On Sparse Multi-Task Gaussian Process Priors for Music Preference Learning

Jens Brehm Nielsen, Bjørn Sand Jensen and Jan Larsen
Technical University of Denmark
Department of Informatics and Mathematical Modelling
Amussens Allé, Building 305, 2800 Lyngby, Denmark,
{jenb, bjje, jl}@imm.dtu.dk

Abstract

In this paper we study pairwise preference learning in a music setting with multi-task Gaussian processes and examine the effect of sparsity in the input space as well as in the actual judgments. To introduce sparsity in the inputs, we extend a classic pairwise likelihood model to support sparse, multi-task Gaussian process priors based on the pseudo-input formulation. Sparsity in the actual pairwise judgments is potentially obtained by a sequential experimental design approach, and we discuss the combination of the sequential approach with the pseudo-input preference model. A preliminary simulation shows the performance on a real-world music preference dataset which motivates and demonstrates the potential of the sparse Gaussian process formulation for pairwise likelihoods.

1 Introduction

Preference learning is aimed at eliciting, modeling and eventually predicting human preference for a given input or normally sets of inputs. In this paper we focus on a relatively robust query type for human preference elicitation suitable for e.g. music applications, namely pairwise comparisons modeled by the likelihood function considered in [11, 1]. This basic likelihood model was first put into the flexible framework of Gaussian processes (GP) priors by Chu *et. al.* [5]. Furthermore, a general multi-task extension to the particular preference setup was proposed in Bonilla *et. al.* [3] based on the multi-task formalism originally developed by Bonilla *et. el.* [2] which supports the inclusion of collaborative or transfer learning between users. GP based models are in turn desirable models for preference learning, however, they all struggle with an inconvenient $\mathcal{O}\left(n^3\right)$ scaling in terms of the number of input instances, n, which makes their use limited for large-scale problems. A number of suggestions have been proposed to resolve this issue for the standard GP regression case.

Our objective is to extend the well-known pairwise likelihood model to allow for explicit sparsity in the input space. This is achieved by extending the pairwise likelihood model in terms of a set of pseudo-inputs (of size $l << n$) which are essentially used to integrate out the function values of the original inputs using the ideas proposed in Snelson *et. al.* [10] for the standard regression case. In effect the multi-task GP prior is now placed over the function values of the pseudo points. Posterior inference relies on a Laplace approximation, and the pseudo-inputs can be found by evidence optimization or be fixed and determined by, e.g., k-means initialization. Secondly, we outline to combine the model with the ideas of Bonilla *et. al.* [3] and include sequential experimental design to ensure that sparsity also persists in terms of the number of actual pairwise comparisons, m, besides

Revision: 2011/12/12. Acknowledgment: This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. This publication only reflects the authors' views.

the sparsity in the associated number of input instances, n. Finally, we evaluate the pseudo-input model on a real-world music preference dataset, examine the multi-task transfer and learning rates and discuss limitations and further improvements of this initial evaluation.

The paper is organized as follows: In Section 2 we review the basic model, provide the pseudo-input extension and discuss option of sequential experimental design. In Section 3 we consider a toy example and present the preliminary results on the music dataset. In Section 4 we discuss the overall findings and outline a number of future research steps.

2 Model & Extensions

We describe the general setup and model in terms of

- a set $\mathcal{A}$ of n_a input instances, e.g. audio tracks, where each input instance i is described by one feature vector $x^{(a)} \in \mathbb{R}^{d_a}$, i.e., $\mathcal{A} = \{x_i^{(a)} | i = 1, ..., n_a\}$.
- a set $\mathcal{U}$ of n_u users, where each user j is described by a feature vector $x^{(u)} \in \mathbb{R}^{d_u}$, i.e., $\mathcal{U} = \{x_j^{(u)} | j = 1, ..., n_u\}$.

The task for a specific user j is to perform a forced choice between two input instances, $x_u^{(a)} \in \mathcal{A}$ and $x_v^{(a)} \in \mathcal{A}$, where $u \neq v$, resulting in a response $y \in \{-1, +1\}$, where $y = +1$ corresponds to a preference for the u'th input, and -1 corresponds to a preference for the v'th input. We acquire m such pairwise comparisons between any two input instances in $\mathcal{A}$ and with any user in $\mathcal{U}$, which results in the set of observations $\mathcal{Y} = \left\{ (y_k; x_{u_k}^{(a)}, x_{v_k}^{(a)}, j_k) | k = 1, ..., m \right\}$.

