Learning Community-based Preferences via Dirichlet Process Mixtures of Gaussian Processes

Ehsan Abbasnejad
ANU & NICTA
Canberra, Australia
eabbasnejad@nicta.com.au

Scott Sanner
NICTA & ANU
Canberra, Australia
ssanner@nicta.com.au

Edwin V. Bonilla
NICTA & ANU
Sydney, Australia
ebonilla@nicta.com.au

Pascal Poupart
University of Waterloo
Waterloo, Canada
ppoupart@cs.uwaterloo.ca

Abstract

Bayesian approaches to preference learning using Gaussian Processes (GPs) are attractive due to their ability to explicitly model uncertainty in users' latent utility functions; unfortunately existing techniques have cubic time complexity in the number of users, which renders this approach intractable for collaborative preference learning over a large user base. Exploiting the observation that user populations often decompose into communities of shared preferences, we model user preferences as an infinite Dirichlet Process (DP) mixture of communities and learn (a) the expected number of preference communities represented in the data, (b) a GP-based preference model over items tailored to *each* community, and (c) the mixture weights representing each user's fraction of community membership. This results in a learning and inference process that scales linearly in the number of users rather than cubicly and additionally provides the ability to analyze individual community preferences and their associated members. We evaluate our approach on a variety of preference data sources including Amazon Mechanical Turk showing that our method is more scalable and as accurate as previous GP-based preference learning work.

1 Introduction

Preference learning has become an important subfield in machine learning transcending multiple disciplines such as economics, operations research and social sciences. A wide range of applications in areas such as recommender systems, autonomous agents, human-computer interaction and e-commerce has motivated machine learning researchers to investigate flexible and effective ways to construct predictive preference models from preference observations [Fürnkranz and Hüllermeier, 2010].

This is a challenging problem since complex relations between users and their preferred items must be uncovered. Furthermore, flexible and principled ways to handle uncertainty over the users' preferences are required in order to balance what the system knows. To address these challenges, non-parametric Bayesian approaches based on Gaussian processes (GPs) [Rasmussen and Williams, 2006] have shown to be effective in real applications [Chu and Ghahramani, 2005a; Bonilla *et al.*, 2010; Platt *et al.*, 2002; Xu *et al.*, 2010]. However, one of the major limitations of preference learning approaches based on GPs is their cubic time complexity in both the number of users and items.

While the number of items in many preference prediction applications may be computationally manageable in a GP framework, the number of users may be much larger and often poses the greatest computational challenge. Fortunately, it is well-known that preferences across a user population often decompose into a smaller number of communities of commonly shared preferences [Postlewaite, 2011].

To exploit community structure in preferences, we propose a novel mixture model of GP-based preference learning. While we might first take a finite mixture model approach to modeling community-based preferences, we note that one of the most difficult parts of modeling communities is determining their number. Fortunately, we can exploit the Dirichlet Process [Müller and Quintana, 2004] framework to build infinite mixture models where the number of mixture components (communities) is inferred automatically. Hence, we model user preferences as an infinite Dirichlet Process (DP) mixture of communities and learn (a) the expected number of preference communities represented in the data, (b) a GP-based preference model over items within each community, and (c) the mixture weights representing each user's fraction of community membership. This results in a learning and inference process that scales linearly in the number of users rather than cubicly and as a side benefit provides the ability to analyze individual communities of preference while also learning how user preferences align with each community and each other.

We evaluate our approach on a variety of data including human preference data collected from Amazon Mechanical Turk showing that our method is more scalable than previous GP-based preference learning and as accurate since it is able to uncover latent shared-preference community structure present in our data.

2 GP-based Preference Learning

In this section, we briefly review the multi-user Gaussian Process based framework for Bayesian preference learning originally proposed by [Bonilla *et al.*, 2010] before modifying it

to a Dirichlet Process mixture in Section 3.

Let $U = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n\}$ be a set of n users and let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ be a set of m items and denote the set of observed preferences of each user $\mathbf{u} \in U$ with $\mathcal{D}^{\mathbf{u}} = \{\mathbf{x}_i \succ \mathbf{x}_j\}$ where $1 \leq i \leq m$ and $1 \leq j \leq m$. Given the preferences $\mathcal{D}^{\mathbf{u}}$ for $\mathbf{u}$, satisfaction of the von Neumann-Morgenstern axioms [Neumann and Morgenstern, 1944] justify the existence of utilities $f_i^{\mathbf{u}} \in \mathbb{R}$ for each item $\mathbf{x}_i \in X$ s.t. $\mathbf{x}_i^{\mathbf{u}} \succ \mathbf{x}_j^{\mathbf{u}} \in D^{\mathbf{u}}$ iff $f_i^{\mathbf{u}} > f_j^{\mathbf{u}}$. In order to model the distribution over these utilities, we denote a latent utility vector $\mathbf{f}$ for *all* users and items with $\mathbf{f} = [f_1^{\mathbf{u}_1}, f_2^{\mathbf{u}_1}, \ldots, f_m^{\mathbf{u}_n}]^T$. Then, we can define the likelihood over all the preferences given the latent functions as:

$$p(\mathcal{D}|\mathbf{f}, \alpha) = \prod_{\mathbf{u} \in U} \prod_{\mathbf{x}_i \succ \mathbf{x}_j \in \mathcal{D}^{\mathbf{u}}} p(\mathbf{x}_i \succ \mathbf{x}_j | f_i^{\mathbf{u}}, f_j^{\mathbf{u}}, \alpha) \quad (1)$$

$$\text{with} \qquad p(\mathbf{x}_i \succ \mathbf{x}_j | f_i^{\mathbf{u}}, f_j^{\mathbf{u}}, \alpha) = \Phi\left(\frac{f_i^{\mathbf{u}} - f_j^{\mathbf{u}}}{\alpha}\right), \quad (2)$$

where $\Phi(x) = \int_{-\infty}^{x} \mathcal{N}(y)dy$ and $\mathcal{N}(y)$ is a zero-mean Gaussian distribution with unit variance and $\alpha \in \mathbb{R}$ is the *discriminal dispersion* hyperparameter (indicating how strongly users discriminate differences of utility) as specified in Thurstone's model of comparative judgment [Thurstone, 1959].

