Gaussian Processes with Monotonicity Constraints for Preference Learning from Pairwise Comparisons

Robert Chin[1], Chris Manzie[2], Alex Ira[2], Iman Shames[2]

Abstract— **In preference learning, it is beneficial to incorporate monotonicity constraints for learning utility functions when there is prior knowledge of monotonicity. We present a novel method for learning utility functions with monotonicity constraints using Gaussian process regression. Data is provided in the form of pairwise comparisons between items. Using conditions on monotonicity for the predictive function, an algorithm is proposed which uses the weighted average between prior linear and maximum a posteriori (MAP) utility estimates. This algorithm is formally shown to guarantee monotonicity of the learned utility function in the dimensions desired. The algorithm is tested in a Monte Carlo simulation case study, in which the results suggest that the learned utility by the proposed algorithm performs better in prediction than the standalone linear estimate, and enforces monotonicity unlike the MAP estimate.**

I. INTRODUCTION

We consider the problem of learning a function which captures the tradeoff between competing objectives. We refer to such functions as utility functions and moreover, we focus on learning functions which are monotonic, for the following motivating reasons.

A. Motivation

1) Sensible Utility Functions: Having monotonicity constraints in the case where preferences are monotonic will ensure sensible use and interpretation of learned utility functions (ie. for utility maximisation). To illustrate, consider an example pertinent to control engineering in which the task is to tune controller parameters to optimise a utility function over time-domain characteristics (see [1]). For simplicity, we shall consider the tradeoff between only two characteristics (or 'features'): the negated overshoot and settling time of a single output step response trajectory (the negation is so that the utility function will be monotonically increasing rather than decreasing). It is reasonable to believe that an engineer would prefer less overshoot holding all else constant, and likewise for settling time. The slope of the contours of the utility function indicate the relative quantity of additional overshoot in which the agent is willing to accept in order to reduce settling time, as doing so keeps utility constant (by 'moving along the contour', so-to-speak). As negated overshoot and settling time are both deemed desirable, then contours should naturally be negatively-sloped. A violation in monotonicity of the learned utility will produce positively-sloped contours, giving the interpretation that the engineer is willing to accept additional overshoot to increase settling time, which is clearly not sensible behaviour.

2) Regularisation: Incorporating prior information is a commonly used technique for regularisation in learning. We can leverage prior knowledge of monotonicity to regularise the estimates and reduce the data requirements that lead to a desired quality of estimate, compared to methods which do not explicitly consider monotonicity. For the latter class of algorithms, monotonicity may not even be guaranteed except in the asymptotic case when there is an arbitrarily large amount of data.

B. Literature Overview

The study of preferences has attracted interest from a number of disciplines. Economists first introduced a rigorous treatment on the theory of preferences in the mid-20$^{\text{th}}$ century [2], [3]. Psychologists have also studied the impact of preferences on decision making [4]. Recently, the task of learning preferences from data has gathered attention in the field of machine learning. An overview to preference learning is given in [5].

The current literature highlights that not all established preference learning methods handle monotonicity constraints or guarantee monotonicity. Monotonicity has previously been considered in Gaussian processes regression [6], where 'virtual' derivative observations were injected into the data, however a guarantee of monotonicity for the posterior mean was not provided. Monotonicity has also previously been considered for active learning [7], however in ordinal classification rather than learning a utility function.

The literature on prefrence learning/estimation from pairwise comparison data is rich. A common justification for data in this form is that it may be more useful or reliable in contrast to data consisting of numeric rating labels [8]. Models for processing pairwise comparison data have been formulated since the 1920s for analysing results of surveys with binary responses [9], [10]. Discrete choice theory ([11], [12]) specifically deals with models of decision making when there are a finite number of decisions, for which paired comparisons is a special case. In preference learning literature, 'learning-to-rank' from pairwise comparisons (ie. providing a preference ranking rather than an explicit utility) is considered in [13], while learning utility functions from Gaussian processes is exhibited in [14], [15]. In [16], monotonic utility functions are estimated from pairwise comparisons, however the estimated function is necessarily

[1]Jointly affiliated with The University of Melbourne, Australia and University of Birmingham, United Kingdom. This author is supported by the Elizabeth and Vernon Puzey scholarship. Corresponding author. Email: chinr@student.unimelb.edu.au

[2]Affiliated with The University of Melbourne, Australia

restricted to be monotonic in all features. Preference learning from pairwise comparisons has also been applied in finding optimal policies for reinforcement learning [17].

C. Contributions

In this paper, we consider preference learning by the learning of a utility function, as opposed to learning-to-rank. We further specialise to the case where the data is provided in the form of pairwise comparisons. Our main contributions relate to learning utility functions with the property of monotonicity. We primarily build on the work of [14] in using Gaussian process regression to learn a utility function from pairwise comparisons, by providing an algorithm which can guarantee monotonicity in as many features as desired. The performance of this algorithm is also analysed and tested in a case study.