Given the two latent function values $\mathbf{f}_k = \left[f_{j_k}\left(x_{u_k}^{(a)}\right), f_{j_k}\left(x_{v_k}^{(a)}\right) \right]$ (associated with a particular user) at the two inputs, we model the observations by a likelihood function $p\left(y_k | \mathbf{f}_k, \boldsymbol{\theta}_{\mathcal{L}}\right)$. The likelihood function is defined by additional parameters $\boldsymbol{\theta}_{\mathcal{L}}$. The function f_{j_k} is an absolute, latent function preserving the preference information over the input space for a particular user j. The function parametrization admits that we directly place a Gaussian process prior on f_{j_k} allowing for a flexible predictive model for the pairwise responses of a particular user. A multi-task setting can be constructed by exploiting an observed feature vector per user. Consequently, we can think of a global latent multi-task preference function $f(x^{(a)}, x^{(u)})$ instead of several individual single-task preference functions $f_j(x^{(a)})$. The multi-task kernel formulation of a GP [2] can hence be formulated as:

$$f_j(x_i^{(a)}) = f(x_i^{(a)}, x_j^{(u)}) \sim \mathcal{GP}\left(0, k(x_i^{(a)}, \cdot) k(x_j^{(u)}, \cdot)\right) = \mathcal{GP}\left(0, k(x_{i,j}, \cdot)\right), \quad (1)$$

where we have joined the audio and user feature into one input instance, $x = \{x^{(a)}, x^{(u)}\}$, and thereby defined the unique set of inputs as $\mathcal{X} = \{\{x_i^{(a)}, x_j^{(u)}\} | i = 1...n_a, j = 1...n_u\}$. Thus, the GP framework constitutes a non-linear, yet very flexible alternative to the more traditional models such as (Generalized) Linear Models. Also, this formulation addresses the multi-task kernel only in the definition of the covariance function - everywhere else, we only think of one input x containing both user and task features simultaneously with a corresponding function value $f(x)$. This definition will be convenient later.

Given a standard Bayesian framework and assuming the likelihood factorizes we now obtain the posterior over the function, i.e.,

$$p\left(\mathbf{f} | \mathcal{X}, \mathcal{Y}, \boldsymbol{\theta}\right) \propto p\left(\mathbf{f} | \mathcal{X}, \boldsymbol{\theta}_{GP}\right) \prod_{k=1}^{m} p\left(y_k | \mathbf{f}_k, \boldsymbol{\theta}_{\mathcal{L}}\right)$$

with $\mathbf{f} = [f(x_1^{(a)}, x_1^{(u)}), f(x_1^{(a)}, x_2^{(u)}), ..., f(x_1^{(a)}, x_{n_u}^{(u)}), ..., ..., f(x_{n_a}^{(a)}, x_{n_u}^{(u)})]^\top$, $\boldsymbol{\theta}_{GP}$ contains the GP hyper-parameters and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\mathcal{L}}, \boldsymbol{\theta}_{GP}\}$. The main computational issue in the single task GP is to calculate/approximate the posterior which poses a $\mathcal{O}\left(n_a^3\right)$ scaling challenge due to the inversion of the kernel matrix. Coupling n_u single task GPs in the covariance structure will further scale this to $\mathcal{O}([n_a n_u]^3)$. In practical preference applications, this is of course a problem and to remedy this we first consider the (standard) pairwise likelihood in Section 2.1.1 and then a sparse extension in Section 2.1.2 allowing for a sparse GP prior with less than $(n_a)(n_u)$ inputs. Finally, we suggest the sequential extension in Section 2.3.

2.1 Likelihood

2.1.1 Pairwise Likelihood (Standard)

Pairwise comparisons are typically modeled by the classic Probit choice model [11, 1], constituting the basis for the so-called pairwise likelihood function given by

$$p\left(y_k|\mathbf{f}_k, \boldsymbol{\theta}_{\mathcal{L}}\right) = \Phi\left(y_k \frac{f_{j_k}\left(x_{u_k}^{(a)}\right) - f_{j_k}\left(x_{v_k}^{(a)}\right)}{\sqrt{2}\sigma}\right), \tag{2}$$

where $\Phi(x)$ defines a cumulative Gaussian (with zero mean and unity variance), and $\boldsymbol{\theta}_{\mathcal{L}} = \{\sigma\}$. The use of a GP prior in connection with this likelihood was first proposed in [5].

