor graph of $f(x, y)$.
This paper is structured as follows. In the next section, we treat steepest descent in conjunction with the sum-product algorithm. In Section 3, we show how the M-step in the expectation maximization algorithm can be implemented by steepest descent. We offer some concluding remarks in Section 4.

2 Sum-product algorithm and steepest descent

In earlier work [3], we briefly touched upon the subject of gradient descent in the context of the sum(mary)-product algorithm. Here, we present a more detailed exposition. As in [3], we start by considering the factor graph depicted in Fig. 1(a), which represents the global function $f(\theta) \triangleq f_A(\theta) f_B(\theta)$. The gradient $\nabla_\theta f(\theta)$ in update rule (5) is given by

$$\nabla_\theta f(\theta) \quad = \quad f_B(\theta) \nabla_\theta f_A(\theta) + f_A(\theta) \nabla_\theta f_B(\theta), \tag{7}$$

and similarly, the gradient $\nabla_\theta \log f(\theta)$ in update rule (6) equals

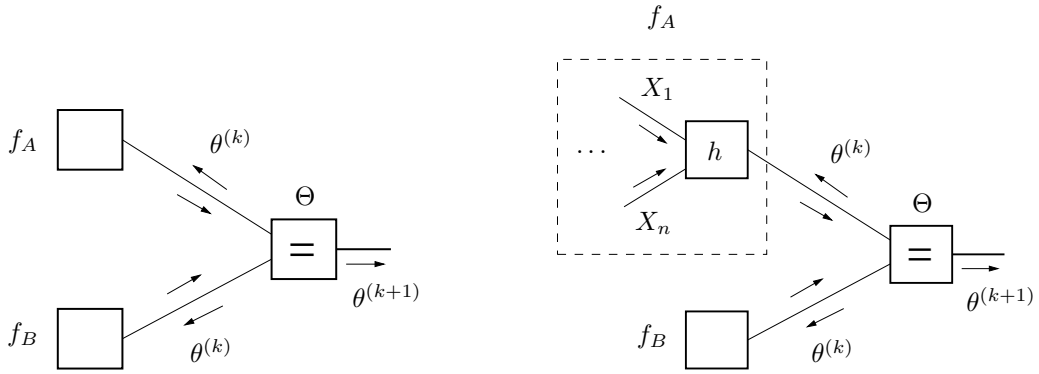
$$\nabla_\theta \log f(\theta) \quad = \quad \nabla_\theta \log f_A(\theta) + \nabla_\theta \log f_B(\theta). \tag{8}$$

Steepest descent according to rule (6) (and similarly (5)) may be viewed as follows:

1. The equality constraint node in Fig. 1(a) broadcasts the estimate $\theta^{(k)}$. Node f_A replies with $\nabla_\theta \log f_A(\theta)|_{\theta^{(k)}}$ and likewise node f_B.
2. A new estimate $\theta^{(k+1)}$ is computed as

$$\hat{\theta}^{(k+1)} = \hat{\theta}^{(k)} + \lambda_k \big(\nabla_\theta \log f_A(\theta)|_{\hat{\theta}^{(k)}} + \nabla_\theta \log f_B(\theta)|_{\hat{\theta}^{(k)}} \big). \tag{9}$$

3. Iterate 1–2.



(a) f_A and f_B as closed boxes.

(b) Opening the box f_A.

Figure 1: Steepest descent on factor graphs.

In Fig. 1(a), the nodes f_A and f_B may be summaries of the subgraph "behind" them, as illustrated in Fig. 1(b): the function f_A is a summary of the dashed box. This box contains a.o. the local node h, which is connected to the equality constraint node Θ. The summary $f_A(\theta)$ is computed from the messages $\mu_{X_k \to h}$, arriving at the node h from the left, according the sum-product rule [2]:

$$f_A(\theta) = \gamma \cdot \int_{x_1,\ldots,x_n} h(x_1, \ldots, x_n, \theta) \cdot \prod_{\ell=1}^{n} \mu_{X_\ell \to h}(x_\ell), \tag{10}$$

where γ is an arbitrary scale factor. The above gradient method requires $\nabla_\theta f_A(\theta)$ and/or $\nabla_\theta \log f_A(\theta)$ (see (9)). In the following, we show how those expressions can be computed. We distinguish three cases:

1. h is an equality constraint node
2. h is differentiable
3. h corresponds to a deterministic mapping.

