Collaborative Gaussian Processes for Preference Learning

Neil Houlsby, Jose Miguel Hernández-Lobato, Ferenc Huszár, Zoubin Ghahramani

Computational and Biological Learning Lab, Department of Engineering, University of Cambridge

UNIVERSITY OF CAMBRIDGE

Multi-user Preference Learning

- General approach: model a user with a latent 'utility' function f such that $f(\mathbf{x}_i) > f(\mathbf{x}_j)$ when item i is preferred to item j, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are feature vectors for the items.
- GP models are popular for *single-user* preference learning [Chu and Ghahramani, 2005].
- Multiple users: leverage shared behavior, may or may not have user features.
- Current approaches require solving at least U Gaussian processes problems, where U is the number of users in the system [Bonilla et al., 2010, Birlutiu et al., 2010].

The Task

- **Goal**: Build a scalable multi-user probabilistic preference learner that may incorporate features if they are available (and useful).
- **Approach**: Combine dimensionality reduction methods from the field of collaborative filtering with flexible Gaussian process models for learning user preferences.
- **Further desiderata**: Efficient inference with preference data, 'active sampling' of item pairs for efficient data collection.

Reformulating Preference Learning as Binary Classification

- Let $\mathbf{x} \in \mathcal{X}$ denote the item features vectors, and $y \in \{-1, +1\}$ the preference labels. $f : \mathcal{X} \mapsto \mathbb{R}$ is a user's 'preference' or utility function. GP preference learning [Chu and Ghahramani, 2005]:

$$\mathcal{P}(y|\mathbf{x}_i, \mathbf{x}_j, f) = \Phi[(f[\mathbf{x}_i] - f[\mathbf{x}_j])y]\,,$$

where Φ = Gaussian c.d.f.

- Define $g : \mathcal{X}^2 \mapsto \mathbb{R}$ as $g(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i) - f(\mathbf{x}_j)$, now

$$\mathcal{P}(y|\mathbf{x}_i, \mathbf{x}_j, g) = \Phi[g(\mathbf{x}_i, \mathbf{x}_j)y]\,.$$

- GP prior on f and a linear operation $\rightarrow$ GP on g; derive the *preference kernel*:

$$k_{\mathrm{pref}}((\mathbf{x}_i, \mathbf{x}_j), (\mathbf{x}_k, \mathbf{x}_l)) = k(\mathbf{x}_i, \mathbf{x}_k) + k(\mathbf{x}_j, \mathbf{x}_l) - k(\mathbf{x}_i, \mathbf{x}_l) - k(\mathbf{x}_j, \mathbf{x}_k)\,.$$

- Puts all *anti-symmetry* constraints into the prior, greatly simplifies inference.

Active Sampling of Item Pairs

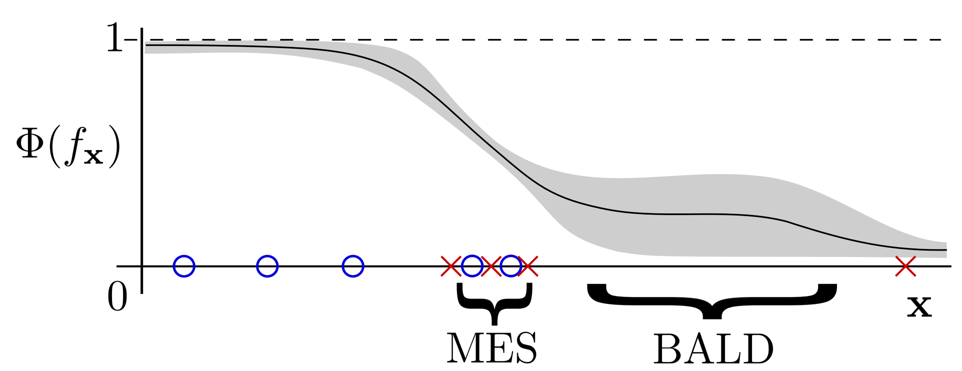

Figure: Toy example with 1D input. Circles and crosses denote labelled data. The plot shows the mean and variance of the predictive distribution. Maximum entropy sampling (MES) samples from the region of highest marginal uncertainty, ignoring the second term in (2). BALD samples from the region of greatest uncertainty in the latent function.