2.1.2 Pairwise Likelihood with Pseudo-Inputs

We extend the standard preference model in Eq. 2 to obtain sparsity in the input space in terms of the effective number of points in the prior and posterior. We generally follow the ideas in [10], i.e., given a set of pseudo-inputs $\bar{\mathbf{X}}$, their functional values $\bar{\mathbf{f}}$ must come from a Gaussian process like the real latent data $\mathbf{f}$. Therefore, we can directly place a Gaussian process prior over $\bar{\mathbf{f}}$

$$p\left(\bar{\mathbf{f}}|\bar{\mathbf{X}}\right) = \mathcal{N}\left(\bar{\mathbf{f}}|\mathbf{0}, \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}\right) \tag{3}$$

where the matrix $\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}$ is the covariance matrix of the l pseudo-inputs collected in the matrix $\bar{\mathbf{X}} = [\bar{x}_1, ..., \bar{x}_l]$. Recall, that we have formulated our multi-task problem only in terms of the covariance function. Therefore, each pseudo-input $\bar{x}$ defines both a task vector $\bar{x}^{(a)} \in \mathbb{R}^{d_a}$ and a user vector $\bar{x}^{(u)} \in \mathbb{R}^{d_u}$, which are stacked to form each of the pseudo-input vectors used in $\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}$. Then the covariance matrix, $\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}$, is again found by the use of the same multi-task covariance function $k(\cdot,\cdot)$ from Eq. 1, i.e., $[\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}]_{i,j} = k(\bar{x}_i, \bar{x}_j)$[1]. The overall idea of the pseudo-input formalism is now to refine the likelihood such that the real $\mathbf{f}$ values that enter directly in the original, non-sparse likelihood function (through f_k), exist only in the form of predictions from the the pseudo-inputs $\bar{\mathbf{f}}(\bar{\mathbf{X}})$. Given the listed assumptions, we formally have that $\mathbf{f}$ and $\bar{\mathbf{f}}$ are jointly Gaussian, i.e.,

$$\begin{bmatrix} \mathbf{f}_k \\ \bar{\mathbf{f}} \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\mathbf{x}_k\mathbf{x}_k} & {\mathbf{K}_{\bar{\mathbf{X}}\mathbf{x}_k}}^\top \\ \mathbf{K}_{\bar{\mathbf{X}}\mathbf{x}_k} & \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}} \end{bmatrix}\right), \tag{4}$$

where we define the following matrices and vectors

$$\mathbf{K}_{\mathbf{x}_k\mathbf{x}_k} = \begin{bmatrix} k(x_{u_k,j_k}, x_{u_k,j_k}) & k(x_{u_k,j_k}, x_{v_k,j_k}) \\ k(x_{v_k,j_k}, x_{u_k,j_k}) & k(x_{v_k,j_k}, x_{v_k,j_k}) \end{bmatrix}, \mathbf{K}_{\bar{\mathbf{X}}\mathbf{x}_k} = [\mathbf{k}_{u_k}, \mathbf{k}_{v_k}]$$

with $[\mathbf{k}_{u_k}]_i = k(\bar{x}_i, x_{u_k,j_k})$ and $[\mathbf{k}_{v_k}]_i = k(\bar{x}_i, x_{v_k,j_k})$. Note, that we have now formally stacked the task and user feature into one input, such that x_{u_k,j_k} and x_{v_k,j_k} contain the task feature for option u and v, respectively, together with the user feature.

From Eq. 4 it is trivial to find the conditional distribution of $\mathbf{f}_k$ given $\bar{\mathbf{f}}$, hence the likelihood can be derived in terms of $\bar{\mathbf{f}}$, i.e. $p\left(y_k|\bar{\mathbf{f}}, \bar{\mathbf{X}}\right)$, by integrating over $\mathbf{f}_k$

$$\begin{aligned} p\left(y_k|x_{u_k,j_k}, x_{v_k,j_k}, \bar{\mathbf{X}}, \bar{\mathbf{f}}, \boldsymbol{\theta}\right) &= \int_{\mathbf{f}_k} p\left(y_k|\mathbf{f}_k, \boldsymbol{\theta}_{\mathcal{L}}\right) p\left(\mathbf{f}_k|\bar{\mathbf{f}}, \bar{\mathbf{X}}\right) d\mathbf{f}_k & (5) \\ &= \int_{\mathbf{f}_k} \Phi\left(y_k \frac{f_{j_k}\left(x_{u_k}^{(a)}\right) - f_{j_k}\left(x_{v_k}^{(a)}\right)}{\sqrt{2}\sigma}\right) \mathcal{N}\left(\mathbf{f}_k|\mu_k, \boldsymbol{\Sigma}_k\right) d\mathbf{f}_k & (6) \\ &= \Phi\left(y_k \frac{\mu_{u_k} - \mu_{v_k}}{\sigma_k^*}\right) & (7) \end{aligned}$$