2.1 Equality constraint node

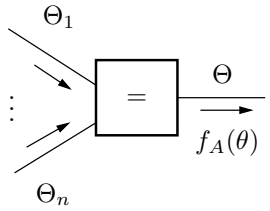


Figure 2: Equality constraint node.

If h is an equality constraint node (see Fig. 2), then the required gradients are computed analogously to (7) and (8):

$$\nabla_\theta f_A(\theta) \quad = \quad \sum_{\ell=1}^{n} \nabla_\theta \mu_{\theta_\ell \to =}(\theta) \prod_{m=1; m \neq \ell}^{n} \mu_{\theta_m \to =}(\theta), \tag{11}$$

and

$$\nabla_\theta \log f_A(\theta) \quad = \quad \sum_{\ell=1}^{n} \nabla_\theta \log \mu_{\Theta_\ell \to \boxminus}(\theta). \tag{12}$$

2.2 Differentiable node function

Let $h(x_1, \ldots, x_n, \theta)$ be differentiable w.r.t. θ. The gradient $\nabla_\theta f_A(\theta)$ can then be computed as follows:

$$\nabla_\theta f_A(\theta) \quad = \quad \gamma \cdot \nabla_\theta \left[\int_{x_1,\ldots,x_n} h(x_1, \ldots, x_n, \theta) \cdot \prod_{\ell=1}^{n} \mu_{X_\ell \to h}(x_\ell) \right] \tag{13}$$

$$= \quad \gamma \cdot \int_{x_1,\ldots,x_n} \nabla_\theta h(x_1, \ldots, x_n, \theta) \cdot \prod_{\ell=1}^{n} \mu_{X_\ell \to h}(x_\ell). \tag{14}$$

Note that in (14), we differentiated under the integral sign; this is allowed, since h is assumed to be differentiable. The update rule (14) can be viewed as applying the sum-product rule to the node $\nabla_\theta h$, as illustrated in Fig. 3. The incoming messages are the standard sum-product summaries $\mu_{X_\ell \to h}$. In

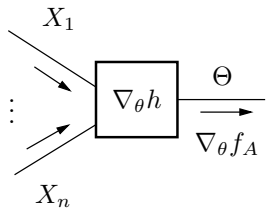


Figure 3: Differentiable node function h.

other words: if h is differentiable, then the differentiation operator does not propagate to the subgraph on the left of h. Instead, it is (only) applied to the local node function h. This is not the case when h corresponds to a deterministic mapping, which is the subject of next subsection.
The gradient $\nabla_\theta \log f_A(\theta)$ equals

$$\nabla_\theta \log f_A(\theta) \quad = \quad \frac{\nabla_\theta f_A(\theta)}{f_A(\theta)}, \tag{15}$$

and is computed from (10) and (14). In order to evaluate $\nabla_\theta \log f_A(\theta)$, the sum-product rule is thus applied twice, i.e., both to h and to $\nabla_\theta h$.
When the variables X_ℓ are discrete (and the alphabet is not "too large"), the expressions (14) and (15) can be evaluated straightforwardly. If on the other hand those variables (or a subset of them) are continuous, the integrals (10) and (14) may be evaluated in several ways [3] (see also [6] for an illustration):

- In some cases, a closed-form expression of (10) or (14) may exist.
- The integrals (10) and (14) can be approximated based on canonical distributions, as for example Gaussian distributions.
- The incoming messages $\mu_{X_\ell \to h}(x_\ell)$ may be hard estimates $\hat{x}_\ell$. The expressions (14) and (15) then reduce to

  $$\nabla_\theta f_A(\theta) = \gamma \cdot \nabla_\theta h(\hat{x}_1, \ldots, \hat{x}_n, \theta) \tag{16}$$

  and

  $$\nabla_\theta \log f_A(\theta) = \frac{\nabla_\theta h(\hat{x}_1, \ldots, \hat{x}_n, \theta)}{h(\hat{x}_1, \ldots, \hat{x}_n, \theta)}. \tag{17}$$

