Message-Passing Formulations of Approximate Inference Algorithms

Boris Yangel

March 3, 2013

Abstract

This note covers message-passing formulations of different approximate inference algorithms such as expectation propagation, variational inference and Gibbs sampling. For each algorithm the general framework is described, as well as the assumptions that lead to the message-passing formulation. Derivations of message formulas are also provided.

1 Notation

We denote groups of random variables by X, possibly with subscripts, while single variables are denoted by x (again, with subscripts). For instance, $x_j \in X_i$ means that the variable x_j belongs to the group of variables X_i. We also use shortened notation $X^{\setminus j}$ for $X \setminus x_j$.

In all the sections $P(X)$ stands for the distribution being approximated, while $Q(X)$ stands for an approximation. Marginals $\int_{X^{\setminus j}} P(X) dX^{\setminus j}$ and $\int_{X^{\setminus j}} Q(X) dX^{\setminus j}$ are denoted by $P(x_j)$ and $Q(x_j)$.

2 Introduction

The task of an approximate inference algorithm is to approximate given unnormalized probability distribution $P(X)$ with some other distribution $Q(X)$ of a simpler form. In practice, $P(X)$ usually corresponds to a posterior distribution over X obtained from some joint distribution $P(X, Z)$ by fixing Z to an observed value, and we are interested in a normalized approximation which will allow to compute posterior expectations or some other characteristics that depend on the normalizer.

It can be useful to provide some factorization of the approximated $P(X)$ explicitly, like

$$P(X) \propto \prod_i f_i(X_i), \tag{1}$$

since this additional structure can then be exploited by inference algorithms. Factorized probability distributions are usually represented using Bayesian networks, Markov random fields or factor graphs, a nice generalization of the former two.

Quality of the approximation is usually measured using Kullback-Leibler (KL) divergence

$$\mathrm{KL}\Big(P_1(X)\,||\,P_2(X)\Big) = \int_X P_1(X)\ln\frac{P_1(X)}{P_2(X)}dX. \tag{2}$$

Note that this measure is non-symmetric, and, thus, gives us two ways to measure the approximation quality. We will call $\mathrm{KL}\big(Q(X)\,||\,P(X)\big)$, acting as a function of Q, *forward* KL-divergence, while $\mathrm{KL}\big(P(X)\,||\,Q(X)\big)$ will be called *reverse* KL-divergence. Both forward and reverse KL-divergence are members of a richer family of distance measures between probability distributions known as α-divergence family [1].

3 Expectation Propagation

3.1 Useful fact about reverse KL-divergence

Let's say we want to approximate $P(X)$ using a *fully-factorized* approximation

$$Q(X) = \prod_j Q(x_j), \tag{3}$$

where each of the factors $Q(x_j)$ comes from a distribution family F_j. If we choose reverse KL-divergence as a quality measure for our approximation, we can write it down as a function of Q as

$$\begin{aligned}
\mathrm{KL}\Big(P(X)\,||\,Q(X)\Big) &= \int_X P(X)\ln\frac{P(X)}{Q(X)}dX = \\
= \text{const} - \int_X P(X)\ln Q(X)dX &= \text{const} - \sum_j \int_X P(X)\ln Q(x_j)dX = \\
= \text{const} - \sum_j \int_{x_j} \ln Q(x_j)\Big[\int_{X^{\setminus j}} P(X)dX^{\setminus j}\Big]dx_j &= \\
= \text{const} + \sum_j \mathrm{KL}\Big(P(x_j)\,||\,Q(x_j)\Big), &
\end{aligned} \tag{4}$$

where by $P(x_j)$ we denote the corresponding marginal of the distribution $P(X)$. That is, in order to fit a fully factorized approximation to a distribution $P(X)$, we need to set each marginal $Q(x_j)$ to a *KL-projection* of a corresponding marginal of the distribution $P(X)$:

$$Q(x_j) = \arg\min_{q\in F_j} \mathrm{KL}\Big(P(x_j)\,||\,q(x_j)\Big). \tag{5}$$

