Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks *Supplementary Material*

José Miguel Hernández-Lobato, Harvard University, USA
Ryan P. Adams, Harvard University, USA

1 Derivation of the gradients

In this section we derive the gradient of the logarithm of the marginal likelihood, that is, $\log Z$, with respect to the means and variances of the network weights in the Gaussian approximation q. In traditional backpropagation we have, for each neuron j, one variable δ_j containing the gradient of the network error with respect to the input or activation for neuron j. In PBP, the corresponding algorithm is very similar, with the difference that we now have two variables for each neuron j instead of only one. We have one variable δ_j^m that contains the gradient of $\log Z$ with respect to the mean of the activation for neuron j. Additionally, there is another variable δ_j^v that contains the gradient of $\log Z$ with respect to the variance of the activation for neuron j.

The mean and variance of the output of unit j are defined as m_j^z and v_j^z, respectively. The mean and variance of the activation or input for unit j are defined as m_j^a and v_j^a, respectively. We have that, becauseof the ReLU activation function,

$$m_j^z = \Phi(\alpha_j)\left[m_j^a + \sqrt{v_j^a}\gamma_j\right]\,, \tag{1}$$

$$v_j^z = m_j^z\left[m_j^a + \sqrt{v_j^a}\gamma_j\right]\Phi(-\alpha_j) + \Phi(\alpha_j)v_j^a(1-\gamma_j^2-\gamma_j\alpha_j)\,, \tag{2}$$

where $\gamma_j = \phi(\alpha_j)/\Phi(\alpha_j)$, $\alpha_j = m_j^a/\sqrt{v_j^a}$ and ϕ and Φ denote the standard Gaussian pdf and cdf. For the single neuron in the last layer we have that $m_j^z = m_j^a$ and $v_j^z = v_j^a$.

We have that m_j^a and v_j^a are given by

$$m_j^a = \frac{1}{\sqrt{|I(j)|}}\sum_{i\in I(j)} m_i^z m_{j,i}^w\,, \tag{3}$$

$$v_j^a = \frac{1}{|I(j)|}\sum_{i\in I(j)}\left\{[m_i^z]^2 v_{j,i}^w + [v_i^z][m_{j,i}^w]^2 + v_i^z v_{j,i}^w\right\}\,, \tag{4}$$

where $I(j)$ is the set of neurons whose output is the input to neuron j, $m_{i,j}^w$ and $v_{i,j}^w$ are the mean and variances of the weight connecting neurons i and j. Therefore,

$$\frac{\partial m_j^a}{\partial m_i^z} = \frac{1}{\sqrt{|I(j)|}}m_{j,i}^w\,, \qquad \frac{\partial m_j^a}{\partial v_i^z} = 0\,, \tag{5}$$

$$\frac{\partial v_j^a}{\partial m_i^z} = \frac{2m_i^z v_{j,i}^w}{|I(j)|}\,, \qquad \frac{\partial v_j^a}{\partial v_i^z} = \frac{[m_{j,i}^w]^2 + v_{j,i}^w}{|I(j)|}\,, \tag{6}$$

and

$$\frac{\partial m_i^a}{\partial m_{i,j}^w} = \frac{m_j^z}{\sqrt{|I(i)|}}\,, \qquad \frac{\partial m_i^a}{\partial v_{i,j}^w} = 0\,, \tag{7}$$

$$\frac{\partial v_i^a}{\partial m_{i,j}^w} = \frac{2v_j^z m_{i,j}^w}{|I(i)|}\,, \qquad \frac{\partial v_i^a}{\partial v_{i,j}^w} = \frac{[m_j^z]^2 + v_j^z}{|I(i)|}\,, \tag{8}$$