- The messages $\mu_{X_\ell \to h}(x_\ell)$ may be lists of samples (a.k.a. "particles") $\{x_\ell^{(i_\ell)}\}$ [3]. Consequently

$$\nabla_\theta f_A(\theta) = \gamma \cdot \sum_{i_1,\ldots,i_n} \nabla_\theta h\big(x_1^{(i_1)}, \ldots, x_n^{(i_n)}, \theta\big), \tag{18}$$

and

$$\nabla_\theta \log f_A(\theta) = \frac{\sum_{i_1,\ldots,i_n} \nabla_\theta h\big(x_1^{(i_1)}, \ldots, x_n^{(i_n)}, \theta\big)}{\sum_{i_1,\ldots,i_n} h\big(x_1^{(i_1)}, \ldots, x_n^{(i_n)}, \theta\big)}, \tag{19}$$

where the sums are taken over the lists of samples.

- When $\mu_{X_\ell \to h}(x_\ell)$ are "quantized messages" [3], i.e., the variables x_ℓ are quantized with quantization levels $x_\ell^{(i_\ell)}$, then

$$\nabla_\theta \log f_A(\theta) = \gamma \cdot \sum_{i_1,\ldots,i_n} \nabla_\theta h\big(x_1^{(i_1)}, \ldots, x_n^{(i_n)}, \theta\big) \mu_{x_1 \to h}\big(x_1^{(i_1)}\big) \ldots \mu_{x_n \to h}\big(x_n^{(i_n)}\big), \tag{20}$$

and

$$\nabla_\theta \log f_A(\theta) = \frac{\sum_{i_1,\ldots,i_n} \nabla_\theta h\big(x_1^{(i_1)}, \ldots, x_n^{(i_n)}, \theta\big) \mu_{x_1 \to h}\big(x_1^{(i_1)}\big) \ldots \mu_{x_n \to h}\big(x_n^{(i_n)}\big)}{\sum_{i_1,\ldots,i_n} h\big(x_1^{(i_1)}, \ldots, x_n^{(i_n)}, \theta\big) \mu_{x_1 \to h}\big(x_1^{(i_1)}\big) \ldots \mu_{x_n \to h}\big(x_n^{(i_n)}\big)}, \tag{21}$$

where the sums are taken over the quantization levels (which may be different for each variable).

- Combinations of the previous options are possible.

2.3 Deterministic mapping

We now consider the case where the local function h corresponds to the deterministic mapping $x_m \triangleq g(x_1, \ldots, x_{m-1}, x_{m+1}, \ldots, x_n, \theta)$, i.e.,

$$h(x_1, \ldots, x_n, \theta) \triangleq \delta\big(x_m - g(x_1, \ldots, x_{m-1}, x_{m+1}, \ldots, x_n, \theta)\big). \tag{22}$$

We assume that g is differentiable. Let $v \triangleq (x_1, \ldots, x_{m-1}, x_{m+1}, \ldots, x_n)$. The message $f_A(\theta)$ is then computed as follows

$$\begin{aligned} f_A(\theta) &= \gamma \cdot \int_{x_1,\ldots,x_n} \delta\big(x_m - g(v,\theta)\big) \cdot \prod_{\ell=1}^{n} \mu_{X_\ell \to h}(x_\ell) && (23) \\ &= \gamma \cdot \int_v \mu_{X_m \to h}\big(g(v,\theta)\big) \cdot \prod_{\ell \neq m} \mu_{X_\ell \to h}(x_\ell). && (24) \end{aligned}$$

