Multi-Task Preference Learning with an Application to Hearing-Aid Personalization

Adriana Birlutiu, Perry Groot, Tom Heskes

Radboud University Nijmegen, Intelligent Systems, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands

Abstract

We present an EM-algorithm for the problem of learning preferences with Gaussian processes in the context of multi-task learning. We validate our approach on an audiological data set and show that predictive results for sound quality perception of normal-hearing and hearing-impaired subjects, in the context of pairwise comparison experiments, can be improved using a hierarchical model.

Key words: preference learning, multi-task learning, hierarchical modeling, Gaussian processes

1. Introduction

There has been a wide interest in learning the preferences of people within artificial intelligence research in the last years [19]. Preference learning is a crucial aspect in modern applications such as decision support systems [14], recommender systems [9, 7], and personalized devices [17, 30].

It is important to optimize the preference learning process in terms of cost/time invested. Many machine learning techniques especially designed for learning optimization, such as multi-task learning, have been little explored in the context of preference learning. Multi-task learning is especially suited to the situation in which data for a specific single scenario is scarce, but data is already available from similar scenarios. An example is evaluating sound quality with hearing-aids: we have gathered sound evaluations for quite some subjects, information that we would like to exploit when learning a model for a new

subject.

The aim of this article is to apply multi-task learning to the context of preference learning. We consider the problem of learning subject preferences not as an individual problem, but in the context of learning from similar tasks with multiple subjects. In this way, the model of different subjects can regularize and influence each other. We demonstrate the usefulness of our model on an audiological data set. We show that the process of learning preferences can be significantly improved by using a hierarchical non-parametric model based on Gaussian processes.

1.1. Related Work

In this section we review some studies from preference learning and multi-task learning related to the work presented in this paper.

1.1.1. Preference Learning

Preference learning has recently received much attention in the machine learning community [22]. In the literature, two approaches are mainly used for representing preference information: *i)* binary preference predicates and *ii)* scoring methods (utility functions) [21, 22]. For example, the first approach solves a ranking problem as an augmented binary classification problem [29, 28, 21, 1]; the second approach uses regression to map instances to target valuations for direct ranking [13, 18, 16]. We focus on the second approach by modeling utility functions using Gaussian processes (GPs). By formulating the preference elicitation process as a probabilistic Bayesian learning problem, one can deal with inconsistencies in subject responses as well as learn biases the subject may have. GPs have been around quite some time [32, 8], nevertheless, their applications have increased considerably over the years and is still the focus of much research [41]. Only recently, GP models have been applied to the problem of eliciting people's preferences [16, 12] or eliciting probability distributions from expert's opinions [26, 27, 38].

1.1.2. Multi-Task Learning

The basic idea in multi-task learning is that models learned on different scenarios have parts in common. In a Bayesian framework this often boils down to the sharing of a hierarchical prior [3, 20, 44]. A typical application scenario for multi-task learning are recommender systems [7, 36], some of these applications combine content information (e.g., features of items) with collaborative information (data from other subjects) [15, 45]. Multi-task learning with Gaussian processes has recently received attention [43, 46, 10, 40]. The learning setting in [15] for conjoint analysis is similar to the one considered in this paper, however, the authors restrict to a linear parametric form for the utility function. The work in [20] extends kernel learning to the multi-task setting. The contribution of this paper is the extension of the multi-task Gaussian processes for regression introduced by [43, 46] to learning from qualitative preference statements. Preliminary results were reported by us in [5].

1.2. Structure of the Article

Section 2 introduces the probabilistic choice model used for learning preferences, which assumes a latent utility function. Section 3 presents three representations for utility functions: *i)* A parametric representation in which multi-task learning can be easily implemented; *ii)* A non-parametric Gaussian process representation; *iii)* A dual representation based on Gaussian processes. Section 4 describes Bayesian learning of the individual utility function. Section 5 presents the multi-task preference learning. We introduce a hierarchical extension to the Bayesian framework and use the Expectation Maximization algorithm for learning a hierarchical prior. Section 6 reports experimental results with the hierarchical model for learning subject preferences in an audiological context. Section 7 presents our conclusions and directions for future work. Appendices A and B give details about the algorithms developed in this paper.

1.3. Notation

Boldface notation is used for vectors and matrices and normal fonts for the components of vectors and matrices or scalars. Superscript is used to distinguish between different vectors or matrices and subscript to address their components. The notation $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is used for a multivariate Gaussian with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$. The transpose of a matrix $\boldsymbol{M}$ is denoted by $\boldsymbol{M}^T$. The zero vector and identity matrix are denoted by $\mathbf{0}$ and $\mathbf{1}$, respectively.

2. Probabilistic Choice Models

Let $X = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$ be a set of N distinct inputs. Typically, every input is represented by a d-dimensional vector of features, $\boldsymbol{x}^i \in \mathbb{R}^d$. Let D^j be a set of N^j observed preference comparisons over instances in X, corresponding to subject j,

$$D^j = \{(\boldsymbol{x}^{i1}, \ldots, \boldsymbol{x}^{iK}, y^i)|1 \leq i \leq N^j, \boldsymbol{x}^{i\cdot} \in X, y^i \in \{1, \ldots, K\}\}$$

where $y^i = k$ means that alternative $\boldsymbol{x}^{ik}$ is preferred from the K inputs presented to subject j. We consider a version of this setup in which the preference data of each subject uses the same set of inputs X, which is known beforehand and remains fixed. This is the standard setup in marketing applications of preference modeling where the same choice panel questions are given to many individual consumers, each subject provides his/her own preferences, and we assume that there is some similarity among the preferences of the subjects.