[1] Notice, that now we have introduced one more use of i and j, besides to index input and users, namely to index element of a matrix. In the following we will keep using both, but when i and j are used to index matrices and vectors, it will be clear from the notation

where $\mu_k = [\mu_{u_k}, \mu_{v_k}]^\top$, $\mu_{u_k} = \mathbf{k}_{u_k}^T \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \bar{\mathbf{f}}$, $\mu_{v_k} = \mathbf{k}_{v_k}^T \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \bar{\mathbf{f}}$ and

$$\mathbf{\Sigma}_k = \left[\begin{array}{cc} \sigma_{u_k u_k} & \sigma_{u_k v_k} \\ \sigma_{v_k u_k} & \sigma_{v_k v_k} \end{array} \right] = \mathbf{K}_{\mathbf{x}_k \mathbf{x}_k} - \mathbf{K}_{\bar{\mathbf{X}}\mathbf{x}_k}^\top \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \mathbf{K}_{\bar{\mathbf{X}}\mathbf{x}_k}$$

Furthermore, $(\sigma_k^*)^2 = 2\sigma^2 + \sigma_{u_k u_k} + \sigma_{v_k v_k} - \sigma_{u_k v_k} - \sigma_{v_k u_k}$, which all together results in the pseudo-input likelihood

$$p\left(y_k | x_{u_k, j_k}, x_{v_k, j_k}, \bar{\mathbf{X}}, \bar{\mathbf{f}}, \boldsymbol{\theta}\right) = \Phi\left(z_k\right), \quad \text{where } z_k = y_k \left(\mathbf{k}_u^T - \mathbf{k}_v^T\right) \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \bar{\mathbf{f}} / \sigma_k^* \qquad (8)$$

2.2 Posterior - Inference & Predictions

Both likelihoods described in Section 2.1 lead to untractable posteriors and call for approximation techniques or sampling methods. Our goal in this initial study is to examine the model and its properties - not to provide the optimal approximation - and we will only explore inference based on the Laplace approximation.

2.2.1 Posterior Approximation

Inference using the Laplace approximation has also been applied in [4] for the standard model. The general solution to the approximation problem can be found by considering the unnormalized log-posterior and the resulting cost function (to be maximized) is given by

$$\psi\left(\bar{\mathbf{f}} | \mathcal{Y}, \mathcal{X}, \bar{\mathbf{X}}, \boldsymbol{\theta}\right) = \log p\left(\mathcal{Y} | \bar{\mathbf{f}}, \mathcal{X}, \bar{\mathbf{X}}, \boldsymbol{\theta}\right) - \frac{1}{2} \bar{\mathbf{f}}^T {\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}}^{-1} \bar{\mathbf{f}} - \frac{1}{2} \log |\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}| - \frac{N}{2} \log 2\pi. \qquad (9)$$

where $[\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}]_{i,j} = k(x_i, x_j)_{\boldsymbol{\theta}_{\mathcal{GP}}}$. We use a damped Newton method with optional linesearch to maximize Eq. (9). The basic damped Newton step (with adaptive damping factor λ) can in this case be calculated without inversion of the Hessian (see [7])

$$\bar{\mathbf{f}}^{new} = \left(\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} + \mathbf{W} - \lambda \mathbf{I}\right)^{-1} \left[(\mathbf{W} - \lambda \mathbf{I}) - \bar{\mathbf{f}} + \nabla \log p(\mathcal{Y} | \bar{\mathbf{f}}, \mathcal{X}, \bar{\mathbf{X}}, \boldsymbol{\theta})\right], \qquad (10)$$

Using the notation $\nabla\nabla_{i,j} = \frac{\partial^2}{\partial f(x_i) \partial f(x_j)}$ we apply the definition $\mathbf{W}_{i,j} = -\sum_k \nabla\nabla_{i,j} \log p(y_k | x_{u_k, j_k}, x_{v_k, j_k}, \bar{\mathbf{X}}, \bar{\mathbf{f}}, \boldsymbol{\theta})$. When converged, the resulting approximation can be shown to be $p\left(\bar{\mathbf{f}} | \mathcal{Y}, \mathcal{X}, \bar{\mathbf{X}}, \boldsymbol{\theta}\right) \approx \mathcal{N}\left(\bar{\mathbf{f}} | \hat{\mathbf{f}}, \left(\mathbf{W} + {\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}}^{-1}\right)^{-1}\right)$. The damped Newton step requires the Jacobian and Hessian of the new pseudo-input log-likelihood, which requires the following derivatives

$$\frac{\partial}{\partial \bar{\mathbf{f}}} p\left(y_k | ...\right) = y_k \frac{\mathcal{N}\left(z_k\right)}{\sigma_k \Phi\left(z_k\right)} \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \left(\mathbf{k}_u - \mathbf{k}_v\right) \qquad (11)$$

$$\frac{\partial^2}{\partial \bar{\mathbf{f}} \bar{\mathbf{f}}^\top} p\left(y_k | ...\right) = -y_k^2 \frac{\mathcal{N}\left(z_k\right)}{\sigma_k^2 \Phi\left(z_k\right)} \left[z_k + \frac{\mathcal{N}\left(z_k\right)}{\Phi\left(z_k\right)}\right] \cdot \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \left(\bar{\mathbf{k}}_u - \bar{\mathbf{k}}_v\right) \left(\bar{\mathbf{k}}_u - \bar{\mathbf{k}}_v\right)^\top \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1}. \qquad (12)$$