As a consequence

$$\begin{aligned} \nabla_\theta f_A(\theta) &= \gamma \cdot \int_v \nabla_\theta \big[\mu_{X_m \to h}\big(g(v,\theta)\big)\big] \cdot \prod_{\ell \neq m} \mu_{X_\ell \to h}(x_\ell) && (25) \\ &= \gamma \cdot \int_v \nabla_\theta g(v,\theta)\ \nabla_{x_m} \mu_{X_m \to h}(x_m)\big|_{g(v,\theta)} \cdot \prod_{\ell \neq m} \mu_{X_\ell \to h}(x_\ell). && (26) \end{aligned}$$

In (25), we assumed that the message $\mu_{X_m \to h}$ is differentiable and therefore differentiation under the integral sign is allowed. Eq. (26) may be viewed as applying the sum-product rule to the node function $\nabla_\theta g$, as illustrated in Fig. 4.

Note that the differentiation operator propagates to the left of h: besides the standard sum-product messages $\mu_{X_\ell \to h}(x_\ell)$ ($\ell \neq m$), also the message $\nabla_{x_m} \mu_{X_m \to h}(x_m)|_{g(v,\theta)}$ is required, which is the *gradient* of a sum-product message! Analogously as in (14) and (15), the update rule (26) can be evaluated in several ways, depending on the datatype of the incoming messages.
For example, when the incoming messages are hard estimates $\hat{x}_\ell$ ($\ell \neq m$) and $\nabla_{x_m} \mu_{X_m \to h}(x_m)|_{g(\hat{v},\theta)}$, where $\hat{v}$ stands for $(\hat{x}_1, \ldots, \hat{x}_{m-1}, \hat{x}_{m+1}, \ldots, \hat{x}_n)$, then

$$\nabla_\theta f_A(\theta) \quad = \quad \gamma \cdot \nabla_\theta g(\hat{v}, \theta) \; \nabla_{x_m} \mu_{X_m \to h}(x_m)|_{g(\hat{v},\theta)} \cdot \prod_{\ell \neq m} \mu_{X_\ell \to h}(\hat{x}_\ell). \tag{27}$$

Eq. (27) is of course nothing else than Leibniz's chain rule for differentiation.

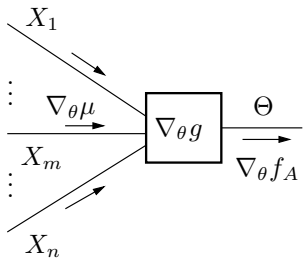


Figure 4: Deterministic mapping g.

2.4 Summary

We have seen that

1. When steepest descent is combined with the sum-product algorithm, gradients of sum-product messages are required.

2. If the local node function h is *differentiable*, the gradient of the outgoing message is computed by the sum-product rule applied to $\nabla_\theta h$, where the incoming messages are standard sum-product messages (see (14)). In other words, the differentiation operator does not propagate through the node h; it is only applied to the local node function h.

3. If the local node h corresponds to a *deterministic mapping* g, the gradient of the outgoing message is computed by the sum-product rule applied to $\nabla_\theta g$ (see (26)). All incoming messages are standard sum-product messages, except for one, which is the *gradient* of an incoming sum-product message μ_m. In this case, the differentiation operator is applied both to the local node function and the incoming message μ_m; in other words, the differentiation operator *propagates* from node h towards the node the message μ_m has been sent from.

4. Differentiation also propagates through the equality constraint node (see (11) and (12)).

5. The three previous observations indicate that along an edge X_ℓ in the factor graph, the following messages may propagate (see Fig. 5):

   - standard sum-product messages,
   - gradients of sum-product messages,
   - or both,

   depending on

- the location of the edges at which the steepest descent update rules are applied
- the kind of nodes the edge X_ℓ is connected to.

