IEEE Conference on Decision and Control 2018

THE UNIVERSITY OF MELBOURNE

Gaussian Processes with Monotonicity Constraints for Preference Learning from Pairwise Comparisons

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

[1]School of Engineering, The University of Melbourne
[2]School of Computer Science, University of Birmingham

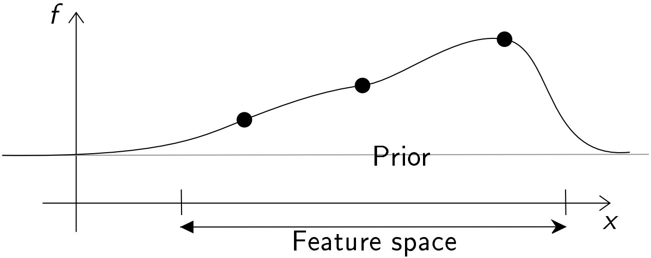
Outline

- Overview
- Motivation
- Review of pairwise preference learning
- Definitions and setup
- Main results
  - Empirical Bayes for prior modelling
  - Monotonicity guarantees
  - Learning algorithm
- Simulation case study
- Summary and future work

Problem Statement

From pairwise comparison data, learn a utility function with monotonicity constraints in desired dimensions.

- Chu/Ghahramani (2005), Gaussian process regression from pairwise comparisons:



- Our approach: introduce two latent utility estimates $\mathbf{f}_{\mathrm{MAP}}$, $\mathbf{f}_{\mathrm{lin}}$.
- Find a convex combination between these which satisfies monotonicity constraints.

Motivation

- *Why preference learning?*
  - Difficult-to-obtain utility or cost functions (requires domain knowledge or expertise).
- *Why pairwise comparisons?*
  - Numeric ratings susceptible to a 'drift effect'.
- *Why monotonicity?*
  - Model features that are desirable.
  - Prior regularisation and for reducing data requirements.

- Psychometrics (1920s)
- Discrete choice theory in economics (1970s).
- Learning-to-rank algorithms (1990s), eg. Google PageRank.
- Chu/Ghahramani (2005)
- Riihimäki/Vehtari (2010)
- Akrour et. al. (2012) & DeepMind/OpenAI (2017)

THE UNIVERSITY OF MELBOURNE

Definition (Ordinal utility functions)

- Ordinal utility function $h : \mathcal{X} \to \mathbb{R}$.
- Represents underlying preferences such that $x_A \preceq x_B \Leftrightarrow h(x_A) \leq h(x_B)$.
- Infinitely many ordinal utility functions for the same preferences.

Definition (Monotonic preferences)

Strict monotonicity at x in dimension j:

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

Weak monotonicity is defined analogously.

- Focus on case where $\mathcal{X}$ is compact subset of $\mathbb{R}^d_{\geq 0}$.

- 'Library' of n distinct items $\mathbb{X} \in \mathcal{X}^n$.

Assumption

(Rating model) The user generates comparisons between $\mathbf{x}_A, \mathbf{x}_B \in \mathbb{X}$ *using the data generating process*

$$v(\mathbf{x}_A) := g(\mathbf{x}_A) + \varepsilon_A$$
$$v(\mathbf{x}_B) := g(\mathbf{x}_B) + \varepsilon_B$$

- User rates $\mathbf{x}_B$ preferred over $\mathbf{x}_A$ when $v(\mathbf{x}_B) > v(\mathbf{x}_A)$.
- $g(\cdot)$ is the underlying utility function.
- $\varepsilon_A, \varepsilon_B \sim \mathcal{N}\left(0, \sigma^2_{\text{noise}}\right)$ are i.i.d. rating noise.
- Noise models inaccuracy in judgement or multiple users.

Maximum Likelihood Estimate of Utilities

- Linear model of the utility function $g(x) = \beta^\top x$.
- Can treat x as being lifted from an original feature space through strictly monotonic transformations.
- Constrained maximum likelihood from M pairwise comparisons:

$$\widehat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ -\sum_{i=1}^{M} \log \Phi \left(\frac{\beta^\top \mathbf{x}_{Bi} - \beta^\top \mathbf{x}_{Ai}}{\sqrt{2}\sigma_{\text{noise}}} \right) \right\}$$

$$\text{s.t. } \beta_j > 0, \forall j \in \mathcal{J}$$

  - $\mathcal{J} \subseteq \{1, \ldots, d\}$: dimensions desired with monotonicity.
  - Convex problem.
  - σ_{noise} can be absorbed into β.
- This gives what we want, but can we do better?