The preference observations from the comparisons described above can be modeled using probabilistic choice models. The main idea behind probabilistic choice models is to assume a latent utility function value $U^j(\boldsymbol{x}^i)$ associated with each input $\boldsymbol{x}^i$ which captures the preference of subject j for $\boldsymbol{x}^i$. In the ideal case, the latent function values are consistent with the preference observations, which in probabilistic terms can be written as $P(y^i = k|\boldsymbol{x}^{i1}, \ldots, \boldsymbol{x}^{iK}, \boldsymbol{\theta}^j) = 1$ if $U^j(\boldsymbol{x}^{ik}) \geq U^j(\boldsymbol{x}^{il})$, $l \neq k$. In practice, however, subjects are often inconsistent

in their responses. In order to deal with these inconsistencies, a standard modeling assumption [11, 31, 25] is that the subject's decision in such a forced-choice comparison follows a multinomial logistic model, which is defined as

$$P(y^i = k|\boldsymbol{x}^{i1}, \ldots, \boldsymbol{x}^{iK}, U^j) = \frac{\exp\left[U^j(\boldsymbol{x}^{ik})\right]}{\sum_{l=1}^{K} \exp\left[U^j(\boldsymbol{x}^{il})\right]} \,. \tag{1}$$

For pairwise comparisons ($K = 2$), Equation (1) is known as the Bradley-Terry model [11]. For multiple choice experiments ($K > 2$), the model is a softmax function [31].

An alternative to the model from Equation (1) is the multinomial probit model, which has been used to learn from pairwise comparisons in [16, 12]. The two models, logistic and probit, give similar predictions, however, for ($K \geq 3$) the probit model is more difficult to handle [33]. For this study we use the multinomial logistic model.

3. Modeling the Utility Function

The probabilistic choice model assumes a latent utility function U^j. This section discusses three representations for the utility function:

1. A parametric representation in which multi-task learning is naturally obtained by introducing a joint prior over parameters (Section 3.1).
2. A non-parametric representation based on Gaussian processes (Section 3.2). Multi-task learning is in this case arguably more complicated since here one has to consider a joint prior over functions.
3. A dual representation of the utility function based on Gaussian processes (Section 3.3). This dual representation has a parametric form on which multi-task learning can be easily implemented by employing the theory of hierarchical modeling for parametric models. We show that this representation preserves properties of the non-parametric Gaussian process representation (Section 3.4).

The second and third representation are graphically illustrated in Figure 1 for the case of pairwise comparisons ($K = 2$). For simplicity, we omit the superscript j when representing individual utility functions.

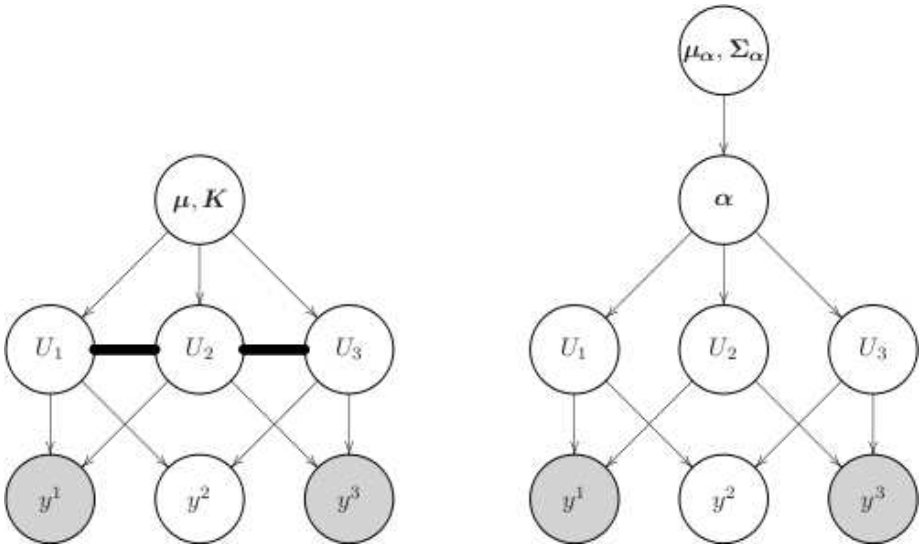

Figure 1: Preference learning based on two representations of the utility function. Left: non-parametric Gaussian process (cf. Section 3.2). Right: parametric Gaussian process (cf. Section 3.3). The observation y^1 of the comparison $\{\boldsymbol{x}^1, \boldsymbol{x}^2\}$ depends on the value associated by the subject's latent utility function with the inputs $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$, $U_1 = U(\boldsymbol{x}^1)$ and $U_2 = U(\boldsymbol{x}^2)$ respectively. The goal is to predict the outcomes of the unseen comparisons (y^2) based on the observed ones (y^1 and y^3). We do this by learning the latent utility function U.

3.1. Parametric Models for Utility Functions

The utility function in the parametric representation is a fixed model, $U(\boldsymbol{x}, \boldsymbol{\theta})$, in which the vector of parameters $\boldsymbol{\theta}$ captures the preferences of the subject. To learn a subject's preferences, we need to learn the parameter $\boldsymbol{\theta}$. Multi-task learning is implemented by introducing a prior distribution over $\boldsymbol{\theta}$. This prior is learned from the data available from all subjects. Since the model $U(\boldsymbol{x}, \boldsymbol{\theta})$ is predefined, this parametric representation is rather limited.

3.2. Non-Parametric Models for Utility Functions

The main advantage of using the Gaussian process formalism in our framework is that it models the utility function in a non-parametric way, allowing more flexibility than having a fixed parametric model. Furthermore, the computational complexity of GPs is independent of the dimension of the inputs but dependent on the number of inputs; this is an advantage when having few data points but of high dimension.

