Divergence measures and message passing

Thomas Minka
Microsoft Research Ltd., Cambridge, UK
MSR-TR-2005-173, December 7, 2005

Abstract

This paper presents a unifying view of message-passing algorithms, as methods to approximate a complex Bayesian network by a simpler network with minimum information divergence. In this view, the difference between mean-field methods and belief propagation is not the amount of structure they model, but only the measure of loss they minimize ('exclusive' versus 'inclusive' Kullback-Leibler divergence). In each case, message-passing arises by minimizing a localized version of the divergence, local to each factor. By examining these divergence measures, we can intuit the types of solution they prefer (symmetry-breaking, for example) and their suitability for different tasks. Furthermore, by considering a wider variety of divergence measures (such as alpha-divergences), we can achieve different complexity and performance goals.

1 Introduction

Bayesian inference provides a mathematical framework for many artificial intelligence tasks, such as visual tracking, estimating range and position from noisy sensors, classifying objects on the basis of observed features, and learning. In principle, we simply draw up a belief network, instantiate the things we know, and integrate over the things we don't know, to compute whatever expectation or probability we seek. Unfortunately, even with simplified models of reality and clever algorithms for exploiting independences, exact Bayesian computations can be prohibitively expensive. For Bayesian methods to enjoy widespread use, there needs to be an array of approximation methods, which can produce decent results in a user-specified amount of time.

Fortunately, many belief networks benefit from an averaging effect. A network with many interacting elements can behave, on the whole, like a simpler network. This insight has led to a class of approximation methods called *variational methods* (Jordan et al., 1999) which approximate a complex network p by a simpler network q, optimizing the parameters of q to minimize information loss. The simpler network q can then act as a surrogate for p in a larger inference process. (Jordan et al. (1999) used convex duality and mean-field as the inspiration for their methods, but other approaches are also possible.) Variational methods are well-suited to large networks, especially ones that evolve through time. A large network can be divided into pieces, each of which is approximated variationally, yielding an overall variational approximation to the whole network. This decomposition strategy leads us directly to message-passing algorithms.

Message passing is a distributed method for fitting variational approximations, which is particularly well-suited to large networks. Originally, variational methods used coordinate-descent schemes (Jordan et al., 1999; Wiegerinck, 2000), which do not scale to large heterogeneous networks. Since then, a variety of scalable message-passing algorithms have been developed, each minimizing a different cost function with different message equations. These include:

- Variational message-passing (Winn & Bishop, 2005), a message-passing version of the mean-field method (Peterson & Anderson, 1987)
- Loopy belief propagation (Frey & MacKay, 1997)
- Expectation propagation (Minka, 2001b)
- Tree-reweighted message-passing (Wainwright et al., 2005b)
- Fractional belief propagation (Wiegerinck & Heskes, 2002)
- Power EP (Minka, 2004)

One way to understand these algorithms is to view their cost functions as free-energy functions from statistical physics (Yedidia et al., 2004; Heskes, 2003). From this

viewpoint, each algorithm arises as a different way to approximate the entropy of a distribution. This viewpoint can be very insightful; for example, it led to the development of generalized belief propagation (Yedidia et al., 2004).

The purpose of this paper is to provide a complementary viewpoint on these algorithms, which offers a new set of insights and opportunities. All six of the above algorithms can be viewed as instances of a recipe for minimizing information divergence. What makes algorithms different is the measure of divergence that they minimize. Information divergences have been studied for decades in statistics and many facts are now known about them. Using the theory of divergences, we can more easily choose the appropriate algorithm for our application. Using the recipe, we can construct new algorithms as desired. This unified view also allows us to generalize theorems proven for one algorithm to apply to the others.

The recipe to make a message-passing algorithm has four steps:

1. Pick an approximating family for q to be chosen from. For example, the set of fully-factorized distributions, the set of Gaussians, the set of k-component mixtures, etc.
2. Pick a divergence measure to minimize. For example, mean-field methods minimize the Kullback-Leibler divergence $\mathrm{KL}(q \,||\, p)$, expectation propagation minimizes $\mathrm{KL}(p \,||\, q)$, and power EP minimizes α-divergence $D_\alpha(p \,||\, q)$.
3. Construct an optimization algorithm for the chosen divergence measure and approximating family. Usually this is a fixed-point iteration obtained by setting the gradients to zero.
4. Distribute the optimization across the network, by dividing the network p into factors, and minimizing local divergence at each factor.

All six algorithms above can be obtained from this recipe, via the choice of divergence measure and approximating family.

The paper is organized as follows:

2 Divergence measures

This section describes various information divergence measures and illustrates how they behave. The behavior of divergence measures corresponds directly to the behavior of message-passing algorithms.

Let our task be to approximate a complex univariate or multivariate probability distribution $p(\mathbf{x})$. Our approximation, $q(\mathbf{x})$, is required to come from a simple predefined family $\mathcal{F}$, such as Gaussians. We want q to minimize a divergence measure $D(p \,||\, q)$, such as KL divergence. We will let p be unnormalized, i.e. $\int_x p(x)dx \neq 1$, because $\int_x p(x)dx$ is usually one of the things we would like to estimate. For example, if $p(x)$ is a Markov random field ($p(\mathbf{x}) = \prod_{ij} f_{ij}(x_i, x_j)$) then $\int_x p(x)dx$ is the partition function. If x is a parameter in Bayesian learning and $p(x)$ is the likelihood times prior ($p(x) \equiv p(x, D) = p(D|x)p_0(x)$ where the data D is fixed), then $\int_x p(x)dx$ is the evidence for the model. Consequently, q will also be unnormalized, so that the integral of q provides an estimate of the integral of p.

There are two basic divergence measures used in this paper. The first is the Kullback-Leibler (KL) divergence:

$$\mathrm{KL}(p \,||\, q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx + \int (q(x) - p(x)) dx \tag{1}$$

This formula includes a correction factor, so that it applies to unnormalized distributions (Zhu & Rohwer, 1995). Note this divergence is asymmetric with respect to p and q. The second divergence measure is a generalization of KL-divergence, called the **α-divergence** (Amari, 1985; Trottini & Spezzaferri, 1999; Zhu & Rohwer, 1995). It is actually a family of divergences, indexed by $\alpha \in (-\infty, \infty)$. Different authors use the α parameter in different ways. Using the convention of Zhu & Rohwer (1995), with α instead of

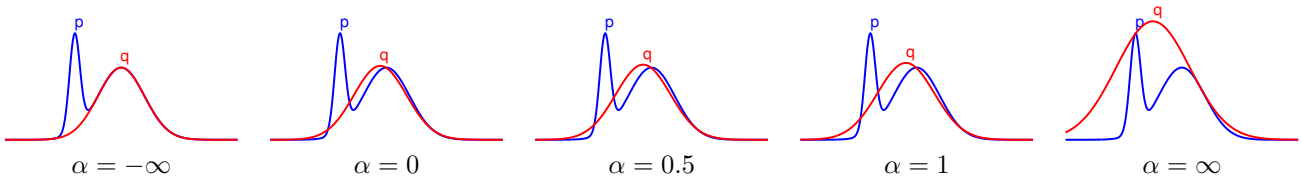


Figure 1: The Gaussian q which minimizes α-divergence to p (a mixture of two Gaussians), for varying α. $\alpha \to -\infty$ prefers matching one mode, while $\alpha \to \infty$ prefers covering the entire distribution.

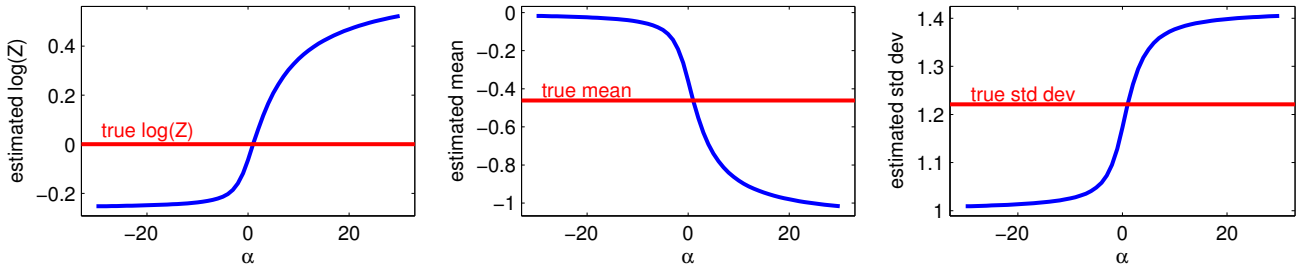


Figure 2: The mass, mean, and standard deviation of the Gaussian q which minimizes α-divergence to p, for varying α. In each case, the true value is matched at $\alpha = 1$.

δ, the formula is:

$$D_\alpha(p \,||\, q) = \frac{\int_x \alpha p(x) + (1-\alpha)q(x) - p(x)^\alpha q(x)^{1-\alpha} dx}{\alpha(1-\alpha)} \tag{2}$$

As in (1), p and q do not need to be normalized. Both KL-divergence and α-divergence are zero if $p = q$ and positive otherwise, so they satisfy the basic property of an error measure. This property follows from the fact that α-divergences are convex with respect to p and q (appendix A). Some special cases:

$$D_{-1}(p \,||\, q) = \frac{1}{2}\int_x \frac{(q(x)-p(x))^2}{p(x)} dx \tag{3}$$

$$\lim_{\alpha \to 0} D_\alpha(p \,||\, q) = \mathrm{KL}(q \,||\, p) \tag{4}$$

$$D_{\frac{1}{2}}(p \,||\, q) = 2\int_x \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx \tag{5}$$

$$\lim_{\alpha \to 1} D_\alpha(p \,||\, q) = \mathrm{KL}(p \,||\, q) \tag{6}$$

$$D_2(p \,||\, q) = \frac{1}{2}\int_x \frac{(p(x)-q(x))^2}{q(x)} dx \tag{7}$$

The case $\alpha = 0.5$ is known as Hellinger distance (whose square root is a valid distance metric), and $\alpha = 2$ is the χ^2 distance. Changing α to $1-\alpha$ swaps the position of p and q.

To illustrate the effect of changing the divergence measure, consider a simple example, illustrated in figures 1 and 2. The original distribution $p(x)$ is a mixture of two Gaussians, one tall and narrow, the other short and wide. The approximation $q(x)$ is required to be a single (scaled) Gaussian, with arbitrary mean, variance, and scale factor. For different values of α, figure 1 plots the global minimum of $D_\alpha(p \,||\, q)$ over q. The solutions vary smoothly with α, the most dramatic changes happening around $\alpha = 0.5$. When α is a large negative number, the best approximation represents only one mode, the one with largest mass (*not* the mode which is highest). When α is a large positive number, the approximation tries to cover the entire distribution, eventually forming an upper bound when $\alpha \to \infty$. Figure 2 shows that the mass of the approximation continually increases as we increase α.