- Classic objective, minimize the entropy $\mathrm{H}[\mathcal{P}]$ of the posterior over the parameters:

$$\mathrm{H}[\mathcal{P}(g|\mathcal{D})] - \mathbb{E}_{\mathcal{P}(y|\mathbf{x}_i,\mathbf{x}_j,\mathcal{D})}\left[\mathrm{H}[\mathcal{P}(g|y, \mathbf{x}_i, \mathbf{x}_j, \mathcal{D})]\right]\,. \qquad (1)$$

- Problems:
  1. Computational: $2n$ posterior updates required (n = number of unseen pairs).
  2. Intractable: hard to compute entropy in parameter space, approximations (e.g. Laplace) required for GPs.
- Solution, re-formulate:

$$\mathrm{H}[\mathcal{P}(y|\mathbf{x}_i, \mathbf{x}_j, \mathcal{D})] - \mathbb{E}_{\mathcal{P}(g|\mathcal{D})}\left[\mathrm{H}\left[\mathcal{P}(y|\mathbf{x}_i, \mathbf{x}_j, g)\right]\right]\,. \qquad (2)$$

- We call this *Bayesian Active Learning with Disagreement*.
  - Eq. (2) requires only 1 posterior update, and entropies of Bernoulli variables only.
  - First term yields 'maximum entropy sampling'. The second discourages locations of inherent uncertainty (Fig. 1).
- For a model with a GP prior on g, Probit likelihood function (exploiting the preference kernel) and a Gaussian approximation (EP, Laplace) to the posterior, Eq. (2) becomes:

$$\mathrm{h}\left[\Phi\left(\frac{\mu_\mathbf{x}}{\sqrt{\sigma^2_\mathbf{x} + 1}}\right)\right] - \frac{C}{\sqrt{\sigma^2_\mathbf{x} + C^2}} \exp\left(\frac{-\mu^2_\mathbf{x}}{2\left(\sigma^2_\mathbf{x} + C^2\right)}\right)\,.$$

where $h[f] = -f \log f - (1 - f) \log(1 - f)$ and $C = \sqrt{\pi \log 2/2}$.

The Model - Collaborative Gaussian Processes

- Introduce a set of 'shared latent functions': $\mathbf{H} = \{h_1 \dots h_D\}$, $D \ll U$.
- For the u-th user, construct the 'user latent function' as:

$$g_u(\mathbf{x}_j, \mathbf{x}_k) = \sum_{d=1}^{D} w_{u,d} h_d(\mathbf{x}_j, \mathbf{x}_k)\,.$$

- $\mathbf{W} = \{\{w_{u,d}\}_{u=1}^{U}\}_{d=1}^{D}$ are the user-specific weights. If user features are available $\mathbf{U} = \{\mathbf{u}_1 \dots \mathbf{u}_U\}$ replace these with functions: $w_d(\mathbf{u})$.

Bayesian Formulation

- The likelihood uses the Probit function:

$$\mathcal{P}(\mathbf{T}^{(\mathcal{D})}|\mathbf{G}^{(\mathcal{D})}) = \prod_{u=1}^{U}\prod_{i=1}^{M_u} \Phi[t_{u,z_{u,i}} g_{u,z_{u,i}}]\,.$$

- GP priors on 'shared latent functions' and weights:

$$\mathcal{P}(\mathbf{H}|\mathbf{X}, \mathcal{L}) = \prod_{j=1}^{D} \mathcal{N}(\mathbf{h}_j|\mathbf{0}, \mathbf{K}_{\mathrm{items}})\,, \qquad \mathcal{P}(\mathbf{W}|\mathbf{U}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{w}_{\cdot,d}|\mathbf{0}, \mathbf{K}_{\mathrm{users}})\,.$$

- Enforce consistency in the matrix factorization:

$$\mathcal{P}(\mathbf{G}^{(\mathcal{D})}|\mathbf{W}, \mathbf{H}) = \prod_{u=1}^{U}\prod_{i=1}^{M_u} \delta[g_{u,z_{u,i}} - \mathbf{w}_u \mathbf{h}_{\cdot,z_{u,i}}]\,.$$