In this model, $p(\mathbf{f}|\mathbf{K})$ is the prior over the latent utilities $\mathbf{f}$ and is defined via a GP with zero-mean function and a covariance matrix $\mathbf{K}$ that factorizes over users and items [Bonilla *et al.*, 2010]. Therefore:

$$p(\mathbf{f}|\mathbf{K}) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{K}), \quad \mathbf{K} = \mathbf{K}_u \otimes \mathbf{K}_x, \quad (3)$$

where $\mathbf{K}$ is a kernel matrix composed of the Kronecker product of the kernel matrix over the users $\mathbf{K}_u$ and the kernel matrix over the items $\mathbf{K}_x$. One interesting feature of this model is the inherent transfer of preferences across users through the correlated prior (which can be *learned* through hyperparameter optimization of $\mathbf{K}$), intended to help the prediction generalize to users $\mathbf{u}$ for which few preferences are recorded in $\mathcal{D}_{\mathbf{u}}$.

The posterior of the latent utility function $\mathbf{f}$ for each user and item given all the preferences is:

$$p(\mathbf{f}|\mathcal{D}, \mathbf{K}, \alpha) = \frac{1}{Z} p(\mathbf{f}|\mathbf{K}) p(\mathcal{D}|\mathbf{f}, \alpha), \quad (4)$$

with Z being the normalizer. This posterior is analytically intractable due to the non-Gaussian nature of the likelihood, requiring approximation as detailed in [Bonilla *et al.*, 2010]. What is simply critical to note for the purposes of this paper is that prediction with this Gaussian Process model requires a matrix inversion of dimensions equivalent to $\mathbf{K}$ (i.e., mn), thus having $O(m^3 n^3)$ time complexity. In the next section we seek to mitigate this computational complexity while at the same time exploiting community structure often present in social preference data [Postlewaite, 2011].

3 Dirichlet Process Mixtures of Community-based Preference GPs

In this section we propose to alter the GP preference based model of Section 2 to model users as a (potentially) infinite mixture of GP-based communities. Hence, we now assume there are an infinite number of possible communities of preference $C = \{1 \ldots \infty\}$ where for every user $\mathbf{u}$ there is a latent indicator $c_{\mathbf{u}} \in C$ indicating to which community $\mathbf{u}$ belongs. Further, we assume each community $c \in C$ has it's own latent utility function over *items only* $\mathbf{f}_c = [f_1^c, f_2^c, \ldots, f_m^c]^T$ since we assume all members of the community share common preferences. Hence, we can now define a likelihood in this model for the preference data conditioned on all latent utilities $\mathbf{f} = [\mathbf{f}_1, \ldots, \mathbf{f}_\infty]$ (where we have redefined $\mathbf{f}$ from the previous section) *and* the vector of latent community indicators for each user $\mathbf{c} = [c_{\mathbf{u}_1}, \ldots, c_{\mathbf{u}_n}]^T$:

$$p(\mathcal{D}|\mathbf{f}, \mathbf{c}, \alpha) = \prod_{\mathbf{u} \in U} \prod_{(i,j) \in \mathcal{D}^{\mathbf{u}}} p(\mathbf{x}_i \succ \mathbf{x}_j | f_i^{c_{\mathbf{u}}}, f_j^{c_{\mathbf{u}}}, \alpha) \quad (5)$$

with $p(\mathbf{x}_i \succ \mathbf{x}_j | f_i^{c_{\mathbf{u}}}, f_j^{c_{\mathbf{u}}}, \alpha)$ as defined in the previous section and a community-specific kernel prior $\mathbf{K}_x^c$ for *each* $\mathbf{f}_c$ ($c \in C$) over *items only*.

$$p(\mathbf{f}_c|\mathbf{K}_x^c) = \mathcal{N}(\mathbf{f}_c; \mathbf{0}, \mathbf{K}_x^c). \quad (6)$$

It is important to emphasize here that one important feature of our infinite community-based mixture model is that the hyper-parameters K_x^c of the utilities f_c are community-dependent and can be optimized as part of the learning process as described in Section 3.3 leading to improved community modeling and generalization over all item preferences within that community.

We assume that $\mathbf{c}$ is generated according to an infinite multinomial distribution with parameters $\boldsymbol{\pi} \in \mathbb{R}^{|C|}$ ($\sum_{i=1}^{|C|} \boldsymbol{\pi}_i = 1$), hence

$$p(\mathbf{c}|\boldsymbol{\pi}) = \prod_{i=1}^{|C|} \boldsymbol{\pi}_i^{n_i} \quad \text{where} \quad n_i = \sum_{\mathbf{u} \in U} \delta_{c_{\mathbf{u}}=i}. \quad (7)$$

