Efficient Learning of Trajectory Preferences Using Combined Ratings and Rankings

Thane Somers, Nicholas R. J. Lawrance and Geoffrey A. Hollinger
Robotics Program, School of Mechanical, Industrial & Manufacturing Engineering
Oregon State University, Corvallis, OR 97331 USA
Email: {somersth, nicholas.lawrance, geoff.hollinger}@oregonstate.edu

***Abstract*—In this paper we propose an approach for modeling and learning human preferences using a combination of absolute (querying an expert for a numerical value) and relative (asking the expert to select the highest-value option from a set) queries. Our approach uses a Gaussian process regression model with an associated likelihood function that can take into account both pairwise preferences and numerical ratings to approximate the user's latent value function from a set of (noisy) queries. We show that using a combination of relative and absolute queries performs better than either query type alone and propose a simple active learning approach to sequentially select informative queries that speed up the learning process when searching for high-value regions of the user's latent value space. We demonstrate the effectiveness of our method on a 1-D function approximation task and on a simulated autonomous surface vehicle performing a lake monitoring mission. These experiments show that our algorithm is able to efficiently learn an operator's mission preferences and use those mission preferences to autonomously plan trajectories that fulfill the operator's goals.**

I. Introduction

Current methods for performing robotic environmental monitoring place a high mental, physical, and time burden on human operators. Reducing this burden requires increasing levels of autonomy. In addition to adapting to and withstanding dangerous, dynamic, and unstructured environments, attaining a greater level of autonomy requires that a robot have a complete picture of the operator's preferences and goals.

Due to environmental complexities, it can be difficult and time-consuming for operators to build controllers for autonomous vehicles performing complex tasks. Furthermore, many tasks do not have a single goal, but rather involve a trade-off between multiple objectives. For instance, an autonomous robot monitoring an ocean environment is required to observe multiple ecological variables while avoiding strong currents and obstacles, all with limited endurance and communication.

With current levels of autonomy, a team of trained experts is required to deploy and operate these robots for the duration of each of these missions. By improving the robot's understanding of the operator's goals, we can reduce this high mental (and often physical) burden on the operators while simultaneously increasing the robot's ability to adapt to unexpected environmental conditions.

Several methods have been proposed to allow the robot to learn and model the expert's goals [20]. With an accurate model of the expert's preferences, the robot can plan actions for itself, leveraging the large amount of research into planning and optimization [11]. By automatically building a controller based on a reward function learned from an expert, the time-consuming complexity of manually programming a controller is removed. Ultimately, these methods combine the expert's domain knowledge with the robot's ability to gather and adapt to new data. However, current approaches, such as learning from demonstration, have several disadvantages. They require the expert to manually provide demonstrations of robot trajectories and are susceptible to noise.

Fig. 1: Commanding aquatic robots, such as the pictured Platypus Lutra autonomous surface vehicle, is currently a slow, hands-on process where operators manually set individual waypoints at the outset of the mission. This burden motivates the need for algorithms that can quickly learn and meet a human operator's goals while adapting to the dynamic aquatic environment.

We propose a novel method that combines absolute and relative queries to learn a Gaussian process (GP) representation of the user's latent reward function over the mission objective space. This method has several advantages over existing frameworks. First, combining the specificity of absolute trajectory ratings with the exploration value of a relative ranking of several trajectories speeds the learning process. Additionally, actively selecting between absolute and relative queries maximizes the information gained from each question. Furthermore, since our approach represents the learned preference function as a GP in the high-level objective space, it is able to incorporate domain knowledge and is resistant to noisy, non-linear inputs.

With experiments in non-linear function approximation and

on simulated lake monitoring trials, we show that our proposed method efficiently learns an expert's preferences and can use those preferences to plan trajectories that meet the expert's goals. Furthermore, we demonstrate the value of adaptively selecting rating and ranking queries in reducing the effort required of the robot operator.

II. Related Work

Learning from demonstration (LfD) methods are often used to learn an operators preferences from a set of human driven demonstrations of near-optimal robot behavior [2]. One of the most well-known LfD algorithms is inverse reinforcement learning (IRL) [1]. IRL assumes that the human is acting as a Markov decision process providing optimal demonstrations, then attempts to find the reward function that matches the policy presented in the user's demonstrations. Another common LfD method is coactive learning, which learns a user's reward function from improvements the user makes to example solutions for a problem. Studied representations of the reward functions include a linear scaling over the features of the proposed control plan [19] and a probabilistic distribution over this linear scaling [20]. These LfD methods often assume the demonstrations are optimal, can perform poorly when the expert's preferences are non-linear [20], and require time-consuming complete demonstrations.