The properties observed in this example are general, and can be derived from the formula for α-divergence. Start with the mode-seeking property for $\alpha \ll 0$. It happens because the valleys of p force the approximation downward. Looking at (3,4) for example, we see that $\alpha \leq 0$ emphasizes q to be small whenever p is small. These divergences are **zero-forcing** because $p(\mathbf{x}) = 0$ forces $q(\mathbf{x}) = 0$. In other words, they avoid "false positives," to an increasing degree as α gets more negative. This causes some parts of p to be excluded. The cost of excluding an $\mathbf{x}$, i.e. setting $q(\mathbf{x}) = 0$, is $p(\mathbf{x})/(1-\alpha)$. Therefore q will keep the areas of largest total mass, and exclude areas with small total mass.

Zero-forcing emphasizes modeling the tails, rather than the bulk of the distribution, which tends to underestimate the variance of p. For example, when p is a mixture of Gaussians, the tails reflect the component which is widest. The optimal Gaussian q will have variance similar to the variance of the widest component, even if there are many overlapping components. For example, if p has 100 identical Gaussians in a row, forming a plateau, the optimal q is only as wide as one of them.

When $\alpha \geq 1$, a different tendency happens. These diver-

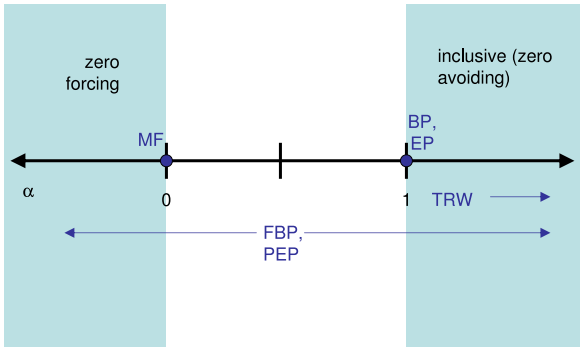


Figure 3: The structure of α-divergences.

gences want to cover as much of p as possible. Following the terminology of Frey et al. (2000), these divergences are **inclusive** ($\alpha < 1$ are **exclusive**). Inclusive divergences require $q > 0$ whenever $p > 0$, thus avoiding "false negatives." If two identical Gaussians are separated enough, an exclusive divergence prefers to represent only one of them, while an inclusive divergence prefers to stretch across both.

Figure 3 diagrams the structure of α space. As shown later, the six algorithms of section 1 correspond to minimizing different α-divergences, indicated on the figure. Variational message-passing/mean-field uses $\alpha = 0$, belief propagation and expectation propagation use $\alpha = 1$, tree-reweighted message-passing can use a variety of $\alpha \geq 1$, while fractional belief propagation and power EP can use any α-divergence.

The divergences with $0 < \alpha < 1$ are a blend of these extremes. They are not zero-forcing, so they try to represent multiple modes, but will ignore modes that are far away from the main mass (how far depends on α).

Now consider the mass of the optimal q. Write $q(x) = \tilde{Z}\bar{q}(x)$, where $\bar{q}$ is normalized, so that $\tilde{Z}$ represents the mass. It is straightforward to obtain the optimum $\tilde{Z}$:

$$\tilde{Z}_\alpha = \begin{cases} \exp\left(\int_x \bar{q}(x) \log \frac{p(x)}{\bar{q}(x)} dx\right) & \text{if } \alpha = 0 \\ \left(\int_x p(x)^\alpha \bar{q}(x)^{1-\alpha} dx\right)^{1/\alpha} & \text{otherwise} \end{cases} \tag{8}$$

This is true regardless of whether $\bar{q}$ is optimal.

Theorem 1 If x is a non-negative random variable, then $E[x^\alpha]^{1/\alpha}$ is nondecreasing in α.

Proof: See appendix B. □

Theorem 2 $\tilde{Z}_\alpha$ is nondecreasing in α. As a consequence,

$$\tilde{Z} \leq \int_x p(x)dx \qquad \text{if } \alpha < 1 \tag{9a}$$

$$\tilde{Z} = \int_x p(x)dx \qquad \text{if } \alpha = 1 \tag{9b}$$

$$\tilde{Z} \geq \int_x p(x)dx \qquad \text{if } \alpha > 1 \tag{9c}$$

Proof: In Th. 1, let $x = p(x)/\bar{q}(x)$ and take the expectation with respect to $\bar{q}(x)$. □

Theorem 2 is demonstrated in figure 2: the integral of q monotonically increases with α, passing through the true value when $\alpha = 1$. This theorem applies to an exact minimization over $\tilde{Z}$, which is generally not possible. But it shows that the $\alpha < 1$ divergence measures tend to underestimate the integral of p, while $\alpha > 1$ tends to overestimate. Only $\alpha = 1$ tries to recover the correct integral.

Now that we have looked at the properties of different divergence measures, let's look at specific algorithms to minimize them.

3 Minimizing α-divergence

This section describes a simple method to minimize α-divergence, by repeatedly minimizing KL-divergence. The method is then illustrated on exponential families and factorized approximations.

3.1 A fixed-point scheme

When q minimizes the KL-divergence to p over a family $\mathcal{F}$, we will say that q is the **KL-projection** of p onto $\mathcal{F}$. As a shorthand for this, define the operator $\text{proj}[\cdot]$ as:

$$\text{proj}[p] \quad = \quad \operatorname*{argmin}_{q \in \mathcal{F}} \text{KL}(p \,||\, q) \tag{10}$$

Theorem 3 Let $\mathcal{F}$ be indexed by a continuous parameter θ, possibly with constraints. If $\alpha \neq 0$:

$$q \text{ is a stationary point of } D_\alpha(p \,||\, q) \tag{11}$$
$$\iff q \text{ is a stationary point of } \text{proj}\left[p(x)^\alpha q(x)^{1-\alpha}\right]$$

Proof: The derivative of the α-divergence with respect to θ is

$$\frac{dD_\alpha(p \,||\, q)}{d\theta} = \frac{1}{\alpha}\left(\int_x \frac{dq(x)}{d\theta}dx - \int_x \frac{p'_\theta(x)}{q(x)}\frac{dq(x)}{d\theta}dx\right) \tag{12}$$

$$\text{where } p'_\theta(x) = p(x)^\alpha q(x)^{1-\alpha} \tag{13}$$

When $\alpha = 1$ (KL-divergence), the derivative is

$$\frac{d\text{KL}(p \,||\, q)}{d\theta} = \int_x \frac{dq(x)}{d\theta}dx - \int_x \frac{p(x)}{q(x)}\frac{dq(x)}{d\theta}dx \tag{14}$$

Comparing (14) and (12), we find that

$$\left.\frac{dD_\alpha(p \,||\, q)}{d\theta}\right|_{\theta=\theta_0} = \frac{1}{\alpha}\left.\frac{d\text{KL}(p'_{\theta_0} \,||\, q)}{d\theta}\right|_{\theta=\theta_0} \tag{15}$$

Therefore if $\alpha \neq 0$, the corresponding Lagrangians must have the same stationary points. □

To find a q satisfying (11), we can apply a fixed-point iteration. Guess an initial q, then repeatedly update it via

$$q'(x) = \text{proj}\big[p(x)^\alpha q(x)^{1-\alpha}\big] \tag{16}$$

$$q(x)^{\text{new}} = q(x)^\epsilon q'(x)^{1-\epsilon} \tag{17}$$

This scheme is heuristic and not guaranteed to converge. However, it is often successful with an appropriate amount of damping (ϵ).

More generally, we can minimize D_α by repeatedly minimizing any other $D_{\alpha'}$ ($\alpha' \neq 0$):

$$q'(x) = \text{argmin}\, D_{\alpha'}(p(x)^{\alpha/\alpha'} q(x)^{1-\alpha/\alpha'} \,||\, q'(x)) \tag{18}$$

3.2 Exponential families

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

$$q(x) \quad = \quad \exp(\textstyle\sum_j g_j(x)\nu_j) \tag{19}$$

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. To work with unnormalized distributions, we make $g_0(x) = 1$ a feature, whose corresponding parameter ν_0 captures the scale of the distribution. To ensure the distribution is proper, there may be constraints on the ν_j, e.g. the variance of a Gaussian must be positive.

KL-projection for exponential families has a simple interpretation. Substituting (19) into the KL-divergence, we find that the minimum is achieved at any member of $\mathcal{F}$ whose expectation of g_j matches that of p, for all j:

$$q = \text{proj}[p] \iff \forall_j \int_x g_j(x)q(x)dx = \int_x g_j(x)p(x)dx \tag{20}$$

For example, if $\mathcal{F}$ is the set of Gaussians, then proj$[p]$ is the unique Gaussian whose mean, variance, and scale matches p. Equation (16) in the fixed-point scheme reduces to computing the expectations of $p(x)^\alpha q(x)^{1-\alpha}$ and setting $q'(x)$ to match those expectations.

3.3 Fully-factorized approximations

A distribution is said to be **fully-factorized** if it can be written as

$$q(\mathbf{x}) = s\prod_i q_i(x_i) \tag{21}$$

We will use the convention that q_i is normalized, so that s represents the integral of q.

KL-projection onto a fully-factorized distribution reduces to matching the marginals of p:

$$q = \text{proj}[p] \iff \forall_i \int_{\mathbf{x}\setminus x_i} q(\mathbf{x})d\mathbf{x} = \int_{\mathbf{x}\setminus x_i} p(\mathbf{x})d\mathbf{x} \tag{22}$$

which simplifies to

$$s = \int_{\mathbf{x}} p(\mathbf{x})d\mathbf{x} \tag{23}$$

$$\forall_i \;\; q_i(x_i) = \frac{1}{s}\int_{\mathbf{x}\setminus x_i} p(\mathbf{x})d\mathbf{x} \tag{24}$$

Equation (16) in the fixed-point scheme simplifies to:

$$s' = \int_{\mathbf{x}} p(\mathbf{x})^\alpha q(\mathbf{x})^{1-\alpha} d\mathbf{x} \tag{25}$$

$$q_i'(x_i) = \frac{1}{s'}\int_{\mathbf{x}\setminus x_i} p(\mathbf{x})^\alpha q(\mathbf{x})^{1-\alpha} d\mathbf{x} \tag{26}$$

$$= \frac{s^{1-\alpha}}{s'} q_i(x_i)^{1-\alpha} \int_{\mathbf{x}\setminus x_i} p(\mathbf{x})^\alpha \prod_{j\neq i} q_j(x_j)^{1-\alpha} d\mathbf{x} \tag{27}$$

In this equation, q is assumed to have no constraints other than being fully-factorized. Going further, we may require q to be in a fully-factorized exponential family. A fully-factorized exponential family has features $g_{ij}(x_i)$, involving one variable at a time. In this case, (20) becomes

$$\begin{aligned} q = \text{proj}[p] \iff \forall_{ij} \int_{\mathbf{x}} g_{ij}(x_i)q(\mathbf{x})d\mathbf{x} \\ = \int_{\mathbf{x}} g_{ij}(x_i)p(\mathbf{x})d\mathbf{x} \end{aligned} \tag{28}$$

This can be abbreviated using a projection onto the features of x_i (which may vary with i):

$$\text{proj}\left[\int_{\mathbf{x}\setminus x_i} q(\mathbf{x})d\mathbf{x}\right] = \text{proj}\left[\int_{\mathbf{x}\setminus x_i} p(\mathbf{x})d\mathbf{x}\right] \tag{29}$$

$$\text{or} \qquad sq_i(x_i) = \text{proj}\left[\int_{\mathbf{x}\setminus x_i} p(\mathbf{x})d\mathbf{x}\right] \tag{30}$$

Equation (16) in the fixed-point scheme becomes:

$$q_i'(x_i) = \frac{1}{s'}\text{proj}\left[\int_{\mathbf{x}\setminus x_i} p(\mathbf{x})^\alpha q(\mathbf{x})^{1-\alpha} d\mathbf{x}\right] \tag{31}$$

$$= \frac{s^{1-\alpha}}{s'}\text{proj}\left[q_i(x_i)^{1-\alpha} \int_{\mathbf{x}\setminus x_i} p(\mathbf{x})^\alpha \prod_{j\neq i} q_j(x_j)^{1-\alpha} d\mathbf{x}\right] \tag{32}$$

Note that $q_i(x_i)^{1-\alpha}$ is inside the projection.