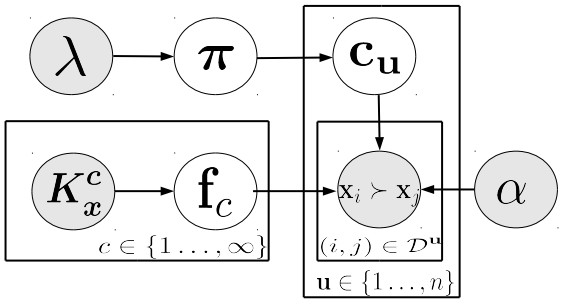


Figure 1: Our proposed generative graphical model for community-based preferences. There is a plate for users $\mathbf{u}$ with community membership indicator $c_{\mathbf{u}} \in C$ and an embedded plate for i.i.d. preference observations $\mathbf{x}_i \succ \mathbf{x}_j$ of user $\mathbf{u}$ depending on the community assignment $c_{\mathbf{u}}$ of $\mathbf{u}$, community $c_{\mathbf{u}}$'s latent utility function $\mathbf{f}_{c_{\mathbf{u}}}$, and the discriminal dispersion parameter α. There is a separate plate for communities $c \in C = \{1 \ldots, \infty\}$ which contains the latent utility function $\mathbf{f}_c$ drawn from a Gaussian Process conditioned on local (optimized) community parameters $\mathbf{K}_x^c$. Finally, each $c_{\mathbf{u}}$ is generated i.i.d. from an infinite multinomial distribution with parameters $\boldsymbol{\pi}$ and Dirichlet Process prior with concentration parameter λ.

Since the number of possible communities $|C|$ is infinite, we resort to Dirichlet processes by defining an infinite Dirichlet prior on the community distribution with concentration parameter λ [Neal, 1998; Rasmussen, 2000]:

$$p(\boldsymbol{\pi}|\lambda) = p(\pi_1, \ldots, \pi_{|C|}|\lambda) = \text{Dirichlet}(\lambda/|C|, \ldots, \lambda/|C|) = \frac{\Gamma(\lambda)}{\Gamma(\lambda/|C|)^{|C|}} \prod_{j=1}^{|C|} \pi_j^{\lambda/|C|-1}. \quad (8)$$

Altogether this generative framework is represented in the graphical model of Figure 1.

Our primary goal in inference is to obtain a sample posterior estimate over $\mathbf{f}$ and $\mathbf{c}$ to be used for future prediction. To this end, we begin by multiplying all of the likelihood and prior factors of our generative model in Figure 1 to obtain a superset of the desired joint posterior parameters:

$$p(\mathbf{f}, \mathbf{c}, \boldsymbol{\pi}|\mathcal{D}, \mathbf{K}_x, \lambda, \alpha) \propto \left[\prod_{c \in C} p(\mathbf{f}_c|\mathbf{K}_x^c)\right] \cdot \left[\prod_{\mathbf{u} \in U} \underbrace{\left[\prod_{(i,j) \in \mathcal{D}^{\mathbf{u}}} p(\mathbf{x}_i \succ \mathbf{x}_j | f_i^{c_\mathbf{u}}, f_j^{c_\mathbf{u}}, \alpha)\right]}_{p(\mathcal{D}^{\mathbf{u}}|\mathbf{f}, \mathbf{c}_\mathbf{u}, \alpha)} p(\mathbf{c}_\mathbf{u}|\boldsymbol{\pi})\right] p(\boldsymbol{\pi}|\lambda) \quad (9)$$

Here of course, we don't necessarily require a posterior estimate over $\boldsymbol{\pi}$ and we will see in Section 3.1 that it is in fact important to marginalize over $\boldsymbol{\pi}$ in the posterior to facilitate Gibbs sampling inference for Dirichlet processes.

However, before we dive into specific details, we first provide a general overview of our posterior inference framework. To perform inference in this Dirichlet Process mixture of GPs, we utilize the joint probability in Equation 9 and devise a collapsed, blocked Gibbs sampler that breaks the Gibbs sampling inference into two distinct steps, for which *different* inference algorithms are appropriate. Specifically, collapsing comes from marginalizing over $\boldsymbol{\pi}$ and blocking stems from repeating joint inference of $\mathbf{f}$ given $\mathbf{c}$. Gibbs sampling then repeats as follows until convergence:

- For each $\mathbf{c}_\mathbf{u}$, infer $p(\mathbf{c}_\mathbf{u}|\mathbf{c}_{\backslash \mathbf{u}}, \mathbf{f}, \mathcal{D}, \lambda, \alpha, \mathbf{K})$ via Gibbs sampling as discussed in Section 3.1.
- Infer $p(\mathbf{f}|\mathbf{c}, \mathcal{D}, \lambda, \alpha, \mathbf{K})$ via Expectation Propagation (EP) as discussed in Section 3.2 with hyperparameters optimized as discussed in Section 3.3.

Here we have merged all kernel hyperparameters into the set $\mathbf{K} = \{\mathbf{K}_x^c | c \in C\}$.

3.1 Inferring Community Membership

In our Gibbs sampler, given $\mathbf{f}$, we now wish to sample $\mathbf{c}$ – the community memberships for all users. Assuming that our blocked Gibbs sampler has already provided us with a sample of $\mathbf{f}$ for some fixed $\mathbf{c}^*$ sampled on the previous iteration, we now wish to sample each new $\mathbf{c}_\mathbf{u}$ in turn for the current iteration provided that we can define $p(\mathbf{c}_\mathbf{u}|\mathbf{c}_{\backslash \mathbf{u}}, \mathbf{f}, \mathcal{D}, \lambda, \alpha)$.