2.2.2 Evidence / Hyperparameter Optimization

Hyperparameters are optimized based on a regularized variant of traditional evidence or maximum likelihood II (ML-II) optimization allowing for simple regularizing priors on the hyperparameters. The reguralization is primarily included for robustness and is in spirit similar to regularized EM algorithms. The details are available in [7], but for completeness we shortly review the process of evidence optimization and comments on the case of the pseudo-input model.

So far we have simply considered the hyper-parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\mathcal{L}}, \boldsymbol{\theta}_{\mathcal{GP}}\}$ and pseudo-inputs $\bar{\mathbf{X}}$ as fixed paraments. However, they have a crucial influence on the model and we will resort to point estimates by iterating between the Laplace approximation with fixed hyper-parameters, i.e., finding $p\left(\bar{\mathbf{f}} | \mathcal{Y}, \mathcal{X}, \bar{\mathbf{X}}, \boldsymbol{\theta}\right)$, followed by an evidence maximization step in which $(\boldsymbol{\theta}, \bar{\mathbf{X}}) = \arg\max_{(\boldsymbol{\theta}, \bar{\mathbf{X}})} p\left(\mathcal{Y} | \boldsymbol{\theta}, \bar{\mathbf{X}}\right)$. The log-evidence, $\log p(\mathcal{Y} | \boldsymbol{\theta}, \bar{\mathbf{X}})$, has to be approximated in our case, which in terms of the existing Laplace approximation yields $\log p\left(\mathcal{Y} | \boldsymbol{\theta}, \bar{\mathbf{X}}\right) \approx \log p(\mathcal{Y} | \hat{\mathbf{f}}, \bar{\mathbf{X}}, \mathcal{X}, \boldsymbol{\theta}) - \frac{1}{2} \hat{\mathbf{f}}^T {\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}}^{-1} \hat{\mathbf{f}} - \frac{1}{2} \log |\mathbf{I} + \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}} \mathbf{W}|$. We perform the optimization step using a standard BFGS method.

The pseudo-input model poses a number of difficulties since $\bar{\mathbf{X}}$ are also to be considered hyperparameter, and the input locations can thus be optimized as outlined above. Typically, this will, as noted in [10][9], lead to a large number of local maxima providing potentially suboptimal solutions, at least when using the proposed gradient method. It is not our aim to resolve nor document this issue, and we will take a pragmatic view and simply accept evidence optimization methods as is. The pseudo-input approach can in some sense be seen as a supervised clustering of the input space, but the optimization of $\bar{\mathbf{X}}$ is heavily influences by the initializations. We recommend starting out with a fixed set of pseudo-inputs initialized by a standard unsupervised clustering, such as k-means like [9], and then attempt an evidence optimization of $\bar{\mathbf{X}}$. We will provide a demonstration of this approach.

2.2.3 Predictions

Predictions of the pairwise judgments for a new experiment $\eta = \{x_u^{(a)*}, x_v^{(a)*}, x^{(u)*}\}$ with $x_u^{(a)*} \in \mathbb{R}^{d_a}$, $x_v^{(a)*} \in \mathbb{R}^{d_a}$ and $x^{(u)*} \in \mathbb{R}^{d_u}$ is given by $p(y|\eta, \mathcal{Y}, \mathcal{X})$. Given the approximated posterior of interest, $p\left(\bar{\mathbf{f}}|\mathcal{Y}, \mathcal{X}, \boldsymbol{\theta}\right)$, the prediction can be made in closed form (see e.g. [5] in the standard case and [7] for the pseudo-input case).

2.3 Sequential Experimental Design

Sequential experiential design - also known as active learning, selective or uncertainty sampling - includes datapoints/queries in a sequential manner by selecting only the most informative experiments/instances in terms of some gain. If the gain is relevant to the task, this effectively reduces the number of real input instances, n, and the number of pairwise comparisons, m, required to obtain a certain performance level compared to random selection of datapoints. Together with the pseudo-input model proposed in Section 2.1.2 this will ensure that we obtain a sparse and close to optimal model in terms of m,n and the effective number of pseudo-inputs l. We formulate the problem as a Bayesian sequential design problem (see e.g. [8]) in terms of a gain function, $G(\cdot)$, the expectation of this gain and the currently observed data $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$, i.e.,

$$\eta_y = \arg\max_{\eta} \sum_{y \in \mathbb{Y}} p(y|\eta, \mathcal{D})\, G\left(y, \eta, p\left(\bar{\mathbf{f}}_{\mathcal{D}\cup\eta}|y, \eta, \mathcal{D}\right), p\left(\bar{\mathbf{f}}_{\mathcal{D}}|y, \eta, \mathcal{D}\right)\right) \tag{13}$$

