6.975 Week 11 Summary: Expectation Propagation

Erik Sudderth

November 20, 2002

1 Introduction

Expectation Propagation (EP) [2, 3] addresses the problem of constructing tractable approximations to complex probability distributions. Let x be a set of random variables of interest, and $p(x)$ be a distribution formed from the product of several "compatibility functions":

$$p(x) \propto \prod_i \psi_i(x) \tag{1}$$

EP is an iterative algorithm which attempts to choose the best approximation to $p(x)$ from within some tractable class of distributions. To ensure that each iteration of EP is computationally feasible, we choose the approximating class to correspond to an exponential family $q(x;\theta)$.

Distributions $p(x)$ of the form shown in equation (1) arise in a huge range of applications. For example, suppose that we have n independent observations y_i of an unobserved random variable x with prior distribution $p_0(x)$. By Bayes' rule, the posterior distribution over x is then given by

$$p\left(x \mid y_1, \ldots, y_n\right) = p_0(x) \prod_{i=1}^{n} p_i(y_i|x) \tag{2}$$

Similarly, the prior or posterior distribution corresponding to any graphical model may be written as in equation (1). For example, a pairwise Markov random field with nodes $\mathcal{V}$, edges $\mathcal{E}$, hidden variables x_s, and local observations y_s has posterior distribution

$$p\left(x \mid y_1, \ldots, y_n\right) = \prod_{s \in \mathcal{V}} \psi_s\left(x_s, y_s\right) \prod_{(s,t) \in \mathcal{E}} \psi_{s,t}\left(x_s, x_t\right) \tag{3}$$

In the remainder of this summary, we focus on inference problems defined by some fixed set of observations y. Thus, rather than explicitly specifying the observations, we assume that the target distribution is written as in equation (1).

2 Exponential Families

An exponential family of distributions $q(x;\theta)$ can be written as

$$q(x;\theta) = \exp\left\{\sum_{\alpha} \theta_\alpha \phi_\alpha(x) - \Phi(\theta)\right\} \tag{4}$$

The parameter vector θ indexes the distributions in the family, each of which corresponds to a different weighting of the *potential functions* $\phi_\alpha(x)$. The *log partition function* $\Phi(\theta)$ ensures that $q(x;\theta)$ is properly normalized for any choice of parameters θ. A wide range of classic distributions, including Gaussian, Poisson, and discrete multinomial, may be written in exponential form. Note that when x is a continuous variable, normalization may only be possible for certain choices of θ (e.g. Gaussian distributions must have nonnegative variance).

Exponential families have a number of properties which simplify standard computations. Expectation Propagation makes extensive use of two of these features. First, if we multiply or divide two exponential distributions, we produce a new distribution which is a member of the same exponential family (although normalizability may be lost). The coefficients of the product or quotient distribution are equal to the sum or difference of the input coefficients.

Second, consider the problem of approximating some arbitrary distribution $p(x)$ with a member of an exponential family. We choose the best approximation by minimizing the following Kullback–Leibler divergence:

$$\theta^* = \arg\min_{\theta} D\left(p(x) \mid\mid q(x;\theta)\right) \tag{5}$$

The optimal solution to this problem is given by moment matching. In particular, θ^* should be chosen so that

$$\int q(x;\theta^*)\phi_\alpha(x)\,dx = \int p(x)\phi_\alpha(x)\,dx \tag{6}$$

for all potential functions $\phi_\alpha(x)$ in the exponential family. For many commonly used exponential families, the mapping between moment and exponential parameterizations is easily computed. Thus, in cases where it is tractable to compute moments of $p(x)$, the minimization of equation (5) has a closed form solution.

3 Assumed Density Filtering

Expectation Propagation can be seen as a method for iteratively refining the solution produced by classic *Assumed Density Filtering* (ADF) methods, such as the extended Kalman filter. Consider the factorized density $p(x)$ of equation (1). ADF begins by choosing $q(x;\theta^1)$ to best approximate the first compatibility function $\psi_1(x)$ according to equation (5). We then proceed through the remaining compatibility functions in order, updating the approximation to $p(x)$ as

$$\theta^i = \arg\min_{\theta} D\left(\psi_i(x)q(x;\theta^{i-1}) \mid\mid q(x;\theta)\right) \tag{7}$$

At each ADF iteration, the current best estimate $q(x;\theta^{i-1})$ of the product distribution is used to guide the incorporation of the next compatibility function $\psi_i(x)$. While this is preferable to constructing independent approximations to each term, it has the undesirable property that the ADF estimate is sensitive to the order in which the compatibility terms are processed. In particular, if the first few terms are “misleading”, so that their product is very different from the true product $p(x)$, ADF may produce extremely poor approximations.