While we could sample $\mathbf{f}$ from $p(\mathbf{f}|\mathbf{c}^*, \mathcal{D}, \lambda, \alpha)$ to compute $p(\mathbf{c}_\mathbf{u}|\mathbf{c}_{\backslash \mathbf{u}}, \mathbf{f}, \mathcal{D}, \lambda, \alpha)$, this seems inefficient given that we can derive the full posterior $p(\mathbf{f}|\mathbf{c}^*, \mathcal{D}, \lambda, \alpha)$ in closed-form given our Gaussian Process inference machinery in Section 3.2. So instead we propose to compute $\mathbb{E}_{p(\mathbf{f}|\mathbf{c}^*, \mathcal{D}, \lambda, \alpha)}[p(\mathbf{c}_\mathbf{u}|\mathbf{c}_{\backslash \mathbf{u}}, \mathbf{f}, \mathcal{D}, \lambda, \alpha)]$.[1]

Now we derive an efficiently computable closed-form for sampling $\mathbf{c}_\mathbf{u}$ where we abbreviate the previous expectation to the shorter form $\mathbb{E}_{\mathbf{f}|\mathbf{c}^*}[p(\mathbf{c}_\mathbf{u}|\mathbf{c}_{\backslash \mathbf{u}}, \mathbf{f}, \mathcal{D}, \lambda, \alpha)]$:[2]

$$\begin{aligned}
\mathbb{E}_{\mathbf{f}|\mathbf{c}^*}[p(\mathbf{c}_\mathbf{u}|\mathbf{c}_{\backslash \mathbf{u}}, \mathbf{f}, \mathcal{D}, \lambda, \alpha)] &\propto \mathbb{E}_{\mathbf{f}|\mathbf{c}^*}\left[\int p(\underbrace{\mathbf{c}_\mathbf{u}, \mathbf{c}_{\backslash \mathbf{u}}}_{\mathbf{c}}, \mathbf{f}, \boldsymbol{\pi}|\mathcal{D}, \lambda, \alpha) d\boldsymbol{\pi}\right] \\
&\propto \mathbb{E}_{\mathbf{f}|\mathbf{c}^*}[p(\mathcal{D}^{\mathbf{u}}|\mathbf{f}, \mathbf{c}_\mathbf{u}, \alpha)] \underbrace{\int p(\mathbf{c}|\boldsymbol{\pi}) p(\boldsymbol{\pi}|\lambda) d\boldsymbol{\pi}}_{p(\mathbf{c}|\lambda) \propto p(\mathbf{c}_\mathbf{u}|\mathbf{c}_{\backslash \mathbf{u}}, \lambda)} \\
&\propto \left[\int \underbrace{p(\mathcal{D}^{\mathbf{u}}|\mathbf{f}, \mathbf{c}_\mathbf{u}, \alpha)}_{\text{Likelihood}} \underbrace{p(\mathbf{f}|\mathbf{c}^*, \mathcal{D}, \lambda, \alpha)}_{\text{Gaussian Process}} d\mathbf{f}\right] \underbrace{p(\mathbf{c}_\mathbf{u}|\mathbf{c}_{\backslash \mathbf{u}}, \lambda)}_{\text{Dirichlet Process}}
\end{aligned} \quad (10)$$

Thus in (10) we arrive at a closed-form computation that is straightforward to compute. In the square brackets, we need only use our GP posterior $f|\mathbf{c}^*$ to compute the product of the probabilities of each of user $\mathbf{u}$'s preferences $\mathbf{x}_i \succ \mathbf{x}_j \in \mathcal{D}^{\mathbf{u}}$ as defined in the next sections. This leaves us only to compute $p(\mathbf{c}_\mathbf{u} = c|\mathbf{c}_{\backslash \mathbf{u}}, \lambda)$ as is standard in Gibbs sampling for Dirichlet processes:

1. If c is an *active community* ($\exists \mathbf{c}_\mathbf{u} \in \mathbf{c}_{\backslash \mathbf{u}}$ s.t. $\mathbf{c}_\mathbf{u} = c$), then
$$p(\mathbf{c}_\mathbf{u} = c|\mathbf{c}_{\backslash \mathbf{u}}, \lambda) = \frac{\sum_{\mathbf{u}' \neq \mathbf{u}} \mathbb{I}[\mathbf{c}_{\mathbf{u}'} = c]}{N - 1 + \lambda} \quad (11)$$
where N is the number of non-empty communities.
2. Else c is a *new community* so
$$p(\mathbf{c}_\mathbf{u} = c|\mathbf{c}_{\backslash \mathbf{u}}, \lambda) = \frac{\lambda}{N - 1 + \lambda} \quad (12)$$

Hence all quantities required to sample $\mathbf{c}_\mathbf{u}$ have now been defined permitting sampling of each $\mathbf{c}_\mathbf{u}$ in turn to complete the community process sampling portion of the Gibbs sampling inference for our model. And the result is intuitive: a user $\mathbf{u}$ is more likely to join a community which provides a higher likelihood on its preference data $D^{\mathbf{u}}$. Additionally, this sampling process displays the well-known "rich-get-richer" effect of Dirichlet Processes since communities with more members have a higher probability of being selected.

3.2 Inferring Community Utility

Once we have sampled the block of all user assignments $\mathbf{c}$, it is now time to sample the block of Gaussian Process latent utilities $\mathbf{f}$ for each active community given $\mathbf{c}$. Thus, we first derive our target conditional distribution:

$$p(\mathbf{f}|\mathbf{c}, \mathcal{D}, \alpha, \mathbf{K}) = \frac{1}{Z} p(\mathcal{D}|\mathbf{f}, \mathbf{c}, \alpha) p(\mathbf{f}|\mathbf{K}) \quad (13)$$

[1] Of course, one can always sample $\mathbf{f}$ and avoid this expectation if preferred, but we conjecture that using the expectation will induce a lower-variance Gibbs sampling process with faster convergence.

[2] A detailed derivation of all math derived in these sections is provided in an online appendix at the authors' web pages.