Learning and modeling user ratings is a well-studied problem across many domains, such as for the well-known Netflix challenge [4]. However, learning ranking models is still an open problem. Several methods have been proposed, including the ELO chess rating system commonly used to rank competitors in multiple games and sports[8]. Rankings have also been incorporated into the latent factor models used in recommender systems [3]. We base our methods for combining rating and ranking queries on previously studied methods for training GPs on ranking inputs [6, 10, 17]. In a robotic handover task, Kupcsik et al. [13] used a similar GP to estimate a human's reward function during policy search. However, they only briefly mention combining query types and do not further discuss the generalizability or limitations of the method.

Active learning methods further improve the convergence rate of supervised learning algorithms. They've been applied across a wide range of domains, including performing IRL in a simple grid world [15], informing reinforcement learning rewards for grasping tasks [7], and selecting poses for underwater inspection [9]. Many of these methods use a form of Bayesian optimization, such as an Upper Confidence Bound (UCB) metric [5]. However, finding the best heuristic has proven challenging, as estimating the value of any given query is highly domain dependent [5]. We demonstrate that actively selecting query types and trajectories provides significant benefit in the user preference domain.

III. Method

In this section, we outline our methods for representing a user's preferences as a GP in the mission objective space and for actively learning that representation using a combination of absolute and relative queries. Our goal is to create a system that efficiently learns a rich representation of the user's preferences by utilizing faster and simpler queries than previous methods while also incorporating domain knowledge to speed up the learning process. Ultimately, this increases the autonomy of the robotic system while simultaneously reducing the human effort required to program and control the robot.

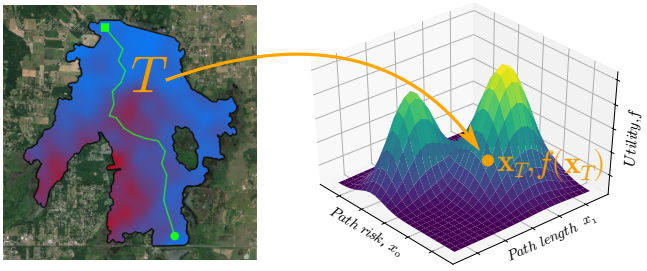


Fig. 2: A trajectory T is mapped to a corresponding objective-space feature vector $\mathbf{x}_T$ (with feature dimensions of path length and accumulated risk). An expert is assumed to have an internal function f that associates an objective-space vector with a scalar utility value $f(\mathbf{x}_T)$.

A. Problem formulation

We assume that the expert has an internal utility model that can associate a trajectory T with a scalar utility value. We represent this model using a fixed objective feature space mapping that maps a trajectory T to a set of k objective values in the form of a vector $\mathbf{x}_T \in \mathbb{R}^k$ in the objective space of the problem. A simple illustration is shown in Fig. 2. The dimensions of the objective space represent features of the trajectory that relate to mission success. For an autonomous underwater vehicle traversing an ocean these could include distance traveled, number of informative samples, and risk measures such as the strength of ocean currents. An example of a set of ratings in the lake monitoring objective space is shown in Fig. 8.

We assume that the user's utility can be modeled as a function $f : \mathbb{R}^k \rightarrow \mathbb{R}$ that maps from a k-dimensional feature space to a scalar utility. The user can be queried about their objective space utility through either absolute or relative queries. For an absolute query, the user is presented with a trajectory and asked to provide a numerical rating $u \in [u_{min}, u_{max}]$, representing a scalar utility measure on a bounded scale.

For relative queries, the user is presented with a set of m trajectories, $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$ and asked to identify the best (highest utility) trajectory $T_j \in \mathcal{T}$. This provides a set of $m-1$ pairwise relationships where the selected trajectory is preferred over each of the other members of the query set, such that $T_j \succ T_i \; \forall \; T_i \in \mathcal{T} \setminus \{T_j\}$, where the trajectory preference relationship is related to the associated utility values, such that $T_j \succ T_i \leftrightarrow f(\mathbf{x}_j) > f(\mathbf{x}_i)$.

