From Belief Propagation to Expectation Propagation

Kevin P. Murphy

6 September 2001

1 Introduction

Expectation Propagation (EP) [Min01b, Min01a] is like Belief Propagation (BP) [YFW01], except that the messages it sends only contain expectations of features, instead of full (marginal) belief states. Hence EP is applicable to problems in which a full (posterior) distribution over a single variable cannot be represented exactly or compactly.

For example, a Gaussian prior combined with n MOG (mixture of Gaussians) likelihoods yields an MOG posterior with $O(2^n)$ modes. We may approximate the posterior with a single Gaussian (moment matching). In Assumed Density Filtering (ADF), we sequentially incorporate each likelihood and project the posterior down to a Gaussian (see Figure 1). Alternatively, we may think of this as simplifying the likelihood term t and then doing an exact Bayesian update. In this case, we may process the data in any order. In EP we are allowed to "go back" and revise our likelihood term approximations in an iterative fashion. In practice, this helps significantly: EP is much more accurate than ADF, and only takes k times more, where k is the number of iterations. Also, Minka showed that EP usually outperforms other deterministic approximation schemes, such as Laplace's method or Variational Bayes.

In this paper, we give an in-depth tutorial on EP, filling in some of the algebraic details omitted from [Min01b].

2 Algorithm

We derive the EP algorithm by a series of transformations of the BP algorithm, as formulated for factor graphs [KFL01].

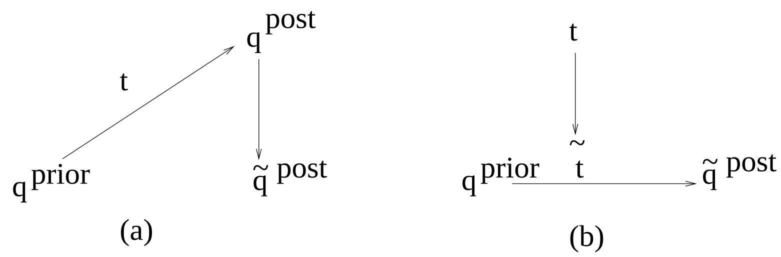

Figure 1: (a) Update then project, (b) Project then update.

In factor graphs, the two key equations are the messages sent from a variable x to a local factor f:

$$\mu_{x \to f}(x) = \prod_{g \in \mathrm{dep}(x) \setminus \{f\}} \mu_{g \to x}(x)$$

and from a factor to a variable

$$\mu_{f \to x}(x) = \sum_{U \setminus \{x\}} f(U) \prod_{y \in U \setminus \{x\}} \mu_{y \to f}(y)$$

where U is the domain of f and $\mathrm{dep}(x)$ are the factors that depend on x.

We can represent these messages as a vector of discrete (possibly unnormalized) probabilities, or as the mean and covariance of a Gaussian random vector, etc. We assume all messages have the same form so they can be combined easily.

Let us define the unnormalized belief at a variable to be $b'(x) = \prod_{f \in \mathrm{dep}(x)} \mu_{f \to x}(x)$. The normalized version of this is $b(x)$. Similarly, let us define $b'_f(u) = \prod_{x \in \mathrm{dom}(f)} \mu_{x \to f}(x)$.

In this case, the product over all-but-one of the messages can be implemented by dividing the belief by the message to be excluded:

$$\mu^{new}_{x \to f}(x) = \frac{b'(x)}{\mu_{f \to x}(x)}$$

For discrete vectors, we define division using $0/0 = 0$. For Gaussian messages, we can define division as in the junction tree algorithm [Mur98] by using the canonical parameterization, i.e., using the precision matrix.

One implementation is as follows. All variables send messages in parallel, then all factors, then all variables absorb in parallel, then all factors. This then repeats. (This is how BP for factor graphs is implemented in BNT.)

initialise messages and beliefs
repeat
    for each x
        for each $f \in \mathrm{dep}(x)$
            $\mu_{x \to f}(x) = b'(x)/\mu_{f \to x}(x)$
    for each f
        for each $x \in \mathrm{dom}(f)$
            $temp_f(u) = b'_f(u)/\mu^{old}_{x \to f}(x)$
            $\mu_{f \to x}(x) = \sum_{u \setminus \{x\}} temp(u) \times f(u)$
    for each x
        $b'(x) = \prod_f \mu_{f \to x}(x)$
    for each f
        $b'_f(u) = \prod_x \mu_{x \to f}(x)$
until messages converged

An alternative implementation is as follows. For each variable x, we update our belief on x one factor at a time.