As seen in Equation 1, since the likelihood is factorized we can take advantage of sequential approximation methods such as *Expectation Propagation* (EP) [Minka, 2001]. EP approximates the posterior $p(\mathbf{f}|\mathbf{c},\mathcal{D},\lambda,\mathbf{K})$ by a tractable distribution $q(\mathbf{f}|\mathbf{c})$. EP assumes that each likelihood term $p(\mathbf{x}_i \succ \mathbf{x}_j|f_i^{\mathbf{c_u}}, f_j^{\mathbf{c_u}}, \alpha)$ can be approximated by a distribution $q(f_i^{\mathbf{c_u}}, f_j^{\mathbf{c_u}}|\boldsymbol{\theta}_{\mathbf{c_u}})$ such that the approximated posterior $q(\mathbf{f}|\mathbf{c})$ factorizes over $q(f_i^{\mathbf{c_u}}, f_j^{\mathbf{c_u}}|\boldsymbol{\theta}_{\mathbf{c_u}})$. Then EP iteratively approximates each $q(f_i^{\mathbf{u}}, f_j^{\mathbf{u}}|\mathbf{c_u})$ in turn by dividing it out from the approximated posterior $q(\mathbf{f}|\mathbf{c})$ (obtaining the cavity distribution), multiplying in the true likelihood $p(\mathbf{x}_i \succ \mathbf{x}_j|f_i^{\mathbf{c_u}}, f_j^{\mathbf{c_u}}, \alpha)$, and projecting the result back to its factorized form by matching its moments to an updated $q(f_i^{\mathbf{c_u}}, f_j^{\mathbf{c_u}}|\boldsymbol{\theta}_{\mathbf{c_u}})$.

This overall procedure is motivated by the aim to minimize the KL$-$divergence between the true posterior $p(\mathbf{f}|\mathcal{D},\mathbf{c},\alpha,\mathbf{K})$ and its approximation $q(\mathbf{f}|\mathbf{c})$.

In the preference learning case we detailed earlier, we can approximate the posterior with a Gaussian:

$$q(\mathbf{f}|\mathbf{c}) = \left[\prod_c \frac{1}{\tilde{Z}^c} p(\mathbf{f}^c|\mathbf{K}^c)\right] \prod_{\mathbf{u}\in U} \prod_{\mathbf{x}_i \succ \mathbf{x}_j \in \mathcal{D}^{\mathbf{u}}\}} q(f_i^{\mathbf{c_u}}, f_j^{\mathbf{c_u}}|\alpha,\mathbf{K}) = \mathcal{N}(\mathbf{f};\boldsymbol{\mu}^c,\boldsymbol{\Sigma}^c). \quad (14)$$

where $\boldsymbol{\mu}^c$ and $\boldsymbol{\Sigma}^c$ denote the mean and covariance of the Gaussian distribution for the community user $\mathbf{u}$ belongs to corresponding to $\boldsymbol{\theta}_{\mathbf{c_u}}$. We are interested in locally approximating each likelihood term in Equation 1 as:

$$p(\mathbf{x}_i \succ \mathbf{x}_j|f_i^{\mathbf{c_u}}, f_j^{\mathbf{c_u}}, \alpha) \approx q(f_i^{\mathbf{c_u}}, f_j^{\mathbf{c_u}}|\boldsymbol{\theta}_{\mathbf{c_u}}) = \tilde{Z}^{\mathbf{u}}_{i,j} \mathcal{N}(f_i^{\mathbf{u}}, f_j^{\mathbf{u}}; \tilde{\boldsymbol{\mu}^c}_{\mathbf{u},[i,j]}, \tilde{\boldsymbol{\Sigma}^c}_{\mathbf{u},[i,j]}), \quad (15)$$

where $\mathcal{N}(f_i^{\mathbf{u}}, f_j^{\mathbf{u}}; \tilde{\boldsymbol{\mu}^c}_{\mathbf{u},[i,j]}, \tilde{\boldsymbol{\Sigma}^c}_{\mathbf{u},[i,j]})$ denotes the local two-dimensional Gaussian over $[f_i^{\mathbf{u}}, f_j^{\mathbf{u}}]^\top$ with mean $\tilde{\boldsymbol{\mu}^c}_{\mathbf{u},[i,j]}$ and covariance $\tilde{\boldsymbol{\Sigma}^c}_{\mathbf{u},[i,j]}$ corresponding to items i and j. Full details are provided in an online appendix.[2]

3.3 Optimizing Kernel Hyper-parameters

One advantage of our model is that the hyper-parameters of the model can be learned independently for each community. Even though we can define distributions over covariance (Inverse-Wishart), here due to the computational cost of a full Bayesian update of $\mathbf{K}$ we instead optimize the hyper-parameters by maximizing the marginal likelihood in a gradient descent algorithm. The marginal likelihood can be obtained from the normalizer $\tilde{Z}^c$ in Equation 14 as:

$$\tilde{Z}^c = \int [p(\mathbf{f}_c|\boldsymbol{\theta}_c)] \prod_{\mathbf{u}\in U} \prod_{\{\mathbf{x}_i \succ \mathbf{x}_j\}\in\mathcal{D}^{\mathbf{u}}} q(f_i^{\mathbf{u}}, f_j^{\mathbf{u}}|\mathbf{c_u}) d\mathbf{f} \quad (16)$$

where both $p(\mathbf{f}_c|\boldsymbol{\theta}_c)$ and $q(f_i^{\mathbf{u}}, f_j^{\mathbf{u}}|\mathbf{c_u})$ are Gaussian distributions and their product produces an unnormalized Gaussian distribution. Therefore, the log likelihood is:

$$\log(\tilde{Z}^c) = -\frac{1}{2}\tilde{\boldsymbol{\mu}^c}^\top (\mathbf{K}^c + \tilde{\boldsymbol{\Sigma}^c})^{-1} \tilde{\boldsymbol{\mu}^c} - \tfrac{1}{2}\log\det(\mathbf{K}^c + \tilde{\boldsymbol{\Sigma}^c}) - \tfrac{n}{2}\log 2\pi \quad (17)$$

The derivative of the marginal likelihood with respect to the kernel hyper-parameters can be used in a gradient descent algorithm to optimize the kernel.