II. PRELIMINARIES

A. Notation

The symbol $\mathbb{R}$ denotes the set of real numbers and $\mathbb{R}_{\geq 0}$ denotes the set of non-negative real numbers. The Cartesian product between sets $\mathbb{S}_1, \ldots, \mathbb{S}_d$ is denoted $\mathbb{S}_1 \times \cdots \times \mathbb{S}_d$. Vectors are denoted by lower case symbols and may be in either bold or italic font and, unless stated otherwise, are to be interpreted as column vectors. Matrices are denoted by upper case symbols and may be in either bold or italic font. I denotes the identity matrix and $\mathbf{1}$ denotes a vector of ones, with dimension clear from context. The matrix transpose operator is denoted by the $\top$ symbol. A multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ is denoted by $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$.

B. Utility Theory

Let $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ denote a topological space. The binary preference relation $\preceq$ denotes a partial ordering on $\mathcal{X}$, where for a pair of items $x_A, x_B \in \mathcal{X}$, the expression $x_A \preceq x_B$ denotes that item x_B is preferred at least as much as x_A. The weak preference relation is denoted by $\prec$.

Definition 1 (Ordinal utility functions): An ordinal utility function $h : \mathcal{X} \to \mathbb{R}$ for $\preceq$ is a function representing the underlying preferences of an agent such that $x_A \preceq x_B \Leftrightarrow h(x_A) \leq h(x_B)$.

Remark 1: There exists infinitely many ordinal utility functions to represent $\preceq$, since for an ordinal utility function $h(x)$ and any strict monotonic transformation $\psi(\cdot)$, then $\psi(h(x))$ is also an ordinal utility function for $\preceq$.

Definition 2 (Monotonic preferences): A differentiable ordinal utility function $h : \mathcal{X} \to \mathbb{R}$ for preferences $\prec$ is strictly monotonic at x in direction j if and only if

$$\frac{\partial h(x)}{\partial x_j} > 0 \tag{1}$$

while weak monotonicity is defined if (1) holds with $\geq$. We focus on the case when $\mathcal{X}$ is a compact subset of $\mathbb{R}^d_{\geq 0}$.

C. Problem Statement

Assumption 1: (Debreu preferences) The users' preferences satisfy the conditions of Debreu's theorem [3], so that there exists a continuous ordinal utility function that can be used to represent them.

Remark 2: Assumption 1 is not overly restrictive, but does serve the purpose of excluding cases where there does not exist a continuous utility function to represent users' preferences. A well-known example of preferences which do not satisfy Assumption 1 are Lexicographic preferences [18].

Assumption 2: (Monotonicity) The users' preferences are strictly monotonic along dimensions given by the index set $\mathcal{J} \subseteq \{1, \ldots, d\}$.

Remark 3: Assumption 2 is not restrictive because it arises from preconditioned knowledge (ie. common sense or otherwise) of features exhibiting monotonicity, which is a motivating factor for the need to learn a monotonic utility function in the first place. For example, it is natural to treat the amount of negated overshoot and settling time as being desirable, hence they should be monotonic in their preferences.

The learning problem is posed as follows:

Problem 1: Given pairwise comparison data $\mathcal{D}$, learn a continuous ordinal utility function that exhibits strict monotonicity in the respective dimensions given by $\mathcal{J}$.[1]

D. Preference Learning from Gaussian Processes

We summarise an approach for learning a utility function from pairwise comparisons using Gaussian processes. The method is adapted from [14], except a generally non-zero prior mean is employed here. Suppose there is a set $\mathbb{X} := \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of n distinct items where each $\mathbf{x}_i \in \mathcal{X}$ for $i = 1, \ldots, n$, from which items to compare are sampled from. Assume the data generating process follows the rating model in Assumption 3.

Assumption 3: (Rating model) The user generates comparisons using the data generating process

$$v(\mathbf{x}_A) := g(\mathbf{x}_A) + \varepsilon_A \tag{2}$$

$$v(\mathbf{x}_B) := g(\mathbf{x}_B) + \varepsilon_B \tag{3}$$

such that when shown items $\mathbf{x}_A, \mathbf{x}_B \in \mathbb{X}$ for comparison, $v(\mathbf{x}_B) > v(\mathbf{x}_A)$ means the user rates $\mathbf{x}_B$ preferred over $\mathbf{x}_A$, where $\varepsilon_A, \varepsilon_B \sim \mathcal{N}\left(0, \sigma^2_{\text{noise}}\right)$ are i.i.d. rating noise terms and $g(\cdot)$ is the underlying utility function of the user. The reverse analogously holds if $v(\mathbf{x}_B) < v(\mathbf{x}_A)$.

