Power EP

Thomas Minka
Microsoft Research Ltd., Cambridge, UK
MSR-TR-2004-149, October 4, 2004

Abstract

This note describes power EP, an extension of Expectation Propagation (EP) that makes the computations more tractable. In this way, power EP is applicable to a wide variety of models, much more than EP. Instead of minimizing KL-divergence at each step, power EP minimizes α-divergence. This minimization turns out to be equivalent to minimizing KL-divergence with the exact distribution raised to a power. By choosing this power to cancel exponents, the problem may be substantially simplified. The resulting approximation is not the same as regular EP, but in practice is still very good, and allows tackling problems which are intractable under regular EP.

1 Introduction

Expectation Propagation (EP) is a method for approximating a complex probability distribution p by a simpler distribution q (Minka, 2001b). To apply it, you write the distribution as a product of terms, $p(x) = \prod_a t_a(x)$, and then you approximate the terms one by one. At each step, you need to evaluate an integral of the form $\int_x t_a(x) q^{\backslash a}(x) dx$ (a definite integral). If the terms t_a are simple enough, this integral can be done quickly, usually analytically.

Unfortunately, there are many functions which cannot be factored into simple terms. For example, consider a product of T densities:

$$p(x) = \frac{1}{x^2+1}\frac{1}{x^2+1} \tag{1}$$

This function naturally factors into two terms, but the integrals $\int_x \frac{1}{x^2+1}\, q^{\backslash a}(x) dx$ are not easy to evaluate, so EP bogs down. On the other hand, the integrals $\int_x (x^2+1) q^{\backslash a}(x) dx$ are easy to evaluate. Using power EP, we can approximate p using only these easy integrals.

Power EP is an extension of EP to employ integrals of the form $\int_x t_a(x)^\beta q(x) dx$, for any powers β that you choose. For example, in the T density problem we can let the powers be $\beta = -1$, which leads to easy integrals. The rest of the algorithm is essentially the same as EP, with a scaling step to account for the powers.

Power EP was originally described by Minka & Lafferty (2002), briefly and without derivation. Later, Wiegerinck & Heskes (2002) discussed an algorithm called "fractional belief propagation", describing its energy function and an interpretation in terms of α-divergence. In fact, fractional belief propagation is a special case of power EP, and all of their results also apply to power EP. This paper expands on both Minka & Lafferty (2002) and Wiegerinck & Heskes (2002), giving a unified perspective on power EP and the different ways of implementing it.

The paper is organized as follows. Section 2 describes the algorithm in detail. Section 3 describes the α-divergence interpretation, and section 4 gives the objective function.

2 EP and power EP

A set of distributions is called an **exponential family** if it can be written as

$$q(x) \quad = \quad \exp(\sum_j g_j(x)\nu_j) \tag{2}$$

where ν_j are the parameters of the distribution and g_j are fixed features of the family, such as $(1, x, x^2)$ in the Gaussian case. Because we will be working with unnormalized distributions, we make 1 a feature, whose corresponding parameter captures the scale of the distribution. The reason to use exponential families is closure under multiplication: the product of distributions in the family is also in the family.

Given a distribution p and an exponential family $\mathcal{F}$, Expectation Propagation tries to find a distribution $q \in \mathcal{F}$ which is "close" to p in the sense of the **KL-divergence**:

$$KL(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx + \int (q(x) - p(x)) dx \tag{3}$$

Here we have written the divergence with a correction factor, so that it applies to unnormalized distributions. The basic operation in EP is **KL-projection**, defined as

$$\text{proj}[p] \quad = \quad \underset{q \in \mathcal{F}}{\text{argmin}}\, KL(p \;||\; q) \tag{4}$$

(In this notation, the identity of $\mathcal{F}$ comes from context.) Substituting the form of q (2) into the KL-divergence, we find that the minimum is the unique member of $\mathcal{F}$ whose expectation of g_j matches that of p, for all j:

$$q = \text{proj}[p] \iff \forall_j \int_x g_j q(x) dx = \int_x g_j p(x) dx \tag{5}$$

For example, if $\mathcal{F}$ is the set of Gaussians, then $\text{proj}[p]$ means computing the mean and variance of p, and returning a Gaussian with that mean and variance.