A Gaussian process (GP) [41] is a collection of random variables, any finite number of which have a joint Gaussian distribution. In our case the random variables are the output values of the utility function and we identify the utility function U with a finite vector $\boldsymbol{U}$. Following the approach of [16] for learning preferences with GPs, we define a GP prior over the utility function, i.e., given $X = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$, the joint distribution over the utility function values is a multivariate Gaussian distribution,

$$\{U(\boldsymbol{x}^1), \ldots, U(\boldsymbol{x}^N)\} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K}) \,. \tag{2}$$

The covariance matrix $\boldsymbol{K}$ is generated by a kernel function κ, $\boldsymbol{K}_{ij} = \kappa(\boldsymbol{x}^i, \boldsymbol{x}^j)$. Possible choices for κ are, for example, the linear kernel κ_{Linear} or the Gaussian kernel κ_{Gauss} defined below,

$$\kappa_{\text{Linear}}(\boldsymbol{x}^i, \boldsymbol{x}^j) = \sum_{l=1}^{d} x_l^i y_l^j \,,$$
$$\kappa_{\text{Gauss}}(\boldsymbol{x}^i, \boldsymbol{x}^j) = \exp\left(-\frac{s}{2}\sum_{l=1}^{d}(x_l^i - y_l^j)^2\right) \,.$$

where s is a length-scale parameter.

A Gaussian process is in fact equivalent to a Bayesian interpretation of linear regression. Let

$$U(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\theta} = \sum_i \theta_i \phi_i(\boldsymbol{x}) \,,$$

be a linear combination of (a possibly infinite number of) basis functions $\phi_i(\cdot)$ where $\boldsymbol{\theta}$ is a weight vector. If the weight vector $\boldsymbol{\theta}$ is drawn from a Gaussian distribution, this induces a probability distribution over functions $U(\cdot) = \boldsymbol{\phi}(\cdot)^T \boldsymbol{\theta}$. This distribution is a Gaussian process.

The left-hand side of Figure 1 is a graphical representation of preference learning using the GP representation of the utility function. The utility function values $U(\boldsymbol{x}^1), \ldots, U(\boldsymbol{x}^N)$ are correlated, and depend on the prior estimates $\boldsymbol{\mu}$ and $\boldsymbol{K}$.

3.3. Dual Formulation of the GP

Inspired by the representer theorem [42] — that makes use of both the parametric model and the flexibility of the GP formalism – we use a dual representation for the utility function. The dual representation has a parametric form on which multi-task learning can be easily implemented by employing the theory of hierarchical modeling for parametric models. In the dual representation, the utility function $U(\boldsymbol{x})$, $\boldsymbol{x} \in X$ is defined as follows

$$U(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \kappa(\boldsymbol{x}, \boldsymbol{x}^i) \,, \tag{3}$$

where $\boldsymbol{x}^i \in X$, κ is the kernel function, and $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}})$. The vector of parameters $\boldsymbol{\alpha}$ with dimension N — the number of inputs — compactly captures the information collected from the data set related to a subject. Even though $\boldsymbol{\alpha}$ is a parameter, it does not specify the form of the utility function — as the representation of the utility function in Equation (3) is data dependent. The data-dependent $\boldsymbol{\alpha}$ parameter can give further insights about the importance of each data point and can be used to obtain sparseness and detect outliers [24].

The right-hand side of Figure 1 is a graphical representation of preference learning using the dual representation of the GP. The utility function is determined by the parameter $\boldsymbol{\alpha}$. Furthermore, $\boldsymbol{\alpha}$ depends on the hierarchical prior estimates $\boldsymbol{\mu}_{\boldsymbol{\alpha}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$. The utility function values $U(\boldsymbol{x}^1), \ldots, U(\boldsymbol{x}^N)$ are conditionally independent given $\boldsymbol{\alpha}$.

3.4. Equivalence of the GP Representations

In this section we analyze the relation between the two Gaussian process representations of the utility function given in Sections 3.2 and 3.3. We show

below that the two representations induce the same Gaussian distribution over the utility function for any subset $Z \subseteq X$.

Let $\boldsymbol{U}_Z$ be the vector $\boldsymbol{U}$ restricted to the index set Z, and let $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}})$ be a Gaussian distributed variable. From Equation (3) follows that $\boldsymbol{U}_{\boldsymbol{Z}}$ is a linear combination of Gaussian distributed variables and has therefore a multivariate Gaussian distribution. The distribution over $\boldsymbol{\alpha}$ induces the following distribution over $\boldsymbol{U}_Z$

$$\boldsymbol{U}_Z \sim \mathcal{N}(\boldsymbol{K}(Z,X)\boldsymbol{\mu}_{\boldsymbol{\alpha}},\, \boldsymbol{K}(Z,X)\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}\boldsymbol{K}(Z,X)^T)\,. \tag{4}$$

The two Gaussian distributions from Equations (4) and (2) restricted to $Z \subseteq X$ are the same when

$$\boldsymbol{K}(Z,X)\boldsymbol{\mu}_{\boldsymbol{\alpha}} = \boldsymbol{\mu}_Z\,,$$
$$\boldsymbol{K}(Z,X)\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}\boldsymbol{K}(Z,X)^T = \boldsymbol{K}(Z,Z)\,,$$

with $\boldsymbol{\mu}_Z$ the vector $\boldsymbol{\mu}$ restricted to the index set Z. This leads to the following result.

Theorem 3.1 (Primal-Dual Equivalence). *The utility model* $U(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \kappa(\boldsymbol{x}, \boldsymbol{x}^i)$ *with* $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}})$ *and* $\boldsymbol{x} \in X = \{\boldsymbol{x^1}, \ldots, \boldsymbol{x^N}\}$ *is equivalent with the standard GP formulation* $\boldsymbol{U} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$ *when*

$$\boldsymbol{K}\boldsymbol{\mu}_{\boldsymbol{\alpha}} = \boldsymbol{\mu}\,, \tag{5}$$
$$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} = \boldsymbol{K}^+\,, \tag{6}$$

with $\boldsymbol{K}^+$ *the pseudo-inverse of* $\boldsymbol{K}$.