The goal is to identify regions of the objective space with high utility using a limited number of queries to the expert. In this work we explore a number of measures to quantify performance, as discussed in the results section. Overall, we

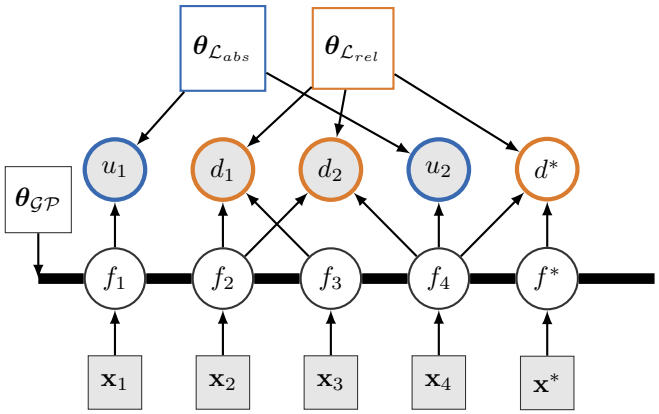


Fig. 3: A graphical model illustration of the problem formulation. Circles represent random variables, boxes are deterministic. Filled shapes are observed. Observations consist of input feature vectors (filled squares $\mathbf{x}$), output absolute utility observations (filled blue circles u) and output pairwise relative relationships (filled orange circles d). The GP (illustrated as the black horizontal bar) generates latent function values (unfilled circles f) conditioned on the hyperparameters of the covariance function θ_k and input locations. The relative and absolute likelihoods, $\mathcal{L}_{rel}$ and $\mathcal{L}_{abs}$, also conditioned on their respective parameters, provide the likelihood of the observations.

are interested in showing that a combination of absolute and relative queries performs better than either query-type alone, and we explore approaches to actively selecting queries.

B. Objective Space Gaussian Process Learning

We estimate the user's preference function f using a GP over the mission objective space of the robot conditioned on relative and absolute queries. Our work draws on a formulation for GP regression that combines absolute and pairwise relative observations into a single modeling framework [17], adapting it to the preference learning domain. The GP predicts the unbounded, unobserved latent function f, from which we use a likelihood function to estimate the probability of observing the input training data. Figure 3 illustrates our approach as a graphical model.

1) Gaussian process latent function: We want to estimate $f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$, the user's (unobserved) latent reward function from a set of observations collected from the user. For ease of notation, we group both absolute and relative queries into a single observation set consisting of input locations $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and output observations $Y = \{d_1, \ldots, d_p, u_1, \ldots, u_q\}$. We assume that each $\mathbf{x} \in X$ is unique but can be referenced by multiple observations (fig. 3). The GP is conditioned on the input observation locations and the GP covariance function hyperparameters $\boldsymbol{\theta}_{\mathcal{GP}}$ (fig. 3). For this work, we use the common squared exponential covariance function

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 exp\left(-\frac{|\mathbf{x} - \mathbf{x}'|^2}{2l^2}\right), \tag{1}$$

with process variance σ_f^2 and length scale l, so $\boldsymbol{\theta}_{\mathcal{GP}} = \{\sigma_f, l\}$. The GP regression is similar to a standard GP regression problem, except that there is no analytic solution for solving for the maximum likelihood posterior. Instead, we use the Laplace approximation that approximates the posterior as a normal distribution, and the system is solved by iteratively searching for the mode of the distribution that maximizes the posterior likelihood,

$$\begin{aligned}\hat{\mathbf{f}} &= \arg\max_{f(X)} p(f(X)|Y) \\ &= \arg\max_{f(X)} p(Y|f(X), \boldsymbol{\theta}_{\mathcal{L}}) p(f(X)|\boldsymbol{\theta}_{\mathcal{GP}}).\end{aligned} \tag{2}$$

Solving for the hyperparameters requires an additional step, repeatedly solving the maximum likelihood $\hat{\mathbf{f}}$ for given hyperparameters, then varying the hyperparameters to maximize the posterior likelihood until both converge. Chu and Ghahramani [6] show that this problem is convex and can be solved using gradient descent. Jensen and Nielsen [17] provide analytic derivatives for the likelihood functions listed below with respect to their hyperparameters.

2) Relative observation likelihood: The formulation for a pairwise preference likelihood function was originally formulated in [6]. We use a *preference* relationship for ranked points, where an input point $\mathbf{x}_i$ is said to be *preferred* over $\mathbf{x}_j$ (written $\mathbf{x}_i \succ \mathbf{x}_j$), if $f(\mathbf{x}_i) \geq f(\mathbf{x}_j)$.