We now compute the gradient of γ_j and α_j with respect to m_j^a and v_j^a:

$$\frac{\partial \alpha_j}{\partial m_j^a} = \frac{1}{\sqrt{v_j^a}}\,, \qquad \frac{\partial \alpha_j}{\partial v_j^a} = \frac{m_j^a}{2 v_j^a \sqrt{v_j^a}}\,, \tag{9}$$

$$\frac{\partial \gamma_j}{\partial m_j^a} = -\left[\gamma_j \alpha_j + \gamma_j^2\right] \frac{\partial \alpha_j}{\partial m_j^a}\,, \qquad \frac{\partial \gamma_j}{\partial v_j^a} = -\left[\gamma_j \alpha_j + \gamma_j^2\right] \frac{\partial \alpha_j}{\partial v_j^a}\,. \tag{10}$$

Then we obtain

$$\frac{\partial m_j^z}{\partial m_j^a} = \frac{\partial \alpha_j}{\partial m_j^a} \phi(\alpha_j) \left[m_j^a + \sqrt{v_j^a} \gamma_j\right] + \Phi(\alpha_j) \left[1 + \sqrt{v_j^a} \frac{\partial \gamma_j}{\partial m_j^a}\right]\,, \tag{11}$$

$$\frac{\partial m_j^z}{\partial v_j^a} = \frac{\partial \alpha_j}{\partial v_j^a} \phi(\alpha_j) \left[m_j^a + \sqrt{v_j^a} \gamma_j\right] + \Phi(\alpha_j) \left[\frac{\gamma_j}{2\sqrt{v_j^a}} + \sqrt{v_j^a} \frac{\partial \gamma_j}{\partial v_j^a}\right]\,, \tag{12}$$

$$\begin{aligned}\frac{\partial v_j^z}{\partial m_j^a} = {} & \frac{\partial m_j^z}{\partial m_j^a} \left[m_j^a + \sqrt{v_j^a} \gamma_j\right] \Phi(-\alpha_j) + m_j^z \left\{ \left[1 + \sqrt{v_j^a} \frac{\partial \gamma_j}{\partial m_j^a}\right] \Phi(-\alpha_j) - \left[m_j^a + \sqrt{v_j^a} \gamma_j\right] \phi(\alpha_j) \frac{\partial \alpha_j}{\partial m_j^a} \right\} + \\ & \phi(\alpha_j) \frac{\partial \alpha_j}{\partial m_j^a} v_j^a (1 - \gamma_j^2 - \gamma_j \alpha_j) - \Phi(\alpha_j) v_j^a \left\{ 2\gamma_j \frac{\partial \gamma_j}{\partial m_j^a} + \frac{\partial \gamma_j}{\partial m_j^a} \alpha_j + \gamma_j \frac{\partial \alpha_j}{\partial m_j^a} \right\}\,,\end{aligned} \tag{13}$$

$$\begin{aligned}\frac{\partial v_j^z}{\partial v_j^a} = {} & \frac{\partial m_j^z}{\partial v_j^a} \left[m_j^a + \sqrt{v_j^a} \gamma_j\right] \Phi(-\alpha_j) + \\ & m_j^z \left\{ \left[\frac{1}{2\sqrt{v_j^a}} \gamma_j + \frac{\partial \gamma_j}{\partial v_j^a} \sqrt{v_j^a}\right] \Phi(-\alpha_j) - \left[m_j^a + \sqrt{v_j^a} \gamma_j\right] \phi(\alpha_j) \frac{\partial \alpha_j}{\partial v_j^a} \right\} + \\ & \phi(\alpha_j) \frac{\partial \alpha_j}{\partial v_j^a} v_j^a (1 - \gamma_j^2 - \gamma_j \alpha_j) + \\ & \Phi(\alpha_j) \left\{ (1 - \gamma_j^2 - \gamma_j \alpha_j) + v_j^a \left\{ -2\gamma_j \frac{\partial \gamma_j}{\partial v_j^a} - \frac{\partial \gamma_j}{\partial v_j^a} \alpha_j - \gamma_j \frac{\partial \alpha_j}{\partial v_j^a}) \right\} \right\}\,.\end{aligned} \tag{14}$$

