Expectation Propagation for the Clutter Problem – Theory and Implementation

Matěj Korvas

May 23, 2013

Abstract

These lecture notes describe the algorithm of Expectation Propagation as applied to the Clutter problem, touching on the underlying theory. Important is also the practical part where I document my implementation of the algorithm.

1 Introduction

Expectation Propagation (EP for short) was introduced in [2] as an iterated version of the previously known Assumed-Density Filtering approximate inference algorithm. In the work [2], the author also shows how EP is applied to the clutter problem.

In the next section, we describe the EP algorithm in general, in Section 3, we formulate the clutter problem and derive formulas used in EP to solve it, and the final Section 4 discusses our implementation of EP applied to the clutter problem.

2 Expectation Propagation

Expectation Propagation is an approximate inference algorithm for graphical probabilistic models that factorise as follows:

$$p(\mathbf{z}, \mathbf{e}) = \prod_i f_i(\mathbf{z}, \mathbf{e}) \tag{1}$$

where $\mathbf{z}$ is the vector of latent variables, $\mathbf{e}$ is the vector of observed variables (evidence), and f_i are factors that depend on a non-empty subset of $\mathbf{z}$ and a subset of $\mathbf{e}$. This factorisation naturally emerges in experiments with i.i.d. observations where f_0 is the prior on $\mathbf{z}$ and f_i the posterior for the i-th observation for $i = 1, \ldots$

TODO: Draw a figure of a graphical model that is typically used for doing EP.

EP approximates the factors f_i with factors $\tilde{f}_i$ that belong to a convenient probability distribution family. The approximation aims to minimise the KL-divergence between a distribution computed using the exact factor f_i, and a distribution using the approximate factor $\tilde{f}_i$. If the approximating distribution family is chosen from the exponential family (which it typically is), minimising the KL-divergence is reduced to *matching moments*, i.e. setting a few moments of the estimating distribution (its sufficient statistics) to the values of corresponding moments of the distribution approximated. Choosing the family from the exponential family has also other benefits, including the fact that this family is closed under the operation of product (this property being assumed in the algorithm), and that Minka [2] proved the existence of a fixed point for the solution provided the family is exponential.

Choosing the approximating family is the first thing done in the algorithm. Next, approximated factors $\tilde{f}_i$ and their product $\mathcal{Q} = \prod_i \tilde{f}_i$ are initialised to uniform. The algorithm then proceeds in iterations, iteratively updating all the approximating factors in each of the outer iterations. When convergence is reached, the normalisation coefficient, an estimate of $p(\mathbf{e})$, is computed. A more detailed exposition of the algorithm follows.

1. Initialisation

All the approximate factors are initialised to uniform, meaning the initial approximation is non-informative. The product $\mathcal{Q}$ of the factors is computed accordingly. Typically, all the factors, as well as their product, are initialised to constant 1.

Factors that already belong to the chosen family can also be computed during initialisation, as such factors are always best approximated by themselves, not needing to be updated iteratively.

2. Outer loop

Following four steps are repeated until convergence.

2.1. Choose a factor $\tilde{f}_i$

Choose a factor to approximate.

2.2. Compute the cavity distribution $\mathcal{Q}^{\setminus i}$

When updating the factor $\tilde{f}_i$, we would ideally want to minimise the KL-divergence between the true distribution and the resulting approximative distribution:

$$\arg\min_{\tilde{f}_i} \mathrm{KL}(p \,||\, \prod_i \tilde{f}_i). \tag{2}$$

However, there we would need to compute moments of p in order to optimise for this KL-divergence. If we were able to do that, we would not need to use approximate inference in the first place, so let us assume this is intractable. In that case, we have to substitute p with an approximation. The approximation used in EP is the following:

$$\hat{p} = \frac{1}{Z_i} f_i \mathcal{Q}^{\setminus i} \tag{3}$$

where

$$\mathcal{Q}^{\setminus i} \propto \prod_{j \neq i} \tilde{f}_j \quad (= \mathcal{Q}/\tilde{f}_i). \tag{4}$$

Here, $\mathcal{Q}^{\setminus i}$ is called the *cavity distribution*, as it is a distribution over $\mathbf{z}$ obtained by multiplying all the approximate factors but the i-th one (thus creating the cavity in the distribution) and normalising (in order to make it a distribution). $\hat{p}$ is defined as a product of the *exact* factor f_i with the rest of the factors *approximated*, normalised to 1, and the cavity distribution needs to be computed in order to express $\hat{p}$.