The rating noise may be interpreted as inaccuracy of the user's judgement (ie. due to imperceptible differences in alternatives or other psychological factors such as fatigue). If there are multiple inhomogeneous users providing ratings (as considered in [10]), then any differences between users' opinions can also be modelled using rating noise. The data

[1]Observe that the monotonicity-constrained preference learning problem is trivial if x is of dimension 1, as then the ordinal utility function can be chosen to be any strictly monotonic continuous function. Thus we are primarily interested in the case where the feature vector is of dimension greater than 1.

consists of M comparisons, denoted by $\mathcal{D} = \{\mathbb{X}_A, \mathbb{X}_B\}$ where $\mathbb{X}_A := \{\mathbf{x}_{A1}, \ldots, \mathbf{x}_{AM}\}$, $\mathbb{X}_B := \{\mathbf{x}_{B1}, \ldots, \mathbf{x}_{BM}\}$ (with indices $Ai, Bi \in \{1, \ldots, n\}$) are sets such that the agent has rated $\mathbf{x}_{Bi}$ preferred over $\mathbf{x}_{Ai}$ for each $i = 1, \ldots, M$. We seek an estimate of the latent utilities $\mathbf{f} := \begin{bmatrix} f(\mathbf{x}_1) & \ldots & f(\mathbf{x}_n) \end{bmatrix}^\top$ from our estimated utility function $f(\cdot)$. Denote the (normalised) difference in utility for the i^{th} comparison by

$$Z_i := \frac{f(\mathbf{x}_{Bi}) - f(\mathbf{x}_{Ai})}{\sqrt{2}\sigma_{\text{noise}}} \tag{4}$$

The estimate of $\mathbf{f}$ is obtained by maximum a posteriori (MAP) estimation. This estimator is a function of the data, denoted by $\mathbf{f}_{\text{MAP}}(\mathcal{D})$. In what follows, this dependency will be suppressed for brevity of notation. Given a kernel function $k(x, x')$ and prior mean function $m(x)$, the MAP estimate via Gaussian process regression is

$$\mathbf{f}_{\text{MAP}} = \text{argmin}_{\mathbf{f}} \left\{ -\sum_{i=1}^{M} \log \Phi(Z_i) + \frac{1}{2} (\mathbf{f} - \mathbf{m}(X))^\top \mathbf{K}^{-1} (\mathbf{f} - \mathbf{m}(X)) \right\} \tag{5}$$

where $X := \begin{bmatrix} \mathbf{x}_1 & \ldots & \mathbf{x}_n \end{bmatrix}^\top$, $\mathbf{m}(X) := \begin{bmatrix} m(\mathbf{x}_1) & \ldots & m(\mathbf{x}_n) \end{bmatrix}^\top$, $\mathbf{K}$ is the $n \times n$ Gram matrix of $\mathbb{X}$ using the kernel $k(\cdot, \cdot)$ and $\Phi(\cdot)$ is the cumulative distribution function of the standard Gaussian. Note that the optimisation problem in (5) is strictly convex and twice differentiable (as shown in [14]) and therefore can be solved in a straightforward manner using Newton-like methods. Once $\mathbf{f}_{\text{MAP}}$ has been obtained, an approximate predictive posterior mean (via a Laplace approximation [19]) at a test location x_* is given by

$$\bar{f}_*(x_*) = m(x_*) + \mathbf{k}_*^\top \mathbf{K}^{-1} (\mathbf{f}_{\text{MAP}} - \mathbf{m}(X)) \tag{6}$$

where $\mathbf{k}(X, x) := \begin{bmatrix} k(\mathbf{x}_1, x) & \ldots & k(\mathbf{x}_n, x) \end{bmatrix}^\top$ and $\mathbf{k}_* := \mathbf{k}(X, x_*)$. As illustrated in Figure 3 from Section IV, this approach generally does not guarantee monotonicity of the learned function.

III. PREFERENCE LEARNING WITH MONOTONICITY CONSTRAINTS

In this section we present the main contributions of the paper.

A. Maximum Likelihood of a Linear Utility Function

By following an empirical Bayes approach (in which a Bayesian prior is obtained from data), a maximum likelihood estimator (MLE) is proposed to obtain a prior mean function $m(x)$. Let $\psi_i : \mathcal{X}_i \to \mathbb{R}$, $i = 1, \ldots, d$ be strictly monotonic functions, and define the operator $\boldsymbol{\psi} : \mathcal{X} \to \mathbb{R}^d, \boldsymbol{\psi}(x) := (\psi_1(x_1), \ldots, \psi_d(x_d))$. We can then construct a model of the utility function linear in the basis $\boldsymbol{\psi}(x)$:

$$g(x) = \beta^\top \boldsymbol{\psi}(x) \tag{7}$$

where β is the vector of parameters to be estimated. By applying the rating model from Assumption 3, the maximum likelihood estimate of β under the monotonicity hypothesis 2 involves solving the constrained problem

$$\widehat{\beta} = \underset{\beta}{\text{argmin}} \left\{ -\sum_{i=1}^{M} \log \Phi \left(\frac{\beta^\top \boldsymbol{\psi}(\mathbf{x}_{Bi}) - \beta^\top \boldsymbol{\psi}(\mathbf{x}_{Ai})}{\sqrt{2}\sigma_{\text{noise}}} \right) \right\} \quad \text{s.t. } \beta_j > 0, \forall j \in \mathcal{J} \tag{8}$$

Using a linear-in-basis form for the utility function is the usual approach taken in discrete choice theory [12], however as we later demonstrate, by using this form only as a prior, the overall estimate can be improved upon.

