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

Adriana Birlutiu, Perry Groot, Tom Heskes

Radboud University Nijmegen, Intelligent Systems, Heyendaalseweg 135, 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 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, 31].

It is important to optimize the preference learning process in terms of cost/time invested. Many machine learning techniques especially designed for optimizing the learning process, 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 [23]. In the literature, two approaches are mainly used for representing preference information: *i)* binary preference predicates and *ii)* scoring methods (utility functions) [22, 23]. For example, the first approach solves a ranking problem as an augmented binary classification problem [30, 29, 22, 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 [33, 8], nevertheless, their applications have increased considerably over the years and is still the focus of much research [44]. 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 [27, 28, 41].

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, 49]. A typical application scenario for multi-task learning are recommender systems [7, 37], which combine content information (e.g., features of items) with collaborative information (data from other subjects) [15, 50]. Multi-task learning with Gaussian processes has recently received attention [46, 51, 10, 43]. The contribution of this paper is the extension of the multi-task Gaussian processes for regression introduced by [46, 51] 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, which represents how subjects choose among a finite set of alternatives. The model assumes a latent utility function that represents subjects' preferences. 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 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. The appendices 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 $\boldsymbol{I}$, 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, $\boldsymbol{x}^i \in \mathbb{R}^d$. We consider M subjects. Let D^j be a set of N^j observed preference comparisons over instances in X, corresponding to subject j, $j = 1, \ldots, M$,

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

where $y^i = p$ means that alternative $\boldsymbol{x}^{ip}$ is preferred from the P inputs presented to a subject. 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 in the general sense that people have some common preferences.

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 $P(y^i = p|\boldsymbol{x}^{i1}, \ldots, \boldsymbol{x}^{iK}, U^j) = 1$ if $U^j(\boldsymbol{x}^{ip}) \geq U^j(\boldsymbol{x}^{il})$, $l \neq p$. In practice, however, subjects are often inconsistent in their responses. A very inconsistent subject will have a high uncertainty associated with the utility function; this uncertainty is directly taken into account in the probabilistic framework. A standard modeling assumption [11, 32, 26] is that the subject's decision in such a forced-choice comparison follows a multi-

nomial logistic model, which is defined as

$$P(y^i = p|\boldsymbol{x}^{i1}, \ldots, \boldsymbol{x}^{iP}, U^j) = \frac{\exp\left[U^j(\boldsymbol{x}^{ip})\right]}{\sum_{l=1}^{P} \exp\left[U^j(\boldsymbol{x}^{il})\right]} \,. \tag{2}$$

For pairwise comparisons, i.e., the subject choosing between one of two presented alternatives, Equation (2) is known as the Bradley-Terry model [11]. The Bradley-Terry model using a dichotomous response scale {*worse, better*} can be extended to a polytomous response scale, such as for instance {*much worse, worse, equal, better, much better*}. The polytomous response scale results in more information from a comparison than a dichotomous response scale and can be modeled using a polytomous Rasch model [52]. The optimal response scale, however, depends on the application domain. The polytomous scale cannot be applied in some domains. For example, in the audiological domain that we consider in Section 6 it is standard practice to use forced-choice pairwise comparisons using a response scale with two or three items since more alternatives or a larger response scale is tiresome for the subject.

An alternative to the model from Equation (2) 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 $P \geq 3$ the probit model is more difficult to handle [34]. For this study we use the multinomial logistic model.

In this probabilistic framework, learning the preferences of a subject j reduces to learning the corresponding utility function U^j. The goal of this paper is to learn the utility functions corresponding to different subjects, jointly, by sharing information between them.

3. Modeling the Utility Function

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 in Appendix A that this representation preserves properties of the non-parametric Gaussian process representation.

For simplicity of notation we omit, in this section and the next one, the superscript j when referring to the individual utility function.

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 with a fixed parametric model. Furthermore, the computational complexity of GPs is independent of the dimension of the data points but dependent on the number of them; this is an advantage when having few data points but of high dimension.

A Gaussian process (GP) [44] 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,

$$\boldsymbol{U} = \{U(\boldsymbol{x}^1), \ldots, U(\boldsymbol{x}^N)\} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{K}) \,. \tag{3}$$

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,

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

where s is a length-scale parameter. The choice of the kernel function depends on our assumptions about properties of the "true" utility function, where "true" refers to how the people evaluate utilities in reality. In some domains, a linear kernel can be good enough; in other domains when a more complex form of the utility function is needed, a Gaussian kernel is more suited.

