Collaborative Gaussian Processes for Preference Learning – Supplementary Material

Neil Houlsby
University of Cambridge
nmth2@cam.ac.uk

Jose Miguel Hernández-Lobato
University of Cambridge
jmh233@cam.ac.uk

Ferenc Huszár
University of Cambridge
fh277@cam.ac.uk

Zoubin Ghahramani
University of Cambridge
zoubin@eng.cam.ac.uk

1 The preference kernel

The mean function μ_{pref} and covariance function k_{pref} of the GP prior on g can be computed from the mean function μ and covariance function k of the GP on f as follows

$$\begin{aligned} k_{\text{pref}}((\mathbf{x}_i,\mathbf{x}_j),(\mathbf{x}_k,\mathbf{x}_l)) &= \text{Cov}[g(\mathbf{x}_i,\mathbf{x}_j),g(\mathbf{x}_k,\mathbf{x}_l)] \\ &= \text{Cov}\left[(f(\mathbf{x}_i)-f(\mathbf{x}_j)),(f(\mathbf{x}_k)-f(\mathbf{x}_l))\right] \\ &= \mathbb{E}\left[(f(\mathbf{x}_i)-f(\mathbf{x}_j))\cdot(f(\mathbf{x}_k)-f(\mathbf{x}_l))\right]-(\mu(\mathbf{x}_i)-\mu(\mathbf{x}_j))\,(\mu(\mathbf{x}_k)-\mu(\mathbf{x}_l)) \\ &= k(\mathbf{x}_i,\mathbf{x}_k)+k(\mathbf{x}_j,\mathbf{x}_l)-k(\mathbf{x}_i,\mathbf{x}_l)-k(\mathbf{x}_j,\mathbf{x}_k)\,, \end{aligned} \tag{1}$$

and

$$\mu_{\text{pref}}(\mathbf{x}_i,\mathbf{x}_j)=\mathbb{E}\left[g([\mathbf{x}_i,\mathbf{x}_j])\right]=\mathbb{E}\left[f(\mathbf{x}_i)-f(\mathbf{x}_j)\right]=\mu(\mathbf{x}_i)-\mu(\mathbf{x}_j)\,. \tag{2}$$

2 Properties of the preference kernel

It is easy to show that the preference kernel k_{pref} generates valid covariance matrices. Additionally, k_{pref} respects the anti-symmetry properties of preference learning. In particular, the prior correlation between $g(\mathbf{x}_i,\mathbf{x}_j)$ and $g(\mathbf{x}_j,\mathbf{x}_i)$ is

$$\text{Corr}(g(\mathbf{x}_i,\mathbf{x}_j),g(\mathbf{x}_j,\mathbf{x}_i))=\frac{k_{\text{pref}}((\mathbf{x}_i,\mathbf{x}_j),(\mathbf{x}_j,\mathbf{x}_i))}{\sqrt{k_{\text{pref}}((\mathbf{x}_i,\mathbf{x}_j),(\mathbf{x}_i,\mathbf{x}_j))}\sqrt{k_{\text{pref}}((\mathbf{x}_j,\mathbf{x}_i),(\mathbf{x}_j,\mathbf{x}_i))}}=-1\,, \tag{3}$$

where we have assumed $\mu_{\text{pref}}=0$ to simplify the derivations. This shows that the value of g at $(\mathbf{x}_i,\mathbf{x}_j)$ is perfectly anti-correlated with the value of g at $(\mathbf{x}_j,\mathbf{x}_i)$ under the prior. From this fact it can be shown that all elements g of the reproducing kernel Hilbert space (RKHS) corresponding to k_{pref} have the property $g(\mathbf{x}_i,\mathbf{x}_j)=-g(\mathbf{x}_j,\mathbf{x}_i)$. Finally, the preference kernel ensures transitivity between pairwise item preferences. In particular, since $g(\mathbf{x}_i,\mathbf{x}_j)=f(\mathbf{x}_i)-f(\mathbf{x}_j)$, we have that if $g(\mathbf{x}_i,\mathbf{x}_j)>0$ then $f(\mathbf{x}_i)>f(\mathbf{x}_j)$ and if also $g(\mathbf{x}_j,\mathbf{x}_k)>0$ then $f(\mathbf{x}_j)>f(\mathbf{x}_k)$ and $f(\mathbf{x}_i)>f(\mathbf{x}_k)$. Therefore, if $g(\mathbf{x}_i,\mathbf{x}_j)>0$ and $g(\mathbf{x}_j,\mathbf{x}_k)>0$ then $g(\mathbf{x}_i,\mathbf{x}_k)>0$.

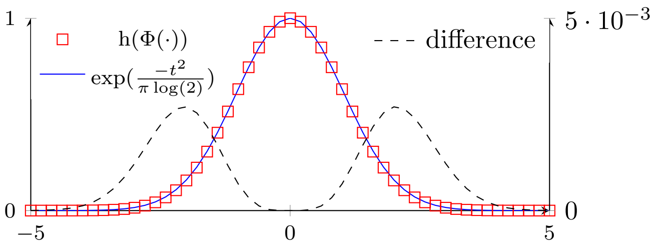

Figure 1: Analytic approximation to the binary entropy of the error function by a squared exponential. The absolute error is always smaller than $3 \cdot 10^{-3}$.

3 Taylor expansion on $\log \mathrm{h}[\Phi(x)]$

The function $\log \mathrm{h}[\Phi(x)]$ can be approximated using

$$
\begin{aligned}
f(x) &= f(0) + \frac{f'(0)x}{1!} + \frac{f''(0)x^2}{2!} + \dots, \\
f(x) &= \log \mathrm{h}[\Phi(x)], \\
f'(x) &= -\frac{1}{\log 2}\frac{\Phi'(x)}{\mathrm{h}[\Phi(x)]}\left[\log \Phi(x) - \log(1-\Phi(x))\right], \\
f''(x) &= \frac{1}{\log 2}\frac{\Phi'(x)^2}{\mathrm{h}[\Phi(x)]^2}\left[\log \Phi(x) - \log(1-\Phi(x))\right] - \frac{1}{\log 2}\frac{\Phi''(x)}{\mathrm{h}[\Phi(x)]}\left[\log \Phi(x) - \log(1-\Phi(x))\right] - \\
&\quad \frac{1}{\log 2}\frac{\Phi'(x)^2}{\mathrm{h}[\Phi(x)]}\left[\frac{1}{\Phi(x)} + \frac{1}{(1-\Phi(x)})\right]. \\
\therefore \log \mathrm{h}[\Phi(x)] &= 1 - \frac{1}{\pi \log 2}x^2 + \mathcal{O}(x^4).
\end{aligned}
$$

Note that the x^3 term will be zero because the function is even. By exponentiating, we obtain

$$
\mathrm{h}[\Phi(x)] \approx \exp\left(-\frac{x^2}{\pi \log 2}\right). \tag{4}
$$

Figure 1 demonstrates the striking accuracy of this approximation. The approximation error is never larger than 0.3%.

4 Expectation propagation and variational Bayes

In this section, we describe in detail the proposed method for approximate inference in the multi-task preference learning model. This method is based on the combination of expectation propagation Minka and Lafferty (2002); van Gerven et al. (2010) and variational inference Stern et al. (2009). We first describe the general version of the method. Finally, in Section 4.4, we describe the version which employs sparse approximations to the covariance matrices $\mathbf{K}_{\text{users}}$ and $\mathbf{K}_{\text{items}}$ for speeding up computations.

The proposed EP method approximates the exact posterior distribution by the following parametric distribution:

$$
\begin{aligned}
\mathcal{Q}(\mathbf{G}^{(\mathcal{D})}, \mathbf{W}, \mathbf{H}) \quad = \quad & \left[\prod_{u=1}^{U}\prod_{d=1}^{D}\mathcal{N}(w_{ud}|m^w_{u,d}, v^w_{u,d})\right]\left[\prod_{d=1}^{D}\prod_{i=1}^{P}\mathcal{N}(h_{d,i}|m^h_{d,i}, v^h_{d,i})\right] \\
& \left[\prod_{u=1}^{N}\prod_{j=1}^{M_u}\mathcal{N}(g_{u,z_{u,j}}|m^g_{u,j}, v^g_{u,j})\right],
\end{aligned}
\tag{5}
$$

where $m^w_{u,d}$, $v^w_{u,d}$, $m^h_{d,i}$, $v^h_{d,i}$, $m^g_{u,j}$, and $v^g_{u,j}$ are free parameters to be determined by EP. The joint distribution of the model parameters and the data $\mathcal{P}(\mathbf{G}^{(\mathcal{D})}, \mathbf{W}, \mathbf{H}, \mathbf{T}^{(\mathcal{D})}, \mathbf{X}, \ell)$ can be factorized into four factors $f_1, \ldots, f_4$, namely,

$$\mathcal{P}(\mathbf{G}^{(\mathcal{D})}, \mathbf{W}, \mathbf{H}, \mathbf{T}^{(\mathcal{D})}, \mathbf{X}, \ell) = \prod_{k=1}^{4} f_a(\mathbf{G}^{(\mathcal{D})}, \mathbf{W}, \mathbf{H}) \,, \tag{6}$$

where $f_1(\mathbf{G}^{(\mathcal{D})}, \mathbf{W}, \mathbf{H}) = \mathcal{P}(\mathbf{T}^{(\mathcal{D})}|\mathbf{G}^{(\mathcal{D})})$, $f_2(\mathbf{G}^{(\mathcal{D})}, \mathbf{W}, \mathbf{H}) = \mathcal{P}(\mathbf{G}^{(\mathcal{D})}|\mathbf{W}, \mathbf{H})$, $f_3(\mathbf{G}^{(\mathcal{D})}, \mathbf{W}, \mathbf{H}) = \mathcal{P}(\mathbf{W}|\mathbf{U})$ and $f_4(\mathbf{G}^{(\mathcal{D})}, \mathbf{W}, \mathbf{H}) = \mathcal{P}(\mathbf{H}|\mathbf{X}, \ell)$. EP approximates each of these exact factors by approximate factors $\hat{f}_1(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})}), \ldots, \hat{f}_4(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})})$ that have the same functional form as (5), namely,

$$\begin{aligned} \hat{f}_a(\mathbf{G}^{(\mathcal{D})}, \mathbf{W}, \mathbf{H}) \quad = \quad & \left[\prod_{u=1}^{U}\prod_{d=1}^{D} \mathcal{N}(w_{ud}|\hat{m}^{a,w}_{u,d}, \hat{v}^{a,w}_{u,d})\right] \left[\prod_{d=1}^{D}\prod_{i=1}^{P} \mathcal{N}(h_{d,i}|\hat{m}^{a,h}_{d,i}, \hat{v}^{a,h}_{d,i})\right] \\ & \left[\prod_{u=1}^{N}\prod_{j=1}^{M_u} \mathcal{N}(g_{u,z_{u,j}}|\hat{m}^{a,g}_{u,j}, \hat{v}^{a,g}_{u,j})\right] \hat{s}_a \,, \end{aligned} \tag{7}$$

where $a = 1, \ldots, 4$ and $\hat{m}^{a,w}_{u,d}$, $\hat{v}^{a,w}_{u,d}$, $\hat{m}^{a,h}_{d,i}$, $\hat{v}^{a,h}_{d,i}$, $\hat{m}^{a,g}_{u,j}$, $\hat{v}^{a,g}_{u,j}$ and $\hat{s}_a$ are free parameters to be determined by EP. The posterior approximation $\mathcal{Q}(\mathbf{w}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})})$ is obtained as the normalized product of the approximate factors $\hat{f}_1, \ldots, \hat{f}_4$, that is,

$$\mathcal{Q}(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})}) \propto \hat{f}_1(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})}) \cdots \hat{f}_4(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})}) \,. \tag{8}$$