Proof: Equation (6) follows directly from the definition of the pseudo-inverse,

$$\boldsymbol{K}\,\boldsymbol{K}^+\,\boldsymbol{K} = \boldsymbol{K}\,.$$

If $\boldsymbol{K}$ is invertible, for any $\boldsymbol{\mu}$ there exists a $\boldsymbol{\mu}_{\boldsymbol{\alpha}}$ that satisfies Equation (5). This property does not necessarily hold if $\boldsymbol{K}$ is not invertible. □

The equivalence between the primal and the dual representations holds when we apply the model in a transductive setting, i.e., only to inputs $\boldsymbol{x} \in X$. The two representations are not equivalent anymore when we apply the model to a new test point $\boldsymbol{x}^* \notin X$.

4. Learning the Utility Function

In order to learn a subject's preferences, we treat the vector of parameters $\boldsymbol{\alpha}$ as a random variable. After performing an experiment and observing its outcome, the posterior distribution over $\boldsymbol{\alpha}$ is computed using Bayes' rule,

$$\begin{aligned} P(\boldsymbol{\alpha}|X, \mathcal{O}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &\propto P(\boldsymbol{\alpha}) P(\mathcal{O}|X, \boldsymbol{\alpha}) \\ &= P(\boldsymbol{\alpha}) \prod_{i=1}^{N} P(y^i|\boldsymbol{x}^{i1}, \ldots, \boldsymbol{x}^{iK}, \boldsymbol{\alpha}) \,, \end{aligned}$$

with $X = \{(\boldsymbol{x}^{i1}, \ldots, \boldsymbol{x}^{iK}),\, i = 1, \ldots, N\}$, observations $\mathcal{O} = \{y^i,\, i = 1, \ldots, N\}$, and likelihood terms as given in Equation (1). We make the common assumption of a Gaussian prior distribution. Note that the maximum of the posterior distribution gives a good estimate for $\boldsymbol{\alpha}$ and can easily be computed. The entire distribution over $\boldsymbol{\alpha}$ is, however, needed in the context of multi-task learning. The exact posterior distribution is intractable, therefore, we approximate it with a Gaussian. The Gaussian approximation is a good approximation of the posterior because with few data points the posterior is close to a Gaussian due to the prior, and with many data points the posterior approaches again a Gaussian as a consequence of the central limit theorem [6]. Two types of approaches exist for approximating the posterior distribution *i)* deterministic methods for approximate inference (e.g., Laplace's method [35], Expectation Propagation [37]); *ii)* methods based on sampling. Since the sampling methods are computationally expensive, and the deterministic methods are known to be very accurate for these types of models [25] we focus on deterministic methods. In Appendix A we present two methods for approximate inference in the probabilistic choice models introduced in Section 2.

5. Multi-Task Preference Learning

In this section we consider learning the utility function in a multi-task setting. We implement the multi-task learning using Bayesian hierarchical modeling. We derive a method for gathering data from previous subjects into a single distribution that is used as a prior distribution for a new subject.

Assume that we have M subjects for which we already learned preferences, each of them with his own set of experiments and responses. The inference problems for all the subjects are coupled by having the same prior over the parameters $\boldsymbol{\alpha}^j$, i.e., we set $P(\boldsymbol{\alpha}^j) = \mathcal{N}(\boldsymbol{\alpha}^j|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ a Gaussian prior with the same $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for all subjects. The posterior distribution for each subject is assumed to be (close to) a Gaussian with mean $\boldsymbol{\mu}^j$ and variance $\boldsymbol{\Sigma}^j$. A penalized version of the maximum likelihood values for the prior mean $\boldsymbol{\mu}$ and the prior variance $\boldsymbol{\Sigma}$, can be obtained by specifying a hyper prior distribution over $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, $P(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We assume a normal-inverse-Wishart distribution as the hyper prior since it is the conjugate prior for the multivariate distribution,

$$P(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu}|\boldsymbol{\mu}_0, \frac{1}{\pi}\boldsymbol{\Sigma})\, \mathcal{IW}(\boldsymbol{\Sigma}|\tau, \boldsymbol{\Sigma}_0)\,.$$

The normal-inverse-Wishart distribution can be specified by means of the scale matrix $\boldsymbol{\Sigma}_0$ with precision τ, and mean $\boldsymbol{\mu}_0$ with precision π. We assume that $\boldsymbol{\mu}_0 = \mathbf{0}$ and $\boldsymbol{\Sigma}_0 = \mathbf{1}$.

EM Algorithm for Learning the Hierarchical Prior

The hierarchical prior is obtained by maximizing the penalized loglikelihood of all data. This optimization is performed by applying the Expectation Maximization algorithm [23, 46], which reduces to the iteration (until convergence) of the following two steps.

E-step: For each subject j, estimate the sufficient statistics (mean $\boldsymbol{\mu}^j$ and covariance matrix $\boldsymbol{\Sigma}^j$) of the posterior distribution over $\boldsymbol{\alpha}^j$, given the current estimates, $\boldsymbol{\mu}^{(t)}$ and $\boldsymbol{\Sigma}^{(t)}$, of the hierarchical prior. The E-step is performed using one of the inference techniques mentioned in Appendix A.

M-step: Re-estimate the parameters of the hierarchical prior:

$$\begin{aligned}\boldsymbol{\mu}^{(t+1)} =& \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\mu}^{j}\,,\\ \boldsymbol{\Sigma}^{(t+1)} =& \frac{1}{\tau+M}\Bigg[\pi\boldsymbol{\mu}^{(t+1)}{\boldsymbol{\mu}^{(t+1)}}^{T} + \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\Sigma}^{j} + \\ & \mathbf{1} + \sum_{j=1}^{M}(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})^{T}\Bigg]\,, \end{aligned} \tag{7}$$

where $\boldsymbol{\mu}^j$ and $\boldsymbol{\Sigma}^j$ are the posterior mean and variance for subject j computed based on the previous prior mean $\boldsymbol{\mu}^{(t)}$ and variance $\boldsymbol{\Sigma}^{(t)}$. The update equation for the variance relates to the variance of a mixture model: the last term on the right-hand side of Equation (7) computes the variance in the individual means and the second term the average of the individual variances in the mixture components.