Remark 4: The convexity of (8) is readily shown using the log-concavity of the Gaussian distribution [20]. Twice differentiability also holds and Newton-like methods can be used to solve (8).

Remark 5: We can fix $\sigma_{\text{noise}} > 0$ without loss of generality as this only affects the scale of the learned utility function (as in [10]). To illustrate, notice that scaling both $g(\cdot)$ and σ_{noise} from Assumption 3 by the same positive constant will not change the distribution of $\mathcal{D}$. The learning problem is unaffected as we are only concerned with learning an ordinal utility function. So for convenience, we may choose $\sigma_{\text{noise}} = 1/\sqrt{2}$.

Remark 6: In numerical implementation of (8), it may be required to enforce monotonicity with the constraint $\beta_j \geq \varrho$ with some small $\varrho > 0$.

After $\widehat{\beta}$ has been obtained, we can express the estimate of the vector of latent utilities as $\Psi(X)\widehat{\beta}$, where $\Psi(X) := \begin{bmatrix} \boldsymbol{\psi}(\mathbf{x}_1) & \ldots & \boldsymbol{\psi}(\mathbf{x}_n) \end{bmatrix}^\top$. We also choose the prior mean function $m(x) = \widehat{\beta}^\top \boldsymbol{\psi}(x)$ and our proposed utility function estimate takes the form

$$\widehat{f}_*(x_*) = \widehat{\beta}^\top \boldsymbol{\psi}(x_*) + \mathbf{k}_*^\top \Omega \mathbf{z} \tag{9}$$

with $\Omega := \mathbf{K}^{-1}$, $\mathbf{z} := \mathbf{y} - X\widehat{\beta}$, and $\mathbf{y}$ is a choice of vector for the latent utilities. Note that this closely resembles the form of the approximate predictive posterior mean in (6).

B. Monotonicity Conditions on Gaussian Process Regression

In this section we state conditions for strict monotonicity of the utility function estimate. As it produces differentiable sample functions of both the prior and posterior, the kernel function being considered is the squared exponential kernel $k(x, x') = \sigma^2 \exp\left[-\frac{1}{2} (x - x')^\top \Lambda^{-1} (x - x') \right]$, where $\Lambda^{-1} = \text{diag}\{\ell_1, \ldots, \ell_d\}$ and $\sigma > 0$, $\ell_1, \ldots, \ell_d > 0$ are hyperparameters. We focus on conditions pertaining to the predictive function (9). Let $\left[\frac{\partial \mathbf{k}_*^\top}{\partial x_*} \right]_j$ denote the j^{th} row of the $d \times n$ matrix $\frac{\partial \mathbf{k}_*^\top}{\partial x_*}$. The predictive function (9) is strictly monotonic at x in direction j if

$$\widehat{\beta}_j \left. \frac{\partial \psi_j(x_j)}{\partial x_j} \right|_{x_j = x_{*j}} + \left[\frac{\partial \mathbf{k}_*^\top}{\partial x_*} \right]_j \Omega \mathbf{z} > 0 \tag{10}$$

For the squared exponential kernel and a choice of prior mean function affine in x (ie. $m(x) = \widehat{\beta}^\top x$), the condition

(10) becomes

$$0 < \widehat{\beta} - \Lambda^{-1} \left[\begin{array}{ccc} (x_* - \mathbf{x}_1)\, k(x_*, \mathbf{x}_1) & \ldots & (x_* - \mathbf{x}_n)\, k(x_*, \mathbf{x}_n) \end{array} \right] \Omega \mathbf{z} \quad (11)$$

C. Monotonicity Constrained Estimates

We propose an approach to conduct preference learning from pairwise comparisons with a guarantee of strict monotonicity along any amount of directions as desired. For simplicity, the form of the prior mean function is chosen to be affine in x. For the condition in (11), observe that when $\widehat{\beta}_j > 0$, the inequality will eventually be satisfied as $\mathbf{z} \to 0$ (or equivalently as $\mathbf{y} \to X\widehat{\beta}$). We state the strict monotonicity guarantee formally in Theorem 1.

Theorem 1: With choice of prior mean function affine in x, let $\mathbf{f}_{\text{lin}} = X\widehat{\beta}$ as obtained from (8) and $\mathbf{f}_{\text{MAP}}$ as from (5). There exists an interval $(\alpha^*, 1]$ with

$$\alpha^* = \max_{j \in \mathcal{J}} \left\{ \frac{\widehat{\beta}_j}{-\widehat{\beta}_j + \gamma_j} \right\} + 1 \quad (12)$$

$$\gamma_j := \min \left\{ 0, \inf_{x \in \mathcal{X}} \left\{ \left[\frac{\partial \mathbf{k}(X, x)^\top}{\partial x} \right]_j \Omega \left(\mathbf{f}_{\text{MAP}} - X\widehat{\beta} \right) \right\} + \widehat{\beta}_j \right\} \quad (13)$$

such that for all $\alpha \in (\alpha^*, 1]$, using the convex combination $\mathbf{y} = \alpha \mathbf{f}_{\text{lin}} + (1-\alpha)\, \mathbf{f}_{\text{MAP}}$ in (9) satisfies (10) for all $j \in \mathcal{J}$ over all $x \in \mathcal{X}$.