The first step of EP is to initialize all the approximate factors $\hat{f}_1, \ldots, \hat{f}_4$ and the posterior approximation $\mathcal{Q}$ to be uniform. In particular, $m^w_{u,d} = m^h_{d,i} = m^g_{u,j} = \hat{m}^{w,a}_{u,d} = \hat{m}^{a,h}_{d,i} = \hat{m}^{g,a}_{u,j} = 0$ and $v^w_{u,d} = v^h_{d,i} = v^g_{u,j} = \hat{v}^{a,w}_{u,d} = \hat{v}^{a,h}_{d,i} = \hat{v}^{a,h}_{u,j} = \infty$ for $a = 1, \ldots, 4$, $u = 1, \ldots, U$, $d = 1, \ldots, D$, $i = 1, \ldots, P$ and $j = 1, \ldots, M_u$. After that, EP refines the parameters of the approximate factors by iteratively minimizing the Kullback-Leibler (KL) divergence between $\mathcal{Q}^{\setminus a}(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})}) f_a(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})})$ and $\mathcal{Q}^{\setminus a}(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})}) \hat{f}_a(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})})$, for $a = 1, \ldots, 4$, where $\mathcal{Q}^{\setminus a}$ is the ratio between $\mathcal{Q}$ and $\hat{f}_a$. That is, EP iteratively minimizes

$$\mathrm{D}_{\mathrm{KL}}(Q^{\setminus a} f_a \| Q^{\setminus a} \hat{f}_a) = \int \left[Q^{\setminus a} f_a \log \frac{Q^{\setminus a} f_a}{Q^{\setminus a} \hat{f}_a} + Q^{\setminus a} \hat{f}_a - Q^{\setminus a} f_a \right] d\mathbf{W}\, d\mathbf{H}\, d\mathbf{G}^{(\mathcal{D})} \tag{9}$$

with respect to $\hat{f}_a$, for $a = 1, \ldots, 4$. The arguments to $Q^{\setminus a} f_a$ and $Q^{\setminus a} \hat{f}_a$ have been omitted in the right-hand side of (9) to improve the readability of the expression. However, the minimization of (9) does not perform well when we have to refine the parameters of $\hat{f}_2$. The reason for this is that the corresponding exact factor f_2 (equation (7) in the main document) is invariant to simultaneous changes in sign, scalings, or rotations of the entries of $\mathbf{W}$ and $\mathbf{H}$. This non-identifiability in the latent space spanned by $\mathbf{W}$ and $\mathbf{H}$ originates multiple modes in the distribution $Q^{\setminus 2} f_2$. The minimization of the direct version of the KL divergence results in an approximation that averages across all of the modes, leading to poor predictive performance. We solve this problem by following an approach similar to the one described by Stern et al. (2009). Instead of minimizing $\mathrm{KL}(\mathcal{Q}^{\setminus 2} f_2 \| \mathcal{Q}^{\setminus 2} \hat{f}_2)$, we refine $\hat{f}_2$ by minimizing the reversed version of the KL divergence, that is, we minimize $\mathrm{KL}(\mathcal{Q}^{\setminus 2} \hat{f}_2 \| \mathcal{Q}^{\setminus 2} f_2)$ with respect to the parameters of $\hat{f}_2$. The reversed version of the divergence has mode seeking properties (Bishop, 2007) and tends to approximate only a single mode of the target distribution, leading to better predictive accuracy.

The EP algorithm iteratively refines the approximate factors until convergence. We assume the algorithm has converged when the absolute value of the change in the parameters $m^g_{u,i}$ of $\mathcal{Q}$, where $u = 1, \ldots, U$ and $i = 1, \ldots, M_u$, is less than a threshold $\delta = 10^{-2}$ between two consecutive cycles of EP, where a cycle consists in the sequential update of all the approximate factors. However, convergence is not guaranteed and EP may end up oscillating without ever stopping (Minka, 2001). This undesirable behavior can be prevented by *damping* the EP updates (Minka and Lafferty, 2002). Let $\hat{f}_a^{\text{new}}$ denote the value of the approximate factor that minimizes the Kullback-Leibler divergence. Damping consists in using

$$\hat{f}_a^{\text{damp}} = \left[\hat{f}_a^{\text{new}}\right]^{\epsilon} \left[\hat{f}_a\right]^{(1-\epsilon)} \,, \tag{10}$$

instead of $\hat{f}_a^{\text{new}}$ for the update of each approximate factor $a = 1, \ldots, 4$. The quantity $\hat{f}_a$ represents in (10) the factor before the update. The parameter $\epsilon \in [0,1]$ controls the amount of damping. The original EP update (that is, without damping) is recovered in the limit $\epsilon = 1$. For $\epsilon = 0$, the approximate factor $\hat{f}_a$ is not modified. To improve the converge of EP, we use a damping scheme with a parameter ϵ that is initialized to 1 and then progressively annealed as recommended by Hernández-Lobato (2010). After each iteration of EP, the value of this parameter is multiplied by a constant $k < 1$. The value selected for k is $k = 0.95$. In the experiments performed, EP performs on average about 50 iterations.

4.1 The EP predictive distribution

EP can also approximate the predictive distribution, given by equation (11) in the main manuscript. For this, we replace the exact posterior with the EP approximation $\mathcal{Q}$. In this way, we obtain

$$\mathcal{P}(t_{u,P+1}|\mathbf{T}^{(\mathcal{D})}, \mathbf{X}, \ell, p_{P+1}) \approx \Phi\left[t_{u,P+1} m^g_{u,P+1} (v^g_{u,P+1} + 1)^{-\frac{1}{2}}\right] \,, \tag{11}$$

where

$$m^g_{u,P+1} = \sum_{d=1}^{D} m^w_{u,d} m^h_{d,P+1} \,, \tag{12}$$

$$v^g_{u,P+1} = \sum_{d=1}^{D} [m^w_{u,d}]^2 v^h_{d,P+1} + \sum_{d=1}^{D} v^w_{u,d} [m^h_{d,P+1}]^2 + \sum_{d=1}^{D} v^w_{u,d} v^h_{d,P+1} \tag{13}$$

and $m^h_{d,P+1}$ and $v^h_{d,P+1}$ for $d = 1, \ldots, D$ are given by

$$m^h_{d,P+1} = \mathbf{k}_\star^{\text{T}} \left[\mathbf{K}_{\text{items}} + \text{diag}[\hat{\mathbf{v}}_d^{h,2}]\right]^{-1} \hat{\mathbf{m}}_d^{h,2} \,, \tag{14}$$

$$v^h_{d,P+1} = k_\star - \mathbf{k}_\star^{\text{T}} \left[\mathbf{K}_{\text{items}} + \text{diag}[\hat{\mathbf{v}}_d^{h,2}]\right]^{-1} \mathbf{k}_\star \,, \tag{15}$$

where $k_\star$ is the prior variance of $h_d(\mathbf{x}_{\alpha(P+1)}, \mathbf{x}_{\beta(P+1)})$, $\mathbf{k}_\star$ is a P-dimensional vector that contains the prior covariances between $h_d(\mathbf{x}_{\alpha(P+1)}, \mathbf{x}_{\beta(P+1)})$ and $h_d(\mathbf{x}_{\alpha(1)}, \mathbf{x}_{\beta(1)}), \ldots, h_d(\mathbf{x}_{\alpha(P)}, \mathbf{x}_{\beta(P)})$ for $d = 1, \ldots, D$, the function $\text{diag}(\cdot)$ applied to a vector returns a diagonal matrix with that vector in its diagonal and the vectors $\hat{\mathbf{m}}_d^{h,2}$ and $\hat{\mathbf{v}}_d^{h,2}$ are given by $\hat{\mathbf{m}}_d^{h,2} = (\hat{m}^{h,2}_{1,d}, \ldots, \hat{m}^{h,2}_{P,d})^{\text{T}}$ and $\hat{\mathbf{v}}_d^{h,2} = (\hat{v}^{h,2}_{1,d}, \ldots, \hat{v}^{h,2}_{P,d})^{\text{T}}$.

4.2 The EP update operations

In this section we describe the EP updates for refining the approximate factors $\hat{f}_1, \ldots, \hat{f}_4$. For the sake of clarity, we only include the update rules with no damping ($\epsilon = 1$). Incorporating the effect of damping in these operations is straightforward. With damping, the natural parameters of the approximate factors become a convex combination of the natural parameters before and after the update with no damping

$$[\hat{v}^{w,a}_{u,d}]^{-1}_{\text{damp}} = \epsilon[\hat{v}^{w,a}_{u,d}]^{-1}_{\text{new}} + (1-\epsilon)[\hat{v}^{w,a}_{u,d}]^{-1}_{\text{old}} \,, \tag{16}$$

$$[\hat{m}^{w,a}_{u,d}]_{\text{damp}}[\hat{v}^{w,a}_{u,d}]^{-1}_{\text{damp}} = \epsilon[\hat{m}^{w,a}_{u,d}]_{\text{new}}[\hat{v}^{w,a}_{u,d}]^{-1}_{\text{new}} + (1-\epsilon)[\hat{m}^{w,a}_{u,d}]_{\text{old}}[\hat{v}^{w,a}_{u,d}]^{-1}_{\text{old}} \,, \tag{17}$$

$$[\hat{v}^{h,a}_{d,i}]^{-1}_{\text{damp}} = \epsilon[\hat{v}^{h,a}_{d,i}]^{-1}_{\text{new}} + (1-\epsilon)[\hat{v}^{h,a}_{d,i}]^{-1}_{\text{old}} \,, \tag{18}$$

$$[\hat{m}^{h,a}_{d,i}]_{\text{damp}}[\hat{v}^{h,a}_{d,i}]^{-1}_{\text{damp}} = \epsilon[\hat{m}^{h,a}_{d,i}]_{\text{new}}[\hat{v}^{h,a}_{d,i}]^{-1}_{\text{new}} + (1-\epsilon)[\hat{m}^{h,a}_{d,i}]_{\text{old}}[\hat{v}^{h,a}_{d,i}]^{-1}_{\text{old}} \,, \tag{19}$$

$$[\hat{v}^{g,a}_{u,j}]^{-1}_{\text{damp}} = \epsilon[\hat{v}^{g,a}_{u,j}]^{-1}_{\text{new}} + (1-\epsilon)[\hat{v}^{g,a}_{u,j}]^{-1}_{\text{old}} \,, \tag{20}$$

$$[\hat{m}^{g,a}_{d,j}]_{\text{damp}}[\hat{v}^{g,a}_{u,j}]^{-1}_{\text{damp}} = \epsilon[\hat{m}^{g,a}_{u,j}]_{\text{new}}[\hat{v}^{g,a}_{u,j}]^{-1}_{\text{new}} + (1-\epsilon)[\hat{m}^{g,a}_{u,j}]_{\text{old}}[\hat{v}^{g,a}_{u,j}]^{-1}_{\text{old}} \,, \tag{21}$$

where $u = 1, \ldots, U$, $d = 1, \ldots, D$, $i = 1, \ldots, P$ and $j = 1, \ldots, M_u$. The subscript *new* denotes the value of the parameter given by the full EP update operation with no damping. The subscript *damp* denotes the parameter value given by the damped update rule. The subscript *old* refers to the value of the parameter before the EP update. The updates for the parameters $\hat{s}_1, \ldots, \hat{s}_4$ are not damped. These parameters are initialized to 1 and are only updated once the EP algorithm has converged.