A Gaussian process is in fact equivalent to a Bayesian interpretation of linear regression (see [44]). Let

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

be a linear combination of (a possibly infinite number of) basis functions $\phi_i(\cdot)$ where $\boldsymbol{\alpha}$ is a weight vector. If the weight vector $\boldsymbol{\alpha}$ is drawn from a Gaussian distribution, this induces a probability distribution over functions $U(\cdot) = \boldsymbol{\phi}(\cdot)^T \boldsymbol{\alpha}$. This distribution is a Gaussian process. From this analogy it follows that a linear kernel is essentially equivalent to a linear parametric model.

A graphical representation of preference learning using the GP representation of the utility function, for the case of pairwise comparisons, is given on the left-hand side of Figure 1. What is inside the plate corresponds to the utility model of one subject. The response y^1 given by a subject to the comparison

$\{\boldsymbol{x}^1, \boldsymbol{x}^2\}$ depends on the values $U(\boldsymbol{x}^1)$, $U(\boldsymbol{x}^2)$ of the subjects' utility function. The goal is to learn the latent utility function, in order to predict the outcomes of the unobserved comparisons (y^2) based on the observed ones (y^1 and y^3). The utility function values corresponding to a given subject, $U(\boldsymbol{x}^1)$, $U(\boldsymbol{x}^2)$, $U(\boldsymbol{x}^3)$, are correlated in the GP formalism since they depend on each other through the kernel (illustrated by the solid bar between them). Furthermore, the utility models, for each subject, depend on the same prior estimates $\boldsymbol{m}$ and $\boldsymbol{K}$.

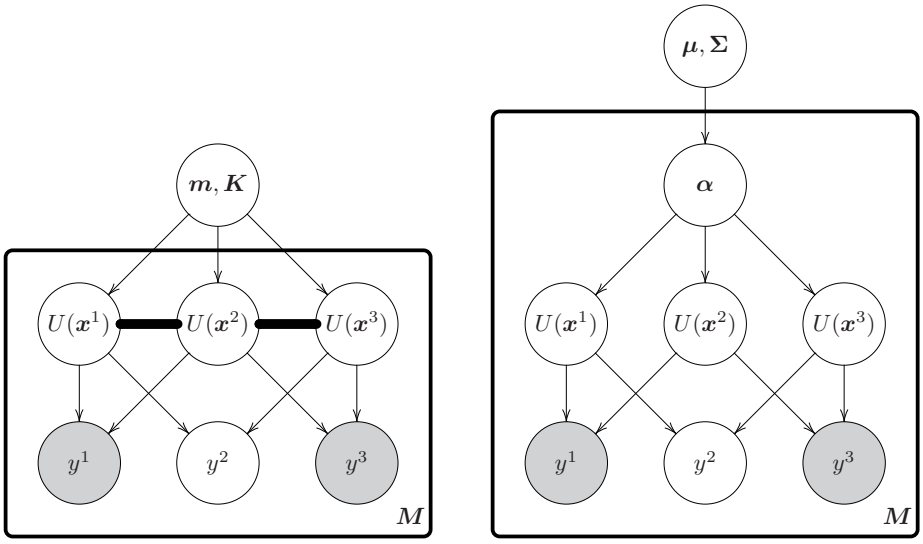

Figure 1: Preference learning based on two representations of the utility function. What is inside the plate corresponds to the utility model of one subject. Left: non-parametric Gaussian process (cf. Section 3.2). Right: parametric Gaussian process (cf. Section 3.3). The observation y^1 of the comparison $\{\boldsymbol{x}^1, \boldsymbol{x}^2\}$ depends on the values $U(\boldsymbol{x}^1)$, $U(\boldsymbol{x}^2)$ of the subjects' utility function. The goal is to learn the latent utility function U in order to predict the outcomes of the unseen comparisons (y^2) based on the observed ones (y^1 and y^3).