for each x
  for each $f \in \mathrm{dep}(x)$
    $\mu_{x \to f}(x) \propto b(x)/\mu_{f \to x}(x)$
    $temp_f(u) = b'_f(u)/\mu^{old}_{x \to f}(x)$
    $\mu_{f \to x}(x) = \sum_{u \backslash x} temp_f(u) \times f(u)$
    $b'_f(u) = temp_f(u) \times \mu_{x \to f}(x)$
    $Z = \mu_{x \to f}(x) \times \mu_{f \to x}(x)$
    $b(x) = \frac{\mu_{x \to f}(x) \times \mu_{f \to x}(x)}{Z}$

We see that the message $\mu_{f \to x}(x)$ is acting like a likelihood term, and the message $b_{old}(x) \stackrel{\text{def}}{=} \mu_{x \to f}(x)$ is like a prior. The term $b(x)$ is the posterior.

Often the result of combining the prior with the likelihood results in a complex posterior. Assumed Density Filtering (ADF) picks the best approximation to the posterior within some specified family. One can then derive what the equivalent message $\mu_{f \to x}(x)$ must have by solving the following equation:

$$b_{new}(x) = \frac{b_{old}(x)\mu_{f \to x}(x)}{Z} \tag{1}$$

This results in the following algorithm, which is EP.

for each x
  for each $f \in \mathrm{dep}(x)$
    $\mu_{x \to f}(x) = b_{old}(x) \propto b(x)/\mu_{f \to x}(x)$
    $temp_f(u) = b'_f(u)/\mu^{old}_{x \to f}(x)$
    $b'_f(u) = temp_f(u) \times \mu_{x \to f}(x)$
    $t_f(x) = \sum_{u \backslash x} temp_f(u) \times f(u)$
    $[b(x), Z] = ADF(b_{old}(x), t_f(x))$
    $\mu_{f \to x}(x) = \frac{b(x)Z}{b_{old}(x)}$

If there is a single variable (as Minka typically assumes), this simplifies to

for each f
  $b_{old}(x) \propto b(x)/\mu_{f \to x}(x)$
  $[b(x), Z] = ADF(b_{old}(x), f(x))$
  $\mu_{f \to x}(x) = \frac{b(x)Z}{b_{old}(x)}$

3 ADF

At this point, let us switch notation to match Minka. Let $t_i(\theta) = f(x)$, the i'th likelihood term; $\tilde{t}_i(\theta) = \mu_{f \to x}(x)$, an approximation to the likelihood, which will be inferred from the approximate posterior. We assume $\theta \in \mathbb{R}^d$.

The exact posterior is given by

$$q_{\text{post}}(\theta) = \frac{q_{\text{prior}}(\theta) t_i(\theta)}{Z_i}$$

where

$$Z_i = \int_\theta q_{\text{prior}}(\theta) t_i(\theta)$$

In ADF, we compute $\tilde{q}_{\text{post}}(\theta)$ to minimize $D(q_{\text{post}}(\theta)||\tilde{q}_{\text{post}}(\theta))$ such that $\tilde{q}_{\text{post}}(\theta)$ is in the required family. This is just maximum likelihood estimation, where the data has distribution q_{post}.

3.1 Gaussian posterior

Let $q_{\text{post}}(\theta)$ be the exact posterior and $\tilde{q}_{\text{post}}(\theta)$ be the approximate posterior. Minka (p14) claims that the best approximate spherical Gaussian posterior of the form

$$\tilde{q}_{\text{post}}(\theta) = N(\theta; \tilde{m}_{\text{post}}, \tilde{v}_{\text{post}} I) = \frac{1}{(2\pi \tilde{v}_{\text{post}})^{d/2}} \exp\left(\frac{-1}{2\tilde{v}_{\text{post}}} (\theta - \tilde{m}_{\text{post}})^T (\theta - \tilde{m}_{\text{post}}) \right)$$

satisfies

$$\begin{aligned} \tilde{m}_{\text{post}} &= \int_\theta q_{\text{post}}(\theta)\theta \\ \tilde{v}_{\text{post}} d + \tilde{m}_{\text{post}}{}^T \tilde{m}_{\text{post}} &= \int_\theta q_{\text{post}}(\theta)\theta^T\theta \end{aligned}$$

or in other words, the expectation constraints

$$\begin{aligned} E_{\tilde{q}_{\text{post}}}[\theta] &= E_{q_{\text{post}}}[\theta] \\ E_{\tilde{q}_{\text{post}}}[\theta^T\theta] &= E_{q_{\text{post}}}[\theta^T\theta] \end{aligned}$$

since $\text{Var}[\theta] = E[\theta^T\theta] - [E\theta]^T[E\theta]$.