When $\alpha = 0$, the fixed-point scheme of section 3.1 doesn't apply. However, there is a simple fixed-point scheme for minimizing KL($q \,||\, p$) when q is fully-factorized and otherwise unconstrained (the other cases are more complicated).

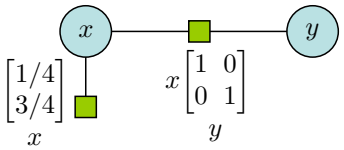


Figure 4: Factor graph for the equality example

With q having form (21), the KL-divergence becomes:

$$\begin{aligned}\mathrm{KL}(q \,||\, p) = s \sum_i \int_{x_i} q_i(x_i) \log q_i(x_i) dx_i \\ - s \int_{\mathbf{x}} \prod_i q_i(x_i) \log p(\mathbf{x}) d\mathbf{x} \\ + s \log s - s + \int_{\mathbf{x}} p(\mathbf{x}) d\mathbf{x} \quad (33)\end{aligned}$$

Zeroing the derivative with respect to $q_i(x_i)$ gives the update

$$q_i(x_i)^{\text{new}} \propto \exp\left(\int_{\mathbf{x}\setminus x_i} \prod_{j\neq i} q_j(x_j) \log p(\mathbf{x}) d\mathbf{x}\right) \quad (34)$$

which is analogous to (27) with $\alpha \to 0$. Cycling through these updates for all i gives a coordinate descent procedure. Because each sub-problem is convex, the procedure must converge to a local minimum.

3.4 Equality example

This section considers a concrete example of minimizing α-divergence over fully-factorized distributions, illustrating the difference between different divergences, and by extension, different message-passing schemes. Consider a binary variable x whose distribution is $p_x(0) = 1/4, p_x(1) = 3/4$. Now add a binary variable y which is constrained to equal x. The marginal distribution for x should be unchanged, and the marginal distribution for y should be the same as for x: $p_y(0) = 1/4$. However, this is not necessarily the case when using approximate inference.

These two pieces of information can be visualized as a factor graph (figure 4). The joint distribution of x and y can be written as a matrix:

$$p(x,y) = \begin{matrix} \\ x \end{matrix} \begin{matrix} \begin{bmatrix} 1/4 & 0 \\ 0 & 3/4 \end{bmatrix} \\ y \end{matrix} \quad (35)$$

This distribution has two modes of different height, similar to the example in figure 1.

Let's approximate this distribution with a fully-factorized q (21), minimizing different α-divergences. This approximation has 3 free parameters: the total mass s, $q_x(0)$, and $q_y(0)$. We can solve for these parameters analytically. By symmetry, we must have $q_y(0) = q_x(0)$. Furthermore, at a fixed point $q = q'$. Thus (27) simplifies as follows:

$$q'_x(x) = \frac{s^{1-\alpha}}{s'} q_x(x)^{1-\alpha} \sum_y p(x,y)^\alpha q_y(y)^{1-\alpha} \quad (36)$$

$$q_x(x)^\alpha = s^{-\alpha} \sum_y p(x,y)^\alpha q_x(y)^{1-\alpha} \quad (37)$$

$$q_x(0)^\alpha = s^{-\alpha} p_x(0)^\alpha q_x(0)^{1-\alpha} \quad (38)$$

$$q_x(0)^{2\alpha-1} = s^{-\alpha} p_x(0)^\alpha \quad (39)$$

$$q_x(1)^{2\alpha-1} = s^{-\alpha} p_x(1)^\alpha \quad (40)$$

$$\left(\frac{q_x(0)}{q_x(1)}\right)^{2\alpha-1} = \left(\frac{p_x(0)}{p_x(1)}\right)^\alpha \quad (41)$$

$$q_x(0) = \begin{cases} \frac{p_x(0)^{\alpha/(2\alpha-1)}}{p_x(0)^{\alpha/(2\alpha-1)} + p_x(1)^{\alpha/(2\alpha-1)}} & \alpha > 1/2 \\ 0 & \alpha \leq 1/2 \end{cases} \quad (42)$$

$$s = p_x(1) q_x(1)^{(1-2\alpha)/\alpha} \quad (43)$$

When $\alpha = 1$, corresponding to running belief propagation, the result is $(q_x(0) = p_x(0), s = 1)$ which means

$$q_{\text{BP}}(x,y) = \begin{matrix} \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix} \\ x \end{matrix} \begin{matrix} \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix} \\ y \end{matrix} = \begin{matrix} \\ x \end{matrix} \begin{matrix} \begin{bmatrix} 1/16 & 3/16 \\ 3/16 & 9/16 \end{bmatrix} \\ y \end{matrix} \quad (44)$$

The approximation matches the marginals and total mass of p. Because the divergence is inclusive, the approximation includes both modes, and smooths over the zeros. It overrepresents the higher mode, making it 9 times higher than the other, while it should only be 3 times higher.

When $\alpha = 0$, corresponding to running mean-field, or in fact when $\alpha \leq 1/2$, the result is $(q_x(0) = 0, s = p_x(1))$ which means

$$q_{\text{MF}}(x,y) = 3/4 \begin{matrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ x \end{matrix} \begin{matrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ y \end{matrix} = \begin{matrix} \\ x \end{matrix} \begin{matrix} \begin{bmatrix} 0 & 0 \\ 0 & 3/4 \end{bmatrix} \\ y \end{matrix} \quad (45)$$

This divergence preserves the zeros, forcing it to model only one mode, whose height is represented correctly. There are two local minima in the minimization, corresponding to the two modes—the global minimum, shown here, models the more massive mode. The approximation does not preserve the marginals or overall mass of p.

At the other extreme, when $\alpha \to \infty$, the result is $(q_x(0) = \frac{\sqrt{p_x(0)}}{\sqrt{p_x(0)}+\sqrt{p_x(1)}}, s = (\sqrt{p_x(0)} + \sqrt{p_x(1)})^2)$ which means

$$q_\infty(x,y) = \frac{(1+\sqrt{3})^2}{4} \begin{matrix} \begin{bmatrix} \frac{1}{1+\sqrt{3}} \\ \frac{\sqrt{3}}{1+\sqrt{3}} \end{bmatrix} \\ x \end{matrix} \begin{matrix} \begin{bmatrix} \frac{1}{1+\sqrt{3}} \\ \frac{\sqrt{3}}{1+\sqrt{3}} \end{bmatrix} \\ y \end{matrix} \quad (46)$$

$$= \begin{matrix} \\ x \end{matrix} \begin{matrix} \begin{bmatrix} 1/4 & \sqrt{3}/4 \\ \sqrt{3}/4 & 3/4 \end{bmatrix} \\ y \end{matrix} \quad (47)$$

As expected, the approximation is a point-wise upper bound to p. It preserves both peaks perfectly, but smooths away the zeros. It does not preserve the marginals or total mass of p.

From these results, we can draw the following conclusions:

- None of the approximations is inherently superior. It depends on what properties of p you care about preserving.
- Fitting a fully-factorized approximation does not imply trying to match the marginals of p. It depends on what properties the divergence measure is trying to preserve. Using $\alpha = 0$ is equivalent to saying that zeros are more important to preserve than marginals, so when faced with the choice, mean-field will preserve the zeros.
- Under approximate inference, adding a new variable (y, in this case) to a model can change the estimation of existing variables (x), even when the new variable provides no information. For example, when using mean-field, adding y suddenly makes us believe that $x = 1$.

4 Message-passing

This section describes a general message-passing scheme to (approximately) minimize a given divergence measure D. Mean-field methods, belief propagation, and expectation propagation are all included in this scheme.

The procedure is as follows. We have a distribution p and we want to find $q \in \mathcal{F}$ that minimizes $D(p \mid\mid q)$. First, we must restrict $\mathcal{F}$ to be an exponential family. Then we will write the distribution p as a product of factors, $p(\mathbf{x}) = \prod_a f_a(\mathbf{x})$, as in a Bayesian network. Each factor will be approximated by a member of $\mathcal{F}$, such that when we multiply these approximations together we get a $q \in \mathcal{F}$ that has a small value of $D(p \mid\mid q)$. The best approximation of each factor depends on the rest of the network, giving a chicken-and-egg problem. This is solved by an iterative message-passing procedure where each factor sends its approximation to the rest of the net, and then recomputes its approximation based on the messages it receives.

The first step is to choose an exponential family. The reason to use exponential families is closure under multiplication: the product of any distributions in the family is also in the family.