If the aim is to find the instance for which the user(s) has/have highest preference, the gain can e.g. be defined as expected improvement [3]. If the aim is a generalization of the preference model for all instances and users, entropy change (reduction) is the natural choice (but not guaranteed to be optimal). The multi-task (-user) and collaborative setting does support specialized gain functions depending, e.g., on user experience, consensus and knowledge, but it is not the aim to develop such concepts here. Since the main focus of the paper is the pseudo-input formulation of the pairwise likelihood, we leave the evaluation of the sequential extension to future research, but consider it a natural part of the general sparse framework outlined.

3 Simulations & Experimental Results

3.1 Example I: Pseudo-Input in 1D

This example is primarily intended to illustrate the basics of the pseudo-input principle in the pairwise case (in a single task setting). The example is based on a deterministic function which defines the pairwise relations, specifically a cosine in $[-2\pi; 2\pi]$ illustrated at the top-left in Figure 1. The seventeen input points are distributed equidistantly throughout the interval. The pairwise dataset $\mathcal{Y}$ is then generated as a complete set of pairwise relations for all input combinations. To model this dataset, we consider three case: A standard model (Section 2.1.1), a sparse model with fixed pseudo-inputs (Section 2.1.2) and a sparse model with optimized pseudo-inputs (Section 2.1.2). The five pseudo-inputs are initialized to $\bar{\mathbf{X}} = [-5, -2, 0, 2, 5]$, i.e. not in the training set. For direct comparison between the three models, we fix the other parameters, i.e., $\boldsymbol{\theta}_{\mathcal{L}}$ and $\boldsymbol{\theta}_{\mathcal{GP}}$, and use a Squared Exponential covariance function in all three cases with variance $\sigma_f = 1$ and lengthscale $\ell = 1$. The results are presented in Figure 1.

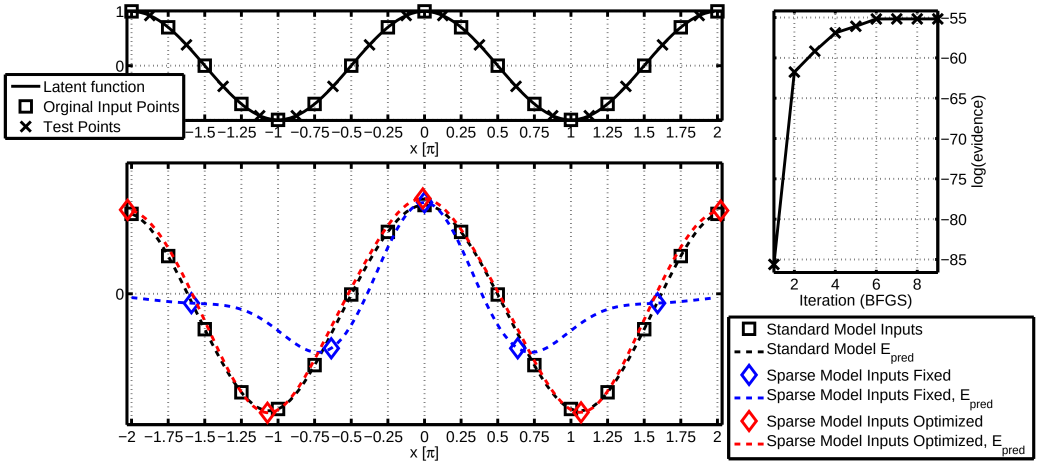

Figure 1: **Top (left)** panel shows a graph of the function from which the true underlying relations are defined. **Top (right)** panel shows the convergence of the evidence optimization. **Bottom (left)** panel shows the input points as markers used the three considered models and the predictive mean (E_{pred}) of the model as dotted graphs.

Given the equidistantly distributed input points and the full pairwise design, the standard model is almost capable of modeling the underlying function, however, the fixed model parameters limits the fit to the original model. Yet, the standard model is the best model we can expect in this case. The sparse model with fixed parameters generally has problems due to the suboptimal placement of the five pseudo-inputs. The optimized version converges to a (possible local) maximum as seen in the right panel of Figure 1 and solves the problem by moving the pseudo-inputs. This provides a better - and almost close to the standard - model despite only requiring 5 points as compared to 17.