Proof: We have strict monotonicity in direction j for all $x \in \mathcal{X}$ if

$$\widehat{\beta}_j + \inf_{x \in \mathcal{X}} \left\{ \left[\frac{\partial \mathbf{k}(X, x)^\top}{\partial x} \right]_j \Omega \left(\mathbf{y} - X\widehat{\beta} \right) \right\} > 0 \quad (14)$$

For ease of notation, define $\eta_j(x) := \left[\frac{\partial \mathbf{k}(X,x)^\top}{\partial x} \right]_j \Omega \left(\mathbf{f}_{\text{MAP}} - X\widehat{\beta} \right)$. Starting from the monotonicity condition (14) and using $\mathbf{y} = \alpha X \widehat{\beta}_j + (1-\alpha)\, \mathbf{f}_{\text{MAP}}$, rearranging for α in the case when $\widehat{\beta}_j + \inf_{x \in \mathcal{X}} \{\eta_j(x)\} \leq 0$ yields

$$\alpha > \frac{\widehat{\beta}_j}{\inf_{x \in \mathcal{X}} \{\eta_j(x)\}} + 1 \quad (15)$$

while noting that $\widehat{\beta}_j > 0$ so $\inf_{x \in \mathcal{X}} \{\eta_j(x)\} < 0$. Let $\alpha_j^* = \widehat{\beta}_j / \inf_{x \in \mathcal{X}} \{\eta_j(x)\} + 1 \geq 0$, which satisfies (15) for all $x \in \mathcal{X}$ if $\alpha \in \left(\alpha_j^*, 1\right]$. For the case when $\widehat{\beta}_j + \inf_{x \in \mathcal{X}} \{\eta_j(x)\} > 0$, it is enough to let $\alpha_j^* = 0$. To combine both cases we write

$$\alpha_j^* = \frac{\widehat{\beta}_j}{-\widehat{\beta}_j + \gamma_j} + 1 \quad (16)$$

where γ_j is given in (13). Then $\alpha^* = \max_{j \in \mathcal{J}} \alpha_j^*$ finds the value such that for all $\alpha \in (\alpha^*, 1]$, strict monotonicity is satisfied in all directions $j \in \mathcal{J}$ over all $x \in \mathcal{X}$, yielding (12). ∎

This result shows that we can find values of α (and hence $\mathbf{y}$) which will satisfy the monotonicity constraint. A value of $\alpha = 1$ is the most conservative, resulting in the predictive function being identical to the prior mean, but is a value that always ensures the monotonicity constraints are satisfied. However, there is merit in choosing an α as low as possible (whilst satisfying monotonicity constraints), as indicated by Corollary 1.

Theorem 2: Use $J(\mathbf{f})$ to denote the cost function in (5). Denote $\mathbf{y}_\alpha := \alpha \mathbf{f}_{\text{lin}} + (1-\alpha)\, \mathbf{f}_{\text{MAP}}$, and define $Z_{i,\text{MAP}}$, $Z_{i,\alpha}$, $Z_{i,\text{lin}}$ as in (4) using the vectors $\mathbf{f}_{\text{MAP}}$, $\mathbf{y}_\alpha$ and $\mathbf{f}_{\text{lin}}$ respectively. Suppose $\mathbf{f}_{\text{MAP}} \neq \mathbf{f}_{\text{lin}}$. Then the negative log likelihoods (or equivalently, the Kullback Leibler divergences from the empirical distribution of comparisons) satisfy

$$-\sum_{i=1}^{M} \log \Phi(Z_{i,\text{MAP}}) < -\sum_{i=1}^{M} \log \Phi(Z_{i,\alpha}) < -\sum_{i=1}^{M} \log \Phi(Z_{i,\text{lin}}) \quad (17)$$

for any $\alpha \in (0,1)$. If $\mathbf{f}_{\text{MAP}} = \mathbf{f}_{\text{lin}}$ then (17) holds but with equality.

Proof: The case with $\mathbf{f}_{\text{MAP}} = \mathbf{f}_{\text{lin}}$ is trivial. For the case $\mathbf{f}_{\text{MAP}} \neq \mathbf{f}_{\text{lin}}$, use by definition $\min_{\mathbf{f}} J(\mathbf{f}) = J(\mathbf{f}_{\text{MAP}}) < J(\mathbf{f}_{\text{lin}})$ along with the strict convexity property of $J(\mathbf{f})$ to establish $J(\mathbf{f}_{\text{MAP}}) < J(\mathbf{y}_\alpha) < J(\mathbf{f}_{\text{lin}})$ for $\alpha \in (0,1)$. Then apply the fact that the weighted norm $\left\| \mathbf{y}_\alpha - X\widehat{\beta} \right\|_{\mathbf{K}} = (1-\alpha) \left\| (\mathbf{f}_{\text{MAP}} - \mathbf{f}_{\text{lin}}) \right\|_{\mathbf{K}} \geq 0$ is strictly monotonically decreasing in α for $\alpha \in (0,1)$. ∎