3.4 Prediction

Given a pair of items $\mathbf{x}_1^*, \mathbf{x}_2^*$ for a particular user, we will be able to determine the predictive distribution over the latent utility functions as:

$$p(f_1^*, f_2^*|\mathcal{D},\alpha,\mathbf{K},\mathbf{u}) = \int_{-\infty}^{\infty} p(f_1^*, f_2^*|\mathbf{f}^{\mathbf{c_u}},\alpha,\mathbf{K}^c) \times p(\mathbf{f}^{\mathbf{c_u}}|\mathcal{D},\alpha,\mathbf{K}^c) d\mathbf{f}^{\mathbf{c_u}} = \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{C}^*) \quad (18)$$

$$\text{with} \quad \boldsymbol{\mu}^* = \mathbf{K}^*(\mathbf{K}^c + \tilde{\boldsymbol{\Sigma}^c})^{-1}\boldsymbol{\mu}^c \quad (19)$$

$$\boldsymbol{C}^* = \boldsymbol{\Sigma}^* - \mathbf{K}^{*\top}(\mathbf{K}^c + \tilde{\boldsymbol{\Sigma}^c})^{-1}\mathbf{K}^*, \quad (20)$$

where $\boldsymbol{\Sigma}^*$ is the 2×2 kernel matrix built from the item pair $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$; $\mathbf{K}^*$ represents the kernel matrix of test items with all the items in the training set; $\mathbf{K}_c^*$ is the kernel matrix of the queried user with other users in the same community; and $\mathbf{K}_x^*$ is the $2\times m$ kernel matrix of the queried pair of items with other items. Subsequently, their preference for a user is determined by integrating out the latent utility functions we have $p(\mathbf{x}_1^* \succ \mathbf{x}_2^*|\mathcal{D},\alpha,\mathbf{K})$ equals:

$$\sum_{\mathbf{c}} p(\mathbf{c}|\lambda) \int\int p(\mathbf{x}_1^* \succ \mathbf{x}_2^*|f_1^*, f_2^*, \mathbf{c}, \alpha, \mathbf{K}) p(f_1^*, f_2^*|\mathcal{D},\mathbf{c},\alpha,\mathbf{K}) df_1^* df_2^* = \sum_{\mathbf{c}} p(\mathbf{c}|\lambda)\Phi\left(\frac{\boldsymbol{\mu}_1^* - \boldsymbol{\mu}_2^*}{\alpha^2 + \boldsymbol{C}_{1,1}^* + \boldsymbol{C}_{2,2}^* - 2\boldsymbol{C}_{1,2}^*}\right). \quad (21)$$

We see that the mean and covariance of the predictive distribution require the inversion of a much smaller matrix leading to $O(n^3)$ time complexity compared to the case where the community of the users is not considered in Section 2 which has $O(n^3m^3)$ time complexity.

Algorithm 1 Blocked Gibbs Sampling Routine

input: $X, U, \mathcal{D}, \lambda, \alpha$
Initialize $\mathbf{c}$ to arbitrary assignments for each user
while not converged **do**
  // Infer community utilities and hyperparameters
  for $\mathbf{c} \in C$ **do**
    1. Obtain $\mathbf{K}^c$ from Eq. 17.
    2. Perform EP to infer $p(\mathbf{f}|\mathbf{c},\mathcal{D},\alpha,\mathbf{K})$ (Sec 3.2).
  end for
  // Sample community membership assignments
  for $\mathbf{u} \in U$ **do**
    3. Sample $\mathbf{c_u}$ from Eq. 10.
  end for
end while

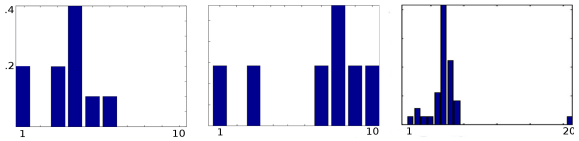
3.5 Final Algorithm

Having found the communities and users' utilities, the algorithm for community-based preference learning is presented in Algorithm 1. After initializing with class assignments for each user, hyperparameters for each community GP are optimized followed by inference of $\mathbf{f}$ and then $\mathbf{c}$, which repeats until convergence.

Table 1: Performance results for Synthetic, Sushi and AMT Car datasets.

Algorithm	Dataset	Accuracy %	Time(s)
[Bonilla *et al.*, 2010]	Synthetic	95.17 ± 3.33	0.05
	Sushi	62.13 ± 5.69	2.10
Full-GP, Section 2	Car	64.00 ± 8.94	0.89
DP Mixture of GPs	Synthetic	100 ± 0	0.04
	Sushi	62.04 ± 5.41	0.09
Section 3.1	Car	64.17 ± 6.97	0.10

Figure 3: Distribution of communities in the datasets . The x-axis is the number of communities; the y-axis is the posterior probability at the last sampling iteration. These values are obtained at iteration 15 of Synthetic and Sushi and iteration 36 of AMT Car.