The first factor to be refined is $\hat{f}_4$. The update operations that minimize $\mathrm{KL}(\mathcal{Q}^{\setminus 4} f_4 \| \mathcal{Q}^{\setminus 4} \hat{f}_4)$ are given by

$$[\hat{v}^{h,4}_{d,i}]_{\text{new}} = \left\{ [v^h_{d,i}]^{-1}_{\text{new}} - [\hat{v}^{h,2}_{d,i}]^{-1}_{\text{old}} \right\}^{-1} , \tag{22}$$

$$[\hat{m}^{h,4}_{d,i}]_{\text{new}} = [\hat{v}^{h,4}_{d,i}]_{\text{new}} \left\{ [m^h_{d,i}]_{\text{new}} [v^h_{d,i}]^{-1}_{\text{new}} - [\hat{m}^{h,4}_{d,i}]_{\text{old}} [\hat{v}^{h,4}_{d,i}]^{-1}_{\text{old}} \right\}^{-1} , \tag{23}$$

for $d = 1, \ldots, D$ and $i = 1, \ldots, P$, where the subscripts *new* and *old* denote the parameter value after and before the update, respectively, and the parameters $[v^h_{d,i}]_{\text{new}}$ and $[m^h_{d,i}]_{\text{new}}$ are the i-th entries in the vectors $[\mathbf{v}^h_d]_{\text{new}}$ and $[\mathbf{m}^h_d]_{\text{new}}$ given by

$$[\mathbf{v}^h_d]_{\text{new}} = \operatorname{diag} \left[\mathbf{\Sigma}^h_d \right] , \tag{24}$$

$$[\mathbf{m}^h_d]_{\text{new}} = \mathbf{\Sigma}^h_d \operatorname{diag}[\hat{\mathbf{v}}^{h,2}_d]^{-1} \hat{\mathbf{m}}^{h,2}_d , \tag{25}$$

where $[\mathbf{\Sigma}^h_d]^{-1} = \mathbf{K}^{-1}_{\text{items}} + \operatorname{diag}[\hat{\mathbf{v}}^{h,2}_d]^{-1}$ and the vectors $\hat{\mathbf{m}}^{h,2}_d$ and $\hat{\mathbf{v}}^{h,2}_d$ are P-dimensional vectors given by $\hat{\mathbf{m}}^{h,2}_d = (\hat{m}^{h,2}_{1,d}, \ldots, \hat{m}^{h,2}_{P,d})^{\mathrm{T}}$ and $\hat{\mathbf{v}}^{h,2}_d = (\hat{v}^{h,2}_{1,d}, \ldots, \hat{v}^{h,2}_{P,d})^{\mathrm{T}}$.

The second factor to be refined by EP is $\hat{f}_3$. The update operations that minimize $\mathrm{KL}(\mathcal{Q}^{\setminus 3} f_3 \| \mathcal{Q}^{\setminus 3} \hat{f}_3)$ are

$$[\hat{v}^{w,3}_{u,d}]_{\text{new}} = \left\{ [v^w_{u,d}]^{-1}_{\text{new}} - [\hat{v}^{w,2}_{u,d}]^{-1}_{\text{old}} \right\}^{-1} , \tag{26}$$

$$[\hat{m}^{w,3}_{u,d}]_{\text{new}} = [\hat{v}^{w,3}_{u,d}]_{\text{new}} \left\{ [m^w_{u,d}]_{\text{new}} [v^w_{u,d}]^{-1}_{\text{new}} - [\hat{m}^{w,3}_{u,d}]_{\text{old}} [\hat{v}^{w,3}_{u,d}]^{-1}_{\text{old}} \right\}^{-1} , \tag{27}$$

for $u = 1, \ldots, U$ and $d = 1, \ldots, D$, where the parameters $[v^w_{u,d}]_{\text{new}}$ and $[m^w_{u,d}]_{\text{new}}$ are the u-th entries in the vectors $[\mathbf{v}^w_d]_{\text{new}}$ and $[\mathbf{m}^w_d]_{\text{new}}$ given by

$$[\mathbf{v}^w_d]_{\text{new}} = \operatorname{diag} \left[\mathbf{\Sigma}^w_d \right] , \tag{28}$$

$$[\mathbf{m}^w_d]_{\text{new}} = \mathbf{\Sigma}^w_d \operatorname{diag}[\hat{\mathbf{v}}^{w,2}_d]^{-1} \hat{\mathbf{m}}^{w,2}_d , \tag{29}$$

where $[\mathbf{\Sigma}^w_d]^{-1} = \mathbf{K}^{-1}_{\text{items}} + \operatorname{diag}[\hat{\mathbf{v}}^{w,2}_d]^{-1}$ and the vectors $\hat{\mathbf{m}}^{w,2}_d$ and $\hat{\mathbf{v}}^{w,2}_d$ are given by $\hat{\mathbf{m}}^{w,2}_d = (\hat{m}^{w,2}_{1,d}, \ldots, \hat{m}^{w,2}_{U,d})^{\mathrm{T}}$ and $\hat{\mathbf{v}}^{w,2}_d = (\hat{v}^{w,2}_{1,d}, \ldots, \hat{v}^{w,2}_{U,d})^{\mathrm{T}}$.

The third factor to be refined by EP is $\hat{f}_2$. For this, we follow the approach used by Stern et al. (2009) and first marginalize $f_2 \mathcal{Q}^{\setminus 2}$ with respect to $\mathbf{G}^{(\mathcal{D})}$. The result of this operation is the auxiliary un-normalized distribution $\mathcal{S}(\mathbf{W}, \mathbf{H})$ given by

$$\begin{aligned} \mathcal{S}(\mathbf{W}, \mathbf{H}) &= \int \prod_{u=1}^{U} \prod_{i=1}^{M_u} \delta[g_{u,z_{u,i}} - \mathbf{w}_u \mathbf{h}_{\cdot,z_{u,i}}] \mathcal{Q}^{\setminus 2}(\mathbf{G}^{(\mathcal{D})}, \mathbf{W}, \mathbf{H}) \, d\mathbf{G}^{(\mathcal{D})} \\ &= \left[\prod_{u=1}^{U} \prod_{i=1}^{M_u} \mathcal{N}(\mathbf{w}_u \mathbf{h}_{\cdot,z_{u,i}} | \hat{m}^{g,1}_{u,i}, \hat{v}^{g,1}_{u,i}) \right] \left[\prod_{u=1}^{U} \prod_{d=1}^{D} \mathcal{N}(w_{u,d} | \hat{m}^{w,3}_{u,d}, \hat{v}^{w,3}_{u,d}) \right] \\ &\quad \left[\prod_{d=1}^{D} \prod_{i=1}^{P} \mathcal{N}(h_{d,i} | \hat{m}^{h,4}_{d,i}, \hat{v}^{h,4}_{d,i}) \right] . \end{aligned} \tag{30}$$

Let $\mathcal{Q}_{\mathbf{W},\mathbf{H}}$ be the posterior approximation (5) after marginalizing $\mathbf{G}^{(\mathcal{D})}$ out. The parameters of $\mathcal{Q}_{\mathbf{W},\mathbf{H}}$, that is, $m^h_{d,i}$, $v^h_{d,i}$, $m^w_{u,d}$ and $v^w_{u,d}$, for $d = 1, \ldots, D$, $u = 1, \ldots, U$ and $i = 1, \ldots, P$, are then optimized to minimize

$\mathrm{KL}(\mathcal{Q}_{\mathbf{W},\mathbf{H}}\|\mathcal{S})$. This can be done very efficiently using the gradient descent method described by Raiko et al. (2007). The resulting EP updates for $\hat{f}_2$ are given by

$$[\hat{v}^{h,2}_{d,i}]_{\text{new}} = \left\{[v^{h}_{d,i}]^{-1}_{\text{new}} - [\hat{v}^{h,2}_{d,i}]^{-1}_{\text{old}}\right\}^{-1} , \tag{31}$$

$$[\hat{m}^{h,2}_{d,i}]_{\text{new}} = [\hat{v}^{h,2}_{d,i}]_{\text{new}} \left\{[m^{h}_{d,i}]_{\text{new}}[v^{h}_{d,i}]^{-1}_{\text{new}} - [\hat{m}^{h,2}_{d,i}]_{\text{old}}[\hat{v}^{h,2}_{d,i}]^{-1}_{\text{old}}\right\}^{-1} , \tag{32}$$

$$[\hat{v}^{w,2}_{u,d}]_{\text{new}} = \left\{[v^{w}_{u,d}]^{-1}_{\text{new}} - [\hat{v}^{w,2}_{u,l}]^{-1}_{\text{old}}\right\}^{-1} , \tag{33}$$

$$[\hat{m}^{w,2}_{u,d}]_{\text{new}} = [\hat{v}^{w,2}_{u,d}]_{\text{new}} \left\{[m^{w}_{u,d}]_{\text{new}}[v^{w}_{u,d}]^{-1}_{\text{new}} - [\hat{m}^{w,2}_{u,d}]_{\text{old}}[\hat{v}^{w,2}_{u,d}]^{-1}_{\text{old}}\right\}^{-1} , \tag{34}$$

$$[\hat{v}^{g,2}_{u,j}]_{\text{new}} = \left\{[v^{g}_{u,j}]^{-1}_{\text{new}} - [\hat{v}^{g,2}_{u,j}]^{-1}_{\text{old}}\right\}^{-1} , \tag{35}$$