3.3. Dual Formulation of the GP

Inspired by the representer theorem [45] — that links the GP to a semi-parametric model — we use a dual representation for the utility function. The

dual representation has a semi-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

$$U(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \kappa(\boldsymbol{x}, \boldsymbol{x}^i) \,, \tag{4}$$

where $\boldsymbol{x}^i \in X$, κ is the kernel function, and $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Equation (4) expresses the utility function as a linear combination of basis functions defined by a kernel centered on the data points. The vector of parameters $\boldsymbol{\alpha}$ with dimension N — the number of inputs — captures the information collected from the data set related to a subject. Even though $\boldsymbol{\alpha}$ is a parameter, it does not induce a fixed form for the utility function — as the representation of the utility function in Equation (4) is data dependent. The parameter $\boldsymbol{\alpha}$ can give further insights about the importance of each data point and can be used to obtain sparseness and detect outliers [25]. The representation of the utility function from Equation (4) is similar to the Relevance Vector Machine (RVM) [48]; the vector of parameters $\boldsymbol{\alpha}$ can give information about which data points (if any) are relevant / prototypes. Furthermore, based on $\boldsymbol{\alpha}$ we can decide which data points to query for labeling next, such as to obtain maximum information in an experimental design / active learning approach. When the number of data points is large, sparsity may be desired for the parameter $\boldsymbol{\alpha}$. In that case, a Laplacian, rather than a Gaussian prior may be more suited.

A graphical representation of preference learning using the dual representation of the GP, for the case of pairwise comparisons, is given on the right-hand side of Figure 1. Analogous to the left-hand side of the figure, what is inside the plate corresponds to the utility model of one subject. The difference with the left-hand side is that the utility function of one subject is determined by the parameter $\boldsymbol{\alpha}$. Note that in this representation the utility function values $U(\boldsymbol{x}^1), \ldots, U(\boldsymbol{x}^N)$ are conditionally independent given $\boldsymbol{\alpha}$. Furthermore, the parameters corresponding to the utility models of different subjects depend on the hierarchical prior estimates $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

The correspondence between the dual (this section) and primal formulation (Section 3.2) is discussed in Appendix A.

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,

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

with inputs $X = \{(\boldsymbol{x}^{i1}, \ldots, \boldsymbol{x}^{iP}),\, i = 1, \ldots, N\}$, preference observations $\mathcal{O} = \{y^i,\, i = 1, \ldots, N\}$, and likelihood terms of the form given in Equation (2). We make the common assumption of a Gaussian prior distribution. The entire posterior distribution, not only a point estimate, of $\boldsymbol{\alpha}$ is needed in the multi-task learning context presented in the next section. 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 [36], Expectation Propagation [39]); *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 [26] we focus on deterministic methods. In Appendix B we present two methods for approximate inference in the probabilistic choice models described in Section 2.

5. Multi-Task Preference Learning

In this section we consider learning the utility function in a multi-task setting. Consider M tasks, each task corresponding to one subject. Let D^j be the data set of subject j, of the form given in Equation (1). The goal is to learn the latent utility functions U^j, $j = 1, \ldots, M$, jointly, sharing information between tasks. 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.

The utility function U^j is parametrized in terms of $\boldsymbol{\alpha}^j$. The inference problems for all the tasks 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 task 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 is 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 $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$.

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 [24, 51], 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 B.

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

$$\begin{aligned}\boldsymbol{\mu}^{(t+1)} &= \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\mu}^{j}\,,\\ \boldsymbol{\Sigma}^{(t+1)} &= \frac{1}{\tau+M}\Bigg[\pi\boldsymbol{\mu}^{(t+1)}{\boldsymbol{\mu}^{(t+1)}}^{T} + \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\Sigma}^{j} + \\ &\qquad \boldsymbol{I} + \sum_{j=1}^{M}(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})^{T}\Bigg]\,, \end{aligned} \tag{5}$$

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 (5) computes the variance between 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. the equivalence stated in Appendix A). The derivation of the EM algorithm is given in Appendix C.

6. Experimental Evaluation

We validated our approach for hierarchical preference learning on an audiological data set. The audiological 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 pairwise comparison listening experiments of the form $(\boldsymbol{x}^1, \boldsymbol{x}^2, y)$, where $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ represent two output sounds obtained by processing the same input sound using two different parameter settings of the hearing aid, and $y = \{1, 2\}$ denotes