Thus, a relative training point consists of a pair of input points $(\mathbf{x}_i, \mathbf{x}_j)$ and an associated binary observation $d \in \{-1, 1\}$, with $d = -1$ signifying to $\mathbf{x}_i \succ \mathbf{x}_j$ and $d = 1$ the opposite. To incorporate noise, we assume that the observations of f are drawn from i.i.d. Gaussian distributions with fixed variance σ_R^2 around the true function f, and the label d represents which sample is larger. The likelihood $\mathcal{L}_{rel}$ of observing a label can be written

$$\mathcal{L}_{rel}\left(d|f(\mathbf{x}_i), f(\mathbf{x}_j), \sigma_R\right) = \Phi\left(d\frac{f(\mathbf{x}_j) - f(\mathbf{x}_i)}{\sigma_R\sqrt{2}}\right) \tag{3}$$

where $\Phi : \mathbb{R} \to (0, 1)$ is the cumulative distribution function for the normal distribution, $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(\gamma; 0, 1)d\gamma$. There is one hyperparameter of (3), $\boldsymbol{\theta}_{\mathcal{L}_{rel}} = \{\sigma_R\}$.

3) Absolute observation likelihood: To incorporate absolute observations, where the expert is queried about a single trajectory with associated feature vector $\mathbf{x}$ and provides a scalar utility value $u \in [0, 1]$, we use a formulation from [10] where the likelihood is represented by a beta distribution,

$$\mathcal{L}_{abs}\left(u|f(\mathbf{x}), \boldsymbol{\theta}_{\mathcal{L}_{abs}}\right) = \text{Beta}\left(\alpha(\mathbf{x}), \beta(\mathbf{x})\right). \tag{4}$$

The beta distribution provides a probability density over a bounded interval $(0, 1)$, and is parameterized by two shape parameters α and β:

$$\alpha(\mathbf{x}) = \nu\mu_B(\mathbf{x}), \tag{5}$$

$$\beta(\mathbf{x}) = (1 - \nu)\mu_B(\mathbf{x}). \tag{6}$$

Since the GP itself maps onto an unbounded scale ($\mathbb{R}$), we also need a function that links the prediction from the latent function f to the observed utility value u. We adopt an approach proposed in [17] that links the mean of the beta distribution μ_B with the mean of the GP prediction $\hat{f}(\mathbf{x})$ using the common probit mean link function,

$$\mu_B(\mathbf{x}) = \Phi\left(\frac{\hat{f}(\mathbf{x})}{\sigma_B\sqrt{2}}\right). \tag{7}$$

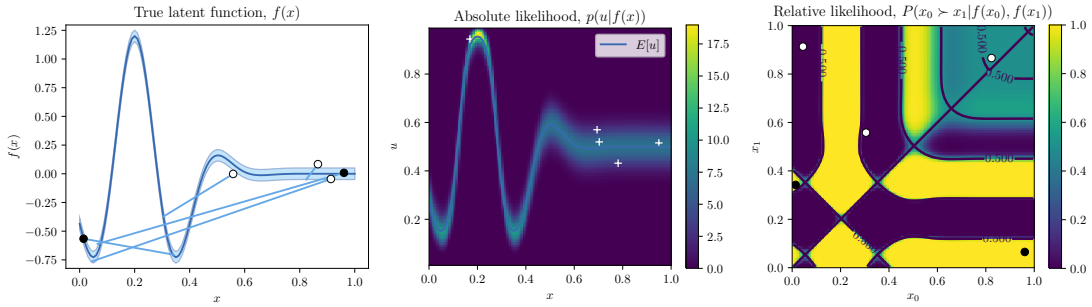


Fig. 4: True latent function, associated likelihoods and samples. The absolute likelihood (center subfigure) illustrates the probability density of u over the domain of $\mathbf{x}$ values, so that each vertical slice of the image is a density $p(u|f(\mathbf{x}))$, with $\sigma_B = 1.0$ and $\nu = 80.0$. The right subfigure shows the likelihood of sampling the class label $d = -1$ ($\mathbf{x}_0 \succ \mathbf{x}_1$) for all pairings of $\mathbf{x}_0, \mathbf{x}_1$ in the domain, with $\sigma_R = 0.1$. The blue lines with circles show pairwise relative samples, where the circle is at the end of the preferred input, and black circles indicate $d = -1$ and white circles $d = 1$. Samples from the absolute function u are shown as white crosses.