$$[\hat{m}^{g,2}_{u,j}]_{\text{new}} = [\hat{v}^{g,2}_{u,j}]_{\text{new}} \left\{[m^{g}_{u,j}]_{\text{new}}[v^{g}_{u,j}]^{-1}_{\text{new}} - [\hat{m}^{g,2}_{u,j}]_{\text{old}}[\hat{v}^{g,2}_{u,j}]^{-1}_{\text{old}}\right\}^{-1} , \tag{36}$$

for $d = 1, \ldots, D$, $u = 1, \ldots, U$, $j = 1, \ldots, M_u$ and $i = 1, \ldots, P$ where $[m^h_{d,i}]_{\text{new}}$, $[v^h_{d,i}]_{\text{new}}$, $[m^w_{u,d}]_{\text{new}}$ and $[v^w_{u,d}]_{\text{new}}$, are the parameters of $\mathcal{Q}$ that minimize $\mathrm{KL}(\mathcal{Q}_{\mathbf{W},\mathbf{H}}\|\mathcal{S})$ and

$$[m^g_{u,j}]_{\text{new}} = \sum_{d=1}^{D} [m^w_{u,d}]_{\text{new}}[m^h_{d,z_{u,j}}]_{\text{new}} , \tag{37}$$

$$[v^g_{u,j}]_{\text{new}} = \sum_{d=1}^{D} [m^w_{u,d}]^2_{\text{new}}[v^h_{d,z_{u,j}}]_{\text{new}} + \sum_{d=1}^{D} [v^w_{u,d}]_{\text{new}}[m^h_{d,z_{u,j}}]^2_{\text{new}} + \sum_{d=1}^{D} [v^w_{u,d}]_{\text{new}}[v^h_{d,z_{u,j}}]_{\text{new}} . \tag{38}$$

The last factor to be refined on each cycle of EP is $\hat{f}_1$. The EP update operations for this factor are

$$[\hat{m}^{g,1}_{u,i}]_{\text{new}} = \hat{m}^{g,2}_{u,i} + \hat{v}^{g,2}_{u,i}[m_{u,i}]^{-1}_{\text{new}} , \tag{39}$$

$$[\hat{v}^{g,1}_{u,i}]_{\text{new}} = \hat{v}^{g,2}_{u,i}\left[\alpha^{-1}_{u,i}[m_{u,i}]^{-1}_{\text{new}} - 1\right] , \tag{40}$$

for $u = 1, \ldots, U$ and $i = 1, \ldots, M_u$, where

$$[m_{u,i}]_{\text{new}} = \hat{m}^{g,2}_{u,i} + \hat{v}^{g,2}_{u,i}\alpha_{u,i} , \tag{41}$$

$$\alpha_{u,i} = \Phi[\beta_{u,i}]^{-1}\phi[\beta_{u,i}]t_{u,i}[\hat{v}^{g,2}_{u,i} + 1]^{-\frac{1}{2}} , \tag{42}$$

$$\beta_{u,i} = t_{u,i}\hat{m}^{g,2}_{u,i}[\hat{v}^{g,2}_{u,i} + 1]^{-\frac{1}{2}} \tag{43}$$

and ϕ and Φ are the density and the cumulative probability functions of a standard Gaussian distribution, respectively.

4.3 The EP approximation of the model evidence

Once EP has converged, we can approximate the evidence of the model, that is, $\mathcal{P}(\mathbf{T}^{(\mathcal{D})}|\mathbf{X},\ell)$, using

$$\mathcal{P}(\mathbf{T}^{(\mathcal{D})}|\mathbf{X},\ell) \approx \int \prod_{a=1}^{4} \hat{f}_a(\mathbf{G}^{(\mathcal{D})},\mathbf{W},\mathbf{H})\, d\mathbf{G}^{(\mathcal{D})}\, d\mathbf{H}\, d\mathbf{W} . \tag{44}$$

For this, we have to compute the value of the parameters $\hat{s}_1, \ldots, \hat{s}_4$. The value of $\hat{s}_1$ is

$$\log \hat{s}_1 = \sum_{u=1}^{U}\sum_{i=1}^{M_u}\left[\log\Phi[\beta_{u,i}] + \frac{1}{2}\log(2\pi) + \frac{1}{2}\log\frac{\hat{v}^{g,1}_{u,i}\hat{v}^{g,2}_{u,i}}{v^g_{u,i}} - \frac{[m^g_{u,i}]^2}{2v^g_{u,i}} + \frac{[\hat{m}^{g,1}_{u,i}]^2}{2\hat{v}^{g,1}_{u,i}} + \frac{[\hat{m}^{g,2}_{u,i}]^2}{2\hat{v}^{g,2}_{u,i}}\right] . \tag{45}$$

The value of $\hat{s}_2$ is given by

$$\log \hat{s}_2 = \log Z_2 + \sum_{u=1}^{U}\sum_{i=1}^{M_u}\left[\frac{1}{2}\log(2\pi) + \frac{1}{2}\log\frac{\hat{v}_{u,i}^{g,1}\hat{v}_{u,i}^{g,2}}{v_{u,i}^{g}} - \frac{[m_{u,i}^{g}]^2}{2v_{u,i}^{g}} + \frac{[\hat{m}_{u,i}^{g,1}]^2}{2\hat{v}_{u,i}^{g,1}} + \frac{[\hat{m}_{u,i}^{g,2}]^2}{2\hat{v}_{u,i}^{g,2}}\right] +$$
$$\sum_{d=1}^{D}\sum_{i=1}^{P}\left[\frac{1}{2}\log(2\pi) + \frac{1}{2}\log\frac{\hat{v}_{d,i}^{h,2}\hat{v}_{d,i}^{h,4}}{v_{d,i}^{h}} - \frac{[m_{d,i}^{h}]^2}{2v_{d,i}^{h}} + \frac{[\hat{m}_{d,i}^{h,2}]^2}{2\hat{v}_{d,i}^{h,2}} + \frac{[\hat{m}_{d,i}^{h,4}]^2}{2\hat{v}_{d,i}^{h,4}}\right] +$$
$$\sum_{u=1}^{U}\sum_{d=1}^{D}\left[\frac{1}{2}\log(2\pi) + \frac{1}{2}\log\frac{\hat{v}_{u,d}^{w,2}\hat{v}_{u,d}^{w,3}}{v_{u,d}^{w}} - \frac{[m_{u,d}^{w}]^2}{2v_{u,d}^{w}} + \frac{[\hat{m}_{u,d}^{w,2}]^2}{2\hat{v}_{u,d}^{w,2}} + \frac{[\hat{m}_{u,d}^{w,3}]^2}{2\hat{v}_{u,d}^{w,3}}\right], \tag{46}$$

where Z_2 is the variational lower bound obtained in the update of $\hat{f}_2$, that is,

$$Z_2 = \int \mathcal{Q}_{\mathbf{W},\mathbf{H}} \log \frac{\mathcal{S}(\mathbf{W},\mathbf{H})}{\mathcal{Q}_{\mathbf{W},\mathbf{H}}(\mathbf{W},\mathbf{H})}\, d\mathbf{W}, d\mathbf{H}\,. \tag{47}$$

The value of $\tilde{s}_3$ is given by

$$\log \hat{s}_3 = \log Z_3 + \sum_{d=1}^{D}\sum_{u=1}^{U}\left[\frac{1}{2}\log(2\pi) + \frac{1}{2}\log\frac{\hat{v}_{u,d}^{w,3}\hat{v}_{u,d}^{w,2}}{v_{u,d}^{w}} - \frac{[m_{u,d}^{w}]^2}{2v_{u,d}^{w}} + \frac{[\hat{m}_{u,d}^{w,3}]^2}{2\hat{v}_{u,d}^{w,3}} + \frac{[\hat{m}_{u,d}^{w,2}]^2}{2\hat{v}_{u,d}^{w,2}}\right], \tag{48}$$

where Z_3 is computed using

$$\begin{aligned}\log Z_3 &= \log \int \mathcal{P}(\mathbf{W}|\mathbf{U})\left[\prod_{u=1}^{U}\prod_{d=1}^{D}\mathcal{N}(w_{u,d}|\hat{m}_{u,d}^{w,2},\hat{m}_{u,d}^{w,2})\right]d\mathbf{W}\\ &= -\frac{DP}{2}\log(2\pi) + \frac{1}{2}\sum_{d=1}^{D}\log|\mathbf{\Sigma}_d^w| - \frac{D}{2}\log|\mathbf{K}_{\text{users}}| - \frac{1}{2}\sum_{u=1}^{U}\sum_{d=1}^{D}\log\hat{v}_{u,d}^{w,2} -\\ &\quad \frac{1}{2}\sum_{u=1}^{U}\sum_{d=1}^{D}\frac{[\hat{m}_{u,d}^{w,2}]^2}{\hat{v}_{u,d}^{w,2}} + \frac{1}{2}\sum_{d=1}^{D}[\mathbf{m}_d^w]^{\text{T}}[\Sigma_d^w]^{-1}\mathbf{m}_d^w\,,\end{aligned} \tag{49}$$

and $[\mathbf{\Sigma}_d^w]^{-1} = \mathbf{K}_{\text{users}}^{-1} + \text{diag}[\hat{\mathbf{v}}_d^{w,2}]^{-1}$, $\mathbf{m}_d^w = \mathbf{\Sigma}_d^w\text{diag}[\hat{\mathbf{v}}_d^{w,2}]^{-1}\hat{\mathbf{m}}_d^{w,2}$ and the vectors $\hat{\mathbf{m}}_d^{w,2}$ and $\hat{\mathbf{v}}_d^{w,2}$ are given by $\hat{\mathbf{m}}_d^{w,2} = (\hat{m}_{1,d}^{w,2},\ldots,\hat{m}_{U,d}^{w,2})^{\text{T}}$ and $\hat{\mathbf{v}}_d^{w,2} = (\hat{v}_{1,d}^{w,2},\ldots,\hat{v}_{U,d}^{w,2})^{\text{T}}$. Finally, the value of $\tilde{s}_4$ is given by

$$\log \hat{s}_4 = \log Z_4 + \sum_{d=1}^{D}\sum_{i=1}^{P}\left[\frac{1}{2}\log(2\pi) + \frac{1}{2}\log\frac{\hat{v}_{d,i}^{h,4}\hat{v}_{d,i}^{h,2}}{v_{d,i}^{h}} - \frac{[m_{d,i}^{h}]^2}{2v_{d,i}^{h}} + \frac{[\hat{m}_{d,i}^{h,4}]^2}{2\hat{v}_{d,i}^{h,4}} + \frac{[\hat{m}_{d,i}^{h,2}]^2}{2\hat{v}_{d,i}^{h,2}}\right], \tag{50}$$