which of the two alternatives was preferred by the subject. The preference data collected in the audiological experiment is related to the overall evaluation of the quality of the sound stimulus presented. Research in audiology [2] shows that intelligibility is an important factor in the perceptual judgment of sound quality by subjects. In order to increase intelligibility, the sound stimulus is being processed, e.g., by reducing noise or increasing the volume in some frequency bands. Sound processing adds, however, different kinds of distortions to the output signal listened to by a subject, thus degrading the comfort and as a result the overall sound quality. The way in which people perceive the quality of the processed sound stimulus varies, even for normal-hearing subjects. 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 from other subjects. To answer this question we compared the hierarchical model with *i)* a pooling method and *ii)* a method which assumes no prior information. In each simulation one subject was left-out (the test subject). The data set for the test 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). In the hierarchical model each subject was characterized by a utility function which describes his/her preferences. The utility function for the test subject was parametrized by the vector $\boldsymbol{\alpha}$ as discussed in Section 3. The EM algorithm described in the previous section was used to gather data from the rest of the subjects in a probability distribution over $\boldsymbol{\alpha}$, which was used as the starting prior. The values of the hyper-parameters of the hierarchical prior were set to $\pi = 0$ and $\tau = 1$. Below we describe the comparison of the pooling method and non-informative prior method with the hierarchical model in more detail.

The pooling method pools all data together and a single model is learned based on all but the test subject, after which data from the test subject is added one by one. A linear kernel was used. For each test subject, we averaged the results using 20 random splits of the data into training (20 data points) and

testing (the remaining data points). Furthermore, the results were averaged within each group of normal-hearing and hearing-impaired subjects. The plots in Figure 2 compare the accuracy obtained using the hierarchical model versus the pooling method. The pooling method works good for normal-hearing subjects but, as we expected, performs worse than the hierarchical model for the hearing-impaired subjects. There is no change in the accuracy of the pooling method as a function of the number of experiments / data points because the few extra data points of the test subject, compared with all the data points from the other subjects, do not really affect the estimate. Note that the variance is higher within the hearing-impaired group due to variations in the audiological conditions.

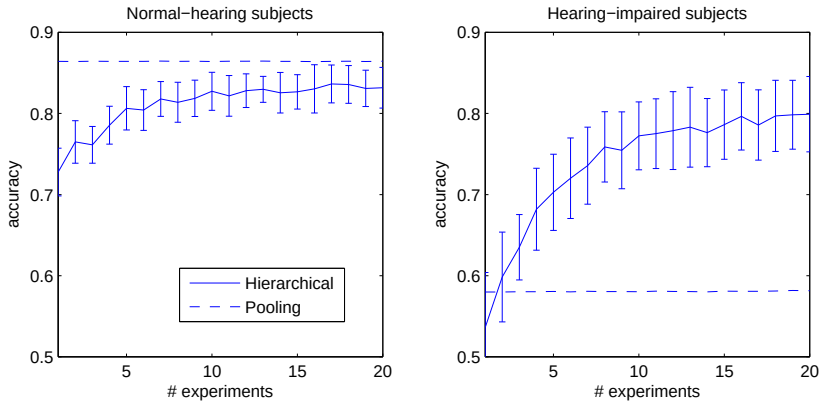

Figure 2: Accuracy for the hierarchical model vs. pooling method. The bars indicate the variance of the accuracy between the subjects, within the two groups of normal-hearing and hearing-impaired subjects.

The non-informative prior method uses a flat prior which assumes no information about the test subject's preferences. For this comparison we only considered the hearing-impaired subjects as the pooling method shows that the normal-hearing subjects are very similar. For each test subject, we averaged the results using 10 random splits of the data into training (450 data points) and testing (the remaining data points). Furthermore, the results were averaged

over all subjects. In Figure 3 left panel we give the percentage of predictions

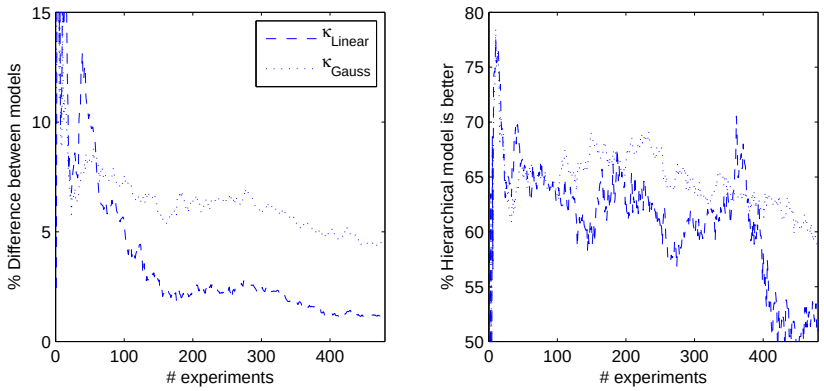

Figure 3: Left: percentage of the number of predictions on which the two models (with the hierarchical and with a flat prior) disagree. Right: percentage of the number of times the prediction accuracy using the hierarchical prior is better than the prediction accuracy with a flat prior. We only considered the hearing-impaired subjects as the pooling method shows that the normal-hearing subjects are very similar.