(a) Synthetic (b) Sushi (c) AMT Car

4 Empirical Evaluation

In this section we empirically evaluate the performance of our algorithm to determine (1) how well the DP mixture of GPs is capable of reducing prediction time vs. Full-GP defined in Section 2, (2) how well the DP mixture of GPs works in accurately learning the preferences of each user compared to Full-GP, (3) whether each community learned has a distinct set of preferences, and (4) how effectively the true number of communities in data is recovered. We perform our evaluation on three datasets: one synthetic and two real-world datasets. The synthetic experiment assesses the effectiveness of our approach in a controlled setting and the real-world datasets include the data obtained from preferences over cars that we have created using Amazon Mechanical Turk as well as a publicly available sushi preference dataset.

It is assumed that we are given a set of users and items with their corresponding features. For each user and item we also augment their features with their ID index, that is, a vector that is only one for that particular user or item (this is a common practice in collaborative filtering). The pairwise item preferences of each user are split into two sets for performing the inference (60%), and testing (40%) where the algorithm performance is evaluated. We use the squared exponential covariance with automatic relevance determination (ARD) for both users and items. We manually tuned λ and α (for Full-GP as well).

Synthetic Dataset: We generated a hypothetical set of 60 users and 10 items and assigned each user to one of four communities. We randomly assigned each item to be liked by one of the communities (high utility) and disliked by the other three communities (low utility). From these communities and their associated utility functions, we generate the preferences of each user (without noise to make a pure synthetic test data set). Our approach is able to converge to exactly four communities with the correct memberships (as shown in Figure 3), demonstrating that our algorithm is effective in recovering latent community structure present in data. As observed in Table 1, the accuracy and the time consumed by the proposed approach is also improved compared to Full-GP.

Sushi Dataset: Here we describe the results on the *sushi* dataset [Kamishima, 2003]. It is a dataset of preferences of people about 10 types of sushis which leads to 45 preferences (pairs of sushis) per user. Each user and item (sushi) is specified by a set of features and where we have categorical features, they are converted to binary ones. Moreover, similar to the collaborative filtering setting, we included the IDs of the users and items as well. We discovered 8 communities in this dataset (as shown in Figure 3 while performing as accurately as Full-GP with two orders of magnitude less running time as shown in Table 1.

Car Preference Dataset using Amazon Mechanical Turk[3]: Amazon Mechanical Turk[4] (AMT) provides an excellent crowdsourcing opportunity for performing online experiments and user studies. Since the number of publicly available preference learning datasets are limited, we set up an experiment in AMT to collect real pair-wise preferences over users. In this experiment users are presented with a choice to prefer a car over another based on their attributes. The car attributes used are:

- Body type: Sedan, SUV
- Engine capacity: 2.5L, 3.5L, 4.5L, etc.
- Transmission: Manual, Automatic
- Fuel consumed: Hybrid, Non-Hybrid

The set is then split into two sets with 60% of all the preferences kept for inference and the rest for testing. The dataset is collected so that 10 unique cars (items) are considered and users are required to answer all 45 possible pair-wise preferences. We targeted US users mainly to have a localized and consequently more meaningful preference dataset. For each user, a set of attributes in terms of general questions (age range, education level, residential region and gender) is collected. Every categorical attribute is converted to binary ones. We collected these preferences from 60 unique users.

As observed in Table 1, the proposed approach is as accurate and faster than Full-GP. Through learning the communities, we can also analyze the most frequently chosen attributes selected by a community as shown in Table 2, where inference converged to 7 communities.

5 Related Work

Probabilistic models for utility functions in preference learning and elicitation have previously been proposed in the machine learning community [Chajewska and Koller, 2000; Guo and Sanner, 2010]. Extensions to non-parametric models have also been developed. In particular, [Chu and Ghahramani, 2005b] proposed a preference learning framework based on Gaussian processes and [Eric *et al.*, 2008] used this

[3] http://users.cecs.anu.edu.au/~u4940058/CarPreferences.html

[4] http://mturk.com

Figure 2: The distribution of preferred cars in each community for the AMT car dataset. Each x-axis position represents a different car and the y-axis the normalized frequency with which that car was preferred to another by a user in the community. Each community is distinct and differs in at least one car attribute.

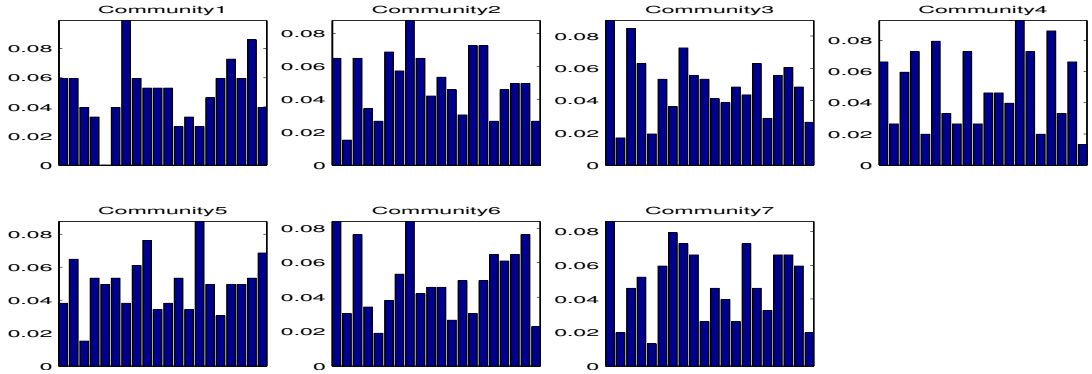


Table 2: Most likely attributes selected in each category by communities discovered in the AMT Car dataset. As observed in Figure 2, some communities are very different while others are similar. For example, community 3 and 7 are only different in the body type they prefer, but are both quite different from community 4.