In each E-step, the distribution over $\boldsymbol{\alpha}^j$ is approximated with a multivariate Gaussian. Therefore, in our hierarchical framework each utility function U^j can still be interpreted as an (approximate) Gaussian process (cf. Section 3.4). The derivation of the EM algorithm is given in Appendix B.

6. Experiments

We validated our approach for hierarchical preference learning on an audiological data set. The data set consists of evaluations of sound quality from 14 normal-hearing and 18 hearing-impaired subjects, which we considered as two separate data sets. Each person was subjected to 576 paired-comparison listening experiments of the form $(\boldsymbol{x}_1, \boldsymbol{x}_2, y)$, where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ represents two input sounds processed with two different settings of the hearing-aid parameters, and $y = \{1, 2\}$ denotes which of the two alternatives was preferred by the subject. A detailed description of the data set can be found in [2].

The goal of the validation was to check whether the preferences of a new subject can be learned more accurately by using the available preferences of

a group of subjects. To answer this question we compared the performances obtained using a hierarchical model learned from the group of subjects versus a model which assumes no information about the new subject's preferences. Each subject was characterized by a utility function which describes his/her preferences. The utility function for the jth subject was parametrized by the vector $\boldsymbol{\alpha}^j$ as discussed in Section 3. In a simulation, the jth subject was left out. The data set for the left-out subject, was split into training (used for learning preferences) and testing (the accuracy of the predictions on the test data was used as a measure of how much we learned about subject's preferences). The EM algorithm described in the previous section was used to gather data from the rest of the subjects in a probability distribution over $\boldsymbol{\alpha}^j$, which was used as the starting prior for the left-out subject. The values of the hyper-parameters of the hierarchical prior were set to $\pi = 0$ and $\tau = 1$. Predictions were made on the test data using a model with a flat prior which assumes no information, and a model which uses the hierarchical prior. For each subject, we averaged the results using 10-folds cross-validation. Furthermore, the results were averaged within each group of normal-hearing and hearing-impaired subjects. The plots on the left-hand side of Figure 2 give the percentage of predictions on which the two models (the one with the hierarchical and the one with the flat prior) disagree, with respect to the total number of predictions made; the dashed line refers to a linear kernel, the dotted line to a Gaussian kernel. As it can be seen from the plots, the difference between the two models decreases as a function of the number of observations. The plots on the right-hand side of Figure 2 show the percentage of correct predictions made using the hierarchical prior, with respect to the number of predictions on which the two models disagree. It can be seen from the plots that especially in the beginning of the learning process, with few observations, the model with a prior learned from the community of other subjects significantly outperforms the model with a flat prior.

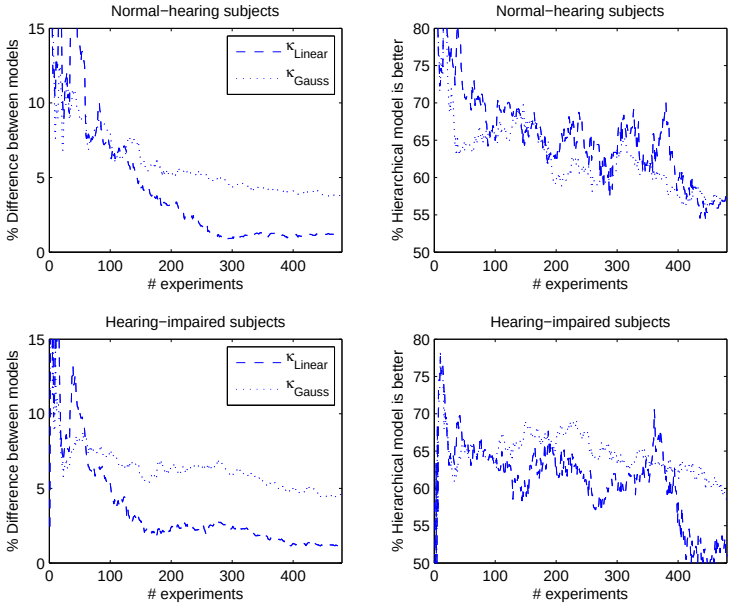

Figure 2: Left: percentage of the number of predictions on which the two models (with the hierarchical and with a flat prior) disagree. Right: percentage of the number of times the prediction accuracy using the hierarchical prior is better than the prediction accuracy with a flat prior. For the Gaussian kernel we set $s = 1$; the results are rather insensitive to the specific choice for this parameter. Top and bottom rows refer to experiments on the data set from normal-hearing and hearing-impaired subjects, respectively.

7. Conclusions and Future Work

We have introduced a hierarchical modeling approach for learning related functions of multiple subjects performing similar tasks using Gaussian processes. A hierarchical prior was used from which model parameters were sampled in order to enforce a similar structure for the utility function of each individual subject.

We are interested in further improvements of the model. Particularly, we plan to investigate how to select, in an active way, the most informative experiments in order to learn subjects' preferences. Furthermore, it might be interesting to automatically cluster, either beforehand or as an integral part of

the algorithm, the subjects into groups with similar behavior—in the current study we manually clustered the data set into two sets of normal-hearing and hearing-impaired subjects.

A. Methods for Approximate Inference

We present two methods for approximate inference suited to the probabilistic choice models introduced in Section 2.

Laplace's method

In the Laplace approximation [35], the posterior distribution is approximated by a Gaussian with mean equal to the maximum a posteriori solution

$$\boldsymbol{\theta}^* \equiv \operatorname*{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \,,$$

where

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log P(k|\boldsymbol{x}^{i1}, \ldots, \boldsymbol{x}^{iK}, \boldsymbol{\theta}) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}) \,,$$

and variance equal to the inverse of the Hessian, the second derivative of $L(\boldsymbol{\theta})$.