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. For the Gaussian kernel we set $s = 1$; the results are rather insensitive to the specific choice for this parameter since the high number of data points dominates the model; this is not always the case, and then an appropriate value for s has to be found. As it can be seen from the plots, the difference between the two models decreases as a function of the number of observations. In Figure 3 right panel we 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.

Furthermore, in order to determine which of the kernels is more suited for this data set, we compared the hierarchical model with a Gaussian kernel and

with a linear kernel. We used the same setup as in the previous comparison. In Figure 4 left panel we give the percentage of predictions on which the two kernels disagree, with respect to the total number of predictions made. In Figure 4 right panel we show the percentage of correct predictions made using the Gaussian kernel, with respect to the number of predictions on which the two kernels disagree. The Gaussian kernel appears to be better overall than the linear kernel for this data set.

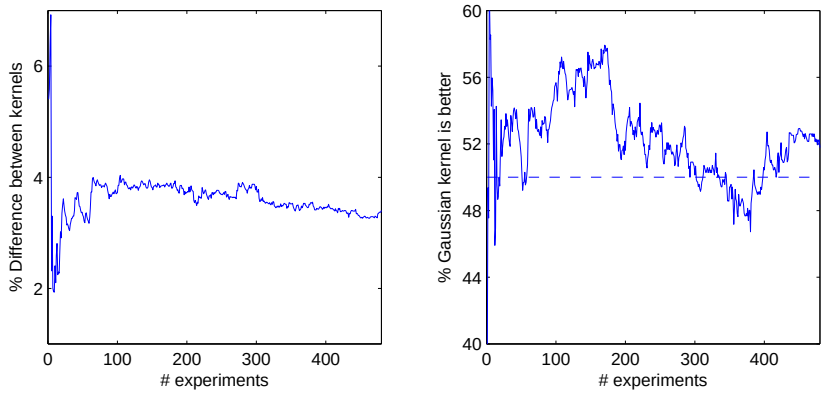

Figure 4: Left: percentage of the number of predictions on which the two hierarchical models (with the Gaussian and with a linear kernel) disagree. Right: percentage of the number of times the prediction accuracy using the Gaussian kernel is better than the prediction accuracy with a linear kernel. We only considered the hearing-impaired group of subjects. The dashed line is drawn for reference of equal performance between the two kernels.

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 ex-

periments 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. For the audiological data set used in this study, we manually clustered the data into two sets of normal-hearing and hearing-impaired subjects since the plots of the maximum-likelihood estimates of the subjects' parameters did not show the need for further subclustering. For other data sets, one could consider replacing the Gaussian prior with a Dirichlet prior [49]. This would lead to automatically clustering of the subjects and would enable the algorithm to identify relatedness among the subjects. In this way the hierarchical prior is learned using those subjects that are more related to the test subject. Another alternative for future research is to compare our approach to other multi-task learning approaches, for example, [38] and [10].

A. Equivalence of the GP Representations

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 $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a Gaussian distributed variable. From Equation (4) follows that $\boldsymbol{U}_Z$ 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 $\boldsymbol{U}_Z$

$$\boldsymbol{U}_Z \sim \mathcal{N}(\boldsymbol{K}(Z,X)\boldsymbol{\mu},\, \boldsymbol{K}(Z,X)\boldsymbol{\Sigma}\boldsymbol{K}(Z,X)^T)\,. \tag{6}$$

The two Gaussian distributions from Equations (6) and (3) restricted to $Z \subseteq X$ are the same when

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

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

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

$$\boldsymbol{K}\boldsymbol{\mu} = \boldsymbol{m}\,, \tag{7}$$

$$\boldsymbol{\Sigma} = \boldsymbol{K}^{+}\,, \tag{8}$$

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

Proof: Equation (8) follows directly from the definition of the pseudo-inverse,

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

If $\boldsymbol{K}$ is invertible, for any $\boldsymbol{m}$ there exists a $\boldsymbol{\mu}$ that satisfies Equation (7). 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 $\boldsymbol{x}^* \notin X$.