We now define the variables δ_j^m and δ_j^v to be

$$\delta_j^m = \frac{\partial \log Z}{\partial m_j^a} = \sum_{k \in O(j)} \left\{ \frac{\partial \log Z}{\partial m_k^a} \frac{\partial m_k^a}{\partial m_j^a} + \frac{\partial \log Z}{\partial v_k^a} \frac{\partial v_k^a}{\partial m_j^a} \right\}\,, \tag{15}$$

$$\delta_j^v = \frac{\partial \log Z}{\partial v_j^a} = \sum_{k \in O(j)} \left\{ \frac{\partial \log Z}{\partial m_k^a} \frac{\partial m_k^a}{\partial v_j^a} + \frac{\partial \log Z}{\partial v_k^a} \frac{\partial v_k^a}{\partial v_j^a} \right\}\,, \tag{16}$$

where the sum is over each neuron k to which neuron j sends signals. The above rules can be recursively written as follows:

$$\delta_j^m = \frac{\partial \log Z}{\partial m_j^a} = \sum_{k \in O(j)} \left\{ \delta_k^m \frac{\partial m_k^a}{\partial m_j^a} + \delta_k^v \frac{\partial v_k^a}{\partial m_j^a} \right\}\,, \tag{17}$$

$$\delta_j^v = \frac{\partial \log Z}{\partial v_j^a} = \sum_{k \in O(j)} \left\{ \delta_k^m \frac{\partial m_k^a}{\partial v_j^a} + \delta_k^v \frac{\partial v_k^a}{\partial v_j^a} \right\}\,, \tag{18}$$

We can then write the required terms $\frac{\partial m_k^a}{\partial m_j^a}$, $\frac{\partial v_k^a}{\partial m_j^a}$, $\frac{\partial m_k^a}{\partial v_j^a}$ and $\frac{\partial v_k^a}{\partial v_j^a}$ as follows:

$$\frac{\partial m_k^a}{\partial m_j^a} = \frac{\partial m_k^a}{\partial m_j^z} \frac{\partial m_j^z}{\partial m_j^a} + \frac{\partial m_k^a}{\partial v_j^z} \frac{\partial v_j^z}{\partial m_j^a}\,, \qquad \frac{\partial v_k^a}{\partial m_j^a} = \frac{\partial v_k^a}{\partial m_j^z} \frac{\partial m_j^z}{\partial m_j^a} + \frac{\partial v_k^a}{\partial v_j^z} \frac{\partial v_j^z}{\partial m_j^a}\,, \tag{19}$$

$$\frac{\partial m_k^a}{\partial v_j^a} = \frac{\partial m_k^a}{\partial m_j^z} \frac{\partial m_j^z}{\partial v_j^a} + \frac{\partial m_k^a}{\partial v_j^z} \frac{\partial v_j^z}{\partial v_j^a}\,, \qquad \frac{\partial v_k^a}{\partial v_j^a} = \frac{\partial v_k^a}{\partial m_j^z} \frac{\partial m_j^z}{\partial v_j^a} + \frac{\partial v_k^a}{\partial v_j^z} \frac{\partial v_j^z}{\partial v_j^a}\,. \tag{20}$$

Table 1: Average Test RMSE in the experiments with deep neural networks.