3.2 EP framework

In the EP framework [2] one approximates distribution $P(X)$ of the form (1) with

$$Q(X) = \frac{1}{\tilde{Z}}\prod_i \tilde{f}_i(X_i), \tag{6}$$

which has the same factorization as $P(X)$. Terms of the approximation are first initialized to some probability distributions and then iteratively refined one-by-one using

$$Q^{\setminus i}(X) \coloneqq \frac{Q(X)}{\tilde{f}_i(X_i)}, \tag{7}$$

$$Q(X) \coloneqq \arg\min_q \mathrm{KL}\Big(\frac{1}{Z} Q^{\setminus i}(X) f_i(X_i) \,||\, q(X)\Big), \tag{8}$$

$$\tilde{f}(X_i) \coloneqq \tilde{Z} \frac{Q(X)}{Q^{\setminus i}(X)}. \tag{9}$$

That is, each factor is removed from the approximation, and then added again in such a way that the new approximation is close to the one having the true factor instead of the approximate one. This process is known as *local* KL-divergence minimization, and it doesn't guarantee to minimize KL-divergence between $P(X)$ and $Q(X)$, although in practice the results are quite similar.

Factors of the approximation are usually constrained to some distribution family, which is rich enough, but allows to compute KL-projection (8) efficiently. The most notable example is exponential family, KL-projection on which can be performed solely by computing the moments of the distribution being approximated [3].

3.3 EP with fully-factorized approximations

Imagine we want to approximate given distribution $P(X)$ of the form (1) with a fully factorized approximation

$$Q(X) = \prod_i \tilde{f}_i(X_i) = \prod_i \prod_{x_j \in X_i} \tilde{f}_{ij}(x_j) = \prod_{x_j} \prod_{i : x_j \in X_i} \tilde{f}_{ij}(x_j) = \prod_{x_j} Q(x_j) \tag{10}$$

using the EP framework. Using the result from section 3.1, the update (8) for factor $\tilde{f}_i(X_i)$ will result in projecting marginals

$$\int_{X^{\setminus j}} \frac{1}{Z} Q^{\setminus i}(X) f_i(X_i) dX^{\setminus j} \tag{11}$$

to a corresponding $Q(x_j)$. This marginals can be represented as

$$\begin{aligned} &\int_{X^{\setminus j}} \frac{1}{Z} Q^{\setminus i}(X) f_i(X_i) dX^{\setminus j} = \\ &= \int_{X^{\setminus j}} \frac{1}{Z} f_i(X_i) \prod_{l \neq i} \prod_{x_m \in X_l} \tilde{f}_{lm}(x_m) dX^{\setminus j} = \\ &= \prod_{\substack{k \neq i \\ x_j \in X_k}} \tilde{f}_{kj}(x_j) \int_{X_i^{\setminus j}} \frac{1}{Z} f_i(X_i) \prod_{x_m \in X_i^{\setminus j}} \prod_{\substack{l \neq i \\ x_m \in X_l}} \tilde{f}_{lm}(x_m) dX_i^{\setminus j} = \tilde{Q}(x_j), \end{aligned} \tag{12}$$

where all the factors not dependent on variables from X_i were integrated to one since the corresponding approximation terms are normalized distributions, and the factors dependent only on x_j were moved out of the integral. Note that if $x_j \notin X_i$, terms under the integral are independent of x_j and integrate to one. Thus, the

corresponding marginals already match $Q(x_j)$, and we should be concerned only about $x_j \in X_i$. For the latter case we now have the an update expression

$$Q(x_j) = \prod_{i:x_j \in X_i} \tilde{f}_{ij}(x_j) \coloneqq \arg\min_q \mathrm{KL}\Big(\tilde{Q}(x_j) \,||\, q(x_j)\Big), \tag{13}$$

or, since on each EP iteration we refine only one factor $\tilde{f}_i$,

$$\tilde{f}_{ij}(x_j) \coloneqq \frac{\arg\min_q \mathrm{KL}\Big(\tilde{Q}(x_j) \,||\, q(x_j)\Big)}{\prod_{\substack{k \neq i \\ x_j \in X_k}} \tilde{f}_{kj}(x_j)}. \tag{14}$$