ADF and EP

Assumed Density Filtering and Expectation Propagation [39, 37] are approximation techniques in which the terms of the likelihood corresponding to the observed data are added in a sequential way. At each step the result of the inclusion is projected back into the assumed density (we choose for the assumed density a Gaussian). The projection is done by minimizing the Kullback-Leibler divergence between the real posterior and the approximate density. For assumed densities in the exponential family this reduces to moment matching, i.e., the new approximate posterior is the Gaussian which has the same mean and variance as the real posterior.

For a linear utility model $U(\boldsymbol{x}, \boldsymbol{\theta}) = \Phi(\boldsymbol{x})^T \boldsymbol{\theta}$ and $K = 2$, the computation of the posterior approximation can be simplified from d dimensions (where d is the dimension of $\boldsymbol{\theta}$) to 1 dimension. The likelihood function depends on $\boldsymbol{\theta}$ only

through its projection onto a particular direction defined by the input $\Phi(\boldsymbol{x})$. The key idea is then to decompose $\boldsymbol{\theta}$ such that one of the components of the decomposition is perpendicular to $\boldsymbol{\Sigma}^{1/2}\Phi(\boldsymbol{x})$ (as was, for example, done in [4]). The computations needed for the normalization constant can be simplified as follows

$$\begin{aligned}&\left\langle g\left(\Phi(\boldsymbol{x})^T\boldsymbol{\theta}\right)\right\rangle_{\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\Sigma})}\\&\quad=\left\langle g\left(\eta\sqrt{\Phi(\boldsymbol{x})^T\boldsymbol{\Sigma}\Phi(\boldsymbol{x})}+\Phi(\boldsymbol{x})^T\boldsymbol{\mu}\right)\right\rangle_{\mathcal{N}(\eta|0,1)},\end{aligned}$$

where g is the logistic function and

$$\left\langle g\left(\Phi(\boldsymbol{x})^T\boldsymbol{\theta}\right)\right\rangle_{\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\Sigma})}=\int g\left(\Phi(\boldsymbol{x})^T\boldsymbol{\theta}\right)\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\Sigma})\,\mathrm{d}\boldsymbol{\theta}\,.$$

Similarly, computing the mean and covariance of the real posterior can be reduced to 1 dimension. The same idea of efficiently updating the posterior distribution is extended to generalized linear models in [34] using the Laplace approximation.

B. EM Derivation

The basic idea in Bayesian hierarchical modeling is to assume that the parameters for individual models are drawn from the same hierarchical prior distribution. We make the common assumption of a Gaussian prior distribution, $P(\boldsymbol{\alpha}^j)=\mathcal{N}(\boldsymbol{\alpha}^j|\boldsymbol{\mu},\boldsymbol{\Sigma})$ with the same $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for all models. This prior is updated using Bayes' rule based on the observations from each scenario, resulting in a posterior distribution for each individual model. Because the posterior is intractable, we approximate it with a Gaussian. The hierarchical prior is obtained by maximizing the log-likelihood of all data in a so-called type-II maximum likelihood approach. This optimization is performed by applying the EM algorithm [23, 46], which reduces to the iteration (until convergence) of the following two steps.

E-step: Estimate the sufficient statistics (mean $\boldsymbol{\mu}^j$ and covariance matrix $\boldsymbol{\Sigma}^j$) of the posterior distribution corresponding to each individual model j, given the current estimates ($\boldsymbol{\mu}^{(t)}$ and $\boldsymbol{\Sigma}^{(t)}$) of the hierarchical prior.

M-step: Re-estimate the parameters of the hierarchical prior:

$$\boldsymbol{\mu}^{(t+1)} = \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\mu}^{j}\,, \tag{8}$$

$$\boldsymbol{\Sigma}^{(t+1)} = \frac{1}{\tau+M}\Bigg[\pi\boldsymbol{\mu}^{(t+1)}\boldsymbol{\mu}^{(t+1)^{T}} + \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\Sigma}^{j} + \mathbf{1} + \sum_{j=1}^{M}(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})^{T}\Bigg]\,. \tag{9}$$

The term $\sum_{j=1}^{M}(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})^{T}$, in Equation (9), measures the variance between the most probable estimates for different subjects, and the term $\frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\Sigma}^{j}$ measures the variance of the probabilities $P(\boldsymbol{\alpha}^{j})$ around these most probable estimates, averaged over all the subjects.

In very high dimensions, some of the eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$ may tend to infinity. For numerical stability, we therefore add a small constant β to the diagonal of $\boldsymbol{\Sigma}^{-1}$, and set

$$\boldsymbol{\Sigma} \leftarrow \left(\boldsymbol{\Sigma}^{-1}+\beta\mathbf{1}\right)^{-1}\,,$$

after each update (9).

Let D^{j} denote the data obtained from subject j, $D=\{D^{1},\ldots,D^{M}\}$ denote the data obtained from all subjects, $\mathcal{A}=\{\boldsymbol{\mu}^{j},\boldsymbol{\Sigma}^{j};\, j=1,\ldots,M\}$ denote all parameters for all subjects, and $\Lambda^{(t)}=\{\boldsymbol{\mu}^{(t)},\Sigma^{(t)}\}$ denote the parameters of the hierarchical prior at the tth iteration. In order to obtain the estimates of the hierarchical prior in the $(t+1)$th iteration, we maximize the penalized log likelihood of all data

$$\begin{aligned}&\log[P(D|\Lambda^{(t+1)})P(\Lambda^{(t+1)})]\\ &\qquad = \log P(D|\Lambda^{(t+1)}) + \log P(\Lambda^{(t+1)})\,.\end{aligned}$$