The next step is to write the original distribution p as a product of nonnegative factors:

$$p(\mathbf{x}) = \prod_a f_a(\mathbf{x}) \tag{48}$$

This defines the specific way in which we want to divide the network, and is not unique. Each factor can depend on several, perhaps all, of the variables of p. By approximating each factor f_a by $\tilde{f}_a \in \mathcal{F}$, we get an approximation divided in the same way:

$$\tilde{f}_a(\mathbf{x}) = \exp(\textstyle\sum_j g_j(\mathbf{x})\tau_{aj}) \tag{49}$$

$$q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x}) \tag{50}$$

Now we look at the problem from the perspective of a given approximate factor $\tilde{f}_a$. Define $q^{\setminus a}(\mathbf{x})$ to be the product of all other approximate factors:

$$q^{\setminus a}(\mathbf{x}) = q(\mathbf{x})/\tilde{f}_a(\mathbf{x}) = \prod_{b \neq a} \tilde{f}_b(\mathbf{x}) \tag{51}$$

Similarly, define $p^{\setminus a}(\mathbf{x}) = \prod_{b \neq a} f_a(\mathbf{x})$. Then factor $\tilde{f}_a$ seeks to minimize $D(f_a p^{\setminus a} \mid\mid \tilde{f}_a q^{\setminus a})$. To make this tractable, assume that the approximations we've already made, $q^{\setminus a}(\mathbf{x})$, are a good approximation to the rest of the network, i.e. $p^{\setminus a} \approx q^{\setminus a}$, at least for the purposes of solving for $\tilde{f}_a$. Then the problem becomes

$$\tilde{f}_a(\mathbf{x}) = \operatorname{argmin} D(f_a(\mathbf{x})q^{\setminus a}(\mathbf{x}) \mid\mid \tilde{f}_a(\mathbf{x})q^{\setminus a}(\mathbf{x})) \tag{52}$$

This problem is tractable, provided we've made a sensible choice of factors. It can be solved with the procedures of section 3. Cycling through these coupled sub-problems gives the message-passing algorithm:

Generic Message Passing

- Initialize $\tilde{f}_a(\mathbf{x})$ for all a.
- Repeat until all $\tilde{f}_a$ converge:
  1. Pick a factor a.
  2. Compute $q^{\setminus a}$ via (51).
  3. Using the methods of section 3:
     $\tilde{f}_a(\mathbf{x})^{\text{new}} =$
     $\operatorname{argmin} D(f_a(\mathbf{x})q^{\setminus a}(\mathbf{x}) \mid\mid \tilde{f}_a(\mathbf{x})q^{\setminus a}(\mathbf{x}))$

This algorithm can be interpreted as message passing between the factors f_a. The approximation $\tilde{f}_a$ is the message that factor a sends to the rest of the network, and $q^{\setminus a}$ is the collection of messages that factor a receives (its "inbox"). The inbox summarizes the behavior of the rest of the network.

4.1 Fully-factorized case

When q is fully-factorized as in section 3.3, message-passing has an elegant graphical interpretation via factor graphs. Instead of factors passing messages to factors, messages move along the edges of the factor graph, between variables and factors, as shown in figure 5. (The case where q is structured can also be visualized on a graph, but a more

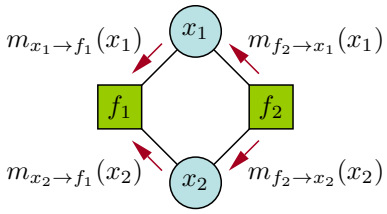


Figure 5: Message-passing on a factor graph

complex type of graph known as a *structured region graph* (Welling et al., 2005).)

Because q is fully-factorized, the approximate factors will be fully-factorized into messages $m_{a \to i}$ from factor a to variable i:

$$\tilde{f}_a(\mathbf{x}) = \prod_i m_{a \to i}(x_i) \tag{53}$$

Individual messages need not be normalized, and need not be proper distributions.

The inboxes $q^{\setminus a}(\mathbf{x})$ will factorize in the same way as q. We can collect all terms involving the same variable x_i, to define messages $m_{i \to a}$ from variable i to factor a:

$$m_{i \to a}(x_i) = \prod_{b \neq a} m_{b \to i}(x_i) \tag{54}$$

$$q^{\setminus a}(\mathbf{x}) = \prod_{b \neq a} \prod_i m_{b \to i}(x_i) = \prod_i m_{i \to a}(x_i) \tag{55}$$

This implies $q_i(x_i) = m_{a \to i}(x_i) m_{i \to a}(x_i)$ for any a.

Now solve (52) in the fully-factorized case. If D is an α-divergence, we can apply the fixed-point iteration of section 3.1. Substitute $p(\mathbf{x}) = f_a(\mathbf{x}) q^{\setminus a}(\mathbf{x})$ and $q(\mathbf{x}) = \tilde{f}_a(\mathbf{x}) q^{\setminus a}(\mathbf{x})$ into (25) to get

$$s' = \int_{\mathbf{x}} f_a(\mathbf{x})^\alpha \tilde{f}_a(\mathbf{x})^{1-\alpha} q^{\setminus a}(\mathbf{x}) d\mathbf{x} \tag{56}$$

$$= \int_{\mathbf{x}} f_a(\mathbf{x})^\alpha \prod_j m_{a \to j}(x_j)^{1-\alpha} m_{j \to a}(x_j) d\mathbf{x} \tag{57}$$

Make the same substitution into (31):

$$q_i'(x_i) = \frac{1}{s'} \text{proj} \left[\int_{\mathbf{x} \setminus x_i} f_a(\mathbf{x})^\alpha \tilde{f}_a(\mathbf{x})^{1-\alpha} q^{\setminus a}(\mathbf{x}) d\mathbf{x} \right] \tag{58}$$

$$m_{a \to i}(x_i)' m_{i \to a}(x_i) = \frac{1}{s'} \times \tag{59}$$

$$\text{proj} \left[\int_{\mathbf{x} \setminus x_i} f_a(\mathbf{x})^\alpha \prod_j m_{a \to j}(x_j)^{1-\alpha} m_{j \to a}(x_j) d\mathbf{x} \right]$$

$$m_{a \to i}(x_i)' = \frac{1}{s' m_{i \to a}(x_i)} \text{proj} \left[m_{a \to i}(x_i)^{1-\alpha} m_{i \to a}(x_i) \int_{\mathbf{x} \setminus x_i} f_a(\mathbf{x})^\alpha \prod_{j \neq i} m_{a \to j}(x_j)^{1-\alpha} m_{j \to a}(x_j) d\mathbf{x} \right] \tag{60}$$

A special case arises if x_i does not appear in $f_a(\mathbf{x})$. Then the integral in (60) becomes constant with respect to x_i and the projection is exact, leaving $m_{a \to i}(x_i)' \propto m_{a \to i}(x_i)^{1-\alpha}$. In other words, $m_{a \to i}(x_i) = 1$. With this substitution, we only need to propagate messages between a factor and the variables it uses.

The algorithm becomes:

Fully-Factorized Message Passing

- Initialize $m_{a \to i}(x_i)$ for all (a, i).
- Repeat until all $m_{a \to i}$ converge:
  1. Pick a factor a.
  2. Compute the messages into the factor via (54).
  3. Compute the messages out of the factor via (60) (if D is an α-divergence), and apply a step-size ϵ (17).

If D is not an α-divergence, then the outgoing message formula will change but the overall algorithm is the same.

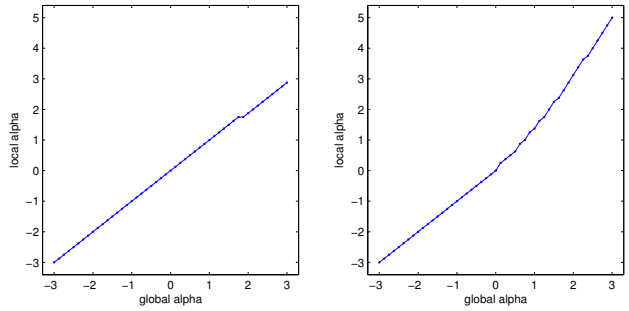
4.2 Local vs. global divergence

The generic message passing algorithm is based on the assumption that minimizing the *local divergences* $D(f_a(x) q^{\setminus a}(x) \,||\, \tilde{f}_a(x) q^{\setminus a}(x))$ approximates minimizing the global divergence $D(p \,||\, q)$. An interesting question is whether, in the end, we are minimizing the divergence we intended, or if the result resembles some other divergence. In the case $\alpha = 0$, minimizing local divergences corresponds exactly to minimizing global divergence, as shown in section 5. Otherwise, the correspondence is only approximate. To measure how close the correspondence is, consider the following experiment: given a global α-divergence index α_G, find the corresponding local α-divergence index α_L which produces the best q according to α_G.

This experiment was carried out with $p(\mathbf{x})$ equal to a 4×4 Boltzmann grid, i.e. binary variables connected by pairwise factors:

$$p(\mathbf{x}) = \prod_i f_i(x_i) \prod_{ij \in \mathcal{E}} f_{ij}(x_i, x_j) \tag{61}$$

The graph $\mathcal{E}$ was a grid with four-connectivity. The unary potentials had the form $f_i(x_i) = [\exp(\theta_{i1}) \exp(\theta_{i2})]$, and the pairwise potentials had the form $f_{ij}(x_i, x_j) = \begin{bmatrix} 1 & \exp(w_{ij}) \\ \exp(w_{ij}) & 1 \end{bmatrix}$. The goal was to approximate p with a fully-factorized q. For a given local divergence α_L,



(a) coupling $\sim U(-1, 1)$ (b) coupling $= 1$

Figure 6: The best local α for minimizing a given global α-divergence, across ten networks with (a) random or (b) positive couplings.

this was done using the fractional BP algorithm of section 7 (all factors used the same α_L). Then $D_{\alpha_G}(p \,||\, q)$ was computed by explicit summation over $\mathbf{x}$ (enumerating all states of the network). Ten random networks were generated with (θ, w) drawn randomly from a uniform distribution over $[-1, 1]$. The results are shown in figure 6(a). For individual networks, the best α_L sometimes differs from α_G when $\alpha_G > 1$ (not shown), but the one best α_L across all 10 networks (shown) is $\alpha_L = \alpha_G$, with a slight downward bias for large α_G. Thus by minimizing localized divergence we are close to minimizing the same divergence globally.

In general, if the approximating family $\mathcal{F}$ is a good fit to p, then we should expect local divergence to match global divergence, since $q^{\backslash a} \approx p^{\backslash a}$. In a graph with random potentials, the correlations tend to be short, so approximating $p^{\backslash a}$ with a fully-factorized distribution does little harm (there is not much over-counting due to loops). If p has long-range correlations, then $q^{\backslash a}$ will not fit as well, and we expect a larger discrepancy between local and global divergence. To test this, another experiment was run with $w_{ij} = 1$ on all edges. In this case, there are long-range correlations and message passing suffers from over-counting effects. The results in figure 6(b) now show a consistent discrepancy between α_G and α_L. When $\alpha_G < 0$, the best $\alpha_L = \alpha_G$ as before. But when $\alpha_G \geq 0$, the best α_L was strictly larger than α_G (the relationship is approximately linear, with slope > 1). To understand why large α_L could be good, recall that increasing α leads to flatter approximations, which try to cover all of p. By making the local approximations flatter, we make the messages weaker, which reduces the over-counting. This example shows that if q is a poor fit to p, then we might do better by choosing a local divergence different from the global one we want to minimize.

We can also improve the quality of the approximation by changing the number of factors we divide p into. In the extreme case, we can use only one factor to represent all of p, in which case the local divergence is exactly the global divergence. By using more factors, we simplify the computations, at the cost of making additional approximations.

4.3 Mismatched divergences

It is possible to run message passing with a different divergence measure being minimized for each factor a. For example, one factor may use $\alpha = 1$ while another uses $\alpha = 0$. The motivation for this is that some divergences may be easier to minimize for certain factors (Minka, 2004). The effect of this on the global result is unclear, but locally the observations of section 2 will continue to hold.

While less motivated theoretically, mismatched divergences are very useful in practice. Henceforth we will allow each factor a to have its own divergence index α_a.