In the previous paragraph, we described ADF as iteratively refining the approximate posterior distribution $q(x;\theta)$. Alternatively, however, each ADF iteration can be seen as first approximating the compatibility function $\psi_i(x)$ by a member of the exponential family $m_i(x)$, and then *exactly* updating the posterior distribution:

$$q(x;\theta^i) = m_i(x)q(x;\theta^{i-1}) \qquad\qquad m_i(x) \propto \frac{q(x;\theta^i)}{q(x;\theta^{i-1})} \tag{8}$$

Note that each $m_i(x)$ is in the same exponential family as $q(x;\theta)$. In the following section, we show how this alternate interpretation naturally leads to the EP algorithm.

4 Expectation Propagation

Consider the ADF algorithm of the previous section. The best approximation $m_i(x)$ to a particular compatibility function $\psi_i(x)$ would be produced by directly minimizing

$$D\left(p(x) \mid\mid m_i(x) \prod_{j\neq i} \psi_j(x)\right) \tag{9}$$

However, because direct computations with $p(x)$ are intractable, ADF must neglect most of the product terms in computing its approximations to the first processed compatibility functions. In standard ADF, these initial approximations are never revisited, and therefore errors may significantly bias the final approximation.

The EP algorithm exploits the interpretation of ADF as approximating compatibility functions (as in equation (8)) to revisit each term approximation multiple times. At later iterations, EP uses its current best estimates of all but one compatibility to improve the exponential approximation to the remaining term. One hopes that by iterating this procedure, EP will converge to a fixed point which approximates $p(x)$ better than the results of any particular ADF ordering.

We initialize the EP algorithm by setting the compatibility approximations to some default values, typically $m_i(x) = 1$. The posterior approximation to $p(x)$ is initialized as $q(x;\theta) \propto \prod_i m_i(x)$. Each iteration of EP then proceeds as follows:

1. Choose some $m_i(x)$ to refine.
2. Remove the effects of $m_i(x)$ from the current posterior estimate $q(x;\theta)$ by dividing and normalizing:
$$q(x;\theta^{\backslash i}) \propto \frac{q(x;\theta)}{m_i(x)} \tag{10}$$
3. Update the exponential approximation to the posterior as $q(x;\theta^*)$, where θ^* is found by computing the projection
$$\theta^* = \arg\min_{\theta} D\left(q(x;\theta^{\backslash i})\psi_i(x) \mid\mid q(x;\theta)\right) \tag{11}$$
4. Refine the exponential approximation to $m_i(x)$ as
$$m_i(x) \propto \frac{q(x;\theta^*)}{q(x;\theta^{\backslash i})} \tag{12}$$

Note that the computational tractability of the EP iteration depends heavily on the properties of exponential families introduced in Section 2.

In many cases, EP produces posterior approximations which compare quite favorably with competing techniques such as ADF. However, there is no guarantee that the EP iteration will converge. In some cases, heuristics can be used to enforce convergence. For example, when approximating with a Gaussian exponential family, Minka improved convergence by constraining all variances to be positive [3]. However, there is currently no framework for understanding how such constraints should be chosen in general, or what effect they will have on the quality of the final approximations.

5 Relationship to Belief Propagation

The loopy Belief Propagation (BP) algorithm can be shown to be a special case of Expectation Propagation. This relationship provides three key benefits. First, it allows much of the recent work analyzing BP to be extended to EP. Second, it provides a mechanism for constructing improved approximations for models where standard BP performs poorly. Finally, EP provides a mechanism for extending BP–style message passing updates to continuous, non–Gaussian models (for which BP is generally intractable).

5.1 EP for Fully Factorized Discrete Approximations

Consider first a standard discrete pairwise Markov random field, with nodes $\mathcal{V}$, edges $\mathcal{E}$, and posterior distribution as in equation (3). For simplicity, we assume that all single–node potentials have been absorbed into one of the neighboring edge potentials, so that $p(x)$ can be rewritten as

$$p(x) = \prod_{(s,t)\in\mathcal{E}} \psi_{s,t}\left(x_s, x_t\right) \tag{13}$$

To derive an EP iteration corresponding to BP, we choose our approximating distribution $q(x;\theta)$ to be fully factorized:

$$q(x;\theta) = \prod_{s\in\mathcal{V}} q_s(x_s) \tag{14}$$

Each $q_s(x_s)$ is chosen to be a general discrete multinomial distribution, and hence places no restrictions on the marginal distributions which EP can represent. This factorization of $q(x;\theta)$ implies that the exponential approximations $m_{s,t}(x_s, x_t)$ to the compatibility functions $\psi_{s,t}\left(x_s, x_t\right)$ also have a factorized form:

$$m_{s,t}(x_s, x_t) = m_{t\to s}(x_s) m_{s\to t}(x_t) \tag{15}$$

As we show below, these marginal approximations to the compatibility functions precisely correspond to the standard BP messages.