Proof:

To simplify notation in the proof, let $p(\theta) = q_{\text{post}}$, $q(\theta) = \tilde{q}_{\text{post}}$, $m = \tilde{m}_{\text{post}}$ and $v = \tilde{v}_{\text{post}}$. c, c' are constants independent of m, v.

$$\begin{aligned} D(p(\theta)||q(\theta)) &= \int_\theta p(\theta) \log q(\theta) + \text{c} \\ &= \int_\theta p(\theta) \left(-(d/2)\log v - \frac{1}{2v}(\theta - m)^T(\theta - m) \right) + \text{c'} \end{aligned}$$

$$\frac{\partial}{\partial m} D(p(\theta)||q(\theta)) = 0 - \int_\theta p(\theta) \frac{1}{2v}(\theta - m)$$

Setting to 0 yields

$$\int_\theta p(\theta)\theta = \int_\theta p(\theta)m = m$$

$$\frac{\partial}{\partial v} D(p(\theta)||q(\theta)) = \frac{-d}{2} \cdot \frac{1}{v} + \int_\theta p(\theta)\frac{1}{2}v^{-2}(\theta - m)^T(\theta - m)$$

Setting to 0 yields

$$\begin{aligned} dv &= \int_\theta p(\theta)\theta^T\theta + m^T m \int_\theta p(\theta) - 2m^T \int_\theta p(\theta)\theta \\ &= \int_\theta p(\theta)\theta^T\theta + m^T m - 2m^T m \end{aligned}$$

∎

On p15, Minka states how to compute the expectations wrt q_{post} as follows. (To simplify notation in this section, let $m = m_{\text{prior}}$ and $v = v_{\text{prior}}$. Also, let $\nabla_m = \nabla_m \log Z(m, v)$ and $\nabla_v = \nabla_v \log Z(m, v)$, where no ambiguity can arise.)

$$\tilde{m}_{\text{post}} = E_{q_{\text{post}}}[\theta] = m + v\nabla_m$$

$$d\tilde{v}_{\text{post}} = E_{q_{\text{post}}}[\theta^T\theta] - E_{q_{\text{post}}}[\theta]^T E_{q_{\text{post}}}[\theta] = vd - v^2\left(\nabla_m^T \nabla_m - 2\nabla_v\right)$$

On p45, Minka states the extensions of these formulae to the case when the approximate prior/posterior is non-spherical:

$$\begin{aligned} \tilde{m}_{\text{post}} &= E_{q_{\text{post}}}[\theta] = m + V\nabla_m \\ \tilde{V}_{\text{post}} &= E_{q_{\text{post}}}[\theta\theta^T] - [E_{q_{\text{post}}}\theta][E_{q_{\text{post}}}\theta]^T = V - V\left(\nabla_m \nabla_m^T - 2\nabla_v\right)V \end{aligned}$$

Note that these relations are a property of the Gaussian distribution, and hold for any $t(\theta)$.

Proof:

We restrict the proof to the spherical case for simplicity.

We will use the following identity in several places. Let $c \stackrel{\text{def}}{=} \frac{-1}{2v}$.

$$\begin{aligned} \nabla_m N(\theta; m, vI) &= \frac{1}{(2\pi v)^{d/2}} \nabla_m e^{c(\theta-m)^T(\theta-m)} \\ &= \frac{1}{(2\pi v)^{d/2}} \times -2c(\theta - m) \times e^{c(\theta-m)^T(\theta-m)} \\ &= \frac{(\theta - m)}{v} N(\theta; m, vI) \end{aligned} \tag{2}$$

Let

$$\begin{aligned}
Z(m, v) &\stackrel{\text{def}}{=} \int_\theta t(\theta) q_{\text{prior}}(\theta) \\
&= \int_\theta t(\theta) N(\theta; m, vI)
\end{aligned}$$

Hence

$$\begin{aligned}
\nabla_m \log Z(m, v) &= \frac{1}{Z} \int_\theta t(\theta) \nabla_m N(\theta; m, vI) \\
&= \frac{1}{Z} \int_\theta t(\theta) \frac{(\theta - m)}{v} q_{\text{prior}}(\theta) \\
&= \frac{1}{Zv} \int_\theta \theta t(\theta) q_{\text{prior}}(\theta) - \frac{m}{Zv} \int_\theta t(\theta) q_{\text{prior}}(\theta) \\
&= \frac{E_{q_{\text{post}}}[\theta]}{v} - \frac{m}{v}
\end{aligned}$$