4.4 Estimating Z

Just as in section 2, we can analytically derive the $\tilde{Z}$ that would be computed by message-passing, for any approximating family. Let $q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x})$, possibly unnormalized, where $\tilde{f}_a(\mathbf{x})$ are any functions in the family $\mathcal{F}$. Define the rescaled factors

$$\tilde{f}'_a(\mathbf{x}) = s_a \tilde{f}_a(\mathbf{x}) \tag{62}$$

$$q'(\mathbf{x}) = \prod_a \tilde{f}'_a(\mathbf{x}) = (\prod_a s_a) q(\mathbf{x}) \tag{63}$$

$$\tilde{Z} = \int_{\mathbf{x}} q'(\mathbf{x}) d\mathbf{x} = \left(\int_{\mathbf{x}} q(\mathbf{x}) d\mathbf{x} \right) \prod_a s_a \tag{64}$$

The scale s_a that minimizes local α-divergence is

$$s_a = \begin{cases} \exp\left(\dfrac{\int_{\mathbf{x}} q(\mathbf{x}) \log \dfrac{f_a(\mathbf{x})}{\tilde{f}_a(\mathbf{x})} d\mathbf{x}}{\int_{\mathbf{x}} q(\mathbf{x}) d\mathbf{x}} \right) & \text{if } \alpha_a = 0 \\ \left(\dfrac{\int_{\mathbf{x}} \left(\dfrac{f_a(\mathbf{x})}{\tilde{f}_a(\mathbf{x})} \right)^{\alpha_a} q(\mathbf{x}) d\mathbf{x}}{\int_{\mathbf{x}} q(\mathbf{x}) d\mathbf{x}} \right)^{1/\alpha_a} & \text{otherwise} \end{cases} \tag{65}$$

Plugging this into (64) gives (for $\alpha_a \neq 0$):

$$\tilde{Z} = \left(\int_{\mathbf{x}} q(\mathbf{x}) d\mathbf{x} \right)^{1 - \sum_a 1/\alpha_a} \times \prod_a \left(\int_{\mathbf{x}} \left(\frac{f_a(\mathbf{x})}{\tilde{f}_a(\mathbf{x})} \right)^{\alpha_a} q(\mathbf{x}) d\mathbf{x} \right)^{1/\alpha_a} \tag{66}$$

Because the mass of q estimates the mass of p, (66) provides an estimate of $\int_{\mathbf{x}} p(\mathbf{x}) d\mathbf{x}$ (the partition function or model evidence). Compared to (8), the minimum of the global divergence, this estimate is more practical to compute since it involves integrals over one factor at a time.

Interestingly, when $\alpha = 0$ the local and global estimates are the same. This fact is explored in section 5.

Theorem 4 For any set of messages $\tilde{f}$:

$$\tilde{Z} \leq \int_{\mathbf{x}} p(\mathbf{x}) d\mathbf{x} \qquad \text{if } \alpha_a \leq 0 \tag{67a}$$

$$\tilde{Z} \geq \int_{\mathbf{x}} p(\mathbf{x}) d\mathbf{x} \qquad \text{if } \begin{array}{r} \alpha_a > 0 \\ \sum_a 1/\alpha_a \leq 1 \end{array} \tag{67b}$$

Proof: Appendix C proves the following generalizations of the Hölder inequality:

$$E[\textstyle\prod_i x_i] \geq \prod_i E[x_i^{\alpha_i}]^{1/\alpha_i} \qquad \text{if } \alpha_i \leq 0 \tag{68a}$$

$$E[\textstyle\prod_i x_i] \leq \prod_i E[x_i^{\alpha_i}]^{1/\alpha_i} \qquad \text{if } \begin{array}{r} \alpha_i > 0 \\ \sum_i 1/\alpha_i \leq 1 \end{array} \tag{68b}$$

Substituting $x_i = f_i/\tilde{f}_i$ and taking the expectations with respect to the normalized distribution $q/\int_{\mathbf{x}} q(\mathbf{x})d\mathbf{x}$ gives exactly the bounds in the theorem. □

The upper bound (67b) is equivalent to that of Wainwright et al. (2005b), who proved it in the case where $p(\mathbf{x})$ was an exponential family, but in fact it holds for any nonnegative $p(\mathbf{x})$. Appendix D provides an alternative proof of (67b), using arguments similar to Wainwright et al. (2005b).

4.5 The free-energy function

Besides its use as an estimate of the model evidence, (66) has another interpretation. As a function of the message parameters $\boldsymbol{\tau}_a$, its stationary points are exactly the fixed points of α-divergence message passing (Minka, 2004; Minka, 2001a). In other words, (66) is the surrogate objective function that message-passing is optimizing, in lieu of the intractable global divergence $D_\alpha(p \,||\, q)$. Because mean-field, belief propagation, expectation propagation, etc. are all instances of α-divergence message passing, (66) describes the surrogate objective for all of them.

Now that we have established the generic message-passing algorithm, let's look at specific instances of it.

5 Mean-field

This section shows that the mean-field method is a special case of the generic message-passing algorithm. In the mean-field method (Jordan et al., 1999; Jaakkola, 2000) we minimize $\mathrm{KL}(q \,||\, p)$, the exclusive KL-divergence. Why should we minimize exclusive KL, versus other divergence measures? Some authors motivate the exclusive KL by the fact that it provides a bound on the model evidence $Z = \int_x p(x)dx$, as shown by (9a). However, theorem 2 shows that there are many other upper and lower bounds we could obtain, by minimizing other divergences. What really makes $\alpha = 0$ special is its computational properties. Uniquely among all α-divergences, it enjoys an equivalence between global and local divergence. Rather than minimize the global divergence directly, we can apply the generic message-passing algorithm of section 4, to get the variational message-passing algorithm of Winn & Bishop (2005). Uniquely for $\alpha = 0$, the message-passing fixed points are exactly the stationary points of the global KL-divergence.

To get variational message-passing, use a fully-factorized approximation with no exponential family constraint (section 3.3). To minimize the local divergence (52), substitute $p(\mathbf{x}) = f_a(\mathbf{x})q^{\setminus a}(\mathbf{x})$ and $q(\mathbf{x}) = \tilde{f}_a(\mathbf{x})q^{\setminus a}(\mathbf{x})$ into the fixed-point scheme for $\alpha = 0$ (34) to get:

$$q_i(x_i)^{\text{new}} \propto \exp\Big(\int_{\mathbf{x}\setminus x_i} \prod_{j\neq i} q_j(x_j) \log f_a(\mathbf{x}) d\mathbf{x}\Big) \times \exp\Big(\int_{\mathbf{x}\setminus x_i} \prod_{j\neq i} q_j(x_j) \log q^{\setminus a}(\mathbf{x}) d\mathbf{x}\Big) \tag{69}$$

$$m_{a\to i}(x_i)^{\text{new}} m_{i\to a}(x_i) \propto \exp\Big(\int_{\mathbf{x}\setminus x_i} \prod_{j\neq i} m_{a\to j}(x_j) m_{j\to a}(x_j) \log f_a(\mathbf{x}) d\mathbf{x}\Big) \times \exp\Big(\Big(\int_{\mathbf{x}\setminus x_i} \prod_{j\neq i} q_j(x_j) d\mathbf{x}\Big) \log m_{i\to a}(x_i)\Big) \tag{70}$$

$$m_{a\to i}(x_i)^{\text{new}} \propto \exp\Big(\int_{\mathbf{x}\setminus x_i} \prod_{j\neq i} m_{a\to j}(x_j) m_{j\to a}(x_j) \log f_a(\mathbf{x}) d\mathbf{x}\Big) \tag{71}$$

Applying the template of section 4.1, the algorithm becomes:

Variational message-passing

- Initialize $m_{a\to i}(x_i)$ for all (a, i).
- Repeat until all $m_{a\to i}$ converge:
  1. Pick a factor a.
  2. Compute the messages into the factor via (54).
  3. Compute the messages out of the factor via (71).

The above algorithm is for general factors f_a. However, because VMP does not project the messages onto an exponential family, they can get arbitrarily complex. (Section 6 discusses this issue with belief propagation.) The only way to control the message complexity is to restrict the factors f_a to already be in an exponential family. This is the restriction on f_a adopted by Winn & Bishop (2005).

Now we show that this algorithm has the same fixed points as the global KL-divergence. Let q have the exponential

form (19) with free parameters ν_j. The global divergence is

$$\mathrm{KL}(q \,||\, p) = \int_{\mathbf{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x} + \int_{\mathbf{x}} (p(\mathbf{x}) - q(\mathbf{x})) d\mathbf{x} \tag{72}$$

Zeroing the derivative with respect to ν_j gives the stationary condition:

$$\frac{d}{d\nu_j} \mathrm{KL}(q \,||\, p) = \int_{\mathbf{x}} g_j(\mathbf{x}) q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x} = 0 \tag{73}$$

Define the matrices $\mathbf{H}$ and $\mathbf{B}$ with entries

$$h_{jk} = \int_{\mathbf{x}} g_j(\mathbf{x}) g_k(\mathbf{x}) q(\mathbf{x}) d\mathbf{x} \tag{74}$$

$$b_{aj} = \int_{\mathbf{x}} g_j(\mathbf{x}) q(\mathbf{x}) \log f_a(\mathbf{x}) d\mathbf{x} \tag{75}$$

Substituting the exponential form of q into (73) gives

$$\mathbf{H}\boldsymbol{\nu} - \sum_a \mathbf{b}_a = 0 \tag{76}$$

In message-passing, the local divergence for factor a is

$$\mathrm{KL}(q(\mathbf{x}) \,||\, f_a(\mathbf{x}) q^{\backslash a}(\mathbf{x})) = \tag{77}$$
$$\int_{\mathbf{x}} q(\mathbf{x}) \log \frac{\tilde{f}_a(\mathbf{x})}{f_a(\mathbf{x})} d\mathbf{x} + \int_{\mathbf{x}} (f_a(\mathbf{x}) - \tilde{f}_a(\mathbf{x})) q^{\backslash a}(\mathbf{x}) d\mathbf{x}$$

Here the free parameters are the τ_{aj} from (49). The derivative of the local divergence with respect to τ_{aj} gives the stationary condition:

$$\int_{\mathbf{x}} g_j(\mathbf{x}) q(\mathbf{x}) \log \frac{\tilde{f}_a(\mathbf{x})}{f_a(\mathbf{x})} d\mathbf{x} = 0 \tag{78}$$

$$\mathbf{H}\boldsymbol{\tau}_a - \mathbf{b}_a = 0 \tag{79}$$

$$\text{where } \sum_a \boldsymbol{\tau}_a = \boldsymbol{\nu} \tag{80}$$

Now we show that the conditions (76) and (79) are equivalent. In one direction, if we have $\boldsymbol{\tau}$'s satisfying (79) and (80), then we have a $\boldsymbol{\nu}$ satisfying (76). In the other direction, if we have a $\boldsymbol{\nu}$ satisfying (76), then we can compute $(\mathbf{H}, \mathbf{B})$ from (74,75) and solve for $\boldsymbol{\tau}_a$ in (79). (If $\mathbf{H}$ is singular, there may be multiple valid $\boldsymbol{\tau}$'s.) The resulting $\boldsymbol{\tau}$'s will satisfy (80), providing a valid message-passing fixed point. Thus a message-passing fixed point implies a global fixed point and vice versa.