We note that

$$\log P(D|\Lambda^{(t+1)}) = \log\left[\frac{P(\mathcal{A},D|\Lambda^{(t+1)})}{P(\mathcal{A}|D,\Lambda^{(t+1)})}\right]\,,\forall\mathcal{A}$$

and thus,

$$\begin{aligned}
&\log P(D|\Lambda^{(t+1)}) + \log P(\Lambda^{(t+1)}) \\
&\quad = \int P(\mathcal{A}|D,\Lambda^{(t)}) \log\left[\frac{P(\mathcal{A},D|\Lambda^{(t+1)})}{P(\mathcal{A}|D,\Lambda^{(t+1)})}\right] \mathrm{d}\mathcal{A} + \\
&\qquad \log P(\Lambda^{(t+1)}) \\
&\quad = Q(\Lambda^{(t+1)},\Lambda^{(t)}) + \log P(\Lambda^{(t+1)}) - \\
&\qquad \int P(\mathcal{A}|D,\Lambda^{(t)}) \log P(\mathcal{A}|D,\Lambda^{(t)})\, \mathrm{d}\mathcal{A}\,,
\end{aligned} \tag{10}$$

with the "full data loglikelihood"

$$\begin{aligned}
&Q(\Lambda^{(t+1)},\Lambda^{(t)}) \\
&\qquad = \int P(\mathcal{A}|\Lambda^{(t)},D) \log P(\mathcal{A},D|\Lambda^{(t+1)})\, \mathrm{d}\mathcal{A}\,,
\end{aligned} \tag{11}$$

The EM algorithm that iteratively maximizes $Q(\Lambda^{(t+1)},\Lambda^{(t)}) + \log P(\Lambda^{(t+1)})$ is guaranteed to converge to a local maximum of the data likelihood since the negative term in Equation (10) can only make things better when $\Lambda^{(t+1)} = \Lambda^{(t)}$.

Different subjects are only coupled through their joint prior, i.e., we have

$$P(\mathcal{A},D|\Lambda^{(t+1)}) = \prod_{j=1}^{M} P(D^j|\boldsymbol{\alpha}^j) P(\boldsymbol{\alpha}^j|\Lambda^{(t+1)})\,.$$

Plugging this into Equation (11) we get

$$\begin{aligned}
&Q(\Lambda^{(t+1)},\Lambda^{(t)}) \\
&\quad = \int P(\mathcal{A}|D,\Lambda^{(t)}) \sum_{j=1}^{M} \log\left[P(D^j|\boldsymbol{\alpha}^j) P(\boldsymbol{\alpha}^j|\Lambda^{(t+1)})\right] \mathrm{d}\mathcal{A}\,, \\
&\quad = \sum_{j=1}^{M} \int P(\boldsymbol{\alpha}^j|D^j,\Lambda^{(t)}) \log P(\boldsymbol{\alpha}^j|\Lambda^{(t+1)})\, \mathrm{d}\boldsymbol{\alpha}^j \; + \\
&\qquad \text{constants independent of } \Lambda^{(t+1)}\,.
\end{aligned}$$

Ignoring these constants and noting that we can skip the index of the integration variable, we get

$$Q(\Lambda^{(t+1)},\Lambda^{(t)}) = M \int \left[\frac{1}{M} P(\boldsymbol{\alpha}|D^j,\Lambda^{(t)})\right] \log P(\boldsymbol{\alpha}|\Lambda^{(t+1)})\, \mathrm{d}\boldsymbol{\alpha}\,.$$

Thus, at each step the following function is maximized

$$
\begin{aligned}
&Q(\Lambda^{(t+1)}, \Lambda^{(t)}) + \log P(\Lambda^{(t+1)}) \\
&\quad = M \int \left[\frac{1}{M} P(\boldsymbol{\alpha} | D^j, \Lambda^{(t)}) \right] \log P(\boldsymbol{\alpha} | \Lambda^{(t+1)}) \, \mathrm{d}\boldsymbol{\alpha} + \\
&\qquad \log P(\Lambda^{(t+1)}) \, .
\end{aligned}
$$

The maximum of this function can be found by computing the gradients with respect to Λ. For the prior over Λ defined as

$$
P(\Lambda) = P(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu} | 0, \frac{1}{\pi} \boldsymbol{\Sigma}) \, \mathcal{IW}(\boldsymbol{\Sigma} | \tau, \mathbf{1}) \, ,
$$

we get the updates from Equations (8) and (9). Note that considering the maximum-likelihood estimate, without the penalization term, i.e., maximizing $Q(\Lambda^{(t+1)}, \Lambda^{(t)})$, has the nice interpretation of the negative Kullback-Leibler divergence (up to again irrelevant constants independent of $\Lambda^{(t+1)}$) between a single Gaussian $P(\boldsymbol{\alpha} | \Lambda^{(t+1)})$ and a mixture of Gaussians, where each of the Gaussians in the mixture corresponds to the posterior of a subject given the previous setting of prior mean and variance. The maximum of this function is then found by moment matching: we have to match the moments of the single Gaussian to the moments of the mixture of Gaussians. □

References

[1] F. Aiolli and A. Sperduti. Learning preferences for multiclass problems. In *Advances in Neural Information Processing Systems 17*, pages 17–24. MIT Press, 2004.

[2] K.H. Arehart, J.M. Kates, C.A. Anderson, and L.O. Harvey Jr. Effects of noise and distortion on speech quality judgments in normal-hearing and hearing-impaired listeners. *Journal of the Acoustical Society of America*, 122(2):1150–1164, August 2007.

[3] B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99, 2003.

[4] D. Barber and C.M. Bishop. Ensemble learning in Bayesian neural networks. *Neural Networks and Machine Learning*, pages 215–237, 1998.

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

[6] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[8] B.J.N. Blight and L. Ott. A Bayesian approach to model inadequacy for polynomial regression. *Biometrika*, 1:79–88, 1975.