This implies

$$\tilde{m}_{\text{post}} = E_{q_{\text{post}}}[\theta] = m + v \nabla_m \log Z(m, v) \tag{3}$$

We will use the following identity in several places. Let $c_\theta \stackrel{\text{def}}{=} -\frac{1}{2}(\theta - m)^T(\theta - m)$.

$$\begin{aligned}
\nabla_v N(\theta; m, vI) &= \nabla_v \frac{1}{(2\pi)^{d/2}} v^{-d/2} e^{c_\theta v^{-1}} \\
&= \frac{1}{(2\pi)^{d/2}} \left(\frac{-d}{2} v^{-d/2-1} e^{c_\theta v^{-1}} + v^{-d/2} c_\theta \cdot -v^{-2} \cdot e^{c_\theta v^{-1}} \right) \\
&= \frac{-d}{2v} \frac{e^{c_\theta v^{-1}}}{(2\pi v)^{d/2}} - \frac{c_\theta}{v^2} \frac{e^{c_\theta v^{-1}}}{(2\pi v)^{d/2}} \\
&= \frac{-d}{2v} N(\theta; m, vI) + \frac{1}{2v^2} (\theta - m)^T (\theta - m) N(\theta; m, vI) \qquad (4)
\end{aligned}$$

Hence

$$\begin{aligned}
\nabla_v \log Z(m, v) &= \frac{1}{Z} \int_\theta \nabla_v t(\theta) N(\theta; m, v) \\
&= \frac{-d}{2vZ} \int_\theta t(\theta) q_{\text{prior}}(\theta) + \frac{1}{2v^2 Z} \int_\theta t(\theta) q_{\text{prior}}(\theta) (\theta - m)^T (\theta - m) \\
&= \frac{-d}{2v} + \frac{1}{2v^2} E_{q_{\text{post}}}[(\theta - m)^T (\theta - m)]
\end{aligned}$$

This implies

$$E_{q_{\text{post}}}[\theta^T \theta] = 2v^2 \nabla_v + dv + m^T m + 2v m^T \nabla_m$$

so

$$d\tilde{v}_{\text{post}} = Eq_{\text{post}}[\theta^T\theta] - Eq_{\text{post}}[\theta]^T Eq_{\text{post}}[\theta] = vd - v^2\left(\nabla_m^T \nabla m - 2\nabla v\right) \quad (5)$$

∎

3.2 Mixture of Gaussians likelihood term

Let us consider the problem of inferring the position of a target where what we see is either a noisy observation of the position, or an outlier (Minka p13). Let θ be the unknown position with prior $N(m_{\text{prior}}, v_{\text{prior}})$ and x be an observation. The likelihood function is

$$t(\theta) = p(x|\theta) = (1-w)N(x;\theta,I) + wN(x;0,10I)$$

The second Gaussian approximates a uniform distribution centered around the origin.

Letting $m = m_{\text{prior}}$ and $v = v_{\text{prior}}$, we have (p15)

$$\begin{aligned} Z(m,v) &= \int_\theta t(\theta) q_{\text{prior}}(\theta) \\ &= (1-w)\int_\theta N(x;\theta,I)N(\theta;m,v) + w\int_\theta N(x;0,10I)N(\theta;m,v) \\ &= (1-w)N(x;m,(v+1)I) + wN(x;0,10I) \end{aligned} \quad (6)$$

Proof:
Consider the following linear dynamical system without observations

$$X_t = AX_{t-1} + W_t$$

where $W_t \sim N(0,W)$ and $X_{t-1} \sim N(\mu,\Sigma)$. Then

$$E[X_t] = AE[X_{t-1}] = A\mu$$

and

$$\text{Cov}[X_t] = A\text{Cov}[X_{t-1}]A^T + W = A\Sigma A^T + W$$

Hence

$$\int_{X_{t-1}} N(X_t; AX_{t-1}, W)N(X_{t-1};\mu,\Sigma) = N(X_t; A\mu, A\Sigma A^T + W)$$

Substituting $\mu = m$, $A = I$, $\Sigma = vI$ and $W = I$ yields the result.