Dataset	**BP$_1$**	**BP$_2$**	**BP$_3$**	**BP$_4$**	**PBP$_1$**	**PBP$_2$**	**PBP$_3$**	**PBP$_4$**
Boston	3.228±0.1951	3.185±0.2365	3.019±0.1848	2.874±0.1570	3.014±0.1800	**2.795±0.1590**	2.938±0.1645	3.088±0.1519
Concrete	5.977±0.2207	5.396±0.1273	5.568±0.1271	5.530±0.1390	5.667±0.0933	**5.241±0.1164**	5.732±0.1075	5.956±0.1597
Energy	1.185±0.1242	0.676±0.0367	**0.628±0.0278**	0.667±0.0321	1.804±0.0481	0.903±0.0482	1.237±0.0592	1.176±0.0552
Kin8nm	0.091±0.0015	0.073±0.0009	0.071±0.0006	0.071±0.0009	0.098±0.0007	**0.071±0.0005**	0.073±0.0007	0.075±0.0008
Naval	0.001±0.0001	**0.001±0.0000**	0.001±0.0001	0.001±0.0001	0.006±0.0000	0.003±0.0001	0.010±0.0013	0.004±0.0011
Power Plant	4.182±0.0402	4.220±0.0744	4.112±0.0378	4.184±0.0591	4.124±0.0345	**4.028±0.0347**	4.065±0.0382	4.075±0.0366
Protein	4.539±0.0288	4.188±0.0313	4.014±0.0326	**3.960±0.0110**	4.688±0.0115	4.251±0.0153	4.094±0.0285	3.970±0.0376
Wine	0.645±0.0098	0.651±0.0108	0.652±0.0101	0.650±0.0158	**0.635±0.0079**	0.643±0.0077	0.641±0.0086	0.637±0.0079
Yacht	1.182±0.1645	1.542±0.1920	1.107±0.0863	1.265±0.1287	1.015±0.0542	**0.848±0.0495**	0.893±0.0991	1.711±0.2288
Year	8.932±NA	8.976±NA	8.933±NA	9.045±NA	**8.869± NA**	8.918±NA	8.874±NA	8.934±NA

Finally, we have that

$$\frac{\partial \log Z}{\partial m^w_{i,j}} = \delta^m_j \frac{\partial m^a_i}{\partial m^w_{i,j}} + \delta^v_j \frac{\partial v^a_i}{\partial m^w_{i,j}}\,, \tag{21}$$

$$\frac{\partial \log Z}{\partial v^w_{i,j}} = \delta^m_j \frac{\partial m^a_i}{\partial v^w_{i,j}} + \delta^v_j \frac{\partial v^a_i}{\partial v^w_{i,j}}\,. \tag{22}$$

2 Results with neural networks including more than one hidden layer

We repeated the experiments from Section 5.1 in the main document for the methods BP and PBP, using neural networks with 2, 3 and 4 hidden layers. We used networks with 50 units in each hidden layer, except in the datasets *Year* and *Protein*, where we used 100. Table 1 shows the average test RMSE and the corresponding standard errors obtained by PBP$_x$ and BP$_x$, where x is the number of hidden layers in the network. PBP has the best overall predictive performance, with PBP$_2$ achieving the best results in 5 datasest. Note that the optimal number of hidden layers in PBP is problem dependent. In datasets such as *Wine* and *Year* one single hidden layer is optimal, while in *Protein* we find that 4 hidden layers is better.

3 Error in the second approximation in equation (12) in the main text

In this section we evaluate the error in the second approximation performed in equation (12) in the main document. This approximation consists in replacing the Student's t density with a Gaussian density that has the same mean and variance. This approximation becomes more and more accurate as the degrees of freedom in the Student's t density increase. This will often be the case as we iterate over the data and we reduce our uncertainty on the value of the noise parameter γ. We evaluated the relative error in $\log Z$ caused by this approximation as PBP iterates over the data of the Boston Housing dataset in the experiments of Section 5.1 in the main document. The left plot in Figure 1 shows the error during the first 100 iterations of PBP over the individual datapoints. The right plot shows the error during the last 100 iterations of the method. We can see that the error is very small in the second case. In particular, at this stage we are highly confident on the value of the noise parameter γ and the parameters α^γ and β^γ in the posterior approximation take relatively high values. This increases the number of degrees of freedom of the Student's t density in equation (12), what improves the quality of the Gaussian approximation.

4 List of approximations

In this section we list all the approximations performed by the method PBP. The list of approximations is

- We use expectation propagation (EP) to adjust a parametric approximation, given by equation (8) in the main document, to the exact posterior distribution, given by equation (3) in the main document.
- In our implementation of EP, we refine the parameters α^γ, β^γ, α^λ and β^λ of the posterior approximation by matching the first and second moments of λ and γ. The KL divergence would be minimized by matching the expectation of the sufficient statistics of a Gamma distribution, but this does not have an analytical solution.