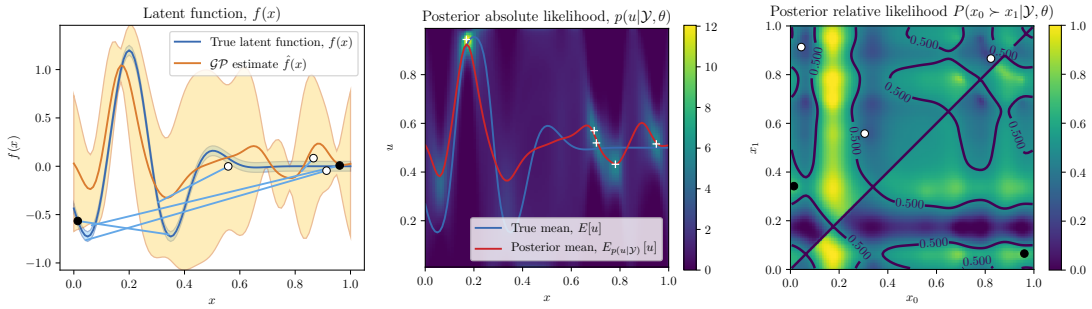


Fig. 5: Posterior estimate with five absolute samples and five pairwise relative samples. The GP estimate of the latent function is shown left with 1σ bounds, and the resulting posterior likelihoods are shown in the center and right subfigures respectively.

The hyperparameters for the absolute likelihood are $\boldsymbol{\theta}_{\mathcal{L}_{abs}} = \{\sigma_B, \nu\}$. σ_B scales how f is mapped to the output range $(0, 1)$, and ν is a precision variable that determines how 'peaky' the beta distribution is.

4) Prediction: For the active learning process, and to identify high-utility paths, we need to be able to generate predictions of both absolute and relative likelihoods from the GP model for unobserved locations X^*. Generating predictions from the GP latent function requires the maximum likelihood solution $\hat{\mathbf{f}}$, and the negative Hessian of the log likelihood W, where

$$W_{i,j} = \sum_k \frac{\partial^2 - \log \mathcal{L}\left(y_i \mid f(\mathbf{x}_i), f(\mathbf{x}_j), \boldsymbol{\theta}_{\mathcal{L}}\right)}{\partial f(\mathbf{x}_i) \partial f(\mathbf{x}_j)}. \tag{8}$$

The latent posterior distribution is $f^* \sim \mathcal{N}(\hat{f}^*, K^*)$, where

$$\hat{f}^* = K_{X,X^*}^T K_{X,X}^{-1} \hat{\mathbf{f}}, \tag{9}$$

$$K^* = K_{X^*,X^*} - K_{X,X^*}^T \left(I + W K_{X,X}\right)^{-1} W K_{X,X^*}. \tag{10}$$

These results can then be used to calculate the output likelihood distributions my marginalizing out $f^*(x)$:

$$p(y^*|\mathbf{x}^*, X, Y) = \int \mathcal{L}(y|f(\mathbf{x}^*), \boldsymbol{\theta}_{\mathcal{L}}) p(f(\mathbf{x}^*)|X, Y) df(\mathbf{x}^*). \tag{11}$$

Figure 4 shows an example of a 'true' one-dimensional latent function, and the resulting sampling likelihood distributions. Figure 5 shows the posterior estimate of the latent function from the GP and resulting posterior likelihood estimates given the training samples from Fig. 4. It is interesting to note the effect of the relative observations which provide general shape information versus the absolute measurements which provide strong estimates of the value of the latent function but only in a small region.

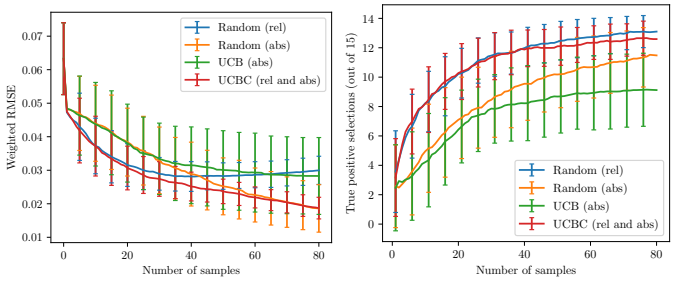
C. Active Selection of Ratings and Rankings

Absolute and relative queries provide different coverage of the objective space. An absolute query learns an accurate utility for a single point while a relative query provides general pairwise comparisons of several points, thus exploring a larger area of the space. Additionally, rankings are intuitively easier to make and users are more confident about their responses [3]. By combining rating and ranking queries, our method is able to make use of the benefits of each.