6. The sum-product messages and their gradients may be represented in various ways.

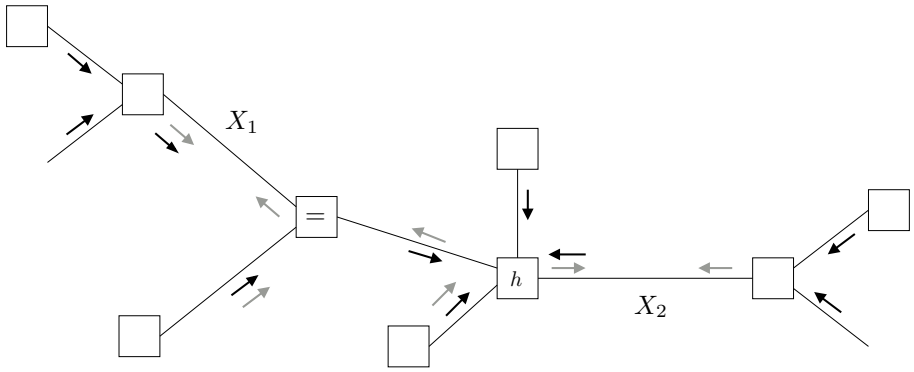


Figure 5: Steepest descent as summary propagation; the arrows represent the messages required for estimating the variables X_1 and X_2 by steepest descent. Some message are standard sum-product messages (black), others are *gradients* of sum-product messages (gray). All node functions are assumed to be differentiable except for the equality constraint node and the node h, which is supposed to correspond to a deterministic mapping.

3 Expectation maximization and steepest descent

In this section, we show how the maximization step in the EM-algorithm can be performed by steepest descent. We start from the exposition in [5], in which is shown how the EM algorithm can be viewed as message passing on factor graphs (see also [7]). Consider the factorization

$$f(x,\theta) \triangleq f_A(\theta) f_B(x,\theta), \tag{28}$$

which is represented by the factor graph of Fig. 6.

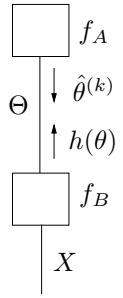


Figure 6: Factor graph of (28).

In this setup, EM amounts to iterative computation of the following messages [5]:
Upwards message $h(\theta)$:

$$h(\theta) = \frac{\int_x f_B(x, \hat{\theta}^{(k)}) \log f_B(x, \theta)}{\int_x \log f_B(x, \hat{\theta}^{(k)})} \tag{29}$$

Downwards message $\hat{\theta}^{(k)}$:

$$\hat{\theta}^{(k)} = \underset{\theta}{\text{argmax}}(\log f_A(\theta) + h(\theta)). \tag{30}$$

The computations (29) and (30) may be simplified when f_A and f_B have "nice" factorizations. Nevertheless, the maximization step (30) may still be intractable. We now show by an example, how in such situations gradient descent can be applied. Let

$$f_A(\theta) \triangleq f_{A_1}(\theta_1) f_{A_2}(\theta_1, \theta_2) \dots f_{A_n}(\theta_{n-1}, \theta_n), \tag{31}$$

and

$$f_B(x, \theta) \triangleq f_{B_0}(x_0) f_{B_1}(x_0, x_1, y_1, \theta_1) f_{B_2}(x_1, x_2, y_2, \theta_2) \dots f_{B_n}(x_{n-1}, x_n, y_n, \theta_n), \tag{32}$$

as illustrated in Fig. 7. The term $h(\theta)$ is given by [5]