3.4 Message formulas

Expression (14) can be interpreted in terms of messages from factors to variables and vise versa. Let us define a message from variable x_j to factor f_i as

$$\mu_{x_j \to f_i}(x_j) = \prod_{\substack{k \neq i \\ x_j \in X_k}} \tilde{f}_{kj}(x_j), \tag{15}$$

and a message from factor f_i to variable x_j as

$$\mu_{f_i \to x_j}(x_j) =$$

$$= \frac{\arg\min_q \mathrm{KL}\Big(\int_{X_i^{\setminus j}} f(X_i) \prod_{k:x_k \in X_i} \mu_{x_k \to f_i}(x_k) dX_i^{\setminus j} \,||\, q(x_j)\Big)}{\mu_{x_j \to f_i}(x_j)}. \tag{16}$$

It can be now seen that the approximation update (14) corresponds to first sending messages from all the variables to the factor f_i, and then using the outgoing message from the factor f_i as a new approximation. So, the outgoing messages from factors represent the approximations, and, thus, we can rewrite (15) as

$$\mu_{x_j \to f_i}(x_j) = \prod_{\substack{k \neq i \\ x_j \in X_k}} \mu_{f_k \to x_j}(x_j). \tag{17}$$

Note that messages do not correspond to a normalized probability distributions, while approximation marginals (products of all the incoming messages to a variable) are guaranteed to be normalized.

3.5 EP and belief propagation

One important thing to note about (16) is that if the projected distribution can be represented exactly in the distribution family of $q(x_j)$ (which is the case, for example, with discrete or Gaussian models), then message from x_j in numerator and denominator cancels out and (16) turns into

$$\mu_{f_i \to x_j}(x_j) = \int_{X_i^{\setminus j}} f(X_i) \prod_{\substack{k \neq j \\ x_k \in X_i}} \mu_{x_k \to f_i}(x_k) dX_i^{\setminus j}, \tag{18}$$

which is a well-known belief propagation (BP) message. Thus, BP is a particular case of EP with fully-factorized approximations. However EP is more general, since it provides a recipe for handling distributions in which BP messages are too complex to deal with.

4 Variational Inference

4.1 Variational inference framework

In variational inference framework one seeks an approximation of the fully-factorized form

$$Q(X) = \prod_j Q(x_j), \tag{19}$$

which minimizes the forward KL-divergence $\mathrm{KL}\big(Q(X)\,||\,P(X)\big)$. It can be shown [3] that in this case approximation factors can be refined one-by-one using the expression

$$Q(x_j) \coloneqq \frac{1}{Z} \exp\big\{\mathbb{E}_{X^{\setminus j}\sim Q(X)} \ln P(X)\big\}, \tag{20}$$

where $\mathbb{E}_{X^{\setminus j}\sim Q(X)}$ denotes the expectation w.r.t. the current approximation over all the variables except x_j.

4.2 Variational message passing

For a factorized distribution of the form (1) update (20) can be written as

$$\begin{aligned} Q(x_j) &\coloneqq \frac{1}{Z} \exp\big\{\mathbb{E}_{X^{\setminus j}\sim Q(X)} \sum_i \ln f_i(X_i)\big\} = \\ &= \frac{1}{Z} \prod_i \exp\big\{\mathbb{E}_{X_i^{\setminus j}\sim Q(X)} \ln f_i(X_i)\big\} = \\ &= \frac{1}{Z'} \prod_{i:x_j\in X_i} \exp\big\{\mathbb{E}_{X_i^{\setminus j}\sim Q(X)} \ln f_i(X_i)\big\}, \end{aligned} \tag{21}$$

where all the factors not dependent on the variable x_j were absorbed in the normalizing constant. If we define message from a variable to a factor as

$$\mu_{x_j\to f_i}(x_j) = Q(x_j) \tag{22}$$

and message from a factor to a variable as

$$\mu_{f_i\to x_j}(x_j) = \int_{X_i^{\setminus j}} \ln f_i(X_i) \prod_{\substack{k\neq j \\ x_k\in X_i}} \mu_{x_k\to f_i}(x_k) dX_i^{\setminus j}, \tag{23}$$

variational update (21) can be seen as a message passing procedure, in which each factor receives messages representing factors of the current approximation from neighboring variables, evaluates the expectation with some variable excluded and sends the result to that variable. The variable then, after receiving results from all the neighboring factors, updates the corresponding approximation factor. So, (22) can be rewritten as

$$\mu_{x_j\to f_i}(x_j) \propto \prod_{k:x_j\in X_k} \mu_{f_k\to x_j}(x_j). \tag{24}$$