2.3. Compute the approximative distribution $\mathcal{Q}_{\textbf{new}}$

Whereas the previous step was concerned with computing the cavity distribution, computing the normalisation coefficient Z_i (as $\int f_i(\mathbf{z})\mathcal{Q}^{\setminus i}(\mathbf{z})\mathrm{d}\mathbf{z}$) and the approximative posterior distribution $\hat{p}$ is reserved for this step.

Having computed $\hat{p}$, we can minimise the KL-divergence to an updated $\mathcal{Q}_{\text{new}}$ restricted to be in the approximating family $\mathcal{F}$:

$$\arg\min_{\mathcal{Q}_{\text{new}} \in \mathcal{F}} \mathrm{KL}(\hat{p} \,||\, \mathcal{Q}_{\text{new}}). \tag{5}$$

As mentioned earlier, this minimisation is achieved by matching moments of $\mathcal{Q}_{\text{new}}$ to those of $\hat{p}$.

2.4. Update the factor

We can see the relation of the f_i, which we wish to approximate, to $\mathcal{Q}_{\text{new}}$ by combining formulas (3) and (5):

$$\mathcal{Q}_{\text{new}} \approx \hat{p} = \frac{1}{Z_i} f_i \mathcal{Q}^{\setminus i}. \tag{6}$$

From here, we easily obtain the formula for the approximation of f_i:

$$f_i \approx \tilde{f}_i = Z_i \frac{\mathcal{Q}_{\text{new}}}{\mathcal{Q}^{\setminus i}}. \tag{7}$$

Thanks to the right hand side of Eq. (7) consisting of a division of distributions from the approximating family (and a coefficient), $\tilde{f}_i$ will also be from that family (provided it is closed under division). Now, the approximate factor gets updated according to Eq. (7), and the outer loop is repeated.

3. Evaluate the normalisation constant

After the algorithm has converged to a set of factors $\{\tilde{f}_i\}$, an approximate posterior $p(\mathbf{z}, \mathbf{e})$ can be computed as a product of the factors, according to the assumptions. If we are interested in $p(\mathbf{e})$, the model evidence, it can be computed now as

$$p(\mathbf{e}) = \int \prod_i \tilde{f}_i(\mathbf{z}) \mathrm{d}\mathbf{z}. \tag{8}$$

3 The Clutter Problem

In the Clutter problem, we assume a sequence of i.i.d. observations being generated either from a normal distribution with an unknown mean with some probability, or from the "clutter" distribution. The model is specified by the following formulas:

$$W_i \overset{\text{i.i.d.}}{\sim} \text{Bern}(\pi) \tag{9}$$

$$X_i \mid \mu \overset{\text{i.i.d.}}{\sim} W_i \, \mathcal{N}(\mathbf{0}, a\mathbf{I}) + (1 - W_i) \, \mathcal{N}(\mu, \mathbf{I}) \tag{10}$$

The π parameter determines the proportion of clutter, W_i select for each observation whether it was generated from the distribution of interest, $\mathcal{N}(\mu, \mathbf{I})$, or the clutter, and finally, μ is the unknown mean of the distribution we are trying to estimate. In the clutter problem, we adopt a broad Gaussian prior on μ:

$$\mu \sim \mathcal{N}(\mathbf{0}, b\mathbf{I}). \tag{11}$$

TODO: draw the graphical model

This problem fits nicely the assumptions for EP:

1. It is intractable to do exact inference to find the value of μ. This is due to the fact that in the Bayesian network, the node for μ has $(N+1)$ independent parent nodes, a prior and the N likelihood factors, of which the N likelihood factors have 2 Gaussian components each. This results in the posterior for μ consisting of 2^N N-dimensional Gaussians, corresponding to the 2^N subsets of observations that could have been generated from the true distribution (as opposed to the clutter).

2. The posterior is a product of factors that depend on a non-empty subset of the latent variables (which is $\{\mu\}$ in this case) and a subset of the observed variables (either $\{X_i\}$ for the likelihood factors, or $\emptyset$ for the prior) – exactly as required.