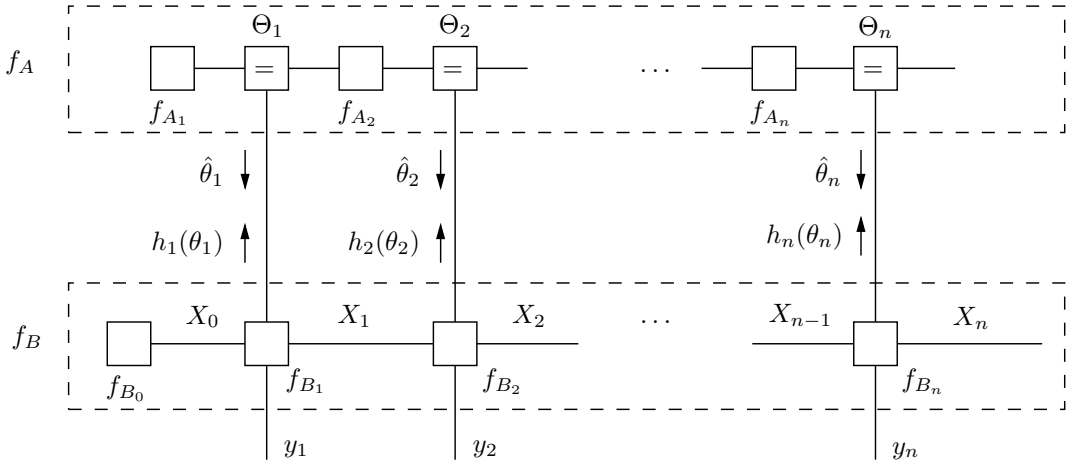


Figure 7: Factor graph of (31) and (32).

$$h(\theta) \quad = \quad \sum_{\ell=1}^{n} h_\ell(\theta_\ell), \tag{33}$$

where

$$h_\ell(\theta_\ell) \quad = \quad \int_{x_{\ell-1}} \int_{x_\ell} p_B(x_{\ell-1}, x_\ell, |y, \hat{\theta}) \log f_{B_\ell}(x_{\ell-1}, x_\ell, y, \theta_\ell), \tag{34}$$

and $p_B(x_{\ell-1}, x_\ell, |y, \hat{\theta})$ is the joint probability distribution of $X_{\ell-1}$ and X_ℓ (conditioned on y and $\hat{\theta}$); the latter can be computed from the sum-product messages $\mu_{X_\ell \to f_{B_\ell}}(x_\ell)$ and $\mu_{X_{\ell-1} \to f_{B_\ell}}(x_{\ell-1})$ as follows

$$p_B(x_{\ell-1}, x_\ell, |y, \hat{\theta}) = \frac{f_{B_\ell}(x_{\ell-1}, x_k, y, \theta_\ell) \mu_{X_\ell \to f_{B_\ell}}(x_\ell) \mu_{X_{\ell-1} \to f_{B_\ell}}(x_{\ell-1})}{\int_{x_{\ell-1}} \int_{x_\ell} f_{B_\ell}(x_{\ell-1}, x_\ell, y, \theta_k) \mu_{X_\ell \to f_{B_\ell}}(x_\ell) \mu_{X_{\ell-1} \to f_{B_\ell}}(x_{\ell-1})}. \tag{35}$$

The downward message $\hat{\theta}$ equals

$$(\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_n)^T = \underset{\theta_1,\theta_2,\ldots,\theta_n}{\operatorname{argmax}} \left[\log f_{A_1}(\theta_1) + \sum_{\ell=2}^{n} \log f_{A_\ell}(\theta_{\ell-1}, \theta_\ell) + \sum_{\ell=1}^{n} h_\ell(\theta_\ell) \right]. \tag{36}$$

The gradient $\nabla_\theta h(\theta) \triangleq \left(\nabla_{\theta_1} h(\theta), \ldots, \nabla_{\theta_n} h(\theta)\right)^T$ required for steepest descent is given by

$$\nabla_{\theta_\ell} h_\ell(\theta_\ell) = \nabla_{\theta_\ell} \left[\int_{x_{\ell-1}} \int_{x_\ell} p_B(x_{\ell-1}, x_\ell, |y, \hat{\theta}) \log f_{B_\ell}(x_{\ell-1}, x_\ell, y, \theta_\ell) \right], \tag{37}$$

$$= \int_{x_{\ell-1}} \int_{x_\ell} p_B(x_{\ell-1}, x_\ell, |y, \hat{\theta}) \nabla_{\theta_\ell} \log f_{B_\ell}(x_{\ell-1}, x_\ell, y, \theta_\ell) \tag{38}$$

$$= \frac{\int_{x_{\ell-1}} \int_{x_\ell} f_{B_\ell}(x_{\ell-1}, x_\ell, y, \hat{\theta}_\ell) \nabla_{\theta_\ell} \log f_{B_\ell}(x_{\ell-1}, x_\ell, y, \theta_\ell) \mu_{X_\ell \to f_{B_\ell}}(x_\ell) \mu_{X_{\ell-1} \to f_{B_\ell}}(x_{\ell-1})}{\int_{x_{\ell-1}} \int_{x_\ell} f_{B_\ell}(x_{\ell-1}, x_\ell, y, \hat{\theta}_\ell) \mu_{X_\ell \to f_{B_\ell}}(x_\ell) \mu_{X_{\ell-1} \to f_{B_\ell}}(x_{\ell-1})}. \tag{39}$$

Note that the rule (39) involves standard sum-product messages. Those messages may again be represented in different ways, such as lists of particles, quantized messages, Gaussian distributions etc. [3].