Note that messages for variational message passing look quite similar to the EP messages. Notable differences are that

- message from a variable to a factor also depends on the incoming message from that factor; EP also has similar "cyclic dependency" in a message from factor to a variable, but it vanishes when no KL projection is required;
- computing message from a factor to a variable involves averaging logarithm of a factor instead of factor itself. For certain problems (like learning the parameters of a Gaussian) it leads to a much simpler message computations than for EP. Several examples are given in [4].

In practice it is useful to restrict all the factors in the model to be conjugate, because otherwise the complexity of the approximations will grow with the number of iterations. A more detailed discussion of this issue, as well as a possible solution, can be found in [4].

5 Gibbs Sampling

5.1 Gibbs sampling framework

Gibbs sampling [3] is quite different from EP and variational inference, since instead of trying to build an approximate representation of $P(X)$, it samples from the true distribution in a MCMC fashion, that is, generates a sequence of samples from distributions that tend to converge to $P(X)$. It is achieved by sampling each variable in turn, conditioned on the rest of the variables:

$$\begin{aligned} x_1^{new} &\sim P(x_1 \mid X^{\setminus 1}), \\ x_2^{new} &\sim P(x_2 \mid X^{\setminus 1\setminus 2}, x_1^{new}), \\ x_3^{new} &\sim P(x_3 \mid X^{\setminus 1\setminus 2\setminus 3}, x_1^{new}, x_2^{new}) \end{aligned}$$

and so on.

5.2 Gibbs sampling as message passing

Let's take a look at the form of the conditional distribution over x_j if $P(X)$ is represented in a factorized form (1):

$$P(x_j \mid X^{\setminus j}) = \frac{1}{Z'} \prod_i f_i(X_i) = \frac{1}{Z''} \prod_{i: x_j \in X_i} f_i(x_j, X_i^{\setminus j}), \tag{25}$$

where all the factors independent of x_j were absorbed in the normalizing constant. So, if we define a message from a factor to a variable as

$$\mu_{f_i \to x_j}(x_j) = f_i(x_j, X_i^{\setminus j}), \tag{26}$$

and a message from a variable to a factor as

$$\mu_{x_j \to f_i}(x_j) = \text{sample}\Big[\frac{1}{Z} \prod_{k: x_j \in X_k} \mu_{f_k \to x_j}(x_j)\Big], \tag{27}$$

then we can think of Gibbs sampling as of message passing procedure, in which every variable receives messages from the neighboring factors, multiplies them together

and normalizes the result to obtain a posterior distribution, which it then uses to sample the new value. This value is then sent back to all the neighboring factors.

In a practical implementation of a message passing framework it would be reasonable to constrain all the incoming messages to a variable to have the same functional form, so that the posterior distribution can be obtained from the incoming messages automatically. This will, however, restrict sampler to models with conjugate factors only. An alternative is to provide a procedure that can sample from distribution represented as any possible product of incoming messages, which seems to be a quite hard goal to achieve.

References

[1] T. P. Minka *et al.*, "Divergence measures and message passing," *Microsoft Research Cambridge, Tech. Rep. MSR-TR-2005-173*, 2005.

[2] T. P. Minka, "Expectation propagation for approximate bayesian inference," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pp. 362–369, Morgan Kaufmann Publishers Inc., 2001.

[3] C. M. Bishop *et al.*, *Pattern recognition and machine learning*, vol. 4. springer New York, 2006.

[4] J. Winn and C. M. Bishop, "Variational message passing," *Journal of Machine Learning Research*, vol. 6, no. 1, p. 661, 2006.