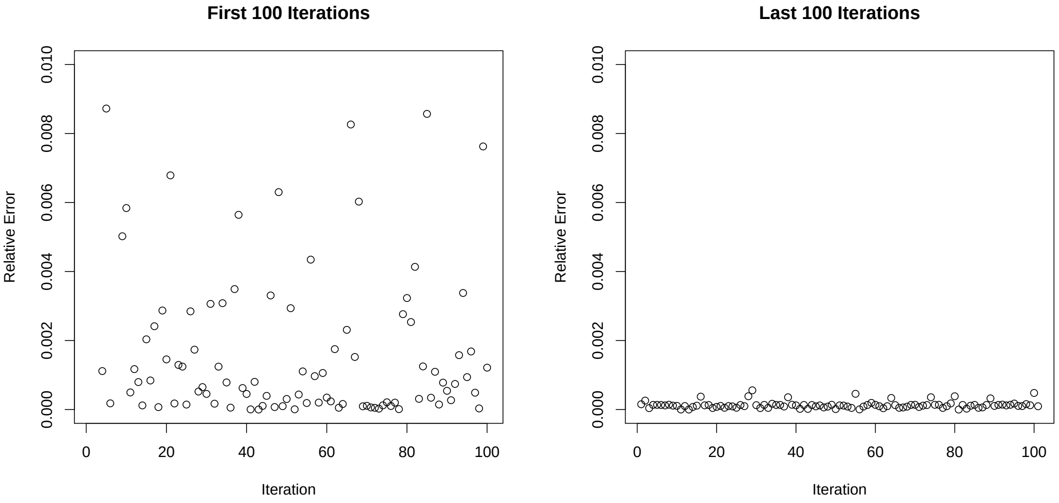

Figure 1: Relative error in the approximation of $\log Z$ as PBP iterates over the data of the Boston Housing dataset. in the experiments of Section 5.1 of the main paper. Left, relative error during the first 100 iterations of the method over the individual datapoints. Right, relative error during the last 100 iterations of the method. We can see that the error is very small during the last iterations. At this stage we are highly confident on the value of the noise parameter γ, the parameters α^γ and β^γ in the posterior approximation take relatively high values and the degrees of freedom of the Student's t density in equation (12) are high, what increases the quality of the Gaussian approximation.

- We approximate the normalization constants in equations (11) and (12) of the main document by replacing a Student's t density with a Gaussian density that has the same mean and variance.
- EP requires to keep in memory one approximate factor for each exact factor in the numerator of the posterior distribution. With massive data the number of exact likelihood factors is very large and keeping in memory all the corresponding approximate factors is inpractical. To avoid this, we do not keep these approximate factors in memory and we do not remove them from the current approximation before processing each datapoint. This is equivalent to doing multiple ADF passes through the data, treating each likelihood factor as a novel example. A disadvantage of this approach is that it can lead to underestimation of the posterior variance when too many iterations are done over the data.

5 Derivations of equations (9) and (10) in the main text

Let the Gamma density be defined as $\text{Gamma}(x|\alpha,\beta) = \beta^\alpha x^{\alpha-1}\exp\{-x\beta\}\Gamma(\alpha)^{-1}$. We denote the normalization constant of $f(x)\text{Gamma}(x|\alpha,\beta)$ by $H(\alpha,\beta)$. In particular,

$$H(\alpha,\beta) = \int f(x)\text{Gamma}(x|\alpha,\beta)\,dx\,. \tag{23}$$

Note that we explicitly write H as function of α and β. Then we have that the first and second moments of the normalized version of $f(x)\text{Gamma}(x|\alpha,\beta)$ are given by

$$\frac{1}{H(\alpha,\beta)}\int xf(x)\text{Gamma}(x|\alpha,\beta)\,dx = \frac{H(\alpha+1,\beta)\alpha}{H(\alpha,\beta)\beta}\,, \tag{24}$$

$$\frac{1}{H(\alpha,\beta)}\int x^2\text{Gamma}(x|\alpha,\beta)\,dx = \frac{H(\alpha+2,\beta)\alpha(\alpha+1)}{H(\alpha,\beta)\beta^2}\,. \tag{25}$$