B. 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 [36], 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

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

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 [42, 39] 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 $U(\boldsymbol{x}, \boldsymbol{\theta}) = \Phi(\boldsymbol{x})^T \boldsymbol{\theta}$, 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 $\boldsymbol{\Sigma}^{1/2}\Phi(\boldsymbol{x})$. 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

$$g(z) = \frac{1}{1 + \exp(-z)} \,,$$

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. For a more detailed description of the method used here see [47, 4]. The same idea of efficiently updating the posterior distribution is extended to generalized linear models in [35] using the Laplace approximation.

Which approximation technique performs better depends on the real posterior distribution. If the posterior distribution has a form close to a Gaussian, the simple Laplace's method gives good results. For more complex posterior distributions, ADF or EP give, in general, better approximations [39]. In the setting presented in this paper, the product between a logistic function and a Gaussian results in a posterior close to a Gaussian, thus the approximation is very accurate and the choice of the approximation method does not have a big influence on the result. In the experimental evaluation we used ADF.

C. 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 will first state the algorithm and then its derivation. 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 [24, 51], 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. The

E-step is performed using one of the inference techniques mentioned in Appendix B.

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

$$\boldsymbol{\mu}^{(t+1)} = \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\mu}^{j}\,, \tag{9}$$

$$\boldsymbol{\Sigma}^{(t+1)} = \frac{1}{\tau+M}\Bigg[\pi\boldsymbol{\mu}^{(t+1)}\boldsymbol{\mu}^{(t+1)^T} + \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\Sigma}^{j} + \boldsymbol{I} + \sum_{j=1}^{M}(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})^T\Bigg]\,. \tag{10}$$

The term $\sum_{j=1}^{M}(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})(\boldsymbol{\mu}^{j}-\boldsymbol{\mu}^{(t+1)})^T$, in Equation (10), measures the variance between the most probable estimates for different subjects; 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

$$\boldsymbol{\Sigma} \leftarrow \left(\boldsymbol{\Sigma}^{-1} + \beta\boldsymbol{I}\right)^{-1}\,, \tag{11}$$

after each update (10). With the update proposed in (11), the eigenvalues of $\boldsymbol{\Sigma}$ remain finite and we never observed problems with numerical stability.

It is common practice to make approximations in the E-step (see e.g., [21, 40]). In theory convergence can then no longer be guaranteed, but in practice, in particular when the approximations are known to be very accurate (as it is our case, see above) it usually works fine.

In the following we give the derivation of the M-step. 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

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

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,

$$\begin{aligned}
&\log P(D|\Lambda^{(t+1)}) + \log P(\Lambda^{(t+1)}) \\
&\quad = \int P(\mathcal{A}|D, \Lambda^{(t)}) \log \left[\frac{P(\mathcal{A}, D|\Lambda^{(t+1)})}{P(\mathcal{A}|D, \Lambda^{(t+1)})} \right] \mathrm{d}\mathcal{A} + \log P(\Lambda^{(t+1)}) \\
&\quad = Q(\Lambda^{(t+1)}, \Lambda^{(t)}) + \log P(\Lambda^{(t+1)}) - \int P(\mathcal{A}|D, \Lambda^{(t)}) \log P(\mathcal{A}|D, \Lambda^{(t+1)}) \, \mathrm{d}\mathcal{A} \,,
\end{aligned} \tag{12}$$

with the "full data loglikelihood"

$$Q(\Lambda^{(t+1)}, \Lambda^{(t)}) = \int P(\mathcal{A}|\Lambda^{(t)}, D) \log P(\mathcal{A}, D|\Lambda^{(t+1)}) \, \mathrm{d}\mathcal{A} \,, \tag{13}$$

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 (12) can only make things better when $\Lambda^{(t+1)} \neq \Lambda^{(t)}$.