where Z_4 is computed using

$$\begin{aligned}\log Z_4 &= \log \int \mathcal{P}(\mathbf{H}|\mathbf{X},\ell)\left[\prod_{d=1}^{D}\prod_{i=1}^{P}\mathcal{N}(h_{d,i}|\hat{m}_{d,i}^{h,2},\hat{m}_{d,i}^{h,2})\right]d\mathbf{H}\\ &= -\frac{DP}{2}\log(2\pi) + \frac{1}{2}\sum_{d=1}^{D}\log|\mathbf{\Sigma}_d^h| - \frac{D}{2}\log|\mathbf{K}_{\text{items}}| - \frac{1}{2}\sum_{d=1}^{D}\sum_{i=1}^{P}\log\hat{v}_{d,i}^{h,2} -\\ &\quad \frac{1}{2}\sum_{d=1}^{D}\sum_{i=1}^{p}\frac{[\hat{m}_{d,i}^{h,2}]^2}{\hat{v}_{d,i}^{h,2}} + \frac{1}{2}\sum_{d=1}^{D}[\mathbf{m}_d^h]^{\text{T}}[\mathbf{\Sigma}_d^h]^{-1}\mathbf{m}_d^h\,,\end{aligned} \tag{51}$$

and $[\mathbf{\Sigma}_d^h]^{-1} = \mathbf{K}_{\text{items}}^{-1} + \text{diag}[\hat{\mathbf{v}}_d^{h,2}]^{-1}$, $\mathbf{m}_d^h = \mathbf{\Sigma}_d\text{diag}[\hat{\mathbf{v}}_d^{h,2}]^{-1}\hat{\mathbf{m}}_d^{h,2}$ and the vectors $\hat{\mathbf{m}}_d^{h,2}$ and $\hat{\mathbf{v}}_d^{h,2}$ are given by $\hat{\mathbf{m}}_d^{h,2} = (\hat{m}_{1,d}^{h,2},\ldots,\hat{m}_{P,d}^{h,2})^{\text{T}}$ and $\hat{\mathbf{v}}_d^{h,2} = (\hat{v}_{1,d}^{h,2},\ldots,\hat{v}_{P,d}^{h,2})^{\text{T}}$. Given $\hat{s}_1,\ldots,\hat{s}_4$, we approximate $\mathcal{P}(\mathbf{T}^{(\mathcal{D})}|\mathbf{X},\ell)$

using

$$
\begin{aligned}
\log \mathcal{P}(\mathbf{T}^{(\mathcal{D})}|\mathbf{X},\ell) \quad \approx \quad & \sum_{i=a}^{4} \log \hat{s}_a - \sum_{u=1}^{U}\sum_{i=1}^{M_u}\left[\frac{1}{2}\log(2\pi) + \frac{1}{2}\log\frac{\hat{v}_{u,i}^{g,1}\hat{v}_{u,i}^{g,2}}{v_{u,i}^{g}} - \frac{[m_{u,i}^{g}]^2}{2v_{u,i}^{g}} + \frac{[\hat{m}_{u,i}^{g,1}]^2}{2\hat{v}_{u,i}^{g,1}} + \frac{[\hat{m}_{u,i}^{g,2}]^2}{2\hat{v}_{u,i}^{g,2}}\right] - \\
& \sum_{d=1}^{D}\sum_{i=1}^{P}\left[\frac{1}{2}\log(2\pi) + \frac{1}{2}\log\frac{\hat{v}_{d,i}^{h,4}\hat{v}_{d,i}^{h,2}}{v_{d,i}^{h}} - \frac{[m_{d,i}^{h}]^2}{2v_{d,i}^{h}} + \frac{[\hat{m}_{d,i}^{h,4}]^2}{2\hat{v}_{d,i}^{h,4}} + \frac{[\hat{m}_{d,i}^{h,2}]^2}{2\hat{v}_{d,i}^{h,2}}\right] - \\
& \sum_{u=1}^{U}\sum_{d=1}^{D}\left[\frac{1}{2}\log(2\pi) + \frac{1}{2}\log\frac{\hat{v}_{u,d}^{w,2}\hat{v}_{u,d}^{w,3}}{v_{u,d}^{w}} - \frac{[m_{u,d}^{w}]^2}{2v_{u,d}^{w}} + \frac{[\hat{m}_{u,d}^{w,2}]^2}{2\hat{v}_{u,d}^{w,2}} + \frac{[\hat{m}_{u,d}^{w,3}]^2}{2\hat{v}_{u,d}^{2,3}}\right]. \qquad (52)
\end{aligned}
$$

Finally, some of the EP updates may generate a negative value for $\hat{v}_{u,i}^{g,a}$, $\hat{v}_{u,d}^{w,a}$ or $\hat{v}_{d,j}^{h,a}$, where $u = 1,\ldots,U$, $i = 1,\ldots,M_u$, $j = 1,\ldots,P$ and $i = 1,\ldots,4$. Negative variances in Gaussian approximate factors are common in many EP implementations (Minka, 2001; Minka and Lafferty, 2002). When this happens, the marginals of the approximate factor with negative variances are not density functions. Instead, they are correction factors that compensate the errors in the corresponding marginals of other approximate factors. However, these negative variances can lead to failure of the proposed EP algorithm. This may happen when we have to compute $\log|\mathbf{\Sigma}_d^h|$ in (51) and some of the $\hat{v}_{d,i}^{h,2}$ are negative. In this case, Σ_d^h may not be positive definite and $|\mathbf{\Sigma}_d^h|$ may be negative. The result is that EP may no longer be able to approximate the model evidence since $\log|\mathbf{\Sigma}_d^h|$ may not be defined in (51). The same may occur for $\log|\mathbf{\Sigma}_d^w|$ in (49). To address this problem, whenever an EP update yields a negative number for any of the $\hat{v}_{u,i}^{g,a}$, $\hat{v}_{u,d}^{w,a}$ or $\hat{v}_{d,j}^{h,a}$, we do not update this parameter, nor the corresponding $\hat{m}_{u,i}^{g,a}$, $\hat{m}_{u,d}^{w,a}$ or $\hat{m}_{d,j}^{h,a}$.

4.4 Sparse approximations to speed up computations

The computational cost of EP is determined by the operations needed to refine the approximate factors $\hat{f}_3$ and $\hat{f}_4$. In particular, computing the vectors $[\mathbf{v}_d^h]_{\text{new}}$ and $[\mathbf{m}_d^h]_{\text{new}}$ in (24) and (25), for $d = 1,\ldots,D$, has cost $\mathcal{O}(DP^3)$. Similarly, the computation of the vectors $[\mathbf{v}_d^w]_{\text{new}}$ and $[\mathbf{m}_d^w]_{\text{new}}$ in (28) and (29), for $d = 1,\ldots,D$, has cost $\mathcal{O}(DU^3)$. These costs can be prohibitive when P or U are very large. Nevertheless, they can be reduced by using sparse approximations to the covariance matrices $\mathbf{K}_{\text{users}}$ and $\mathbf{K}_{\text{items}}$. We use the fully independent training conditional or FITC approximation, also known as the sparse pseudo-input GP (SPGP) Snelson and Ghahramani (2005). With FITC, the $U \times U$ covariance matrix $\mathbf{K}_{\text{users}}$ is approximated by $\mathbf{K}'_{\text{users}} = \mathbf{Q}_{\text{users}} + \text{diag}(\mathbf{K}_{\text{users}} - \mathbf{Q}_{\text{users}})$, where $\mathbf{Q}_{\text{users}} = \mathbf{K}_{\text{users},U,U_0}\mathbf{K}^{-1}_{\text{users},U_0,U_0}\mathbf{K}^{\text{T}}_{\text{users},U,U_0}$. In this expression, $\mathbf{K}_{\text{users},U_0,U_0}$ is an $U_0 \times U_0$ covariance matrix given by the evaluation of the covariance function for the users at all possible pairs of $U_0 < U$ locations or *user pseudo-inputs* $\{\mathbf{u}'_1,\ldots,\mathbf{u}'_{U_0}\}$, where $\mathbf{u}'_i \in \mathcal{U}$ for $i = 1,\ldots,U_0$, and $\mathbf{K}_{\text{users},U,U_0}$ is an $U \times U_0$ matrix with the evaluation of the covariance function for the users at all possible pairs of original user feature vectors and user pseudo-inputs, that is, $(\mathbf{u}_i,\mathbf{u}'_j)$, for $i = 1,\ldots,U$ and $j = 1,\ldots,U_0$. Similarly, the $P \times P$ covariance matrix $\mathbf{K}_{\text{items}}$ is also approximated by $\mathbf{K}'_{\text{items}} = \mathbf{Q}_{\text{items}} + \text{diag}(\mathbf{K}_{\text{items}} - \mathbf{Q}_{\text{items}})$, where $\mathbf{Q}_{\text{items}} = \mathbf{K}_{\text{items},P,P_0}\mathbf{K}^{-1}_{\text{items},P_0,P_0}\mathbf{K}^{\text{T}}_{\text{items},P,P_0}$, $\mathbf{K}_{\text{items},P_0,P_0}$ is a $P_0 \times P_0$ covariance matrix given by the evaluation of the preference kernel at all possible pairs of $P_0 < P$ locations or *item-pair pseudo-inputs* $\{(\mathbf{x}'_1,\mathbf{x}''_1),\ldots,(\mathbf{x}'_{P_0},\mathbf{x}''_{P_0})\}$, where $\mathbf{x}'_i,\mathbf{x}''_i \in \mathcal{X}$ for $i = 1,\ldots,P_0$, and $\mathbf{K}_{\text{items},P,P_0}$ is a $P \times P_0$ matrix with the evaluation of the preference kernel at all possible combinations of feature vectors for the original item pairs and item-pair pseudo-inputs, that is, $((\mathbf{x}_{\alpha(i)},\mathbf{x}_{\beta(i)}),(\mathbf{x}'_j,\mathbf{x}''_j))$, for $i = 1,\ldots,P$ and $j = 1,\ldots,P_0$.