3.2 Example II: Music Preference Data

In order to provide some initial insight into pairwise music preference learning, we consider a publicly available dataset [6]. Specifically, it consist of 10 test subjects, but only 9 with full user meta-data, 30 audio tracks with 10 audio tracks per genre [2]. The genres are Classical, Heavy Metal and Rock/Pop. The design of the experiment is based on a partial version of a complete pairwise design, hence only 155 out of the 420 combinations was evaluated by each of the 10 subjects. We extract standard audio features from the audio tracks, specifically the Mel-Frequency Cepstral Coefficients, MFCCs, (26 dimensions, including delta coefficients), which we project to a 6 dimensional space using PCA. Each track is subsequently modeled by a Gaussian with mean vector, $\mu^{(a)}$, and covariance matrix, $\Sigma^{(a)}$. The feature vector is then constructed as $x^{(a)} = \left[\mu^{(a)}, diag\left(\Sigma^{(a)}\right)\right]^{\top}$.

We define the correlation structure of tracks by considering a general purpose covariance function for audio that easily integrates user features and metadata types for the audio, such as audio features, tags, lyrics etc. It is defined as

$$k\left(x, x'\right) = \left(\sum\nolimits_{\ell=1}^{K_a} k_\ell\left(x^{(a)}, x^{(a)\prime}\right)\right) k_u\left(x^{(u)}, x^{(u)\prime}\right), \tag{14}$$

The first factor is the sum of all the K_a covariance functions defining the correlation structure of the audio inputs, $x^{(a)}$. The second factor, or multi-task part, is a general covariance function defining the covariance function for the user metadata part, $x^{(u)}$. We include only audio features, and e.g. not tags and lyrics, thus $K_a = 1$ and apply a standard squared exponential isotropic covariance function for the audio part. The user kernel is defined by a standard squared exponential kernel between the user features (age and the three prior genre preferences) available in vector form.

[2] The small-scale nature of the dataset is not optimal, yet it has not been possibly to obtain a larger dataset containing both features (or audio) and ratings, and especially the desire to consider pairwise comparisons of music tracks seems to be a novel consideration in music preference modeling.

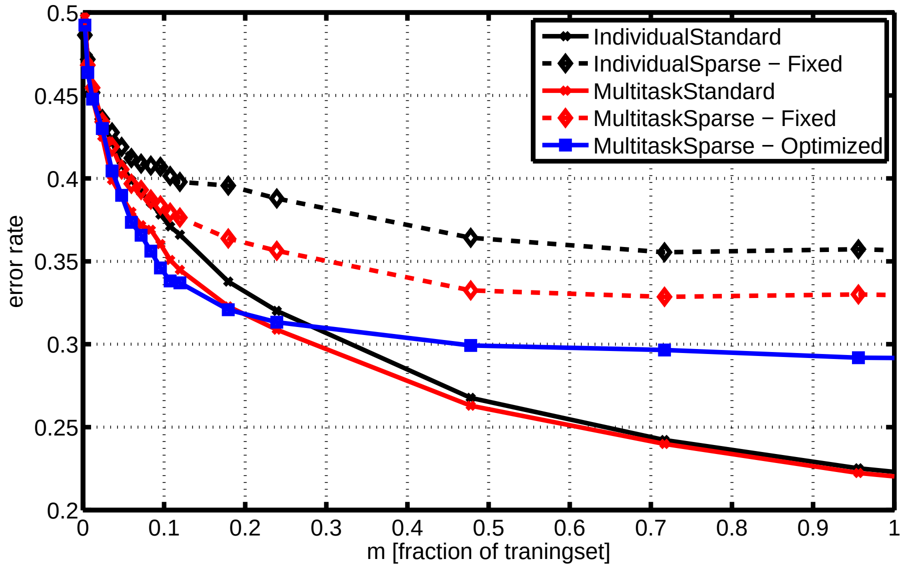

Figure 2: Learning curves averaged over 10 repetitions and 5-folds. $\bar{\mathbf{X}}$ is learned once on the full training set in each fold. A fraction of one corresponds to 80% of all comparisons. Sparse models are limited to 10% of the original number of inputs

3.2.1 Results

We concentrate on two of the most imminent questions which are the performance difference between (sparse) pseudo-input Model versus (dense) standard model, and the difference between individual modeling versus multi-task modelling.

We include a typical example of the learning curves by fixing all model parameters except the pseudo-inputs. Based on initial experiments, we fix the covariance parameters to: $\sigma^{(a)} = 3$, $\ell^{(a)} = 4$, $\sigma^{(u)} = 1.5$, and $\sigma^{(u)} = 1.5$. and the likelihood parameter, $\sigma_{\mathcal{L}} = 1$.

We consider the specific case of 27 pseudo-inputs (10% of total inputs points) in the $2 * 6 + 4 = 16$ dimensional input. This is based on a pure genre assumption, i.e., each of the nine users track preference can be described by single value pr. genre $(9 \cdot 3)$. Multi-task models effectively implies more points per genre if transfer can be exploited between users. The pseudo-inputs are initialized by k-means in the full input space (all audio tracks, all user features).