Remark 7: It is not trivial to find a class of problems such that there is a reasonable probability that $\mathbf{f}_{\text{MAP}} = \mathbf{f}_{\text{lin}}$. To do so requires investigating the intersection of the supports for the sampling distributions of $\mathbf{f}_{\text{MAP}}$ and $\mathbf{f}_{\text{lin}}$, which may or may not be the empty set. The supports of the sampling distributions themselves have cardinality in the order of 2^M, so even if the intersection is not the empty set, the probability should still be very small for reasonably large M. Hence we argue that for most problems of interest (ie. where M is not small), the probability that $\mathbf{f}_{\text{MAP}} = \mathbf{f}_{\text{lin}}$ will either be zero or can be considered negligibly small.

Corollary 1: If $\mathbf{f}_{\text{MAP}} \neq \mathbf{f}_{\text{lin}}$, then for all $\alpha' < \alpha$

$$-\sum_{i=1}^{M} \log \Phi(Z_{i,\alpha'}) < -\sum_{i=1}^{M} \log \Phi(Z_{i,\alpha}) \quad (18)$$

The importance of results (17) and (18) is the revelation that we should choose α as low as possible (whilst satisfying monotonicity constraints, ie. $\alpha > \alpha^*$) and that the resulting empirical fit as measured by the likelihood will be an improvement over using $\mathbf{f}_{\text{lin}}$. A procedure based on this principle which finds a weighting between the linear and MAP estimates to guarantee strict monotonicity is described in Algorithm 1. As for how choice of α affects generalisation performance of the utility estimate on unseen examples, the theoretical investigation of this is out of scope for the paper since it is the focus of ongoing work. However, this is tested on a simulation case study in Section IV.

Algorithm 1 Preference Learning with Strict Monotonicity Constraints

Require: Data set $\mathcal{D}$, minimal distinct items matrix X, monotonicity constraint index set $\mathcal{J}$

Choose hyperparameters $\sigma > 0$, $\sigma_{\text{noise}} = 1/\sqrt{2}$, $\ell_1, \ldots, \ell_d > 0$
Choose $\epsilon \in (0, 1]$
Obtain estimate $\widehat{\beta}$ via (8) for the model $\beta^\top x$
Choose $a_1 > 0$
$\widehat{\beta} \leftarrow a_1 \widehat{\beta}$, $\sigma_{\text{noise}} \leftarrow a_1 \sigma_{\text{noise}}$ ▷ Normalise to a nominated scale
$\mathbf{f}_{\text{lin}} \leftarrow X\widehat{\beta}$
Obtain $\mathbf{f}_{\text{MAP}}$ via (5) using the prior mean $\widehat{\beta}^\top x$
$\alpha \leftarrow \alpha^*$ from (12)
$\alpha \leftarrow \min\{\alpha + \epsilon, 1\}$ ▷ For strict monotonicity
$\mathbf{y} \leftarrow \alpha \mathbf{f}_{\text{lin}} + (1 - \alpha)\, \mathbf{f}_{\text{MAP}}$
Compute the estimated utility function with (9)

IV. SIMULATION CASE STUDIES

In this section we demonstrate and validate findings from Section III using simulations on the following case study. Suppose the agent's true utility function can be specified by

$$q : [0,1] \times [0,1] \to \mathbb{R}, q(x) = \mathbf{\Phi}\left(x; \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix}, \begin{bmatrix} 0.07 & 0 \\ 0 & 0.05 \end{bmatrix}\right) \tag{19}$$

where we designate x_1 as negated and normalised overshoot, x_2 as negated and normalised settling time, and $\mathbf{\Phi}(x; \boldsymbol{\mu}, \Sigma)$ denotes the (monotonic) multivariate cumulative density function of $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ in x. A contour plot of the function is shown in Figure 1. An evenly spaced grid of 25 (5×5) points as are selected as the set of distinct items to construct X. To generate $\mathcal{D}$, 90 unique random pairs of points from this grid are sampled (out of a possible 300) and compared according to (3) with $\sigma_{\text{noise}} = 0.1$. Algorithm 1 is then applied with $\mathcal{J} = \{1, 2\}$, the scaling factor a_1 was also chosen such that $\widehat{\beta}^\top \mathbf{1} = 1$, while ϵ was set as 0.01. The kernel hyperparameters were heuristically selected as $\sigma = \sigma_{\text{noise}}$, $\ell_1 = 0.2$, $\ell_2 = 0.2$. With dimension $d = 2$, finding each γ_j in (12) is straightforward via an exhaustive search.