- Use $\widehat{\beta}^\top x$ as a prior mean in Gaussian process regression.
- Based on the Laplace approximation of posterior, the utility estimate $\bar{f}_* (x_*)$ at test point x_* takes the form:

$$\bar{f}_* (x_*) = \widehat{\beta}^\top x_* + \mathbf{k}_*^\top \mathbf{K}^{-1} (\mathbf{y} - \mathbf{f}_{\mathrm{lin}})$$

  - $X \in \mathbb{R}^{n \times d}$: matrix containing distinct items from $\mathbb{X}$.
  - $\mathbf{f}_{\mathrm{lin}} := X\widehat{\beta}$
  - $\mathbf{y} \in \mathbb{R}^n$: latent vector of utilities for points in $\mathbb{X}$.
  - $\mathbf{k}_*$, $\mathbf{K}$: Gram matrices from Gaussian process kernel $k(\cdot, \cdot)$.
- One choice of $\mathbf{y}$ is the maximum a posteriori estimate:

$$\mathbf{f}_{\mathrm{MAP}} = \mathrm{argmin}_{\mathbf{f}} \left\{ -\log \mathcal{L}(\mathbf{f}) + \frac{1}{2} \|\mathbf{f} - \mathbf{f}_{\mathrm{lin}}\|^2_{\mathbf{K}^{-1}} \right\}$$

- However, this does not guarantee monotonicity of $\bar{f}_* (x_*)$.

- Consider squared exponential kernel:

$$k\left(x, x'\right) = \sigma^2 \exp\left[-\frac{1}{2}\left(x - x'\right)^\top \Lambda^{-1}\left(x - x'\right)\right]$$

  - σ, Λ are hyperparameters.
  - Produces smooth sample paths of the Gaussian process.
- Condition for strict monotonicity of $\bar{f}_*\left(x_*\right)$ in dimension j:

$$\widehat{\beta}_j + \left[\frac{\partial \mathbf{k}\left(X, x\right)^\top}{\partial x}\right]_j \mathbf{K}^{-1}\left(\mathbf{y} - \mathbf{f}_{\text{lin}}\right) > 0, \forall x \in \mathcal{X}$$

  - Derivative tends to $\widehat{\beta}_j > 0$ as $\mathbf{y} \to \mathbf{f}_{\text{lin}}$.

Theorem

There exists an interval $(\alpha^*, 1]$ *with*

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

$$\gamma_j := \min \left\{ 0, \inf_{x \in \mathcal{X}} \left\{ \left[\frac{\partial \mathbf{k}\left(X, x\right)^\top}{\partial x} \right]_j \mathsf{K}^{-1} \left(\mathbf{f}_{MAP} - \mathbf{f}_{\mathrm{lin}} \right) \right\} + \widehat{\beta}_j \right\}$$

where for all $\alpha \in (\alpha^*, 1]$, *choosing* $\mathbf{y} = \alpha \mathbf{f}_{\mathrm{lin}} + (1 - \alpha)\, \mathbf{f}_{\mathrm{MAP}}$ *satisfies monotonicity constraints over the feature space.*

- How should α be chosen?

Choice of α

Theorem

Suppose $\mathbf{f}_{\mathrm{MAP}} \neq \mathbf{f}_{\mathrm{lin}}$. Then the negative log likelihoods satisfy for any $\alpha \in (0,1)$:

$$-\log \mathcal{L}\left(\mathbf{f}_{\mathrm{MAP}}\right) < -\log \mathcal{L}\left(\alpha \mathbf{f}_{\mathrm{lin}} + (1-\alpha)\,\mathbf{f}_{\mathrm{MAP}}\right) < -\log \mathcal{L}\left(\mathbf{f}_{\mathrm{lin}}\right)$$

Corollary

For all $\alpha' < \alpha$:

$$-\log \mathcal{L}\left(\alpha' \mathbf{f}_{\mathrm{lin}} + (1-\alpha')\,\mathbf{f}_{\mathrm{MAP}}\right) < -\log \mathcal{L}\left(\alpha \mathbf{f}_{\mathrm{lin}} + (1-\alpha)\,\mathbf{f}_{\mathrm{MAP}}\right)$$

- Should choose α as low as possible whilst satisfying monotonicity constraints.