Now, write the original distribution p as a product of terms:

$$p(x) = \prod_a t_a(x) \tag{6}$$

This defines the specific way in which we want to divide the network, and is not unique. By approximating each term t_a by $\tilde{t}_a$, we get an approximation divided in the same way:

$$q(x) = \prod_a \tilde{t}_a(x) \tag{7}$$

For each term t_a, the algorithm first computes $q^{\backslash a}(x)$, which represents the "rest of the distribution." Then it minimizes KL-divergence over $\tilde{t}_a$, holding $q^{\backslash a}$ fixed. This process can be written succinctly as follows:

$$q^{\backslash a}(x) \quad = \quad q(x)/\tilde{t}_a(x) \tag{8}$$

$$\tilde{t}_a(x)^{new} \quad = \quad \text{argmin}\, KL(t_a(x)q^{\backslash a}(x) \;||\; \tilde{t}_a(x)q^{\backslash a}(x)) \tag{9}$$

$$= \quad \text{proj}\left[t_a(x)q^{\backslash a}(x)\right] / q^{\backslash a}(x) \tag{10}$$

In the Gaussian case, both $q(x)$ and $\tilde{t}_a(x)$ will be scaled Gaussians, so algorithmically (8) means "divide the Gaussians and their corresponding scale factors to get a new scaled Gaussian, and call it $q^{\backslash a}(x)$." Similarly, (9) means "construct a scaled Gaussian whose moments match $t_a(x)q^{\backslash a}(x)$ and divide it by $q^{\backslash a}(x)$, to get a new scaled Gaussian which replaces $\tilde{t}_a(x)$." This is the basic EP algorithm.

Power EP extends the algorithm to use the following update (for any desired n_a):

$$q^{\backslash a}(x) \quad = \quad q(x)/\tilde{t}_a(x)^{1/n_a} \tag{11}$$

$$\tilde{t}_a(x)^{new} \quad = \quad \left(\text{proj}\left[t_a(x)^{1/n_a} q^{\backslash a}(x)\right] / q^{\backslash a}(x)\right)^{n_a} \tag{12}$$

To see this another way, define the "fractional term" f_a:

$$f_a(x) = t_a(x)^{1/n_a} \qquad \tilde{f}_a(x) = \tilde{t}_a(x)^{1/n_a} \tag{13}$$

The update for $\tilde{f}$ is:

$$q^{\backslash fa}(x) = q(x)/\tilde{f}_a(x) \tag{14}$$

$$\tilde{f}_a(x)^{new} = \text{proj}\left[f_a(x)q^{\backslash fa}(x)\right]/q^{\backslash fa}(x) \tag{15}$$

This is exactly like EP, except we then scale $\tilde{f}$ in order to get $\tilde{t}$.

For example, suppose we want to approximate $p(x) = a(x)/b(x)$ with $q(x) = \tilde{a}(x)/\tilde{b}(x)$. For $a(x)$ we use $n = 1$, and for $b(x)$ we use $n = -1$. Then the updates are:

$$q^{\backslash a}(x) = \frac{1}{\tilde{b}(x)} \tag{16}$$

$$\tilde{a}(x)^{new} = \text{proj}\left[a(x)\, q^{\backslash a}(x)\right]/q^{\backslash a}(x) \tag{17}$$

$$q^{\backslash b}(x) = \frac{\tilde{a}(x)}{\tilde{b}(x)^2} \tag{18}$$

$$\tilde{b}(x)^{new} = \text{proj}\left[b(x)\, q^{\backslash b}(x)\right]/q^{\backslash b}(x) \tag{19}$$

Notice the old $\tilde{b}(x)$ is contained in $q^{\backslash b}(x)$, creating a feedback loop. By iterating these equations, we get the following algorithm for power EP:

Power EP

- Initialize $\tilde{f}_a(x)$ for all a, and set $q(x)$ to their product.
- Repeat until convergence:
  For each a:

$$q^{\backslash fa}(x) = q(x)/\tilde{f}_a(x) \tag{20}$$

$$q'(x) = \text{proj}\left[f_a(x)q^{\backslash fa}(x)\right] \tag{21}$$

$$\tilde{f}_a(x)^{new} = \tilde{f}_a(x)^{1-\gamma}\left(\frac{q'(x)}{q^{\backslash fa}(x)}\right)^{\gamma} = \tilde{f}_a(x)\left(\frac{q'(x)}{q(x)}\right)^{\gamma} \tag{22}$$

$$q(x)^{new} = q(x)\left(\frac{\tilde{f}_a(x)^{new}}{\tilde{f}_a(x)}\right)^{n_a} = q(x)\left(\frac{q'(x)}{q(x)}\right)^{n_a\gamma} \tag{23}$$

In (22), a damping factor γ has been used, which doesn't change the fixed points, but can help convergence. Choosing $\gamma = 1/n_a$ makes $q(x)^{new} = q'(x)$, which is convenient computationally and also tends to have good convergence properties.

To get an intuitive sense of what power EP is doing, consider the case of positive integer n_a. This means that one fractional term $f_a(x)$ is repeated, say, 3 times. Power EP approximates each copy with the same $\tilde{f}_a(x)$. To update it, power EP removes *one* copy of $f_a(x)$, computes a new $\tilde{f}_a(x)$ for it, and then replicates that approximation to all the copies. If we applied regular EP across all copies, then the approximations would not be synchronized, but at convergence they would be the same, yielding the same answer as power EP (this is shown in section 4). Of course, in the non-integer case this construction is not possible and there is no equivalent EP algorithm.

The above discussion has focused on EP, but in fact the mean-field method has a very similar structure. All you do is reverse the direction of the divergence in (9). This way of writing the mean-field updates is known as Variational Message Passing (Winn & Bishop, 2005).

3 Alpha-divergence

This section shows how power EP can be understood in terms of minimizing α-divergence instead of KL-divergence. This interpretation is originally due to Wiegerinck & Heskes (2002), who mentioned it in the context of fractional belief propagation.

The α-divergence (Amari, 1985; Trottini & Spezzaferri, 1999) is a generalization of KL-divergence with the following formula:

$$D_\alpha(p \mid\mid q) \quad = \quad \frac{4}{1-\alpha^2}\left(1 - \int_x p(x)^{(1+\alpha)/2} q(x)^{(1-\alpha)/2} dx\right) \tag{24}$$

The α-divergence is ≥ 0, with equality iff $p = q$. It is convex with respect to p and q. When $\alpha = 1$, it reduces to $KL(p \mid\mid q)$, and when $\alpha = -1$, it is $KL(q \mid\mid p)$. Consider the following algorithm, call it α-EP:

α-EP

- Initialize $\tilde{t}_a(x)$ for all a, and set $q(x)$ to their product.
- Repeat until convergence:
  For each a:

$$q^{\backslash a}(x) \quad = \quad q(x)/\tilde{t}_a(x) \tag{25}$$

$$q(x)^{new} \quad = \quad \operatorname*{argmin}_{q \in \mathcal{F}} D_{\alpha_a}(t_a(x) q^{\backslash a}(x) \mid\mid q(x)) \tag{26}$$

$$\tilde{t}_a(x)^{new} \quad = \quad q(x)^{new}/q^{\backslash a}(x) \tag{27}$$

This algorithm is just like EP except it minimizes α-divergence at each step. It turns out that this algorithm has exactly the same fixed points as power EP. The power EP algorithm given in the previous section is just a particular way of implementing α-EP.

To see the equivalence, set $\alpha_a = (2/n_a) - 1$. Then the minimization in (26) becomes

$$\operatorname*{argmin}_{q \in \mathcal{F}} \qquad \frac{n_a^2}{n_a - 1}\left(1 - \int_x \left(t_a(x) q^{\backslash a}(x)\right)^{1/n_a} q(x)^{1-1/n_a} dx\right) \tag{28}$$

This function is convex in the parameters of $q(x)$, so the minimum is unique. Setting the gradient to zero gives a condition for the minimum:

$$q(x) \quad = \quad \text{proj}\left[\left(t_a(x) q^{\backslash a}(x)\right)^{1/n_a} q(x)^{1-1/n_a}\right] \tag{29}$$

The right hand side is equivalent to $q'(x)$ from (21), so this amounts to $q(x) = q'(x)$, which is true at any fixed point of (23). Thus the inner loop of power EP is equivalent to minimizing α-divergence.