From the discussion in section 2, we expect that in multimodal cases this method will represent the most massive mode of p. When the modes are equally massive, it will pick one of them at random. This symmetry-breaking property is discussed by Jaakkola (2000). Sometimes symmetry-breaking is viewed as a problem, while other times it is exploited as a feature.

6 Belief Propagation and EP

This section describes how to obtain loopy belief propagation (BP) and expectation propagation (EP) from the generic message-passing algorithm. In both cases, we locally minimize $\mathrm{KL}(p \,||\, q)$, the inclusive KL-divergence. Unlike the mean-field method, we do not necessarily minimize global KL-divergence exactly. However, if inclusive KL is what you want to minimize, then BP and EP do a better job than mean-field.

To get loopy belief propagation, use a fully-factorized approximation with no explicit exponential family constraint (section 3.3). This is equivalent to using an exponential family with lots of indicator features:

$$g_{ij}(x_i) = \delta(x_i - j) \tag{81}$$

where j ranges over the domain of x_i. Since $\alpha = 1$, the fully-factorized message equation (60) becomes:

$$m_{a \to i}(x_i)' \propto \int_{\mathbf{x} \backslash x_i} f_a(\mathbf{x}) \prod_{j \neq i} m_{j \to a}(x_j) d\mathbf{x} \tag{82}$$

Applying the template of section 4.1, the algorithm is:

Loopy belief propagation

- Initialize $m_{a \to i}(x_i)$ for all (a, i).
- Repeat until all $m_{a \to i}$ converge:
  1. Pick a factor a.
  2. Compute the messages into the factor via (54).
  3. Compute the messages out of the factor via (82), and apply a step-size ϵ.

It is possible to improve the performance of belief propagation by clustering variables together, corresponding to a partially-factorized approximation. However, the cost of the algorithm grows rapidly with the amount of clustering, since the messages get exponentially more complex.

Because BP does not project the messages onto an exponential family, they can have unbounded complexity. When discrete and continuous variables are mixed, the messages in belief propagation can get exponentially complex. Consider a dynamic Bayes net with a continuous state whose dynamics is controlled by discrete hidden switches (Heskes & Zoeter, 2002). As you go forward in time, the state distribution acquires multiple modes due to the unknown switches. The number of modes is multiplied at every time step, leading to an exponential increase in message complexity through the network. The only way to control the complexity of BP is to restrict the factors to already be in an exponential family. In practice, this limits BP to fully-discrete or fully-Gaussian networks.

Expectation propagation (EP) is an extension of belief propagation which fixes these problems. The essential dif-

ference between EP and BP is that EP imposes an exponential family constraint on the messages. This is useful in two ways. First, by bounding the complexity of the messages, it provides practical message-passing in general networks with continuous variables. Second, EP reduces the cost of clustering variables, since you don't have to compute the exact joint distribution of a cluster. You could fit a jointly Gaussian approximation to the cluster, or you could fit a tree-structured approximation to the cluster (Minka & Qi, 2003).

With an exponential family constraint, the fully-factorized message equation (60) becomes:

$$m_{a\to i}(x_i)' \propto \frac{1}{m_{i\to a}(x_i)} \text{proj}\left[m_{i\to a}(x_i) \int_{\mathbf{x}\setminus x_i} f_a(\mathbf{x}) \prod_{j\neq i} m_{j\to a}(x_j) d\mathbf{x}\right] \quad (83)$$

Applying the template of section 4.1, the algorithm is:

Expectation propagation

- Initialize $m_{a\to i}(x_i)$ for all (a, i).
- Repeat until all $m_{a\to i}$ converge:
  1. Pick a factor a.
  2. Compute the messages into the factor via (54).
  3. Compute the messages out of the factor via (83), and apply a step-size ϵ.

7 Fractional BP and Power EP

This section describes how to obtain fractional belief propagation (FBP) and power expectation propagation (Power EP) from the generic message-passing algorithm. In this case, we locally minimize any α-divergence.

Previous sections have already derived the relevant equations. The algorithm of section 4.1 already implements Power EP. Fractional BP excludes the exponential family projection. If you drop the projection in the fully-factorized message equation (60), you get:

$$m_{a\to i}(x_i)' \propto m_{a\to i}(x_i)^{1-\alpha} \times \int_{\mathbf{x}\setminus x_i} f_a(\mathbf{x})^\alpha \prod_{j\neq i} m_{a\to j}(x_j)^{1-\alpha} m_{j\to a}(x_j) d\mathbf{x} \quad (84)$$

Equating $m_{a\to i}(x_i)$ on both sides gives

$$m_{a\to i}(x_i)' \propto \left(\int_{\mathbf{x}\setminus x_i} f_a(\mathbf{x})^\alpha \prod_{j\neq i} m_{a\to j}(x_j)^{1-\alpha} m_{j\to a}(x_j) d\mathbf{x}\right)^{1/\alpha} \quad (85)$$

which is the message equation for fractional BP.

8 Tree-reweighted message passing

This section describes how to obtain tree-reweighted message passing (TRW) from the generic message-passing algorithm. In the description of Wainwright et al. (2005b), tree-reweighted message passing is an algorithm for computing an upper bound on the partition function $Z = \int_x p(x)dx$. However, TRW can also be viewed as an inference algorithm which approximates a distribution p by minimizing α-divergence. In fact, TRW is a special case of fractional BP.

In tree-reweighted message passing, each factor f_a is assigned an *appearance probability* $\mu(a) \in (0, 1]$. Let the power $\alpha_a = 1/\mu(a)$. The messages $M_{ts}(x_s)$ in Wainwright et al. (2005b) are equivalent to $m_{a\to i}(x_i)^{\alpha_a}$ in this paper. In the notation of this paper, the message equation of Wainwright et al. (2005b) is:

$$m_{a\to i}(x_i)^{\alpha_a} \propto \int_{\mathbf{x}\setminus x_i} f_a(\mathbf{x})^{\alpha_a} \prod_{j\neq i} \frac{\prod_{b\neq a} m_{b\to j}(x_j)}{m_{a\to j}(x_j)^{\alpha_a - 1}} d\mathbf{x} \quad (86)$$

$$= \int_{\mathbf{x}\setminus x_i} f_a(\mathbf{x})^{\alpha_a} \prod_{j\neq i} m_{a\to j}(x_j)^{1-\alpha_a} m_{j\to a}(x_j) d\mathbf{x} \quad (87)$$

This is exactly the fractional BP update (85). The TRW update is therefore equivalent to minimizing local α-divergence. The constraint $0 < \mu(a) \leq 1$ requires $\alpha_a \geq 1$. Note that (86) applies to factors of any degree, not just pairwise factors as in Wainwright et al. (2005b).

The remaining question is how to obtain the upper bound formula of Wainwright et al. (2005b). Because $\sum_a \mu(a) \neq 1$ in general, the upper bound in (67b) does not directly apply. However, if we redefine the factors in the bound to correspond to the trees in TRW, then (67b) gives the desired upper bound.

Specifically, let A be any subset of factors f_a, and let $\mu(A)$ be a normalized distribution over all possible subsets. In TRW, $\mu(A) > 0$ only for spanning trees, but this is not essential. Let $\mu(a)$ denote the appearance probability of factor a, i.e. the sum of $\mu(A)$ over all subsets containing factor a. For each subset A, define the *factor-group* f_A according to:

$$f_A(\mathbf{x}) = \prod_{a\in A} f_a(\mathbf{x})^{\mu(A)/\mu(a)} \quad (88)$$

These factor-groups define a valid factorization of p:

$$p(\mathbf{x}) = \prod_A f_A(\mathbf{x}) \quad (89)$$

This is true because of the definition of $\mu(a)$. Similarly, if

$q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x})$, then we can define approximate factor-groups $\tilde{f}_A$ according to:

$$\tilde{f}_A(\mathbf{x}) = \prod_{a \in A} \tilde{f}_a(\mathbf{x})^{\mu(A)/\mu(a)} \tag{90}$$

which provide another factorization of q:

$$q(\mathbf{x}) = \prod_A \tilde{f}_A(\mathbf{x}) \tag{91}$$

Now plug this factorization into (66), using powers $\alpha_A = 1/\mu(A)$:

$$\tilde{Z} = \prod_A \left(\int_{\mathbf{x}} \left(\frac{f_A(\mathbf{x})}{\tilde{f}_A(\mathbf{x})} \right)^{\alpha_A} q(\mathbf{x}) d\mathbf{x} \right)^{1/\alpha_A} \tag{92}$$

Because $\sum_A \mu(A) = 1$, we have $\sum_A 1/\alpha_A = 1$. By theorem 4, (92) is an upper bound on Z. When restricted to spanning trees, it gives exactly the upper bound of Wainwright et al. (2005b) (their equation 16). To see the equivalence, note that $\exp(\Phi(\theta(T)))$ in their notation is the same as $\int_{\mathbf{x}} \left(\frac{f_A(\mathbf{x})}{\tilde{f}_A(\mathbf{x})} \right)^{\alpha_A} q(\mathbf{x}) d\mathbf{x}$, because of their equations 21, 22, 58, and 59.

In Wainwright et al. (2005b), it was observed that TRW sometimes achieves better estimates of the marginals than BP. This seems to contradict the result of section 3, that $\alpha = 1$ is the best at estimating marginals. However, in section 4.2, we saw that sometimes it is better for message-passing to use a local divergence which is different from the global one we want to minimize. In particular, this is true when the network has purely attractive couplings. Indeed, this was the good case for TRW observed by Wainwright et al. (2005b) (their figures 7b and 9b).

9 Choosing a divergence measure

This section gives general guidelines for choosing a divergence measure in message passing. There are three main considerations: computational complexity, the approximating family, and the inference goal.

First, the reason we make approximations is to save computation, so if a divergence measure requires a lot of work to minimize, we shouldn't use it. Even among α-divergences, there can be vast differences in computational complexity, depending on the specific factors involved. Some divergences also have lots of local minima, to trap a would-be optimizer. So an important step in designing a message passing algorithm should be to determine which divergences are the easiest to minimize on the given problem.

Next we have the approximating family. If the approximating family is a good fit to the true distribution, then it doesn't matter which divergence measure you use, since all will give similar results. The only consideration at that point is computational complexity. If the approximating family is a poor fit to the true distribution, then you are probably safest to use an exclusive divergence, which only tries to model one mode. With an inclusive divergence, message passing probably won't converge at all. If the approximating family is a medium fit to the true distribution, then you need to consider the inference goal.