We now describe how to refine the third and fourth approximate factors when $\mathbf{K}_{\text{users}}$ and $\mathbf{K}_{\text{items}}$ are replaced by $\mathbf{K}'_{\text{users}}$ and $\mathbf{K}'_{\text{items}}$, respectively. The required operations are can be efficiently implemented using the formulas described in (Naish-Guzman and Holden, 2007) and (Gredilla, 2010). In particular, let $\mathbf{K}'_{\text{users}} = \mathbf{D} + \mathbf{P}\mathbf{R}^{\text{T}}\mathbf{R}\mathbf{P}^{\text{T}}$, where $\mathbf{D} = \text{diag}(\mathbf{K}_{\text{users}} - \mathbf{Q}_{\text{users}})$, $\mathbf{P} = \mathbf{K}_{\text{users},U,U_0}$ and $\mathbf{R}$ is the upper Cholesky factor of $\mathbf{K}^{-1}_{\text{users},U_0,U_0}$, that is, $\mathbf{K}^{-1}_{\text{users},U_0,U_0} = \mathbf{R}^{\text{T}}\mathbf{R}$. This Cholesky factor can be computed using

$$
\mathbf{R} = \text{rot180}(\text{chol}(\text{rot180}(\mathbf{K}_{\text{users},U_0,U_0}))^{\text{T}} \setminus \mathbf{I}), \qquad (53)
$$

where $\mathbf{I}$ is the identity matrix, rot180$(\cdot)$ rotates an $m \times m$ square matrix 180° so that the element in position (i,j) is moved to position $(m-i+1, m-j+1)$, $\mathbf{A} \setminus \mathbf{a}$ denotes the solution to the linear system $\mathbf{A}\mathbf{x} = \mathbf{a}$ and chol$(\cdot)$ returns the upper Cholesky factor of its argument. The matrix $\mathbf{\Sigma}_d^w$, required to compute the vectors $[\mathbf{v}_d^w]_{\text{new}}$ and $[\mathbf{m}_d^w]_{\text{new}}$ in (28) and (29), can the be encoded efficiently using $\mathbf{\Sigma}_d^w = \mathbf{D}_d^{\text{new}} + \mathbf{P}_d^{\text{new}}[\mathbf{R}_d^{\text{new}}]^{\text{T}}\mathbf{R}_d^{\text{new}}[\mathbf{P}_d^{\text{new}}]^{\text{T}}$, where

$$\mathbf{D}_d^{\text{new}} = \left(\mathbf{I} + \mathbf{D}\text{diag}[\hat{\mathbf{v}}_d^{w,2}]^{-1}\right)^{-1}\mathbf{D}\,, \tag{54}$$

$$\mathbf{P}_d^{\text{new}} = \left(\mathbf{I} + \mathbf{D}\text{diag}[\hat{\mathbf{v}}_d^{w,2}]^{-1}\right)^{-1}\mathbf{P}\,, \tag{55}$$

$$\mathbf{R}_d^{\text{new}} = \text{rot180}(\text{chol}(\text{rot180}(\mathbf{I} + \mathbf{R}\mathbf{P}^{\text{T}}\text{diag}[\hat{\mathbf{v}}_d^{w,2}]^{-1}(\mathbf{I} + \mathbf{D}\text{diag}[\hat{\mathbf{v}}_d^{w,2}]^{-1})^{-1}\mathbf{P}\mathbf{R}^{\text{T}})^{\text{T}}) \setminus \mathbf{R} \tag{56}$$

and $\hat{\mathbf{v}}_d^{w,2}$ is given by $\hat{\mathbf{v}}_d^{w,2} = (\hat{v}_{1,d}^{w,2}, \ldots, \hat{v}_{U,d}^{w,2})^{\text{T}}$. The matrix $\mathbf{\Sigma}_d^h$, required to compute the vectors $[\mathbf{v}_d^h]_{\text{new}}$ and $[\mathbf{m}_d^h]_{\text{new}}$ in (24) and (25), can the be efficiently encoded in a similar manner. For this, we only have to replace $\hat{\mathbf{v}}_d^{w,2}$ by $\hat{\mathbf{v}}_d^{h,2} = (\hat{v}_{d,1}^{h,2}, \ldots, \hat{v}_{d,P}^{h,2})^{\text{T}}$ and $\mathbf{K}_{\text{users},U_0,U_0}$ and $\mathbf{K}_{\text{users},U,U_0}$ by $\mathbf{K}_{\text{items},P_0,P_0}$ and $\mathbf{K}_{\text{items},P,P_0}$, respectively. These alternative representations of $\mathbf{\Sigma}_d^w$ and $\mathbf{\Sigma}_d^h$ allow us to update $\hat{f}_3$ and $\hat{f}_4$ in $\mathcal{O}(dU_0^2U)$ and $\mathcal{O}(dP_0^2P)$ operations, respectively.

We also describe the new update for $\log Z_3$. Instead of using (49), we now use the following expression

$$\begin{aligned}\log Z_3 = \sum_{d=1}^{D} \Bigg[&-\frac{U}{2}\log(2\pi) + \log|\mathbf{R}_d^{\text{new}}| - \log|\mathbf{R}| - \frac{1}{2}\sum_{u=1}^{U}\log\left(\hat{v}_{u,d}^{w,2} + d_u\right) + \\ &\frac{1}{2}\sum_{u=1}^{U}\hat{m}_{u,d}^{w,2}([m_d^w]_{\text{new}})_u - \frac{1}{2}\sum_{u=1}^{U}\frac{[\hat{m}_{u,d}^{w,2}]^2}{\hat{v}_{u,d}^{w,2}}\Bigg]\,,\end{aligned} \tag{57}$$

where d_u is the u-th entry in the diagonal of $\mathbf{D}$ and $([m_d^w]_{\text{new}})_u$ is the u-th entry in the vector $[\mathbf{m}_d^w]_{\text{new}}$. The analogous update for $\log Z_4$ is given by

$$\begin{aligned}\log Z_4 = \sum_{d=1}^{D} \Bigg[&-\frac{P}{2}\log(2\pi) + \log|\mathbf{R}_d^{\text{new}}| - \log|\mathbf{R}| - \frac{1}{2}\sum_{i=1}^{P}\log\left(\hat{v}_{d,i}^{h,2} + d_i\right) + \\ &\frac{1}{2}\sum_{i=1}^{P}\hat{m}_{d,i}^{h,2}([m_d^h]_{\text{new}})_i - \frac{1}{2}\sum_{i=1}^{P}\frac{[\hat{m}_{d,i}^{h,2}]^2}{\hat{v}_{d,i}^{h,2}}\Bigg]\,,\end{aligned} \tag{58}$$

where d_i is the i-th entry in the diagonal of $\mathbf{D}$ and $([m_d^h]_{\text{new}})_i$ is the i-th entry in the vector $[\mathbf{m}_d^h]_{\text{new}}$. Note that d_i, $\mathbf{R}$ and $\mathbf{R}_d^{\text{new}}$ in (58) refer to the matrices needed for working with the efficient encoding of $\mathbf{K}'_{\text{items}}$. By contrast, d_u, $\mathbf{R}$ and $\mathbf{R}_d^{\text{new}}$ in (57) refer to the same matrices, but for working with the efficient encoding of $\mathbf{K}'_{\text{users}}$.

Finally, to compute the predictive distribution, instead of (14) and (15), we use

$$m_{d,P+1}^h = \mathbf{k}_\star^{\text{T}}\boldsymbol{\gamma}_d^{\text{new}}\,, \tag{59}$$

$$v_{d,P+1}^h = d_\star + \parallel \mathbf{R}_d^{\text{new}}\mathbf{k}_\star \parallel^2\,, \tag{60}$$

where $\mathbf{k}_\star$ is a P_0-dimensional vector that contains the prior covariances between $h_d(\mathbf{x}_{\alpha(P+1)}, \mathbf{x}_{\beta(P+1)})$ and the value of latent function h_d at the item-pair pseudo-inputs, that is, $h_d(\mathbf{x}'_1, \mathbf{x}''_1), \ldots, h_d(\mathbf{x}'_{P_0}, \mathbf{x}''_{P_0})$, $\boldsymbol{\gamma}_d^{\text{new}} = [\mathbf{R}_d^{\text{new}}]^{\text{T}}\mathbf{R}_d^{\text{new}}[\mathbf{P}_d^{\text{new}}]^{\text{T}}\text{diag}[\hat{\mathbf{v}}_d^{h,2}]^{-1}\hat{\mathbf{m}}_d^{h,2}$, $d_\star = k_\star - \mathbf{p}_\star^{\text{T}}\mathbf{R}^{\text{T}}\mathbf{R}\mathbf{p}_\star$ and finally, $k_\star$ is the prior variance of $h_d(\mathbf{x}_{\alpha(P+1)}, \mathbf{x}_{\beta(P+1)})$. Note that in all of these formulas, $\mathbf{R}_d^{\text{new}}$, $\mathbf{R}_d^{\text{new}}$ and $\mathbf{R}^{\text{T}}$ refer to the matrices needed for working with the efficient encoding of $\mathbf{K}'_{\text{items}}$.

5 Performance of BALD on GP binary classification problems

BALD is evaluated in a series of GP binary classification tasks with real-world data. In these experiments BALD is compared with several related algorithms for active learning with GPs: random sampling, Maximum Entropy Sampling Sebastiani and Wynn (2000), Query by Committee Freund et al. (1997), the Informative Vector Machine Lawrence et al. (2002) and an SVM-based approach Tong and Koller (2001). These algorithms and their relation to BALD are described in the following paragraphs.

Recall that the central objective of information theoretic active learning for classification is

$$\mathrm{H}[\mathcal{P}(g|\mathcal{D})] - \mathbb{E}_{\mathcal{P}(y|\mathbf{x},\mathcal{D})}\left[\mathrm{H}[\mathcal{P}(g|y,\mathbf{x},\mathcal{D})]\right] , \tag{61}$$

where g is the classifier latent function, $\mathbf{x}$ is a new feature vector, y is the corresponding label and $\mathcal{D}$ contains the data observed so far. BALD uses the following equivalent reformulation

$$\mathrm{H}[\mathcal{P}(y|\mathbf{x},\mathcal{D})] - \mathbb{E}_{\mathcal{P}(g|\mathcal{D})}\left[\mathrm{H}\left[\mathcal{P}(y|\mathbf{x},g)\right]\right] . \tag{62}$$

Maximum Entropy Sampling (MES) (Sebastiani and Wynn, 2000) is similar to BALD in the sense that it also works explicitly in data space (that is, using equation (62)). MES was proposed for regression models with input-independent observation noise. In this scenario, the noise in the target variable y does not depend on the input $\mathbf{x}$ and the second term in equation (62) is constant and can be safely ignored. However, if noise in the target variable is not input-independent, MES will tend to sample regions of the input space where uncertainty in g is low but uncertainty in the labels (because of observation noise) is high, as illustrated in Figure 1 of the main manuscript.

The Query by Committee (QBC) approach makes a different approximation to (62) (Freund et al., 1997). QBC samples parameters from the posterior (called committee members). These parameters are then used to perform a deterministic vote on the outcome of each candidate $\mathbf{x}$. The $\mathbf{x}$ with the most balanced vote is selected for the next active inclusion in the training set. This objective is termed the 'principle of maximal disagreement'. QBC is similar to BALD when the objective used by BALD is approximated by sampling from the posterior, with the exception that BALD uses a probabilistic measure of disagreement (equation (62)). Note that the deterministic vote criterion used by QBC does not take into account the confidence of the learning method on its predictions. Because of this, QBC can exhibit the same pathologies as MES.