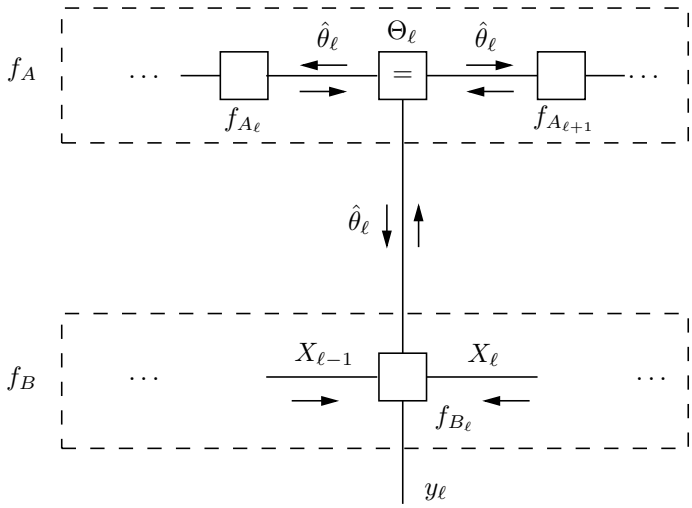


Figure 8: Steepest descent as summary propagation.

Expectation maximization, in which the M-step is performed by steepest descent, may then be formulated as follows (see Fig. 8):

1. The equality constraint nodes Θ_ℓ broadcast the estimates $\hat{\theta}_\ell^{(k)}$.
2. The nodes f_{A_ℓ} and $f_{A_{\ell+1}}$ reply with $\nabla_{\theta_\ell} \log f_{A_\ell}|_{\hat{\theta}^{(k)}}$ and $\nabla_{\theta_\ell} \log f_{A_{\ell+1}}\big|_{\hat{\theta}^{(k)}}$ respectively.
3. A forward and backward sum(mary)-product sweep is performed in the box f_B.
4. The nodes f_{B_ℓ} reply with $\nabla_{\theta_\ell} h|_{\hat{\theta}^{(k)}}$, computed according to (39).
5. The new estimate $\hat{\theta}^{(k+1)}$ is computed:

$$\hat{\theta}_\ell^{(k+1)} = \hat{\theta}_\ell^{(k)} + \lambda_k \left(\nabla_{\theta_\ell} \log f_{A_\ell}|_{\hat{\theta}^{(k)}} + \nabla_{\theta_\ell} \log f_{A_{\ell+1}}\big|_{\hat{\theta}^{(k)}} + \nabla_{\theta_\ell} h|_{\hat{\theta}^{(k)}} \right). \tag{40}$$

6. Iterate 1–5.

As usual, several update schedules are possible. For example, Step 3 does not need to be carried out at each iteration. Moreover, forward-only message passing amounts to recursive algorithms, known as "recursive EM" or "online EM" [8] – [12]. In [8] [9], recursive algorithms for fixed parameter estimation are derived based on EM in conjunction with steepest descent. Those algorithms are used in [10] for online estimation of the parameters of hidden Markov models, which are models in which f_B is of the form (32). It is common practice to extend the algorithms of [8] – [10] to *time-varying* parameters by introducing some ad-hoc "forgetting" mechanism. We illustrated by the example (31)–(32), how parameters with non-trivial priors can be treated in a rigorous way (see also [5] [7] [11] [12]).
The example (31)–(32) can easily be extended to general functions f_A and f_B. The gradient of the h-message leaving the generic node $g(z_1, z_2, \ldots, z_n, \theta_\ell)$ (cf. Fig. 9) is given by