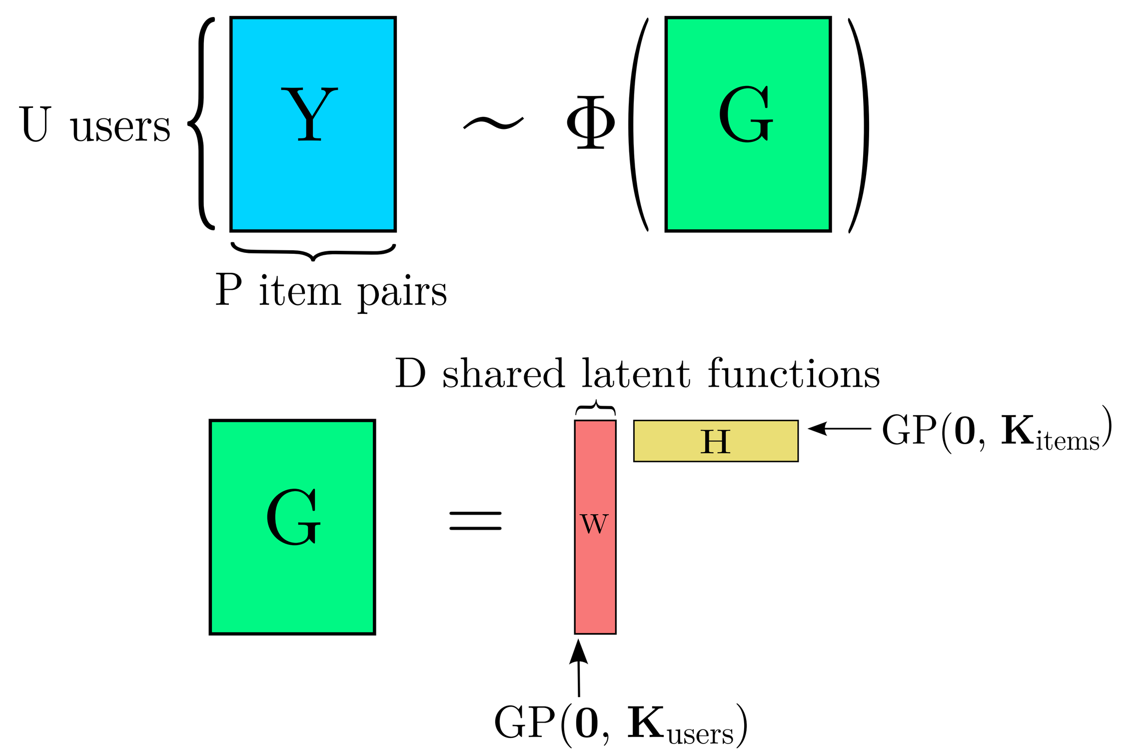

Figure: Graphical depiction of the matrices used in the proposed model.

Table: Notation

$\mathcal{L}$	List of P item pairs evaluated by users.
$\mathcal{D}$	Indices of particular items in $\mathcal{L}$ evaluated by each user.
$\mathbf{X}, \mathbf{U}$	Set of item and user feature vectors respectively.
$\mathbf{T}^{(\mathcal{D})} \in \{-1, +1\}^{U \times P}$	Matrix of preference evaluations corresponding to items in $\mathcal{D}$.
$\mathbf{G}^{(\mathcal{D})} \in \mathbb{R}^{U \times P}$	Matrix of 'user preference functions'.
$\mathbf{H} \in \mathbb{R}^{D \times P}$	Matrix of 'shared preference functions'.
$\mathbf{W} \in \mathbb{R}^{U \times D}$	Matrix of user weights.

Inference

- Hybrid scheme: combine expectation propagation and variational Bayes.
- Approximate the posterior with a fully factorised product of Gaussians Q:

$$\overbrace{\mathcal{P}(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})}|\mathbf{T}^{(\mathcal{D})}, \mathbf{X}, \mathcal{L})}^{\text{posterior distribution}} = \frac{\overbrace{\mathcal{P}(\mathbf{T}^{(\mathcal{D})}|\mathbf{G}^{(\mathcal{D})})}^{f_1}\overbrace{\mathcal{P}(\mathbf{G}^{(\mathcal{D})}|\mathbf{W}, \mathbf{H})}^{f_2}\overbrace{\mathcal{P}(\mathbf{W}|\mathbf{U})}^{f_3}\overbrace{\mathcal{P}(\mathbf{H}|\mathbf{X}, \mathcal{L})}^{f_4}}{\mathcal{P}(\mathbf{T}^{(\mathcal{D})}|\mathbf{X}, \mathcal{L})}\,.$$

$$\approx \mathcal{Q}(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})}) = \prod_{i=1}^{4} \hat{f}_i(\mathbf{W}, \mathbf{H}, \mathbf{G}^{(\mathcal{D})})\,.$$