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

1: Obtain estimate $\widehat{\beta}$ via MLE
2: $\mathbf{f}_{\text{lin}} \leftarrow X\widehat{\beta}$
3: Choose hyperparameters σ, Λ
4: Obtain $\mathbf{f}_{\text{MAP}}$ via MAP using prior mean $\widehat{\beta}^\top x$
5: $\tilde{\alpha} \leftarrow \min\{\alpha^* + \epsilon, 1\}$ with small $\epsilon > 0$
6: $\mathbf{y} \leftarrow \tilde{\alpha}\mathbf{f}_{\text{lin}} + (1 - \tilde{\alpha})\,\mathbf{f}_{\text{MAP}}$
7: Estimate utility function with $\bar{f}_*(x_*) = \widehat{\beta}^\top x_* + \mathbf{k}_*^\top \mathbf{K}^{-1}(\mathbf{y} - \mathbf{f}_{\text{lin}})$

Case Study

- 2 dimensional example with features x_1, x_2 over $[0, 1] \times [0, 1]$.
- $\mathcal{J} = \{1, 2\}$
- Randomly generate 90 comparisons from the grid.

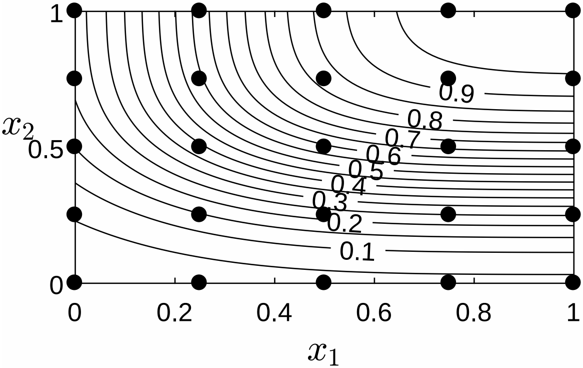


Figure: Contour plot of underlying utility.

- Using $\mathbf{f}_{\mathrm{MAP}}$ as the latent utility vector, the utility estimate does not satisfy monotonicity constraints.

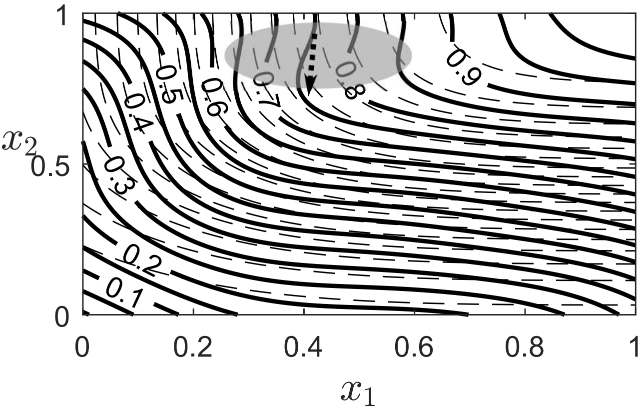


Figure: Thick line: utility estimate.

- Using $\mathbf{y}_{\tilde{\alpha}} := \tilde{\alpha}\mathbf{f}_{\text{lin}} + (1 - \tilde{\alpha})\,\mathbf{f}_{\text{MAP}}$ as the latent utility vector, the utility estimate does satisfy monotonicity constraints.

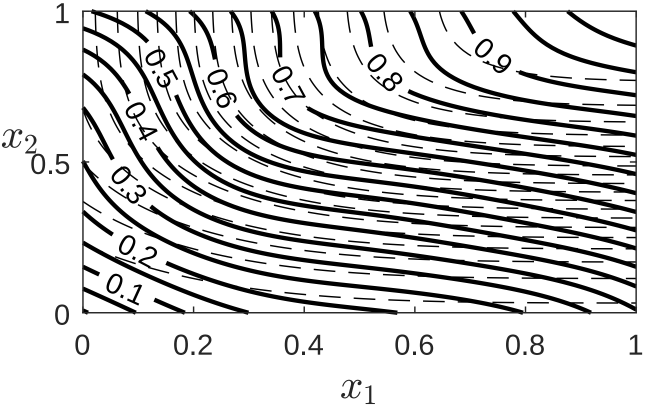


Figure: Thick line: utility estimate.

Results

- 1000 Monte-Carlo simulations.
- Validation on seen and unseen pairs.

Table: Average prediction accuracy

	$\mathbf{f}_{\text{MAP}}$	$\mathbf{y}_{\tilde{\alpha}}$	$\mathbf{f}_{\text{lin}}$
Dominated pairs	98.43%	100%	100%
Non-dominated pairs	89.32%	85.93%	76.76%
Overall	94.79%	94.37%	90.70%

- Same hierarchy as log likelihoods.

Summary and Future Work

- Contributions
  - Selection of monotonic prior
  - Monotonicity guarantees
  - Learning algorithm
- Future work
  - Hyperparameter selection
  - Confidence estimates
  - Scalability
  - Generalisation error
  - Application in control

Code available at `https://github.com/rzch/gp_monotonicity`