To provide some insight into the generalization properties of the relatively small dataset, we use a 5-fold cross-validation (CV) scheme. In each of the five CV we use one fold as test (279 observations), and 4 fold for training (837 observations). We evaluate the learning curves for a number of training set sizes, m, by selecting a random subsets of the full set. This is done 10 times for each m.

The preliminary results presented in Fig. 2 yields a few noticeable observations. Comparing the standard multi-task versus standard individual, we observe a minor benefit in the multi-task/collaborative model versus modeling users individually, thus some (useful) transfer is present. We furthermore observe that as more and more data is observed the individual model performs almost equally well as the multi-task. This is expected and individual models will in the limit outperform a multi-task model, but the exact point at which the individual models outperforms a multi-task model is difficult to estimate beforehand.

The second point to notice is the difference between the standard multi-task and the sparse multi-task. From a m-fraction of 0.0125 the sparse model contains less points than standard model (on average) and with approximately less than a 20% of the training set, the sparse model is fully capable to compete with the standard multi-task model. After 20% of the pairwise comparisons ($m = 0.2$) approximately 80% of all real inputs points has been observed. After this point the sparse model seems to lack the flexibility to fully describe the preferences. Whether this is due to a general characterize of the music preference problem or the fixed hyperparameters is so far unexplored, but we speculate that a full hyperparameter optimization will further minimize the gap between the sparse and the non-sparse model in this pairwise case.

The exact shape and absolute level of the learning curves are found to be sensitive to the exact prior parameters including $\bar{\mathbf{X}}$, and a robust scheme is to be derived to ensure robust and generalizable results. Despite its limitations the included case study suggests that the sparse pairwise model can provide some computation relief without scarifying all of the performance - also in the multi-task case - but there is a large number of model combinations still to be evaluated in future work.

4 Discussion & Conclusion

We derived a sparse version of the pairwise likelihood model using the pseudo-input formulation, and applied the Laplace approximation. We suggest to examine Expectation Propagation and (sequential) MCMC methods for more efficient and exact approximations. The pseudo-inputs are optimized using an evidence optimization approach which in general is challenging due to local maximum of the evidence, which is to be examined in the future. For now we rely on a ”good” initialization. In the final step we suggested that the pairwise pseudo-input model should be combined with a sequential experimental design to reduce the actual number of pairwise experiments.

A synthetic example was used to show the effect of the pseudo-inputs and evidence optimization. As motivating example we presented a multi-task problem, namely a music preference problem. This typically requires a sparse approximation both in terms of input (tracks) as evaluated and in terms of the number of comparisons users have to perform, but the evaluation of the latter is considered future work on a larger dataset. We see the pseudo-input model as a useful tool in examining clustering properties of features and users in GP based preference learning, but this will probably require more elaborate inference methods and kernels.

In conclusion this workshop contribution serves primarily as a presentation of the pairwise likelihood in a pseudo-input formulation with the sequential design as an additional suggested option.

References

[1] R. D. Bock and J. V. Jones. The Measurement and Prediction of Judgment and Choice. 1968.

[2] E. Bonilla, K. Ming Chai, and C. Williams. Multi-task gaussian process prediction. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 153–160. MIT Press, Cambridge, MA, 2008.

[3] E. Bonilla, S. Guo, and S. Sanner. Gaussian Process Preference Elicitation. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 262–270. 2010.

[4] W. Chu and Z. Ghahramani. Extensions of Gaussian Processes for ranking: semi-supervised and active learning. In *Workshop Learning to Rank at Advances in Neural Information Processing Systems 18*, 2005.

[5] W. Chu and Z. Ghahramani. Preference Learning with Gaussian Processes. *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 137–144, 2005.

[6] B. S. Jensen, J. S. Gallego, and J. Larsen. A Predictive Model of Music Preference using Pairwise Comparisons - Supporting Material and Dataset. www.imm.dtu.dk/pubdb/p.php?6143.

[7] B. S. Jensen and J. B. Nielsen. Pairwise Judgements and Absolute Ratings with Gaussian Process Priors. Technical report, November 2011.

[8] D. V. Lindley. On a Measure of the Information Provided by an Experiment. *The Annals of Mathematical Statistics*, (4):986–1005, 1956.

[9] Y. Qi, A. Abdel-Gawad, and T. Minka. Sparse-posterior gaussian processes for general likelihoods. In *Proceedings of the Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, 2010.

[10] E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using Pseudo-Inputs. *Advances in neural information processing*, 2006.

[11] L. L. Thurstone. A Law of Comparative Judgement. *Psychological Review*, 34, 1927.