- Iteratively refine $\hat{f}_1, \hat{f}_3, \hat{f}_4$ with EP i.e. minimize $\mathrm{KL}[\mathcal{Q}^{\setminus i} f_i || \mathcal{Q}^{\setminus i} \hat{f}_i]$ with respect to the parameters.
- Refine $\hat{f}_2$ (the matrix factorization) with VB i.e. reverse the direction of the KL divergence.
- Further speed-up achieved using the FITC approximation.

Small Scale Experiments

Table: Related algorithms and their computational complexities.
P = num pairs, U = num users, M_u = num preferences from user u .

Notation	Algorithm	Complexity of Inference
CPU	Collaborative Preference (with user features)	$\mathcal{O}(DU^3 + DP^3 + D\sum_u M_u)$
CP	Collaborative Preference (without user features)	$\mathcal{O}(DP^3 + D\sum_u M_u)$
Bi	[Birlutiu et al., 2010]	$\mathcal{O}(UP^3)$
Bo	[Bonilla et al., 2010]	$\mathcal{O}((\sum_u M_u)^3)$
SU	Single-user [Chu and Ghahramani, 2005]	$\mathcal{O}(\sum_u M_u^3)$

Table: Average test error with 100 users.

Dataset	CPU	CP	BI	BO	SU
Synthetic	0.162	0.180	0.175	**0.157**	0.226
Sushi	0.171	0.163	**0.160**	0.266	0.187
MovieLens	0.182	**0.166**	0.168	0.302	0.217
Election	0.199	0.123	**0.077**	0.401	0.300
Jura	0.159	**0.153**	**0.153**	0.254	0.181

Table: Training times (s) with 100 users.

Dataset	CPU	CP	BI	BO	SU
Synthetic	7.793	9.498	22.524	311.574	0.927
Sushi	5.694	4.307	20.028	215.136	0.817
MovieLens	5.313	4.013	19.366	69.048	0.604
Election	13.134	12.408	20.880	120.011	0.888
Jura	3.762	2.404	15.234	88.502	0.628

Large Scale Experiments

Table: Test error for each method and active learning strategy with at most 1000 users.
$-\mathbf{B}$ = BALD, $-\mathbf{E}$ = MES, $-\mathbf{R}$ = random sampling

Dataset	CPU-B	CPU-E	CPU-R	CP-B	CP-E	CP-R	SU-B	SU-E	SU-R
Synthetic	**0.135**	**0.135**	0.139	**0.153**	0.160	0.173	**0.249**	0.259	0.268
Sushi	**0.148**	0.153	0.178	**0.144**	0.151	0.176	**0.179**	0.197	0.212
MovieLens	**0.170**	0.176	0.199	**0.163**	0.170	0.195	**0.225**	0.235	0.248
Election	0.202	**0.158**	0.224	**0.097**	**0.093**	0.151	**0.332**	0.346	0.338
Jura	**0.143**	**0.141**	0.168	**0.138**	**0.138**	0.169	0.176	**0.166**	0.197