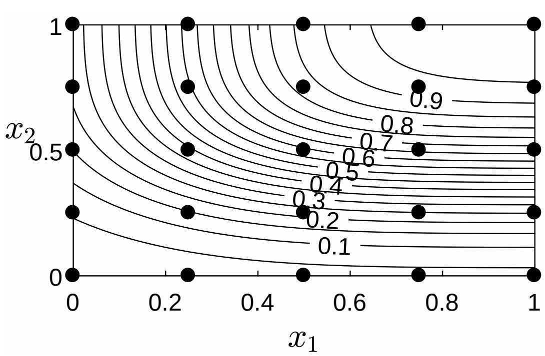

Fig. 1. A countour plot of the utility function used in the data generating process. The grid of points designates the locations used to generate pairwise comparison data.

A. Single Example

The results of a single experiment are depicted graphically in Figures 2, 3, 4. We observe in Figure 3 that the utility function estimate using $\mathbf{f}_{\text{MAP}}$ violates the monotonicity constraints, as indicated from the positively sloped contours in the shaded region (hence at some points the engineer is willing to increase both overshoot and settling time). By using $\mathbf{y}_{\alpha^*+\epsilon}$ as the latent utility vector, Figure 4 shows an estimated utility function which exhibits strict monotonicity.

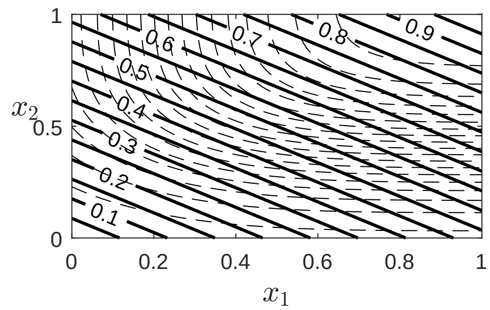

Fig. 2. A countour plot of the estimated utility function with $\mathbf{f}_{\text{lin}}$ (thick lines) along with the original utility function $q(x)$ (dashed lines).

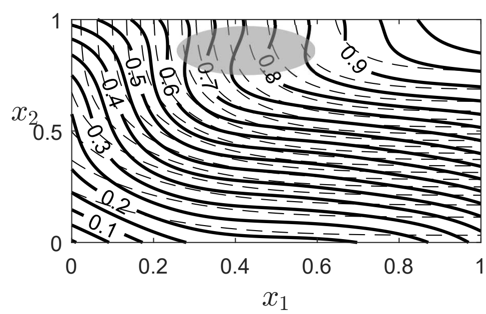

Fig. 3. A countour plot of the estimated utility function with $\mathbf{f}_{\text{MAP}}$ (thick lines) along with the original utility function $q(x)$ (dashed lines). The posterior utility function is not strictly monotonic, as indicated by positively sloped contours (shaded).

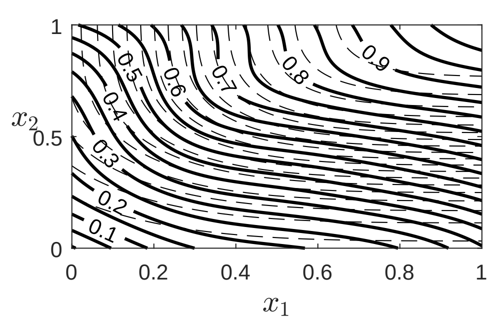

Fig. 4. A countour plot of the estimated utility function with $\mathbf{y}_{\alpha^*+\epsilon}$ (thick lines) along with the original utility function $q(x)$ (dashed lines). The learned utility function now exhibits strict monotonicity.

B. Monte Carlo Simulations

A method is devised to evaluate the performance of the learned utility functions on unseen examples as follows: for an evenly spaced grid of 81 (9×9) points, we find the percentage of correctly predicted preference comparisons by the learned utility function across all possible pairwise comparisons in the grid (3240 in total). We also split these pairs into a subset of monotonic comparisons (1944 pairs), for which the predicted rating is obvious under Assumption 2, and a subset of non-monotonic comparisons (1944 pairs). Table I displays the results from 1000 simulations.

TABLE I

AVERAGE PREDICTION ACCURACY ACROSS 1000 SIMULATIONS

Latent utility vector	$\mathbf{f}_{\text{MAP}}$	$\mathbf{y}_{\alpha^*+\epsilon}$	$\mathbf{f}_{\text{lin}}$
Monotonic subset	98.43%	100%	100%
Non-monotonic subset	89.32%	85.93%	76.76%
Overall	94.79%	94.37%	90.70%

It is of no surprise that the utility estimates from using $\mathbf{y}_{\alpha^*+\epsilon}$ and $\mathbf{f}_{\text{lin}}$ achieve perfect prediction accuracy on monotonic comparisons. Table I and Figure 5 also suggest that for this case study, the hierarchy from (17) and (18) extends to performance on unseen examples, although the accuracy of $\mathbf{f}_{\text{MAP}}$ is very close to the accuracy of $\mathbf{y}_{\alpha^*+\epsilon}$. It is noteworthy that in 212 out of the 1000 simulations, $\mathbf{y}_{\alpha^*+\epsilon}$ achieved a higher overall prediction accuracy than $\mathbf{f}_{\text{MAP}}$. Also in 45 of the simulations, $\alpha^* = 0$ (refer to Figure 6).