For some tasks, there are uniquely suited divergence measures. For example, χ^2 divergence is well-suited for choosing the proposal density for importance sampling (appendix E). If the task is to compute marginal distributions, using a fully-factorized approximation, then the best choice (among α-divergences) is inclusive KL ($\alpha = 1$), because it is the only α which strives to preserve the marginals. Papers that compare mean-field versus belief propagation at estimating marginals invariably find belief propagation to be better (Weiss, 2001; Minka & Qi, 2003; Kappen & Wiegerinck, 2001; Mooij & Kappen, 2004). This is because mean-field is optimizing for a different task. Just because the approximation is factorized does not mean that the factors are supposed to approximate the marginals of p—it depends on what divergence measure they optimize. The inclusive KL should also be preferred for estimating the integral of p (the partition function, see section 2) or other simple moments of p.

If the task is Bayesian learning, the situation is more complicated. Here $\mathbf{x}$ is a parameter vector, and $p(\mathbf{x}) \equiv p(\mathbf{x}|D)$ is the posterior distribution given training data. The predictive distribution for future data y is $\int_{\mathbf{x}} p(y|\mathbf{x})p(\mathbf{x})d\mathbf{x}$. To simplify this computation, we'd like to approximate $p(\mathbf{x})$ with $q(\mathbf{x})$ and predict using $\int_{\mathbf{x}} p(y|\mathbf{x})q(\mathbf{x})d\mathbf{x}$. Typically, we are not interested in $q(\mathbf{x})$ directly, but only this predictive distribution. Thus a sensible error measure is the divergence between the predictive distributions:

$$D\left(\int_{\mathbf{x}} p(y|\mathbf{x})p(\mathbf{x})d\mathbf{x} \,\middle|\middle|\, \int_{\mathbf{x}} p(y|\mathbf{x})q(\mathbf{x})d\mathbf{x} \right) \tag{93}$$

Because $p(y|\mathbf{x})$ is a fixed function, this is a valid objective for $q(\mathbf{x})$. Unfortunately, it is different from the divergence measures we've looked at so far. The measures so far compare p to q point-by-point, while (93) takes averages of p and compares these to averages of q. If we want to use algorithms for α-divergence, then we need to find the α most similar to (93).

Consider binary classification with a likelihood of the form $p(y = \pm 1|\mathbf{x}, \mathbf{z}) = \phi(y\mathbf{x}^{\mathrm{T}}\mathbf{z})$, where $\mathbf{z}$ is the input vector, y is the label, and ϕ is a step function. In this case, the predictive probability that $y = 1$ is $Pr(\mathbf{x}^{\mathrm{T}}\mathbf{z} > 0)$ under the (normalized) posterior for $\mathbf{x}$. This is equivalent to projecting the unnormalized posterior onto the line $\mathbf{x}^{\mathrm{T}}\mathbf{z}$, and measuring the total mass above zero, compared to below zero. These one-dimensional projections might look like the distributions in figure 1. By fitting a Gaussian to $p(\mathbf{x})$, we make all these projections Gaussian, which may alter

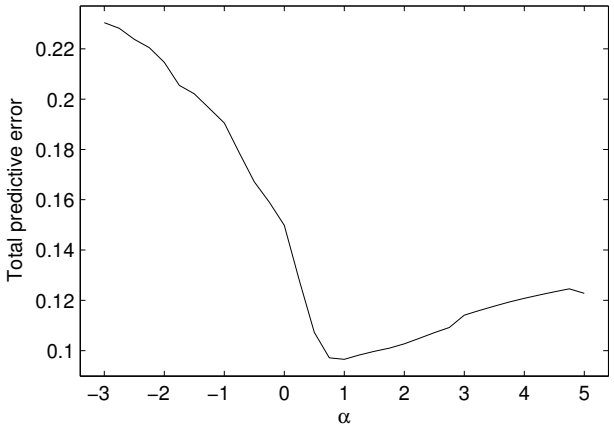


Figure 7: Average predictive error for various alpha-divergences on a mock classification task.

the total mass above/below zero. A good $q(\mathbf{x})$ is one which preserves the correct mass on each side of zero; no other properties matter.

To find the α-divergence which best captures this error measure, we ran the following experiment. We first sampled 210 random one-dimensional mixtures of two Gaussians (means from $\mathcal{N}(0, 4)$, variances from squaring $\mathcal{N}(0, 1)$, scale factors uniform on $[0, 1]$). For each one, we fit a Gaussian by minimizing α-divergence, for several values of α. After optimization, both p and q were normalized, and we computed $p(x > 0)$ and $q(x > 0)$. The predictive error was defined to be the absolute difference $|p(x > 0) - q(x > 0)|$. (KL-divergence to $p(x > 0)$ gives similar results.) The average error for each α value is plotted in figure 7. The best predictions came from $\alpha = 1$ and in general from the inclusive divergences versus the exclusive ones. Exclusive divergences perform poorly because they tend to give extreme predictions (all mass on one side). So $\alpha = 1$ seems to be the best substitute for (93) on this task.

A task in which exclusive divergences are known to do well is Bayesian learning of mixture models, where each component has separate parameters. In this case, the predictive distribution depends on the posterior in a more complex way. Specifically, the predictive distribution is invariant to how the mixture components are indexed. Thus $\int_{\mathbf{x}} p(y|\mathbf{x})p(\mathbf{x})d\mathbf{x}$ is performing a non-local type of averaging—over all ways of permuting the elements of $\mathbf{x}$. This is hard to capture with a point-wise divergence. For example, if our prior is symmetric with respect to the parameters and we condition on data, then any mode in the posterior will have a mirror copy corresponding to swapping components. Minimizing an inclusive divergence will waste resources by trying to represent all of these identical modes. An exclusive divergence, however, will focus on one mode. This doesn't completely solve the problem, since there may be multiple modes of the likelihood even for one component ordering, but it performs well in practice. This is an example of a problem where, because of the complexity of the posterior, it is safest to use an exclusive divergence. Perhaps with a different approximating family, e.g. one which assumes symmetrically placed modes, inclusive divergence would also work well.

10 Future work

The perspective of information divergences offers a variety of new research directions for the artificial intelligence community. For example, we could construct information divergence interpretations of other message-passing algorithms, such as generalized belief propagation (Yedidia et al., 2004), max-product versions of BP and TRW (Wainwright et al., 2005a), Laplace propagation (Smola et al., 2003), and bound propagation (Leisink & Kappen, 2003). We could improve the performance of Bayesian learning (section 9) by finding more appropriate divergence measures and turning them into message-passing algorithms. In networks with long-range correlations, it is difficult to predict the best local divergence measure (section 4.2). Answering this question could significantly improve the performance of message-passing on hard networks. By continuing to assemble the pieces of the inference puzzle, we can make Bayesian methods easier for everyone to enjoy.

Acknowledgment

Thanks to Martin Szummer for corrections to the manuscript.

References

Ali, S. M., & Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *J Royal Stat Soc B*, *28*, 131–142.

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

Frey, B. J., & MacKay, D. J. (1997). A revolution: Belief propagation in graphs with cycles. *NIPS*.

Frey, B. J., Patrascu, R., Jaakkola, T., & Moran, J. (2000). Sequentially fitting inclusive trees for inference in noisy-OR networks. *NIPS 13*.

Heskes, T. (2003). Stable fixed points of loopy belief propagation are minima of the Bethe free energy. *NIPS*.

Heskes, T., & Zoeter, O. (2002). Expectation propagation for approximate inference in dynamic Bayesian networks. *Proc UAI*.

Jaakkola, T. (2000). Tutorial on variational approximation methods. *Advanced mean field methods: theory and practice*. MIT Press.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Learning in Graphical Models*. MIT Press.

Kappen, H. J., & Wiegerinck, W. (2001). Novel iteration schemes for the cluster variation method. *NIPS 14*.

Leisink, M. A. R., & Kappen, H. J. (2003). Bound propagation. *J Artificial Intelligence Research*, *19*, 139–154.

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. (2004). Power EP (Technical Report MSR-TR-2004-149). Microsoft Research Ltd.

Minka, T. P., & Qi, Y. (2003). Tree-structured approximations by expectation propagation. *NIPS*.

Mooij, J., & Kappen, H. (2004). Validity estimates for loopy belief propagation on binary real-world networks. *NIPS*.

Peterson, C., & Anderson, J. (1987). A mean field theory learning algorithm for neural networks. *Complex Systems*, *1*, 995–1019.

Smola, A., Vishwanathan, S. V., & Eskin, E. (2003). Laplace propagation. *NIPS*.

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

Wainwright, M. J., Jaakkola, T. S., & Willsky, A. S. (2005a). MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. *IEEE Transactions on Information Theory*. To appear.

Wainwright, M. J., Jaakkola, T. S., & Willsky, A. S. (2005b). A new class of upper bounds on the log partition function. *IEEE Trans. on Information Theory*, *51*, 2313–2335.

Weiss, Y. (2001). Comparing the mean field method and belief propagation for approximate inference in MRFs. *Advanced Mean Field Methods*. MIT Press.

Welling, M., Minka, T., & Teh, Y. W. (2005). Structured Region Graphs: Morphing EP into GBP. *UAI*.

Wiegerinck, W. (2000). Variational approximations between mean field theory and the junction tree algorithm. *UAI*.

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.

Yedidia, J. S., Freeman, W. T., & Weiss, Y. (2004). Constructing free energy approximations and generalized belief propagation algorithms (Technical Report). MERL Research Lab. `www.merl.com/publications/TR2004-040/`.

Zhu, H., & Rohwer, R. (1995). Information geometric measurements of generalization (Technical Report NCRG/4350). Aston University.

A Ali-Silvey divergences

Ali & Silvey (1966) defined a family of convex divergence measures which includes α-divergence as a special case. These are sometimes called f-divergences because they are parameterized by the choice of a convex function f. Some properties of the α-divergence are easier to prove by thinking of it as an instance of an f-divergence. With appropriate corrections to handle unnormalized distributions, the general formula for an f-divergence is

$$D_f(p \,||\, q) = \frac{1}{f''(1)} \int q(x) f\left(\frac{p(x)}{q(x)}\right) + (f'(1) - f(1))q(x) - f'(1)p(x)dx \quad (94)$$

where f is any convex or concave function (concave functions are turned into convex ones by the f'' term). Evaluating $f''(r)$ at $r = 1$ is arbitrary; only the sign of f'' matters. Some examples:

$$\mathrm{KL}(q \,||\, p) : f(r) = \log(r) \qquad \begin{matrix} f'(1) = 1 \\ f''(1) = -1 \end{matrix} \quad (95)$$

$$\mathrm{KL}(p \,||\, q) : f(r) = r\log(r) \qquad \begin{matrix} f'(1) = 1 \\ f''(1) = 1 \end{matrix} \quad (96)$$

$$D_\alpha(p \,||\, q) : f(r) = r^\alpha \qquad \begin{matrix} f'(1) = \alpha \\ f''(1) = -\alpha(1-\alpha) \end{matrix} \quad (97)$$

The L_1 distance $\int |p(x) - q(x)|dx$ can be obtained as $f(r) = |r - 1|$ if we formally define $(f'(1) = 0, f''(1) = 1)$, for example by taking a limit. The f-divergences are a large class, but they do not include e.g. the L_2 distance $\int (p(x) - q(x))^2 dx$.