As before, we initialize the compatibility approximations of equation (15) to some default values. Using the factorized form of $q(x;\theta)$, we may then individually initialize each of the terms in the factorized posterior approximation as

$$q_s(x_s) \propto \prod_{t \in \Gamma(s)} m_{t \to s}(x_s) \tag{16}$$

where $\Gamma(s)$ is the set of nodes neighboring s. An iteration of EP then proceeds as follows:

1. Choose some $m_{s,t}(x_s, x_t)$ to refine. Since $m_{s,t}(x_s, x_t)$ involves only x_s and x_t, the approximate distributions for all other nodes are unaffected by the EP update, and are thus neglected below.

2. Remove the effects of $m_{s,t}(x_s, x_t)$ from the current posterior estimate $q(x;\theta)$ by dividing and normalizing:

$$q_{s \backslash t}(x_s) \propto \frac{q_s(x_s)}{m_{t \to s}(x_s)} = \prod_{u \in \Gamma(s) \backslash t} m_{u \to s}(x_s) \tag{17}$$

$$q_{t \backslash s}(x_t) \propto \frac{q_t(x_t)}{m_{s \to t}(x_t)} = \prod_{v \in \Gamma(t) \backslash s} m_{v \to t}(x_t) \tag{18}$$

3. Update the exponential approximation to the posterior by determining the appropriate marginal distributions:

$$q_s(x_s) = \sum_{x_t} \psi_{s,t}\left(x_s, x_t\right) q_{s \backslash t}(x_s) q_{t \backslash s}(x_t) \tag{19}$$

$$q_t(x_t) = \sum_{x_s} \psi_{s,t}\left(x_s, x_t\right) q_{s \backslash t}(x_s) q_{t \backslash s}(x_t) \tag{20}$$

4. Refine the exponential approximation to $m_{s,t}(x_s, x_t)$ as

$$m_{t \to s}(x_s) \propto \frac{q_s(x_s)}{q_{s \setminus t}(x_s)} = \sum_{x_t} \psi_{s,t}\left(x_s, x_t\right) \prod_{v \in \Gamma(t) \setminus s} m_{v \to t}(x_t) \tag{21}$$

$$m_{s \to t}(x_t) \propto \frac{q_t(x_t)}{q_{t \setminus s}(x_t)} = \sum_{x_s} \psi_{s,t}\left(x_s, x_t\right) \prod_{u \in \Gamma(s) \setminus t} m_{u \to s}(x_s) \tag{22}$$

Notice that step 4 of the EP update is precisely equivalent to the standard BP message update of both messages along the chosen edge (s, t).

5.2 Free Energy Interpretation of EP

Much of the recent analysis of loopy BP has come from an association with the minimization of a certain "free energy", subject to marginalization constraints [5]. EP can be derived from a similar free energy, where the marginalization constraints are replaced by expectation constraints [1–3]. This connection has allowed much of the theoretical work on BP to be extended to EP. In particular, EP can be shown to have at least one fixed point for any product distribution. In addition, it is possible to construct "double loop" algorithms which provably minimize the free energy [1].

5.3 Higher Order Approximations

BP was derived by applying EP to a fully factorized approximation. It is possible to derive similar algorithms for approximations which are not fully factorized but still tractable, such as a spanning tree [3]. For a discussion of the connections between these higher–order approximations and region–based Kikuchi approximations [5], see Wainwright [4]. At this point, it is not clear which method of creating improved approximations will be more effective in practice.

5.4 Continuous Non–Gaussian Models

For continuous non–Gaussian graphical models, the integral equations prescribed by loopy BP typically have no closed form solutions. However, it is straightforward to apply EP to such models using, for example, a Gaussian approximation to the final posterior distributions. The only change is that the marginalization of step 3 must be supplemented by a KL projection operation; see [1].

References

[1] T. Heskes and O. Zoeter. Expectation propagation for approximate inference in dynamic Bayesian networks. In *Uncertainty in Artificial Intelligence 18*, pages 216–223, 2002.

[2] T. P. Minka. Expectation propagation for approximate Bayesian inference. In *Uncertainty in Artificial Intelligence 17*, pages 362–369, 2001.

[3] T. P. Minka. *A Family of Algorithms for Approximate Bayesian Inference.* PhD thesis, Massachusetts Institute of Technology, January 2001.

[4] M. J. Wainwright. *Stochastic Processes on Graphs with Cycles: Geometric and Variational Approaches.* PhD thesis, Massachusetts Institute of Technology, January 2002.

[5] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report 2002-35, MERL, August 2002.