Using Equation 2, we have

$$
\begin{aligned}
\nabla_m \log Z &= \frac{1}{Z}\nabla_m (1-w)N(x;m,(v+1)I) \\
&= \frac{1-w}{Z}\frac{x-m}{v+1}N(x;m,(v+1)I) \\
&= r\frac{x-m}{v+1} \qquad (7)
\end{aligned}
$$

where we have introduced

$$
r \stackrel{\text{def}}{=} \frac{(1-w)N(x;m,(v+1)I)}{Z}
$$

Using Equation 4, we have

$$
\begin{aligned}
\nabla_v \log Z &= \frac{1}{Z}\nabla_v (1-w)N(x;m,(v+1)I) \\
&= \frac{1-w}{Z}\left(\frac{-d}{2(v+1)}N(x;m,(v+1)I) + \frac{(x-m)^T(x-m)}{2(v+1)^2}N(x;m,(v+1)I)\right) \\
&= -\frac{rd}{2(v+1)} + \frac{r(x-m)^T(x-m)}{2(v+1)^2} \qquad (8)
\end{aligned}
$$

Hence after incorporating the i'th data point x_i into the prior $N(\theta;m,v)$, the new posterior is computed as follows. Substituting Equation 7 into 3 yields

$$
\tilde{m}_{\text{post}} = m + v_{\text{prior}}\nabla_m = m + vr_i\frac{x_i - m}{v+1} \qquad (9)
$$

Substituting Equation 8 into 5 yields

$$
\begin{aligned}
d\tilde{v}_{\text{post}} &= vd - v^2\left(\nabla_m^T\nabla_m - 2\nabla_v\right) \\
&= vd - v^2\left(r_i^2\frac{(x_i-m)^T(x_i-m)}{(v+1)^2} + \frac{2r_i d}{2(v+1)} - \frac{2r_i(x_i-m)^T(x_i-m)}{2(v+1)^2}\right) \\
&= vd - \frac{v^2 r_i d}{v+1} - r_i(1-r_i)\frac{v^2(x_i-m)^T(x_i-m)}{(v+1)^2} \qquad (10)
\end{aligned}
$$

The final ADF algorithm is therefore

1. Initialize prior $m_{\text{prior}} = 0$, $v_{\text{prior}} = 100$.
2. For each data point x_i, compute the new posterior using Equations 9 and 10.

Minka gives the following intuitive interpretation of this algorithm: for each data point, we compute its probability r_i of not being clutter, make a soft update to our estimate m, and change our confidence v in the estimate.

It is clear that this algorithm will depend on the order in which the data is processed. This limitation is overcome in EP.

The final EP algorithm in this case is as follows.

1. The term approximations (messages from factors to the variable) have the form

$$\tilde{t}_i(\theta) = s_i \exp(-\frac{1}{2v_i}(\theta - m_i)^T(\theta - m_i))$$

   Initialise all messages to $(m_i, v_i, s_i) = (0, \infty, 1)$.

2. Initialise the belief to the prior $(\tilde{m}_{\text{post}}, \tilde{v}_{\text{post}}, \tilde{s}_{\text{post}}) = (0, 100, (2\pi\tilde{v}_{\text{post}})^{-d/2})$.

3. Until all messages (m_i, v_i, s_i) converge:
   for $i = 1, \ldots, n$

   (a) $[m_{\text{prior}}, v_{\text{prior}}, s_{\text{prior}}] = \text{divide}([\tilde{m}_{\text{post}}, \tilde{v}_{\text{post}}, \tilde{s}_{\text{post}}], [m_i, v_i, s_i])$

   (b) $[\tilde{m}_{\text{post}}, \tilde{v}_{\text{post}}, \tilde{s}_{\text{post}}, Z] = \text{ADF-MOG}([m_{\text{prior}}, v_{\text{prior}}, s_{\text{prior}}], x_i)$

   (c) $temp = \tilde{s}_{\text{post}} \times Z$

   (d) $[m_i, v_i, s_i] = \text{divide}([\tilde{m}_{\text{post}}, \tilde{v}_{\text{post}}, temp], [m_{\text{prior}}, v_{\text{prior}}, s_{\text{prior}}])$

The ADF-MOG step is implemented by equations 9, 10 and 6. Note that Z is the normalization constant for the exact posterior, and is different from $\tilde{s}_{\text{post}}$.

References

[KFL01] F. Kschischang, B. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans Info. Theory*, February 2001.

[Min01a] T. Minka. Expectation propagation for approximate Bayesian inference. In *UAI*, 2001.

[Min01b] T. Minka. *A family of algorithms for approximate Bayesian inference.* PhD thesis, MIT, 2001.

[Mur98] K. P. Murphy. Inference and learning in hybrid Bayesian networks. Technical Report 990, U.C. Berkeley, Dept. Comp. Sci, 1998.

[YFW01] J. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *IJCAI*, 2001.