[9] J. Blythe. Visual exploration and incremental utility elicitation. In *Eighteenth national conference on Artificial intelligence*, pages 526–532, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[10] E. Bonilla, K.M. 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.

[11] R.A. Bradley and M.E. Terry. Rank analysis of incomplete block designs, I. the method of paired comparisons. *Biometrika*, 39:324–345, 1952.

[12] E. Brochu, N. de Freitas, and A. Ghosh. Active preference learning with discrete choice data. In J.C. Platt, Y. Koller, D. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 409–416. MIT Press, Cambridge, MA, 2008.

[13] R. Caruana, S. Baluja, and T. Mitchell. Using the future to sort out the present: Rankprop and multitask learning for medical risk evaluation. In *In Advances in Neural Information Processing Systems 8*, pages 959–965, 1996.

[14] U. Chajewska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. In *In Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 363–369, 2000.

[15] O. Chapelle and Z. Harchaoui. A machine learning approach to conjoint analysis. In Lawrence K.S., Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 257–264. MIT Press, Cambridge, MA, 2005.

[16] W. Chu and Z. Ghahramani. Preference Learning with Gaussian Processes. In *Proceedings of the 22nd International Conference on Machine Learning*, volume 119 of *ACM International Conference Proceeding Series*, pages 137–144, Bonn, Germany, 2005.

[17] M. Clyde, P. Müller, and G. Parmigiani. Optimal designs for heart defibrillators. *Case Studies in Bayesian Statistics II*, 105:278–292, 1993.

[18] K. Crammer and Y. Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems 14*, pages 641–647. MIT Press, 2001.

[19] J. Doyle. Prospects for preferences. *Computational Intelligence*, 20(2):111–136, 2004.

[20] T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[21] J. Fürnkranz and E. Hüllermeier. Pairwise preference learning and ranking. In *Proceedings of the 14th European Conference on Machine Learning*, pages 145–156, Cavtat, Croatia, 2003. Springer-Verlag.

[22] J. Fürnkranz and E. Hüllermeier. Preference learning. *Künstliche Intelligenz*, 19(1):60–61, 2005.

[23] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis, Second Edition*. Chapman & Hall/CRC, July 2003.

[24] T. van Gestel, J.A.K. Suykens, G. Lanckriet, A. Lambrechts, B. de Moor, and J. Vandewalle. Bayesian framework for least-squares support vector machine classifiers, Gaussian processes, and kernel Fisher discriminant analysis. *Neural Computation*, 14:1115–1147, 2002.

[25] M. Glickman and S. Jensen. Adaptive paired comparison design. *Journal of Statistical Planning and Inference*, 127:279–293, 2005.

[26] J.P. Gosling. *Elicitation: A Nonparametric View*. PhD thesis, Department of Probability and Statistics, School of Mathematics and Statistics, 2005.

[27] J.P. Gosling, J.E. Oakley, and A. O'Hagan. Nonparametric elicitation for heavy-tailed prior distributions. *Bayesian Analysis*, 2:693–718, 2007.

[28] S. Har-Peled, D. Roth, and D. Zimak. Constraint classification: A new approach to multiclass classification and ranking. In *In Advances in Neural Information Processing Systems 15*, pages 365–379, 2002.

[29] R. Herbrich, T. Graepel, P. Bollmann-Sdorra, and K. Obermayer. Learning preference relations for information retrieval, 1998.

[30] T. Heskes and B. de Vries. Incremental utility elicitation for adaptive personalization. In K. Verbeeck, K. Tuyls, A. Nowé, B. Manderick, and B. Kuijpers, editors, *Proceedings of the Seventeenth Belgium-Netherlands Conference on Artificial Intelligence*, pages 127–134, Brussels, 2005. Koninklijke Vlaamse Academie van België voor Wetenschappen en Kunsten.

[31] B. Kanninen. Optimal design for multinomial choice experiments. *Journal of Marketing Research*, 39:307–317, 2002.

[32] G.S. Kimmeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41:495–502, 1970.

[33] J. Kropko and G. Rabinowitz. Choosing between multinomial logit and multinomial probit models for analysis of unordered choice data. Paper

presented at the annual meeting of the MPSA Annual National Conference, Palmer House Hotel, Hilton, Chicago, 2008.

[34] J. Lewi, R. Butera, and L. Paninski. Efficient active learning with generalized linear models. *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.

[35] D.J.C. Mackay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002.

[36] B. Marlin. Modeling user rating profiles for collaborative filtering. In *Proceedings of the 17th Annual Conference on Neural Information Processing Systems (NIPS03)*. MIT Press, 2003.

[37] T. Minka. *A family of approximation methods for approximate Bayesian inference*. PhD thesis, MIT, 2001.

[38] F.A. Moala and A. O'Hagan. Elicitation of Multivariate Prior Distributions: A nonparametric Bayesian approach. Submitted to the *Journal of Statistical Planning and Inference*, 2009.

[39] M. Opper and O. Winther. Tractable approximations for probabilistic models: The adaptive Thouless-Anderson-Palmer mean field approach. *Physical Review Letters*, 2001.

[40] J. Platt, C. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a Gaussian process prior for automatically generating music playlists. In *In Advances in Neural Information Processing Systems*, pages 1425–1432. MIT Press, 2002.

[41] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[42] B. Schölkopf, R. Herbrich, and A.J. Smola. A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational*

Learning Theory and 5th European Conference on Computational Learning Theory, pages 416–426, London, UK, 2001. Springer-Verlag.

[43] A. Schwaighofer, V. Tresp, and K. Yu. Learning Gaussian process kernels via hierarchical bayes. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1209–1216, Cambridge, MA, 2005. MIT Press.

[44] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.

[45] K. Yu, A. Schwaighofer, V. Tresp, W.Y. Ma, and H.J. Zhang. Collaborative ensemble learning: Combining collaborative and content-based information filtering. In *In Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, pages 616–623, 2003.

[46] K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.