The trajectories T_i for each query q_i are selected based on the upper confidence bound (UCB) of the GP's estimate of the user's rating for T_i:

$$x_{t+1} = \arg\max_x \hat{f}^*(x) + \gamma \sqrt{K^*(x)} \tag{12}$$

UCB is well suited to our method as it selects trajectories that have both a high level of uncertainty and are also likely to be highly rated [5]. In learning a user's preferences, it is most important for the robot to be confident that it understands which trajectories have high utility. These trajectories are often difficult to learn as they comprise a small, localized portion of the objective space. Thus, once a region of the objective



(a) Weighted RMS error. (b) True positive (out of 15).

Fig. 6: Results for simulated randomized wave function problem. Lines show mean and error bars show one standard deviation over 100 randomized trials.

space has been found to be poorly rated, there is little value in continuing to explore it.

IV. Experiments and Results

A. Randomized trials for active learning

To demonstrate the advantage of combining ratings and rankings in a single framework we compare the performance of different query selection algorithms on a learning task. We generate 'truth' functions as a sum of three random sinusoidal wave features:

$$f(x) = \sum_{k=1}^{3} a_k \cos(\pi f_k (x - o_k)) \exp(-d_k (x - o_k)^2) \quad (13)$$

where the frequency, amplitude, offset and damping are uniformly randomly selected from the intervals $f_k \in [10, 30]$, $a_k \in [0.6, 1.2]$, $o_k \in [0.1, 0.9]$ and $d_k \in [250.0, 350.0]$ respectively. The input space is limited to $x \in [0, 1]$. In each trial instance, we sample a wave function that can be (noisily) queried, and provide one randomly placed absolute training sample as initial data. Each tested method sequentially selects query locations and samples the true function until the maximum number of queries is reached. All methods use the same GP formulation and hyperparameters, and differ only in their active selection algorithm. We compare our proposed UCB Combined method (labeled UCBC in plots) to a random absolute-only sampler that randomly selects a single rating query at each step, a random relative-only sampler that randomly selects five points for a ranking query and a pure UCB method that greedily selects a rating query based on the upper confidence bound. Our experiments used $\gamma = 3$ based on hand tuning for best performance.

To measure performance we use a weighted RMSE (similar to [16]) over a uniformly distributed set of $n = 100$ test points which calculates the RMS error between the predicted absolute rating $\mathbf{u}_{\text{est}}$ and the true absolute rating $\mathbf{u}_{\text{true}}$ weighted by the magnitude of the rating, such that high valued points are weighted higher than low-valued points. We modify the method used in [16] by weighting the squared error by the maximum of the predicted and the actual ratings. This ensures that if the method predicts a high value where the truth is low, or vice-versa, this will adversely affect the performance score:

$$WRMS = \sqrt{\frac{\sum \left((\mathbf{u}_{\text{est}} - \mathbf{u}_{\text{true}})^2 \cdot \max(\mathbf{u}_{\text{est}}, \mathbf{u}_{\text{true}})\right)}{n}}. \quad (14)$$

Figure 6a shows the WRMS for each method over 100 trials.

We are also interested in how well each method would select high-value points given a fixed number of observations. We identified the 15 points with the highest ratings from the 100 uniform samples of each true function, and after each observation selection, we queried each method for their 15 highest rated points to compare to. Figure 6b shows the number of true positive selections. This metric shows methods that correctly identify the high-value areas, but don't necessarily correctly estimate the rating magnitude.

B. Simulated Lake Monitoring Trials

In these experiments, we study the use of our method on a simulated autonomous surface vehicle (ASV) monitoring a lake environment. The ASV must travel from a start location to a goal location while planning a trajectory that balances the distance traveled with the amount of information sampled along the trajectory. The goal is to allow the ASV to autonomously plan its mission trajectories while maintaining the same balance of objectives, distance and information gathered, that the operator would.