The Informative Vector Machine (IVM) (Lawrence et al., 2002) is also motivated by information theory. This method was originally designed for sub-sampling a dataset and not for addressing online active learning problems. The IVM requires that the target variables y are observed prior to making a query and it is therefore not applicable online active learning tasks. Nonetheless, BALD can be applied to the dataset sub-sampling problem for which the IVM is designed, it is simply equipped with less information. The IVM works with equation (61) instead of (62). Entropies for the latent function g are calculated approximately in the marginal subspace corresponding to the observed data points. For this, the IVM employs a Gaussian approximation to the posterior distribution at these locations. The posterior approximation must be updated to evaluate the entropy decrease after the inclusion of each candidate data point. If there are n candidate inputs under consideration, a total of $\mathcal{O}(n)$ posterior updates are required. By contrast, BALD only requires $\mathcal{O}(1)$ updates. In practice, the IVM approach is infeasible in sophisticated models such as the proposed multi-task approach.

Finally, Tong and Koller (2001) propose an algorithm for active learning with support vector machines. This method approximates the version space (the set of hyperplanes consistent with the data) with a simpler object, such as a hypersphere. The algorithm selects the data point whose dual plane is closest to bisecting this hypersphere.

We now describe the experimental procedure used to compare BALD to these approaches. The datasets were divided randomly into pool and test sets. Each algorithm was initialized with two data points, one from each class, drawn randomly from the pool. The algorithms select points sequentially, and their classification error was assessed on the test set after each query. The procedure was repeated for several random splits of the data to assess statistical significance. Figure 1 provides a summary of the results. BALD can be seen to outperform consistently the alternative algorithms across many datasets. The closest competitor is Maximum Entropy Sampling, which we use as a benchmark active learning algorithm for use with the multi-task preference model in the main paper.

Dataset	**BALD**	**Random**	**Entropy**	**QBC$_2$**	**QBC$_{100}$**	**IVM**	**SVM**
austra	**18.54 ± 2.94**	44.15 ± 12.63	22.46 ± 6.20	68.38 ± 1.38	29.31 ± 5.06	28.46 ± 6.58	55.00 ± 1.00
cancer	**16.80 ± 0.59**	22.20 ± 1.25	21.10 ± 0.48	39.65 ± 0.41	18.95 ± 1.34	21.35 ± 0.50	24.40 ± 8.30
crabs	9.80 ± 0.58	11.40 ± 1.29	**9.20 ± 0.49**	17.00 ± 1.26	10.20 ± 0.97	13.60 ± 1.86	23.20 ± 7.29
letter D v. P	**45.30 ± 1.14**	92.10 ± 2.41	51.50 ± 0.83	48.80 ± 1.34	49.10 ± 1.38	51.00 ± 0.84	N/A
letter E v. F	**30.17 ± 1.11**	71.50 ± 17.72	34.33 ± 0.42	44.67 ± 2.12	30.67 ± 1.65	33.00 ± 2.27	N/A
vehicle	**33.20 ± 2.11**	75.30 ± 7.38	36.60 ± 1.74	85.20 ± 7.16	35.00 ± 1.80	38.20 ± 2.00	41.60 ± 1.64
wine	**8.80 ± 0.37**	26.60 ± 8.57	10.80 ± 1.66	36.40 ± 8.36	12.60 ± 1.78	20.40 ± 9.92	23.80 ± 3.48
wdbc	**18.15 ± 0.37**	47.00 ± 1.46	22.55 ± 1.05	43.85 ± 1.39	23.40 ± 1.05	21.40 ± 0.85	45.70 ± 1.75

Table 1: Performance of BALD and other active learning algorithms on several binary classification datasets from the UCI repository. Entries indicate the number of datapoints (plus or minus one standard error of the mean) required to achieve 95% of the predictive performance achieved by including the entire pool set. Bold typeface indicates the best performing algorithm for each dataset. N/A indicates that the corresponding algorithms did not meet the 95% performance level by the end of the simulation.

6 Detailed description of the analyzed datasets

The following paragraphs describe the datasets used for the experiments in Section 2.1 of the main document.

Synthetic. We generated 10 items with feature vectors $\mathbf{x}_i = (x_{i1}, x_{i2})$, where x_{i1} and x_{i2} are uniformly distributed with zero mean and unit variance, for $i = 1, \ldots, 10$. The user preferences are obtained using $D = 5$ latent functions $h_1, \ldots, h_5$ sampled from a Gaussian process with zero mean and preference kernel given by a squared exponential kernel with unit length-scale. The preferences for the u-th user are generated according to the sign of $g_u(\mathbf{x}_i, \mathbf{x}_j) = \sum_{d=1}^{5} w'_d(\mathbf{u}_u) h_d(\mathbf{x}_i, \mathbf{x}_j) + \epsilon_{ij}$, where $\epsilon_{ij} \sim \mathcal{N}(0, 1)$, the user features $\mathbf{u}_u$ are generated in the same manner as the feature vectors for the items and the functions $w'_1, \ldots, w_D$ follow the same prior distribution as $h_1, \ldots, h'_5$.

Jura. This dataset contains concentration measurements for 7 heavy metals in soils of the Swiss Jura region at 359 locations (Atteia et al., 1994; Webstet et al., 1994). We standardized the measurements of each heavy metal to have zero mean and unit standard deviation across the whole dataset. The standardized measurements are used as utility values to generate preferences for any pair of heavy metals at each location. Therefore, in this dataset, the locations correspond to users and each heavy metal represents a different item. To generate the item features, we randomly singled out 20 locations. The item features are given by the standardized measurements obtained at these locations. The user features correspond to the x and y coordinates for the measurements as well as the rock and land type.

MovieLens. This dataset contains 1 million ratings from 6,000 users on 4,000 movies. A total of 10 movies were randomly selected from the 50 movies with most ratings. We also selected those users with at least 7 ratings on these 10 movies. The remaining missing ratings were estimated using a nearest neighbor method. The ratings for each user were used as utility values in order to generate preferences for each pair of movies. The features for each user are gender, age and occupation. The features for each movie are genres such as *action*, *comedy* or *adventure*.

Sushi. This dataset contains complete rankings given by 5,000 users on 10 different types of sushi (Kamishima et al., 2005), where each sushi includes as features style, major group, minor group, heaviness, consumption frequency, normalized price and sale frequency. The different users are also represented by a set of features which include gender, age and geographical/regional information.

Election. This dataset contains the votes obtained by 8 political parties (items) at 650 constituencies (users) in the 2010 general elections in the UK. We only kept data for those constituencies with at least votes for more than 6 parties. Missing votes were estimated using a nearest neighbor method. To generate feature vectors for each item, we randomly singled out 20 constituencies and used the corresponding votes as features. The features for each 'user' are the corresponding coordinates (latitude and longitude) of the centroid of the constituency on the map.

7 Model of Birlutiu et al.

As in the single-task preference learning model, a different GP classifier is fitted to the data generated by each user. However, the different classifiers are now connected by a common GP prior for the latent preference functions which is optimized to fit the data Birlutiu et al. (2009). Let g_u be the u-th user's latent preference function and let $\boldsymbol{g}_u$ be the k-dimensional vector with the evaluation of this function at all the observed pairs of items, that is, $k = P$. Let $\bar{\boldsymbol{\mu}}$ and $\bar{\boldsymbol{\Sigma}}$ denote the prior mean and prior covariance matrix of $\boldsymbol{g}_u$. Then $\bar{\boldsymbol{\mu}}$ and $\bar{\boldsymbol{\Sigma}}$ are iteratively refined by an EM algorithm which iterates the following steps:

E-step Estimate the sufficient statistics (mean $\boldsymbol{\mu}_u$ and covariance matrix $\boldsymbol{\Sigma}_u$) of the posterior distribution of $\boldsymbol{g}_u$ for user $u = 1, \ldots, U$, given the current estimates at step t of the parameters $\bar{\boldsymbol{\mu}}^{(t)}$ and $\bar{\boldsymbol{\Sigma}}^{(t)}$ of the common GP prior.

M-step Re-estimate the parameters of the GP prior using

$$\bar{\boldsymbol{\mu}}^{(t+1)} = \frac{1}{U}\sum_{u=1}^{U} \boldsymbol{\mu}_u \,,$$

$$\bar{\boldsymbol{\Sigma}}^{(t+1)} = \frac{1}{U}\sum_{u=1}^{U} (\bar{\boldsymbol{\mu}}^{(t)} - \boldsymbol{\mu}_u)^{\mathrm{T}}(\bar{\boldsymbol{\mu}}^{(t)} - \boldsymbol{\mu}_u) + \frac{1}{U}\sum_{u=1}^{U} \boldsymbol{\Sigma}_u \,.$$

On the first iteration of the EM algorithm we fix $\bar{\boldsymbol{\mu}}^{(0)} = \mathbf{0}$ and compute $\bar{\boldsymbol{\Sigma}}^{(0)}$ by evaluating a preference covariance function at all the possible pairs of items. This preference covariance function is generated by a squared exponential kernel with unit lengthscale. The computational cost of the EM algorithm is rather high since each iteration requires the inversion of U covariance matrices of dimension $P \times P$, where P is the total number of observed item pairs. To reduce the computational burden, we limit the number of iterations of the EM algorithm to 20. In our experiments, increasing the number of EM iterations above 20 did not lead to improvements in the predictive performance of this method.