Instantiating the general Eq. (1) for the Clutter problem, we get the following:

$$p((\mu), (x_1, \ldots, x_N)) = p(\mu) \cdot \prod_{i=1}^{N} p(\mu \mid x_i). \tag{12}$$

In Eq. (12), the generic f_0 is instantiated as the prior $p(\mu)$, and the generic $f_i, i = 1, \ldots$ as the likelihood $p(\mu \mid x_i)$. In the following, we may use one or the other notation, whichever is more convenient.

We choose to approximate the factors, and hence also their product, by (unnormalised) spherical Gaussians, with one stipulation: the factors approximating the likelihoods may have their σ^2 parameter negative. This is an inherent property of the algorithm, and we discuss it later in Section 4. Still, each factor $\tilde{f}_i$ can be represented by the triple $\langle \tilde{s}_i, \tilde{m}_i, \tilde{v}_i \rangle$, describing its scale ($\int \tilde{f}_i(\mathbf{z})\mathrm{d}\mathbf{z}$), mean, and variance, respectively:

$$\tilde{f}_i = \tilde{s}_i \, \mathcal{N}(\tilde{\mathbf{m}}_i, \tilde{v}_i \mathbf{I}). \tag{13}$$

Besides that, also the approximate posterior has the same form, and we shall denote its parameters as follows:

$$\mathcal{Q} = \mathcal{N}(\mathbf{m}, v\mathbf{I}). \tag{14}$$

Note that $\mathcal{Q}$ is an approximating *distribution*, i.e. it is normalised to 1.

Since $\tilde{f}_0$, the prior, already is a spherical Gaussian, its parameters can be set as part of initialisation:

$$\tilde{s}_0 = 1 \qquad \tilde{\mathbf{m}}_0 = \mathbf{0} \qquad \tilde{v}_0 = b. \tag{15}$$

This factor is exact and need not be updated anymore.

What remains is expressing the formulas (4), (5), (7) for a factor $\tilde{f}_i, i = 1, \ldots,$ and (8). The following paragraphs are devoted to this.

Update formula for the cavity distribution

The general formula is as follows:

$$\mathcal{Q}^{\setminus i} \propto \mathcal{Q}/\tilde{f}_i. \tag{4 – repeated}$$

After substituting the values of $\mathcal{Q}$ and $\tilde{f}_i$, represented as shown in Eqs. (14) and (13), respectively, we obtain the following:

$$\mathcal{Q}^{\setminus i} \propto \frac{\mathcal{N}(\mathbf{m}, v\mathbf{I})}{\tilde{s}_i\,\mathcal{N}(\tilde{\mathbf{m}}_i, \tilde{v}_i\mathbf{I})}. \tag{16}$$

The parameters of $\mathcal{Q}^{\setminus i}$ can be computed using the formula for the ratio of Gaussians,

$$\mathcal{N}(\mu_1, \mathbf{\Sigma}_1)/\,\mathcal{N}(\mu_2, \mathbf{\Sigma}_2) = C\,\mathcal{N}(\mu, \mathbf{\Sigma}) \tag{17}$$

where

$$\begin{aligned} \mathbf{\Sigma} &= \left(\mathbf{\Sigma}_1^{-1} - \mathbf{\Sigma}_2^{-1}\right)^{-1} \\ \mu &= \mathbf{\Sigma}\left(\mathbf{\Sigma}_1^{-1}\mu_1 - \mathbf{\Sigma}_2^{-1}\mu_2\right) \\ C &= \sqrt{\frac{|\mathbf{\Sigma}|\,|\mathbf{\Sigma}_2|}{(2\pi)^d|\mathbf{\Sigma}_1|}}\exp\left\{-\frac{1}{2}\left(\mu_1^{\mathrm{T}}\mathbf{\Sigma}_1^{-1}\mu_1 - \mu_2^{\mathrm{T}}\mathbf{\Sigma}_2^{-1}\mu_2 - \mu^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mu\right)\right\}. \end{aligned} \tag{18}$$