The environment consists of a simulated information field over a lake, a diverse set of trajectories across it (Fig. 7), and their associated objective scores (Fig. 8). The information field is generated using a sum of 2D Gaussians with added Perlin noise [18]. The information objective score is calculated as a path integral of the information field along the trajectory. Two motion planners, STOMP [11] and RRT [12], are used to provide path diversity. In order to cover the objective space, the paths are planned using a weighted linear combination of the objectives as a cost function $cost = v_1 + w * v_2$. By varying w, trajectories in different parts of the objective space can be created. 200 paths, 100 from each planner, were generated for each training and test set.

We designed a simulated user that represents a human operator performing an environmental sampling mission. As shown in fig. 8, the user wants to score at least 150 on the information gathering objective. Above that, it attempts to minimize the distance traversed. Given these non-linear user preferences, the utility u of a trajectory is encoded by the following equations:

$$u = \begin{cases} 1, \text{ if information} < 125 \\ 2, \text{ if } 125 < \text{information} < 150 \\ \lceil 5 - (\text{distance} - 450)/50 \rceil, \text{ otherwise.} \end{cases}$$

For an absolute rating query, the user rated the presented trajectory on a five-point Likert scale, with 1 being an unacceptable trajectory and 5 being an excellent trajectory [14]. In a relative ranking query, the user is asked to select the best trajectory from a set of five. These scales and set sizes were selected as they represent a good balance of gaining specific information without requiring lengthy consideration

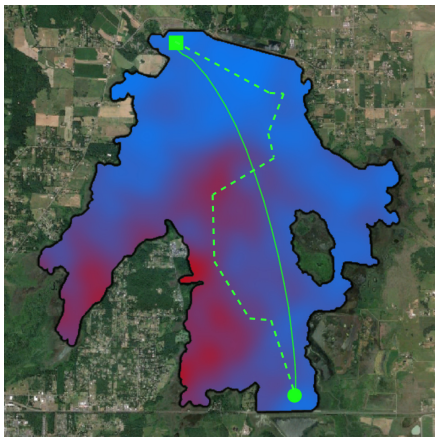


Fig. 7: Two example lake monitoring trajectories from the trajectory pool are shown superimposed over a simulated objective field representing the information gained by traveling over each part of the lake. Red areas represent locations with higher information. The dashed trajectory is longer but gathers more information.

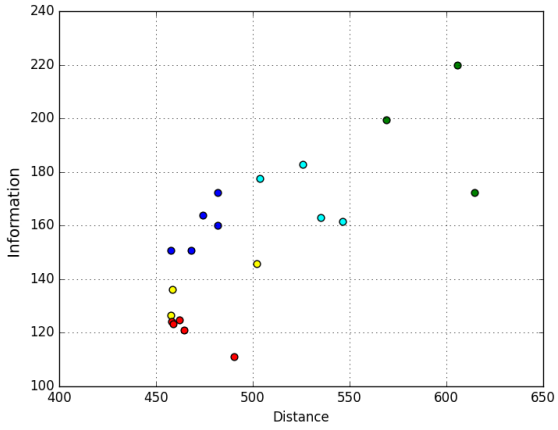


Fig. 8: Trajectory ratings for the simulated user plotted over the mission objective space. Markers are colored by rating: red = 1 (a poor trajectory), yellow = 2, green = 3, cyan = 4, and blue = 5 (an excellent trajectory). The simulated user wants to gather 150 information samples. Above that threshold, lower distances are preferred.

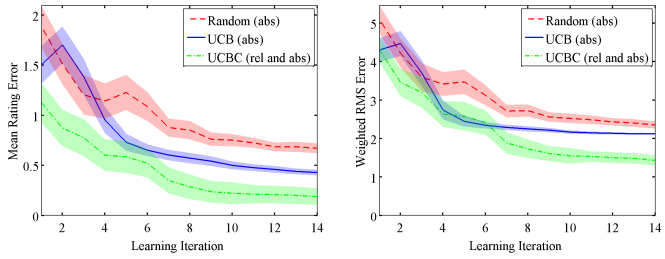
by a human user. The ratings are linearly scaled to the range $[0, 1]$ for use in the GP.

We compare our proposed combined method of choosing between absolute and relative queries (labeled UCBC in plots) to randomly selecting only absolute queries and to actively selecting only absolute queries based on the UCB in eqn. 12. 20 learning trials were run with 14 total queries each.