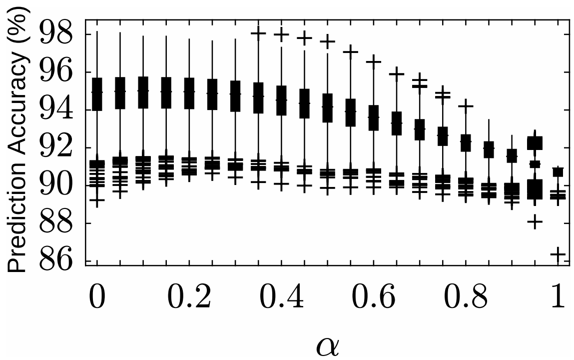

Fig. 5. A boxplot of prediction accuracy in α from 1000 simulations.

V. CONCLUSION

We address the need to learn a utility function from pairwise comparisons with a guarantee of monotonicity, which conventional Gaussian process regression lacks. By specifying a prior function using linear utility functions, it is shown that the proposed algorithm can obtain a higher likelihood compared to the prior whilst satisfying monotonicity constraints. This hierarchy is demonstrated in a numerical case study to translate to higher prediction accuracy on test examples.

MATLAB code for Algorithm 1 and scripts used to produce the results in Section IV are available at `https://github.com/rzch/gp_monotonicity`.

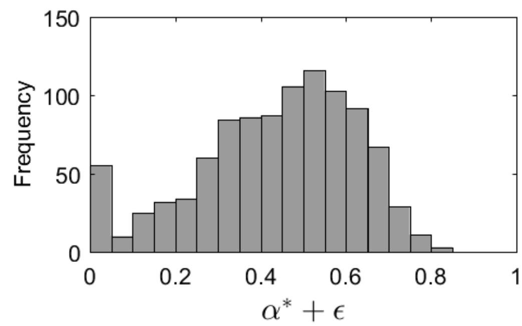

Fig. 6. A histogram for $\alpha^* + \epsilon$ from 1000 simulations.

REFERENCES

[1] A. S. Ira, I. Shames, C. Manzie, R. Chin, D. Nesic, H. Nakada, and T. Sano, "A machine learning approach for tuning model predictive controllers," in *15th International Conference on Control, Automation, Robotics and Vision*, 2018.

[2] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton University Press, 1955.

[3] G. Debreu, "Representation of a preference ordering by a numerical function," in *Decision Process*. John Wiley, 1954.

[4] A. Jern, C. G. Lucas, and C. Kemp, "Evaluating the inverse decision-making approach to preference learning," in *Advances in Neural Information Processing Systems*, 2011.

[5] J. Fürnkranz and E. Hüllermeier, "Preference learning: An introduction," in *Preference Learning*. Springer Berlin Heidelberg, 2010, pp. 1–17.

[6] J. Riihimäki and A. Vehtari, "Gaussian processes with monotonicity information," in *International Conference on Artificial Intelligence and Statistics*, 2010.

[7] N. Barile and A. Feelders, "Active learning with monotonicity constraints," in *SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, apr 2012, pp. 756–767.

[8] K. Tsukida and M. R. Gupta, "How to analyze paired comparison data," University of Washington, Tech. Rep., 2011.

[9] L. L. Thurstone, "A law of comparative judgment," *Psychological Review*, vol. 34, no. 4, pp. 273–286, 1927.

[10] F. Mosteller, "Remarks on the method of paired comparisons: I. the least squares solution assuming equal standard deviations and equal correlations," *Psychometrika*, vol. 16, no. 1, pp. 3–9, mar 1951.

[11] D. McFadden, "Conditional logit analysis of qualitative choice behavior," in *Frontiers in Econometrics*. Academic Press, 1974.

[12] K. E. Train, *Discrete Choice Methods with Simulation*. Cambridge University Press, 2009.

[13] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker, "Label ranking by learning pairwise preferences," *Artificial Intelligence*, vol. 172, no. 16-17, pp. 1897–1916, nov 2008.

[14] W. Chu and Z. Ghahramani, "Preference learning with gaussian processes," in *22nd International Conference on Machine Learning*, 2005.

[15] E. Brochu, N. de Freitas, and A. Ghosh, "Active preference learning with discrete choice data," in *Advances in Neural Information Processing Systems*, 2007.

[16] I. Dewancker, M. McCourt, and S. Ainsworth, "Interactive preference learning of utility functions for multi-objective optimization," in *Future of Interactive Learning Machines Workshop at NIPS*, 2016.

[17] C. Wirth, R. Akrour, G. Neumann, and J. Frnkranz, "A survey of preference-based reinforcement learning methods," *Journal of Machine Learning Research*, vol. 18, no. 136, pp. 1–46, 2017.

[18] H. Hayakawa, "Lexicographic preferences and consumer theory," *Journal of Behavioral Economics*, vol. 7, no. 1, pp. 17–51, jun 1978.

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

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.