As the result, we can express $\mathcal{Q}^{\setminus i}$ in terms of its parameters $\mathbf{m}^{\setminus \mathbf{i}}$ (mean) and $v^{\setminus i}$ ($v^{\setminus i}\mathbf{I}$ being the variance) as follows:

$$\mathbf{m}^{\setminus \mathbf{i}} = v^{\setminus i}(\mathbf{m}v^{-1} - \tilde{\mathbf{m}}_i\tilde{v}_i^{-1}) \qquad (v^{\setminus i})^{-1} = v^{-1} - \tilde{v}_i^{-1}. \tag{19}$$

Update formula for $\mathcal{Q}$

In computing $\mathcal{Q}_{\text{new}}$ according to Eq. (5), we have to compute $\hat{p}$ and then its first and second moment in order to arrive at the spherical normal distribution minimising the KL-divergence to $\hat{p}$. In the definition of $\hat{p}$ in Eq. (3), the quantity Z_i is yet to be computed. It is the normalisation constant of $f_i\mathcal{Q}^{\setminus i}$, i.e.:

$$Z_i = \int f_i(\mu)\mathcal{Q}^{\setminus i}(\mu)\mathrm{d}\mu. \tag{20}$$

Parameters of $\mathcal{Q}^{\setminus i}$ were obtained in the previous step, and f_i was defined as the likelihood of x_i (cf. Eq. (10)):

$$f_i(\mu) = w_i\,\mathcal{N}(x_i; \mathbf{0}, a\mathbf{I}) + (1 - w_i)\,\mathcal{N}(x_i; \mu, \mathbf{I}). \tag{21}$$

Substituting to Eq. (20), we get

$$
\begin{align}
Z_i &= \int \left[w_i\, \mathcal{N}(x_i; \mathbf{0}, a\mathbf{I}) + (1 - w_i)\, \mathcal{N}(x_i; \mu, \mathbf{I})\right]\, \mathcal{N}(\mu; \mathbf{m}^{\backslash \mathbf{i}}, v^{\backslash i}\mathbf{I}) \mathrm{d}\mu \tag{22} \\
&= \int w_i\, \mathcal{N}(x_i; \mathbf{0}, a\mathbf{I})\, \mathcal{N}(\mu; \mathbf{m}^{\backslash \mathbf{i}}, v^{\backslash i}\mathbf{I}) \mathrm{d}\mu \nonumber \\
&\quad + \int (1 - w_i)\, \mathcal{N}(x_i; \mu, \mathbf{I})\, \mathcal{N}(\mu; \mathbf{m}^{\backslash \mathbf{i}}, v^{\backslash i}\mathbf{I}) \mathrm{d}\mu \tag{23} \\
&= w_i\, \mathcal{N}(x_i; \mathbf{0}, a\mathbf{I}) + (1 - w_i) \int \mathcal{N}(x_i - \mu; \mathbf{0}, \mathbf{I})\, \mathcal{N}(\mu; \mathbf{m}^{\backslash \mathbf{i}}, v^{\backslash i}\mathbf{I}) \mathrm{d}\mu \tag{24} \\
&= w_i\, \mathcal{N}(x_i; \mathbf{0}, a\mathbf{I}) + (1 - w_i)\, \mathcal{N}\Big(x_i; \mathbf{m}^{\backslash \mathbf{i}}, (v^{\backslash i} + 1)\mathbf{I}\Big) \tag{25}
\end{align}
$$

where, going from (24) to (25), we used the result [1] about convolution of Gaussians.

Update formula for $\tilde{f}_i$

Formula for the normalisation constant

TODO: how the results are read off

4 Implementation

TODO: how some parameters were instantiated, how uniform factors were expressed as Gaussians (infinite variance)

TODO: general properties of the implementation: Python, uses numpy, can be configured inside the source code, can be asked to draw plots interactively

TODO: example pictures from the algorithm

TODO: problems encountered (negative variance, infinities, unsensible normalisation constants)... but the algorithm converges well. Include some statistics of convergence properties (or, say, distance of the estimated mean from the true one)

References

[1] BROMILEY, P. Products and convolutions of Gaussian distributions. Internal Report 2003-003, TINA Vision, 2003.

[2] MINKA, T. P. Expectation Propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence* (2001), Morgan Kaufmann Publishers Inc., pp. 362–369.