Thus, each moment can be easily approximated given a procedure to approximate the normalization constant $H(\alpha,\beta)$. For this, we only have to substitute $H(\alpha,\beta)$, $H(\alpha+1,\beta)$ and $H(\alpha+2,\beta)$ in the previous expressions with their corresponding approximations. Note that the mean and variance of Gamma$(x|\alpha,\beta)$ are given by α/β and α/β^2, respectively. We can then find the new parameters α^{new} and β^{new} of a Gamma distribution that has the same mean and variance as the normalized version of $f(x)$Gamma$(x|\alpha,\beta)$ by solving the system of equations given by

$$\frac{\alpha^{\text{new}}}{\beta^{\text{new}}} = \frac{H(\alpha+1,\beta)\alpha}{H(\alpha,\beta)\beta}\,, \qquad \frac{\alpha^{\text{new}}}{[\beta^{\text{new}}]^2} = \frac{H(\alpha+2,\beta)\alpha(\alpha+1)}{H(\alpha,\beta)\beta^2} - \left\{\frac{H(\alpha+1,\beta)\alpha}{H(\alpha,\beta)\beta}\right\}^2\,, \tag{26}$$

Let $Z = H(\alpha,\beta)$, $Z_1 = H(\alpha+1,\beta)$ and $Z_2 = H(\alpha+2,\beta)$. Then

$$\alpha^{\text{new}} = \left[ZZ_2Z_1^{-2}(\alpha{+}1)/\alpha - 1.0\right]^{-1}\,, \tag{27}$$

$$\beta^{\text{new}} = \left[Z_2Z_1^{-1}(\alpha{+}1)/\beta - Z_1Z^{-1}\alpha/\beta\right]^{-1}\,. \tag{28}$$

6 EP updates for the approximate factors corresponding to the prior

The only prior factors that need to be processed multiple times using expectation propagation are the factors in equation (2) in the main document. The other Gamma priors on λ and γ have the same functional form as the posterior approximation q. This means that they need to be incorporated only once into q since any removal and posterior re-incorporation of these factors would not produce any improvement in q.

We re-write here the expression for the prior factors than need to be processed multiple times, that is, equation (2) from the main document:

$$p(\mathcal{W}|\lambda) = \prod_{l=1}^{L}\prod_{i=1}^{V_l}\prod_{j=1}^{V_{l-1}+1}\mathcal{N}(w_{ij,l}|0,\lambda^{-1})\,. \tag{29}$$

We also re-write here the expression for the posterior approximation q:

$$q(\mathcal{W},\gamma,\lambda) = \left[\prod_{l=1}^{L}\prod_{i=1}^{V_l}\prod_{j=1}^{V_{l-1}+1}\mathcal{N}(w_{ij,l}|m_{ij,l},v_{ij,l})\right]\text{Gamma}(\gamma|\alpha^\gamma,\beta^\gamma)\text{Gamma}(\lambda|\alpha^\lambda,\beta^\lambda)\,. \tag{30}$$

We denote each exact factor in (29) by

$$f_{ij,l}(w_{ij,l},\lambda) = \mathcal{N}(w_{ij,l}|0,\lambda^{-1})\,. \tag{31}$$

Each of these exact factors is approximated by a corresponding approximate factor given by

$$\tilde{f}_{ij,l}(w_{ij,l},\lambda) = \mathcal{N}(w_{ij,l}|\tilde{m}_{ij,l},\tilde{v}_{ij,l})\text{Gamma}(\lambda|\tilde{\alpha}_{ij,l},\tilde{\beta}_{ij,l})\,, \tag{32}$$