The derivatives with respect to p and q are:

$$\frac{dD_f(p \,||\, q)}{dp(x)} = \frac{1}{f''(1)} \left(f'\left(\frac{p(x)}{q(x)}\right) - f'(1) \right) \quad (98)$$

$$\frac{dD_f(p \,||\, q)}{dq(x)} = \frac{1}{f''(1)} \left(f\left(\frac{p(x)}{q(x)}\right) - f(1) - \frac{p(x)}{q(x)} f'\left(\frac{p(x)}{q(x)}\right) + f'(1) \right) \quad (99)$$

Therefore the divergence and its derivatives are zero at $p = q$. It can be verified by direct differentiation that D_f is jointly convex in (p, q) (the Hessian is positive semidefinite), therefore it must be ≥ 0 everywhere.

As illustrated by (95,96), you can swap the position of p and q in the divergence by replacing f with $rf(1/r)$ (which is convex if f is convex). Thus p and q can be swapped in the definition (94) without changing the essential family.

B Proof of Theorem 1

Theorem 1 (Liapunov's inequality) If x is a non-negative random variable, and we have two real numbers $\alpha_2 > \alpha_1$, then:

$$E[x^{\alpha_2}]^{1/\alpha_2} \geq E[x^{\alpha_1}]^{1/\alpha_1} \tag{100}$$

where $\alpha = 0$ is interpreted as the limit

$$\lim_{\alpha \to 0} E[x_i^{\alpha}]^{1/\alpha} = \exp(E[\log x_i]) \tag{101}$$

Proof: It is sufficient to prove the cases $\alpha_1 \geq 0$ and $\alpha_2 \leq 0$ since the other cases follow by transitivity. If f is a convex function, then Jensen's inequality tells us that

$$E[f(x^{\alpha_1})] \geq f(E[x^{\alpha_1}]) \tag{102}$$

If $\alpha_2 > \alpha_1 > 0$, then $f(x) = x^{\alpha_2/\alpha_1}$ is convex, leading to:

$$E[x^{\alpha_2}] \geq E[x^{\alpha_1}]^{\alpha_2/\alpha_1} \tag{103}$$

$$E[x^{\alpha_2}]^{1/\alpha_2} \geq E[x^{\alpha_1}]^{\alpha_1} \tag{104}$$

If $0 > \alpha_2 > \alpha_1$, then $f(x) = x^{\alpha_2/\alpha_1}$ is concave, leading to:

$$E[x^{\alpha_2}] \leq E[x^{\alpha_1}]^{\alpha_2/\alpha_1} \tag{105}$$

$$E[x^{\alpha_2}]^{1/\alpha_2} \geq E[x^{\alpha_1}]^{\alpha_1} \tag{106}$$

If $\alpha_2 > \alpha_1 = 0$, Jensen's inequality for the logarithm says

$$E[\log x^{\alpha_2}] \leq \log E[x_i^{\alpha_2}] \tag{107}$$

$$\alpha_2 E[\log x_i] \leq \log E[x_i^{\alpha_2}] \tag{108}$$

$$E[\log x_i] \leq \frac{1}{\alpha_2} \log E[x_i^{\alpha_2}] \tag{109}$$

$$\exp(E[\log x_i]) \leq E[x_i^{\alpha_2}]^{1/\alpha_2} \tag{110}$$

If $0 = \alpha_2 > \alpha_1$, Jensen's inequality for the logarithm says

$$E[\log x^{\alpha_1}] \leq \log E[x_i^{\alpha_1}] \tag{111}$$

$$\alpha_1 E[\log x_i] \leq \log E[x_i^{\alpha_1}] \tag{112}$$

$$E[\log x_i] \geq \frac{1}{\alpha_1} \log E[x_i^{\alpha_1}] \tag{113}$$

$$\exp(E[\log x_i]) \geq E[x_i^{\alpha_1}]^{1/\alpha_1} \tag{114}$$

This proves all cases. □

C Hölder inequalities

Theorem 5 For any set of non-negative random variables $x_1, ..., x_n$ (not necessarily independent) and a set of positive numbers $\alpha_1, ..., \alpha_n$ satisfying $\sum_i 1/\alpha_i \leq 1$:

$$E[\textstyle\prod_i x_i] \leq \prod_i E[x_i^{\alpha_i}]^{1/\alpha_i} \tag{115}$$

Proof: Start with the case $\sum_i 1/\alpha_i = 1$. By Jensen's inequality for the logarithm we know that

$$\log(\sum_i \frac{x_i^{\alpha_i}}{\alpha_i}) \geq \sum_i \frac{1}{\alpha_i} \log(x_i^{\alpha_i}) = \sum_i \log(x_i) \tag{116}$$

Reversing this gives:

$$\prod_i x_i \leq \sum_i x_i^{\alpha_i}/\alpha_i \tag{117}$$

Now consider the ratio of the lhs of (115) over the rhs:

$$\frac{E[\prod_i x_i]}{\prod_i E[x_i^{\alpha_i}]^{1/\alpha_i}} = E\left[\prod_i \frac{x_i}{E[x_i^{\alpha_i}]^{1/\alpha_i}}\right] \tag{118}$$

$$\leq E\left[\sum_i \frac{1}{\alpha_i} \frac{x_i^{\alpha_i}}{E[x_i^{\alpha_i}]}\right] \quad \text{by (117)} \tag{119}$$

$$= \sum_i \frac{1}{\alpha_i} \frac{E[x_i^{\alpha_i}]}{E[x_i^{\alpha_i}]} = 1 \tag{120}$$

Now if $\sum_i 1/\alpha_i < 1$, this means some α_i is larger than needed. By Th. 1, this will only increase the right hand side of (115). □

Theorem 6 For any set of non-negative random variables $x_1, ..., x_n$ and a set of non-positive numbers $\alpha_1, ..., \alpha_n \leq 0$:

$$E[\textstyle\prod_i x_i] \geq \prod_i E[x_i^{\alpha_i}]^{1/\alpha_i} \tag{121}$$

where the case $\alpha_i = 0$ is interpreted as the limit in (101).

Proof: By Th.1, this inequality is tightest for $\alpha_i = 0$. By Jensen's inequality for the logarithm, we know that

$$\log E[\textstyle\prod_i x_i] \geq E[\log \prod_i x_i] = \sum_i E[\log x_i] \tag{122}$$

This proves the case $\alpha_i = 0$ for all i:

$$E[\textstyle\prod_i x_i] \geq \prod_i \exp(E[\log x_i]) \tag{123}$$

By Th.1, setting $\alpha_i < 0$ will only decrease the right hand side. □

D Alternate upper bound proof

Define an exponential family with parameters $(\boldsymbol{\nu}, \mathbf{a})$:

$$p(x; \boldsymbol{\nu}, \mathbf{a}) = Z(\boldsymbol{\nu}, \mathbf{a})^{-1} \exp(\textstyle\sum_k a_k \log f_k(x) + \sum_j \nu_j g_j(x)) \quad (124)$$

where $\log Z(\boldsymbol{\nu}, \mathbf{a}) =$

$$\log \int_x \exp(\textstyle\sum_k a_k \log f_k(x) \ + \ \sum_j \nu_j g_j(x)) dx \quad (125)$$

Because it is the partition function of an exponential family, $\log Z$ is convex in $(\boldsymbol{\nu}, \mathbf{a})$. Define a set of parameter vectors $((\boldsymbol{\lambda}_1, \mathbf{a}_1), ..., (\boldsymbol{\lambda}_n, \mathbf{a}_n))$ and non-negative weights $c_1, ..., c_n$ which sum to 1. Then Jensen's inequality says

$$\log Z\left(\textstyle\sum_i c_i \boldsymbol{\lambda}_i, \sum_i c_i \mathbf{a}_i\right) \leq \sum_i c_i \log Z(\boldsymbol{\lambda}_i, \mathbf{a}_i) \quad (126)$$

$$\text{where } \sum_i c_i = 1 \quad (127)$$

Because it is a sum of convex functions, this upper bound is convex in $((\boldsymbol{\lambda}_1, \mathbf{a}_1), ..., (\boldsymbol{\lambda}_n, \mathbf{a}_n))$. The integral that we are trying to bound is $\int_x p(x)dx = Z(\mathbf{0}, \mathbf{1})$. Plugging this into (126) and exponentiating gives

$$\int_x p(x)dx \leq \prod_i Z(\boldsymbol{\lambda}_i, \mathbf{a}_i)^{c_i} \quad (128)$$

$$\text{provided that } \sum_i c_i \boldsymbol{\lambda}_i = \mathbf{0} \quad (129)$$

$$\sum_i c_i \mathbf{a}_i = \mathbf{1} \quad (130)$$

Choose $\mathbf{a}_i$ to be the vector with $1/c_i$ in position i and 0 elsewhere. This satisfies (130) and the bound simplifies to:

$$\int_x p(x)dx \leq \prod_i \left(\int_x f_i(x)^{1/c_i} \exp(\textstyle\sum_j \lambda_{ij} g_j(x)) dx \right)^{c_i} \quad (131)$$

$$\text{provided that } \sum_i c_i \boldsymbol{\lambda}_i = \mathbf{0} \quad (132)$$

To put this in the notation of (66), define

$$c_i = 1/\alpha_i \quad (133)$$

$$\boldsymbol{\lambda}_i = \sum_j \boldsymbol{\tau}_j - \alpha_i \boldsymbol{\tau}_i \quad (134)$$

where $\boldsymbol{\tau}_i$ is the parameter vector of $\tilde{f}_i(x)$ via (49). This definition automatically satisfies (132) and makes (131) reduce to (67b), which is what we wanted to prove.

E Alpha-divergence and importance sampling

Alpha-divergence has a close connection to importance sampling. Suppose we wish to estimate $Z = \int_x p(x)dx$. In importance sampling, we draw n samples from a normalized proposal distribution $q(x)$, giving $x_1, ..., x_n$. Then Z is estimated by:

$$\tilde{Z} = \frac{1}{n} \sum_i \frac{p(x_i)}{q(x_i)} \quad (135)$$

This estimator is unbiased, because

$$E[\tilde{Z}] = \frac{1}{n} \sum_i \int_x \frac{p(x)}{q(x)} q(x) dx = Z \quad (136)$$

The variance of the estimate (across different random draws) is

$$\text{var}(\tilde{Z}) = \frac{1}{n^2} \int_x \frac{p(x)^2}{q(x)^2} q(x) dx - \frac{1}{n^2} Z^2 \quad (137)$$

An optimal proposal distribution minimizes $\text{var}(\tilde{Z})$, i.e. it minimizes $\int_x \frac{p(x)^2}{q(x)} dx$ over q. This is equivalent to minimizing α-divergence with $\alpha = 2$. Hence the problem of selecting an optimal proposal distribution for importance sampling is equivalent to finding a distribution with small α-divergence to p.