4 The objective function

EP does not directly minimize the KL-divergence $KL(p \parallel q)$, but rather a surrogate objective function, described by Minka (2001b; 2001a). This objective extends straightforwardly to power EP. As described in section 2, power EP can be interpreted as taking the fractional term $f_i(x) = t_i(x)^{1/n_i}$ and copying it n_i times. The objective function has exactly this form: we take the EP objective and copy each term n_i times.

The power EP objective is

$$\min_{\hat{p}_i} \max_{q} \sum_i n_i \int_x \hat{p}_i(x) \log \frac{\hat{p}_i(x)}{f_i(x)} \quad + \quad (1 - \sum_i n_i) \int_x q(x) \log q(x) \tag{30}$$

$$\text{such that} \int_x g_j(x) \hat{p}_i(x) dx \quad = \quad \int_x g_j(x) q(x) dx \tag{31}$$

$$\int_x \hat{p}_i(x) dx \quad = \quad 1 \tag{32}$$

$$\int_x q(x) dx \quad = \quad 1 \tag{33}$$

Recall that g_j are the features of the exponential family $\mathcal{F}$, so (31) means that $\text{proj}[\hat{p}_i(x)] = q(x)$.

The dual objective is

$$\min_{\nu} \max_{\lambda} \; (\sum_i n_i - 1) \log \int_x \exp(\sum_j g_j(x) \nu_j) dx$$

$$- \sum_{i=1}^{n} n_i \log \int_x f_i(x) \exp(\sum_j g_j(x) \lambda_{ij}) dx \tag{34}$$

$$\text{such that } (\sum_i n_i - 1) \nu_j = \sum_i n_i \lambda_{ij} \tag{35}$$

Define

$$\tau_{ij} \quad = \quad \nu_j - \lambda_{ij} \tag{36}$$

then it follows that

$$\nu_j \quad = \quad \sum_i n_i \tau_{ij} \tag{37}$$

Thus we can make the following definitions:

$$\tilde{f}_i(x) \quad = \quad \exp(\sum_j g_j(x) \tau_{ij}) \tag{38}$$

$$q(x) \quad = \quad \exp(\sum_j g_j(x) \nu_j) = \prod_i \tilde{f}_i(x)^{n_i} \tag{39}$$

By taking gradients, the stationary points of (34) are found to be:

$$q(x) \quad = \quad \text{proj}\left[f_i(x) q(x) / \tilde{f}_i(x) \right] \qquad \text{for all } i \tag{40}$$

The right hand side is equivalent to $q'(x)$ from (21), so this amounts to $q(x) = q'(x)$ for all i, which is true at any fixed point of power EP. Thus the power EP fixed points are exactly the stationary points of (34). The same result follows similarly for (30).

When the n_i are positive integers, the power EP objective is equivalent to the EP objective with repeated terms. Thus both algorithms converge to the same solutions for this case.

References

Amari, S. (1985). *Differential-geometrical methods in statistics*. Springer-Verlag.

Minka, T. P. (2001a). The EP energy function and minimization schemes. `resarch.microsoft.com/~minka/`.

Minka, T. P. (2001b). Expectation propagation for approximate Bayesian inference. *UAI* (pp. 362–369).

Minka, T. P., & Lafferty, J. (2002). Expectation-propagation for the generative aspect model. *UAI*.

Trottini, M., & Spezzaferri, F. (1999). A generalized predictive criterion for model selection (Technical Report 702). CMU Statistics Dept. `www.stat.cmu.edu/tr/`.

Wiegerinck, W., & Heskes, T. (2002). Fractional belief propagation. *NIPS 15*.

Winn, J., & Bishop, C. (2005). Variational message passing. *Journal of Machine Learning Research*, *6*, 661–694.