Initialliy all the $\tilde{f}_{ij,l}$ are uniform, that is, $\tilde{m}_{ij,l} = 0$, $\tilde{v}_{ij,l} = \infty$, $\tilde{\alpha}_{ij,l} = 1$ and $\tilde{\beta}_{ij,l} = 0$. EP starts to incorporate all the $f_{ij,l}$ into q once it has already incorporated the Gamma priors for λ and γ. The first time $f_{ij,l}$ is incorporated into q we update $\tilde{f}_{ij,l}$ and q as follows:

$$\tilde{m}_{ij,l} = 0\,, \qquad \tilde{v}_{ij,l} = \beta_0^\lambda/(\alpha_0^\lambda - 1)\,, \tag{33}$$

$$m_{ij,l} = 0\,, \qquad v_{ij,l} = \beta_0^\lambda/(\alpha_0^\lambda - 1)\,, \tag{34}$$

where α_0^λ and β_0^λ are the parameters of the Gamma prior on λ. These rules guarantee the matching of means and variances on $w_{ij,l}$ after approximating the Student's t density in equation (11) in the main document with a Gaussian that has the same mean and variance.

On successive iterations, we refine $\tilde{f}_{ij,l}$ by first removing this approximate factor from q to obtain a cavity distribution. This cavity is computed as the ratio of q and $\tilde{f}_{ij,l}$. The cavity marginal distribuion on $w_{ij,l}$ and λ is therefore

$$q^{\setminus ij,l}(w_{ij,l},\lambda) = \mathcal{N}(w_{ij,l}|m^{\setminus ij,l}, v^{\setminus ij,l})\text{Gamma}(\lambda|\alpha_\lambda^{\setminus ij,l}, \beta_\lambda^{\setminus ij,l})\,, \tag{35}$$

where

$$v^{\setminus ij,l} = \left[v_{ij,l}^{-1} - \tilde{v}_{ij,l}^{-1}\right]^{-1}\,, \qquad m^{\setminus ij,l} = v^{\setminus ij,l}\left[m_{ij,l}v_{ij,l}^{-1} - \tilde{m}_{ij,l}\tilde{v}_{ij,l}^{-1}\right]\,, \tag{36}$$

$$\alpha_\lambda^{\setminus ij,l} = \alpha^\lambda - \tilde{\alpha}_{ij,l} + 1\,, \qquad \beta_\lambda^{\setminus ij,l} = \beta^\lambda - \tilde{\beta}_{ij,l}\,, \tag{37}$$

After this, we update the parameters of q to match moments between $q(w_{ij,l},\lambda)$ and the normalized version of $f(w_{ij,l},\lambda)q^{\setminus ij,l}(w_{ij,l},\lambda)$. For this, we use expression (11) in the main text to approximate the normalization constant of $f(w_{ij,l},\lambda)q^{\setminus ij,l}(w_{ij,l},\lambda)$. This last step is obtained by replacing q in equation (11) in the main text with the cavity distribution. Equations (6), (7), (9) and (10) from the main document are then used to obtain the new parameters $m_{ij,l}$, $v_{ij,l}$, α^λ and β^λ for the posterior aproximation. Finally, we update the parameters for the approximate factor $\tilde{f}_{ij,l}$ using

$$\tilde{v}_{ij,l} = \left[v_{ij,l}^{-1} - [v^{\setminus ij,l}]^{-1}\right]^{-1}\,, \qquad \tilde{m}_{ij,l} = \tilde{v}_{ij,l}\left[m_{ij,l}v_{ij,l}^{-1} - m^{\setminus ij,l}[v^{\setminus ij,l}]^{-1}\right]\,, \tag{38}$$

$$\tilde{\alpha}_{ij,l} = \alpha^\lambda - \alpha_\lambda^{\setminus ij,l} + 1\,, \qquad \tilde{\beta}_{ij,l} = \beta^\lambda - \beta_\lambda^{\setminus ij,l}\,. \tag{39}$$