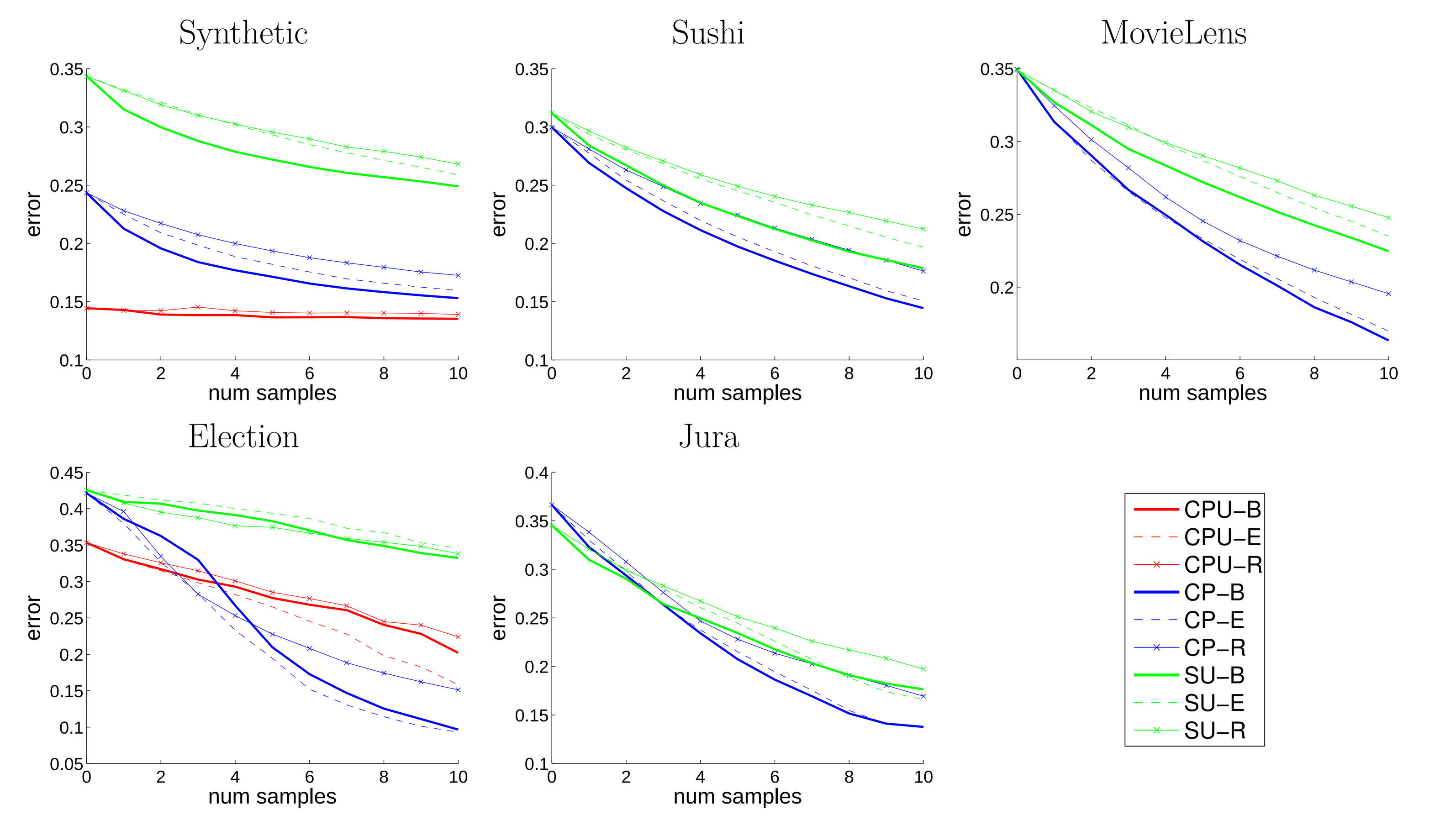

Figure: Average test error for CPU, CP and SU, using the strategies BALD (-B), entropy (-E) and random (-R) for active learning. For clarity, the curves for CPU are included only in the Synthetic and Election datasets.

Acknowledgements

NH is a recipient of the Google Europe Fellowship in Statistical Machine Learning, and this research is supported in part by this Google Fellowship. JMH is supported by Infosys Labs, Infosys Limited.

References

[Birlutiu et al., 2010] Birlutiu A, Groot P, Heskes T. 2010.
Multi-task preference learning with an application to hearing aid personalization.
Neurocomputing 73:1177 – 1185.

[Bonilla et al., 2010] Bonilla EV, Guo S, Sanner S. 2010.
Gaussian process preference elicitation.
In: Advances in Neural Information Processing Systems 23. pp 262–270.

[Chu and Ghahramani, 2005] Chu W, Ghahramani Z. 2005.
Preference learning with Gaussian processes.
In: Proceedings of the 22nd international conference on Machine learning. pp 137–144.