$$\nabla_{\theta_m} h(\theta_m) = \frac{\int_{z_1,\ldots,z_n} g(z_1, z_2, \ldots, z_n, \hat{\theta}_m) \nabla_{\theta_m} \log g(z_1, z_2, \ldots, z_n, \theta_m) \prod_{\ell=1}^{n} \mu_{Z_\ell \to \theta_m}(z_\ell)}{\int_{z_1,\ldots,z_n} g(z_1, z_2, \ldots, z_n, \hat{\theta}_m) \prod_{\ell=1}^{n} \mu_{Z_\ell \to \theta_m}(z_\ell)}. \tag{41}$$

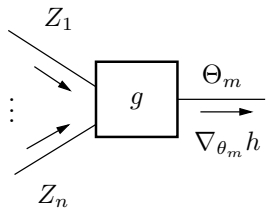


Figure 9: Generic node g.

An illustration of the above procedure can be found in [13], where it is applied to the problem of code-aided phase estimation.

4 Conclusion

We elaborated on previous work about steepest descent in the context of the sum(mary) product algorithm; we have shown in more detail how steepest descent can be used for estimation on factor graphs. Within the setting of summary propagation on factor graphs, it becomes clear how steepest descent can be combined with other powerful algorithms, such as Kalman filters/smoothers, the sum-product algorithm, expectation maximization and particle filters. Such an approach is for example useful in the context of iterative signal processing such as code-aided channel estimation.

References

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society,* B 39, pp. 1–38, 1977.

[2] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Proc. Mag.,* Jan. 2004, pp. 28–41.

[3] H.-A. Loeliger, "Some remarks on factor graphs", *Proc. 3rd International Symposium on Turbo Codes and Related Topics*, 1–5 Sept., 2003.

[4] D. P. Bertsekas, *Nonlinear programming,* Athena Scientific, Belmont, MA, 1995.

[5] J. Dauwels, S. Korl, and H.-A. Loeliger, "Expectation maximization on factor graphs", submitted to *IEEE Int. Symp. Information Theory, 2005.* Available online at `http://www.dauwels.com/files/EM_FG.pdf`.

[6] J. Dauwels, and H.-A. Loeliger, "Phase estimation by message passing", *Proc. 2004 IEEE Int. Conf. on Communications,* June 20–24, Paris, France, pp. 523–527, 2004.

[7] A. W. Eckford and S. Pasupathy, "Iterative multiuser detection with graphical modeling" *IEEE International Conference on Personal Wireless Communications,* Hyderabad, India, 2000.

[8] D. M. Titterington, "Recursive parameter estimation using incomplete data," R. Roy. Stat. Soc., vol. 46(B), pp. 256–267, 1984.

[9] E. Weinstein, M. Feder, A. V. Oppenheim, "Sequential algorithms for parameter estimation based on the Kullback-Leibler information measure," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 38, Sept. 1990, pp. 1652-1654.

[10] V. Krishnamurthy, J.B. Moore, "On-line estimation of hidden Markov model parameters based on the Kullback-Leibler information measure," *IEEE Transactions on Signal Processing,* vol. 41, Aug. 1993, pp. 2557–2573.

[11] H. Zamiri-Jafarian and S. Pasupathy, "EM-based recursive estimation of channel parameters," *IEEE Transactions on Communications,* vol. 47, Sept. 1999, pp. 1297–1302.

[12] L. Frenkel and M. Feder, "Recursive estimate-maximize (EM) algorithms for time varying parameters with applications to multiple target tracking," *IEEE International Conference on Acoustics, Speech, and Signal Processing,* vol. 3, 9–12 May, 1995, pp. 2068–2071.

[13] J. Dauwels, S. Korl, and H.-A. Loeliger, "Expectation maximization for phase estimation," Draft in preparation for the *Eighth International Symposium on Communication Theory and Applications (ISCTA'05).* Available online at `http://www.dauwels.com/files/EM_phase.pdf`.