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 (13) 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, noting that we can skip the index of the integration variable, and dropping the superscript notation for $\Lambda^{(t+1)}$, we obtain

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

The prior over Λ is a normal-inverse-Wishart distribution,

$$P(\Lambda) = P(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu}|0, \frac{1}{\pi}\boldsymbol{\Sigma}) \, \mathcal{IW}(\boldsymbol{\Sigma}|\tau, \boldsymbol{\Sigma}_0) \, ,$$

with the inverse Wishart distribution with scale matrix $\boldsymbol{\Sigma}_0^{-1}$ defined as

$$\mathcal{IW}(\boldsymbol{\Sigma}|\tau, \boldsymbol{\Sigma}_0^{-1}) \propto \det(\boldsymbol{\Sigma}_0^{-1})^{\frac{\tau}{2}} \det(\boldsymbol{\Sigma})^{-\frac{\tau+d+1}{2}} \exp\left[-\frac{1}{2} \mathrm{Tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1}) \right]$$

At each step the following function is maximized with respect to Λ

$$\begin{aligned} & Q(\Lambda,\Lambda^{(t)}) + \log P(\Lambda) \\ & \quad = M \int \left[\frac{1}{M} \sum_{j=1}^{M} P(\boldsymbol{\alpha}|D^j, \Lambda^{(t)}) \right] \log P(\boldsymbol{\alpha}|\Lambda) \, \mathrm{d}\boldsymbol{\alpha} \; + \log P(\Lambda) \, . \end{aligned} \tag{14}$$

The maximum is found by computing the gradients of the expression (14)from above with respect to $\Lambda = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and setting these to zero. We compute below the gradient of (14) with respect to $\boldsymbol{\Sigma}$. We start by writing down only the terms in (14) which depend on $\boldsymbol{\Sigma}$.

$$\begin{aligned} \mathcal{QP}(\boldsymbol{\Sigma}) = & \int \sum_{j=1}^{M} P(\boldsymbol{\alpha}|D^j, \Lambda^{(t)}) \left[-\log\det(\boldsymbol{\Sigma})^{1/2} - \frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu}) \right] \mathrm{d}\boldsymbol{\alpha} \\ & - \frac{\tau+d+1}{2} \log\det(\boldsymbol{\Sigma}) - \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1}) - \frac{1}{2}\log\det(\boldsymbol{\Sigma}) - \frac{\pi}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ = & \left(-\frac{1}{2} \sum_{j=1}^{M} \int P(\boldsymbol{\alpha}|D^j, \Lambda^{(t)}) \, \mathrm{d}\boldsymbol{\alpha} - \frac{\tau+d+1}{2} - \frac{1}{2} \right) \log\det(\boldsymbol{\Sigma}) - \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1}) \\ & - \frac{1}{2} \int \sum_{j=1}^{M} P(\boldsymbol{\alpha}|D^j, \Lambda^{(t)})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu}) \, \mathrm{d}\boldsymbol{\alpha} - \frac{\pi}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ = & -\frac{\tau+d+2+M}{2} \log\det(\boldsymbol{\Sigma}) - \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1}) \\ & - \frac{1}{2} \int \sum_{j=1}^{M} P(\boldsymbol{\alpha}|D^j, \Lambda^{(t)})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu}) \, \mathrm{d}\boldsymbol{\alpha} \frac{\pi}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \frac{\pi}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \end{aligned}$$

Taking the derivatives with respect to $\boldsymbol{\Sigma}$ of each of these terms, we get

$$\begin{aligned}
\frac{\partial \log\det(\boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}} &= \det(\boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-T}\frac{1}{\det(\boldsymbol{\Sigma})} = \boldsymbol{\Sigma}^{-1}\,, \\
\frac{\partial \mathrm{Tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1})}{\partial \boldsymbol{\Sigma}} &= -\boldsymbol{\Sigma}^{-T}\boldsymbol{\Sigma}_0^{-T}\boldsymbol{\Sigma}^{-T} = -\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1}\,, \\
\frac{\partial \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}{\partial \boldsymbol{\Sigma}} &= \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^T\boldsymbol{\mu}\boldsymbol{\Sigma}\,, \\
\frac{\partial}{\partial \boldsymbol{\Sigma}}\int \sum_{j=1}^{M} P(\boldsymbol{\alpha}|D^j,\Lambda^{(t)})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu})\,\mathrm{d}\boldsymbol{\alpha} & \\
= -\int \sum_{j=1}^{M} P(\boldsymbol{\alpha}|D^j,\Lambda^{(t)})\boldsymbol{\Sigma}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}\,\mathrm{d}\boldsymbol{\alpha}\,. &
\end{aligned}$$