	SUV, Sedan	**Automatic, Manual**	**Engine Capacity**	**Hybrid, Non-Hybrid**
1	SUV	Automatic	3.5L	Non-Hybrid
2	Sedan	Manual	2.5L	Non-Hybrid
3	Sedan	Automatic	2.5L	Non-Hybrid
4	SUV	Manual	4.5L	Non-Hybrid
5	Sedan	Manual	2.5L	Hybrid
6	SUV	Automatic	4.5L	Non-Hybrid
7	SUV	Automatic	2.5L	Non-Hybrid

model for active learning with discrete choice data. Multi-user GP-based preference models have been given by [Birlutiu *et al.*, 2010] and [Bonilla *et al.*, 2010]. However, none of these methods directly address the efficiency problem nor discovered any community structure.

Stochastic blocked models [Nowicki and Snijders, 2001; Airoldi *et al.*, 2008] are another class of related approaches that model the dependencies between users and infer the graphical structure under which users interact. Such models implicitly find the communities users belong to, even though not directly applied to preference learning in the setting we discussed here.

Furthermore, [Rasmussen and Ghahramani, 2002] proposed the infinite mixture of Gaussian processes for regression and later extended in [Meeds and Osindero, 2005] that is similar to the mixture of Gaussian processes detailed here. Our work however further extends the mixture of GPs to social preference learning.

6 Conclusion

We exploited the observation that user populations often decompose into communities of shared preferences and modeled user preferences as an infinite Dirichlet Process (DP) mixture of communities. The resulting inference algorithm scales linearly in the number of users unlike previous Gaussian Process preference learning approaches that scaled cubicly in the number of users. We evaluated our approach on a variety of preference data sources including Amazon Mechanical Turk showing that our method is more scalable and as accurate as previous work with only a small number of inferred communities, validating our community-based modeling approach.

For future work, we note that the differing levels of similarity between communities observed in Figure 2 and Table 2 suggest that a hierarchical non-parametric Bayesian approach may be useful for directly modeling the differing relationships between communities.

Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

[Airoldi *et al.*, 2008] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9, 2008.

[Birlutiu *et al.*, 2010] Adriana Birlutiu, Perry Groot, and Tom Heskes. Multi-task preference learning with an application to hearing aid personalization. In *Neurocomputing*, volume 73, 2010.

[Bonilla *et al.*, 2010] Edwin V. Bonilla, Shengbo Guo, and Scott Sanner. Gaussian process preference elicitation. In *NIPS*, pages 153–160, 2010.

[Chajewska and Koller, 2000] Urszula Chajewska and Daphne Koller. Utilities as random variables: Density estimation and structure discovery. In *UAI*, pages 63–71. Morgan Kaufmann Publishers Inc., 2000.

[Chu and Ghahramani, 2005a] Wei Chu and Zoubin Ghahramani. Extensions of gaussian processes for ranking: semi-supervised and active learning. In *Workshop on Learning to Rank at NIPS*, 2005.

[Chu and Ghahramani, 2005b] Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *ICML*, pages 137–144, New York, NY, USA, 2005. ACM.

[Eric *et al.*, 2008] Brochu Eric, Nando De Freitas, and Abhijeet Ghosh. Active preference learning with discrete choice data. In *NIPS*, pages 409–416. 2008.

[Fürnkranz and Hüllermeier, 2010] Johannes Fürnkranz and Eyke Hüllermeier. *Preference Learning*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[Guo and Sanner, 2010] Shengbo Guo and Scott Sanner. Real-time multiattribute Bayesian preference elicitation with pairwise comparison queries. In *AISTATS*, 2010.

[Kamishima, 2003] Toshihiro Kamishima. Nantonac collaborative filtering: recommendation based on order responses. In *ACM SIGKDD*, pages 583–588, 2003.

[Meeds and Osindero, 2005] Edward Meeds and Simon Osindero. An alternative innite mixture of gaussian process experts. In *Advances in Neural Information Processing Systems*, 2005.

[Minka, 2001] Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In *UAI*, volume 17, pages 362–369, 2001.

[Müller and Quintana, 2004] Peter Müller and Fernando A. Quintana. Nonparametric bayesian data analysis. *Statistical Science*, 19:95–110, 2004.

[Neal, 1998] R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. Technical report, Dept. of Statistics, University of Toronto, 1998.

[Neumann and Morgenstern, 1944] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[Nowicki and Snijders, 2001] Krzysztof Nowicki and Tom A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96, 2001.

[Platt *et al.*, 2002] John C Platt, Christopher J C Burges, Steven Swenson, Christopher Weare, and Alice Zheng. Learning a Gaussian Process Prior for Automatically Generating Music Playlists. In *NIPS*, 2002.

[Postlewaite, 2011] Andrew Postlewaite. Social norms and social assets. *Annual Review of Economics*, 3:239–259, 2011.

[Rasmussen and Ghahramani, 2002] Carl Edward Rasmussen and Zoubin Ghahramani. Innite mixtures of gaussian process experts. In *Advances in Neural Information Processing Systems*, 2002.

[Rasmussen and Williams, 2006] Carl Edward Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. 2006.

[Rasmussen, 2000] Carl Edward Rasmussen. The infinite gaussian mixture model. In *Advances in Neural Information Processing Systems*, volume 12, page 554560, 2000.

[Thurstone, 1959] L.L. Thurstone. *The Measurement of Values*. The University of Chicago Press, Chicago, 1959.

[Xu *et al.*, 2010] Zhao Xu, Kristian Kersting, and Thorsten Joachims. Fast active exploration for link-based preference learning using Gaussian processes. In *ECML PKDD*, 2010.