8 Model of Bonilla et al.

An alternative multi-task model for learning pairwise user preferences is described by Bonilla et al. (2010). This approach is based on the assumption that users with similar characteristics or feature vectors should have similar preferences. In particular, there is a single large latent function g which depends on both the features of the two items to be compared, $\mathbf{x}_i$ and $\mathbf{x}_j$, and the specific feature vector for the user who makes the comparison, $\mathbf{u}_u$. Within the framework of the preference kernel, the likelihood function for Bonilla's model is

$$\mathcal{P}(y|\mathbf{u}_u, \mathbf{x}_i, \mathbf{x}_j, g) = \Phi(yg(\mathbf{x}_i, \mathbf{x}_j, \mathbf{u}_u) \tag{63}$$

and the prior for the utility function g is a Gaussian process with zero mean and covariance function

$$k_{\text{Bonilla}}((\mathbf{u}_u, \mathbf{x}_i, \mathbf{x}_j), (\mathbf{u}_s, \mathbf{x}_k, \mathbf{x}_l),) = k_{\text{users}}(\mathbf{u}_u, \mathbf{u}_s) k_{\text{pref}}((\mathbf{x}_i, \mathbf{x}_j), (\mathbf{x}_k, \mathbf{x}_l)) \,, \tag{64}$$

where k_{pref} is a preference kernel and k_{users} is a covariance function for user features. This latter function will be large when $\mathbf{u}_u$ and $\mathbf{u}_s$ are similar to each other and small otherwise. Therefore, the effect of k_{users} in (64) is to favor that users with similar feature vectors agree on their preferences. The preference kernel allows us to do efficient approximate inference in Bonilla's model using any standard implementation of expectation propagation for the binary classification problem with GPs. However, the computational cost of Bonilla's method is rather high. When the preference kernel is used, the cost of this technique is $\mathcal{O}((\sum_{u=1}^{U} M_u)^3)$, where U is the total number of users and M_u is the number of pairs evaluated by the u-th user. By contrast, when the standard framework for preference learning with GPs is used, the cost of this method is $\mathcal{O}((\sum_{u=1}^{U} M_u')^3)$, where M_u' denotes the number of different items evaluated by the u-th user. In practice,

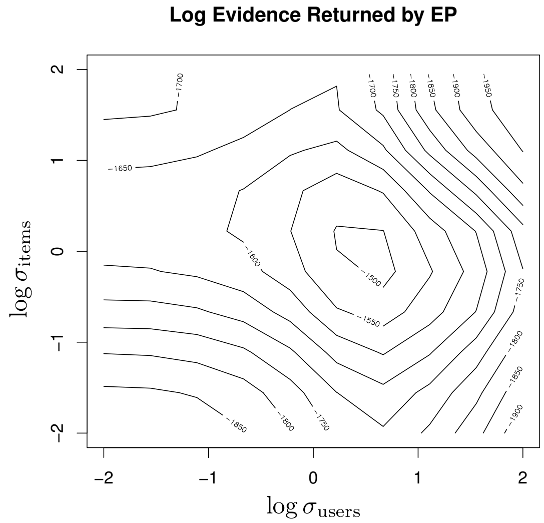

Figure 2: Logarithm of the evidence returned by EP when run on the first training set of the experiments with synthetic data. Different values are considered for the lengthscale parameters σ_{users} and σ_{items}. The synthetic data are generated using $\log \sigma_{\text{users}} = 0$ and $\log \sigma_{\text{items}} = 0$. The highest evidence returned by EP corresponds to values of $\log \sigma_{\text{users}}$ and $\log \sigma_{\text{items}}$ close to zero.

Table 2: Average test error with 100 users.

Dataset	**CPU**	**CP**	**BI**	**BO**	**SU**
Synthetic	**0.163±0.007**	0.182±0.007	0.255±0.016	0.169±0.008	0.228±0.008
Sushi	0.125±0.009	**0.115±0.010**	0.196±0.011	0.253±0.008	0.153±0.010
MovieLens	0.187±0.008	**0.166±0.006**	0.205±0.009	0.314±0.007	0.223±0.008
Election	0.232±0.015	**0.154±0.015**	**0.153±0.011**	0.385±0.017	0.309±0.014
Jura	0.162±0.015	**0.154±0.011**	0.188±0.022	0.181±0.014	0.185±0.013

Bonilla's method is infeasible when we have observations for more than a few hundred users. Additionally, this method requires that feature vectors are available for the different users and that users with similar feature vectors generate similar preference observations. When these conditions do not hold, Bonilla's method may lead to poor predictive performance.

9 Tuning the kernel lengthscale

In this section we perform an additional experiment to show that approximation of the model evidence given by EP can be used to tune the kernel hyper-parameters in the proposed multi-task model. For this, we use the synthetic dataset described in the main document. Figure 2 shows a contour plot of the log-evidence returned by EP when run on the first training set of the experiments with synthetic data and 100 users. Different values are considered for the lengthscale parameters σ_{users} and σ_{items}. The synthetic data are generated using $\log \sigma_{\text{users}} = 0$ and $\log \sigma_{\text{items}} = 0$. The highest evidence returned by EP corresponds to values of $\log \sigma_{\text{users}}$ and $\log \sigma_{\text{items}}$ close to zero. In this experiment we are running EP using a total of 20 latent functions, while the data are generated using only 5 latent functions. As mentioned in the main document, the proposed multi-task model seems to be robust to over-fitting and over-estimation of the number of latent functions does not seem to harm predictive performance.

Table 3: Test error for each method and active learning strategy with at most 1000 users.

Dataset	CPU-B	CPU-E	CPU-R	CP-B	CP-E	CP-R	SU-B	SU-E	SU-R
Synthetic	**0.136±0.004**	**0.136±0.004**	0.14±0.005	**0.154±0.004**	0.161±0.004	0.179±0.004	**0.253±0.005**	0.263±0.007	0.276±0.006
Sushi	**0.150±0.004**	0.155±0.005	0.185±0.004	**0.143±0.05**	0.151±0.05	0.179±0.06	**0.183±0.06**	0.199±0.05	0.215±0.05
MovieLens	**0.172±0.005**	0.177±0.006	0.200±0.006	**0.165±0.005**	0.171±0.006	0.196±0.005	**0.233±0.006**	0.24±0.004	0.253±0.007
Election	0.221±0.015	**0.188±0.016**	0.265±0.014	**0.107±0.016**	**0.103±0.0010**	0.184±0.015	**0.331±0.014**	0.349±0.017	0.344±0.013
Jura	**0.143±0.009**	**0.144±0.011**	0.17±0.011	**0.138±0.009**	**0.138±0.0010**	0.17±0.011	0.183±0.013	**0.169±0.012**	0.201±0.012

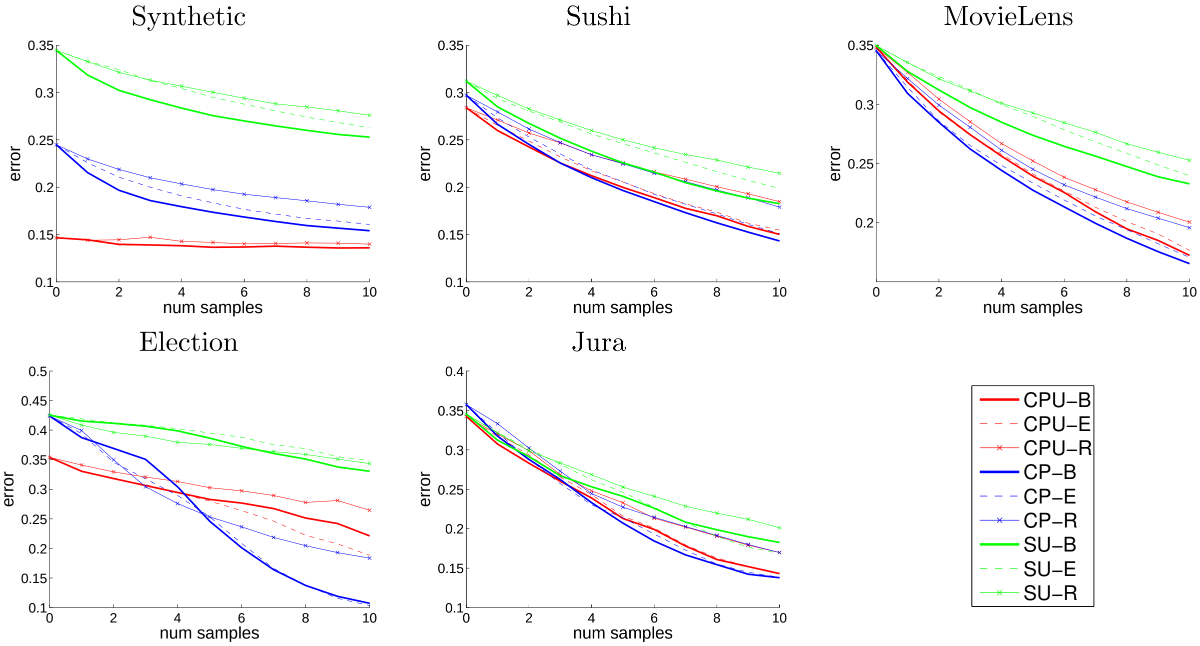

Figure 3: Average test error for CPU, CP and SU, using the strategies BALD (-B), entropy (-E) and random (-R) for active learning.

10 Complete figures for active learning on large datasets

Figure 9 shows the learning curves for all methods on all the datasets. Tables 2 and 3 are reproductions of the tables in the results section of the main paper with information regarding the standard deviations of the results.

References

Atteia, O., Dubois, J.-P., and Webster, R. (1994). Geostatistical analysis of soil contamination in the swiss jura. *Environmental Pollution*, 86(3):315 – 327.

Birlutiu, A., Groot, P., and Heskes, T. (2009). Multi-task preference learning with gaussian processes. In *Proceedings of the 17th European Symposium on Artificial Neural Networks (ESANN)*, pages 123–128.

Bishop, C. M. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer.

Bonilla, E. V., Guo, S., and Sanner, S. (2010). Gaussian process preference elicitation. In *Advances in neural information processing systems*, pages 262–270.

Freund, Y., Seung, H., Shamir, E., and Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine learning*, 28(2):133–168.

Gredilla, M. L. (2010). *Sparse Gaussian Processes for Large-scale Machine Learning*. PhD thesis, Universidad Carlos III de Madrid.

Hernández-Lobato, J. M. (2010). *Balancing Flexibility and Robustness in Machine Learning: Semi-parametric Methods and Sparse Linear Models*. PhD thesis, Universidad Autónoma de Madrid.

Kamishima, T., Kazawa, H., and Akaho, S. (2005). Supervised ordering - an empirical survey. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005)*, pages 673–676.

Lawrence, N., Seeger, M., and Herbrich, R. (2002). Fast sparse gaussian process methods: The informative vector machine. *Advances in neural information processing systems*, 15:609–616.

Minka, T. (2001). *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology.

Minka, T. and Lafferty, J. (2002). Expectation-propagation for the generative aspect model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 352–359.

Naish-Guzman, A. and Holden, S. B. (2007). The generalized fitc approximation. In *Advances in Neural Information Processing Systems 20*.

Raiko, T., Ilin, A., and Juha, K. (2007). Principal component analysis for large scale problems with lots of missing values. In Kok, J., Koronacki, J., Mantaras, R., Matwin, S., Mladenic, D., and Skowron, A., editors, *Machine Learning: ECML 2007*, volume 4701 of *Lecture Notes in Computer Science*, pages 691–698. Springer Berlin / Heidelberg.

Sebastiani, P. and Wynn, H. (2000). Maximum entropy sampling and optimal Bayesian experimental design. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1):145–157.

Snelson, E. and Ghahramani, Z. (2005). Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*.

Stern, D. H., Herbrich, R., and Graepel, T. (2009). Matchbox: large scale online bayesian recommendations. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 111–120, New York, NY, USA. ACM.

Tong, S. and Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66.

van Gerven, M., Cseke, B., de Lange, F., and Heskes, T. (2010). Efficient Bayesian multivariate fMRI analysis using a sparsifying spatio-temporal prior. *Neuroimage*, 50:150–161.

Webstet, R., Atteia, O., and Dubois, J.-P. (1994). Coregionalization of trace metals in the soil in the swiss jura. *European Journal of Soil Science*, 45(2):205–218.