Collecting the terms from above and setting the derivative to zero, we obtain

$$\boldsymbol{\Sigma} = \frac{1}{\tau+d+2+M}\left[\pi\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma}_0^{-1} + \int\sum_{j=1}^{M} P(\boldsymbol{\alpha}|D^j,\Lambda^{(t)})(\boldsymbol{\alpha}-\boldsymbol{\mu})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\,\mathrm{d}\boldsymbol{\alpha}\right].$$

For each subject j, $P(\boldsymbol{\alpha}|D^j,\Lambda^{(t)})$ is the posterior distribution resulting from the hierarchical prior with the previous estimates $\Lambda^{(t)}$. This posterior is approximated to a Gaussian, $\mathcal{N}(\boldsymbol{\alpha}|\boldsymbol{\mu}^j,\boldsymbol{\Sigma}^j)$, in the previous E-step. Then,

$$\begin{aligned}
&\int\sum_{j=1}^{M} P(\boldsymbol{\alpha}|D^j,\Lambda^{(t)})(\boldsymbol{\alpha}-\boldsymbol{\mu})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\,\mathrm{d}\boldsymbol{\alpha} \\
&\quad= \sum_{j=1}^{M}\int \mathcal{N}(\boldsymbol{\alpha}|\boldsymbol{\mu}^j,\boldsymbol{\Sigma}^j)\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \boldsymbol{\alpha}\boldsymbol{\mu}^T - \boldsymbol{\mu}\boldsymbol{\alpha}^T + \boldsymbol{\mu}\boldsymbol{\mu}^T\right)\mathrm{d}\boldsymbol{\alpha} \\
&\quad= \sum_{j=1}^{M}\boldsymbol{\Sigma}^j + \sum_{j=1}^{M}\boldsymbol{\mu}^j(\boldsymbol{\mu}^j)^T - \sum_{j=1}^{M}\boldsymbol{\mu}^j\boldsymbol{\mu}^T - \sum_{j=1}^{M}\boldsymbol{\mu}(\boldsymbol{\mu}^j)^T + \sum_{j=1}^{M}\boldsymbol{\mu}\boldsymbol{\mu}^T\,,
\end{aligned}$$

and thus,

$$\boldsymbol{\Sigma} = \frac{1}{\tau+d+2+M}\left[\pi\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma}_0^{-1} + \sum_{j=1}^{M}\boldsymbol{\Sigma}^j + \sum_{j=1}^{M}(\boldsymbol{\mu}^j-\boldsymbol{\mu})(\boldsymbol{\mu}^j-\boldsymbol{\mu})^T\right],$$

which is the biased estimator of the variance and where $\boldsymbol{\mu}$ is the new mean found in the M-step. To obtain an unbiased estimator we consider

$$\boldsymbol{\Sigma} = \frac{1}{\tau+M}\left[\pi\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma}_0^{-1} + \sum_{j=1}^{M}\boldsymbol{\Sigma}^j + \sum_{j=1}^{M}(\boldsymbol{\mu}^j-\boldsymbol{\mu})(\boldsymbol{\mu}^j-\boldsymbol{\mu})^T\right],$$

which for $\mathbf{\Sigma}_0 = \boldsymbol{I}$ gives Equation (10). The update for the mean is obtained in a similar way and leads to (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 *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 *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] B. Frey and A. Kannan. Accumulator Networks: Suitors of Local Probability Propagation. Advances in Neural Information Processing Systems 13, pages 486–492, 2001.

[22] 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.

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

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

[25] 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.

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

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

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

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

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

[31] 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.

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

[33] 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.

[34] 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.

[35] 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.

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

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

[38] C. Micchelli and M. Pontil Kernels for Multi-task Learning. Advances in Neural Information Processing Systems, 2004.

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

[40] T. Minka. Expectation-Propagation for the Generative Aspect Model. Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, 2002, pages 352–359.

[41] 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.

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

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

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

[45] 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.

[46] 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.

[47] M. Seeger Notes on Minka's expectation propagation for Gaussian process classification. Technical Report 2002.

[48] M. Tipping. Sparse Bayesian Learning and the Relevance Vector Machine Journal of Machine Learning Research, year 2001, volume 1, pages 211–244.

[49] 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.

[50] 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.

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

[52] Applied Rasch Measurement: A Book of Examples. Editors: S. Alagumalai, D. Curtis and N. Hungi. Springer, 2005, ISBN 978-1-4020-3076-5.