To measure the performance of these methods, after each query we calculate the rating prediction error of the learned GP on the trajectories in the test set that would be rated five by the simulated user. Additionally, we calculate the WRMS error, as in equation 14. These error metrics measure a method's ability to learn the user's preferred region.

The mean error and WRMS of each method are given in Figures 9a and 9b. These results show that our method learns and identifies high-utility trajectories with fewer queries than methods using ratings alone.

We made several qualitative observations of the algorithm's performance during the trials which help explain these results. As compared with random queries, active learning reduces the number of queries about uninformative portions of the objective space. However, in some trials, the learner would fail to query about a trajectory with a user rating of five, having estimated that four was the highest possible rating. Incorporating relative queries alleviates this issue by allowing a much larger number of trajectories to be examined. Overall, these results show that our method can successfully generate highly-rated trajectories while improving learning times, which can lead to reduced operator burden and more efficient human-robot teaming.



(a) Error on paths with a true rating of five

(b) Weighted RMS error

Fig. 9: Mean GP estimation error and weighted RMS error across 20 trials for the simulated information-gathering user. The shaded region shows the standard error of the mean. UCBC is the proposed method that combines rankings and ratings.

V. Conclusion and Future Directions

In this work, we proposed a novel preference-learning algorithm. We showed that a robot can efficiently obtain a rich representation of an operator's preferences by actively combining simple rating and ranking queries with a GP to learn the human user's preferred trade-offs among mission objectives. This representation could be used in a wide variety of domains, including aquatic robotics, where the dynamic environment and limited communication necessitate a complete understanding of mission goals. Our experiments showed that multiple query types and principled active learning can significantly improve the convergence rate of preference learning.

Our work suggests several directions for further study. Large-scale user studies are needed to further define the learning algorithm's capabilities and to study how well it copes with differing levels of user expertise. Additionally, methods for incorporating other query types, such as complete demonstrations and fully-ordered rankings into the GP should be explored to further broaden the method's capabilities. Complementing this, techniques for identifying relevant objective features should be examined. Ultimately, this framework and its underlying principles have the potential to reduce the cost, time, and operator burden of deploying and controlling autonomous robots.

Acknowledgments

This work was supported in part by NSF grant IIS-1317815 and Office of Naval Research grant N00014-14-1-0905.

References

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pages 1–8, 2004.

[2] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57 (5):469–483, 2009.

[3] Suhrid Balakrishnan and Sumit Chopra. Two of a kind or the ratings game? Adaptive pairwise preferences and latent factor models. In *Proceedings of the IEEE Conference on Data Mining*, pages 725–730, 2010.

[4] Robert M. Bell and Yehuda Koren. Lessons from the Netflix prize challenge. *ACM Sspecial Interest Group on Knowledge Discovery and Data Mining Explorations Newsletter*, 9(2):75, 2007.

[5] Alexandra Carpentier, Alessandro Lazaric, Mohammad Ghavamzadeh, Rémi Munos, and Peter Auer. Upper-Confidence-Bound Algorithms for Active Learning in Multi-armed Bandits. In *Proceedings of the International Conference on Algorithmic Learning Theory*, pages 189–203, 2011.

[6] Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. *Proceedings of the International Conference on Machine Learning*, pages 137–144, 2005.

[7] Christian Daniel, Oliver Kroemer, Malte Viering, Jan Metz, and Jan Peters. Active reward learning with a novel acquisition function. *Autonomous Robots*, 39(3): 389–405, 2015.

[8] Arpad E. Elo. *The Rating of Chess Players, Past and Present*. Arco Publishers, 1978.

[9] Geoffrey A Hollinger, Brendan Englot, Franz S Hover, Urbashi Mitra, and Gaurav S Sukhatme. Active planning for underwater inspection and the benefit of adaptivity. *The International Journal of Robotics Research*, 32(1): 3–18, 2012.

[10] Bjorn Sand Jensen, Jens Brehm Nielsen, and Jan Larsen. Bounded Gaussian process regression. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, sep 2013.

[11] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. STOMP: Stochastic trajectory optimization for motion planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4569–4574. IEEE, may 2011.

[12] James J Kuffner and Steven M LaValle. RRT-connect: An efficient approach to single-query path planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 995–1001, 2000.

[13] Andras Kupcsik, David Hsu, and Wee Sun Lee. Learning Dynamic Robot-to-Human Object Handover from Human Feedback. In *Proceedings of the International Symposium on Robotics Research*, pages 1–11, Genoa, Italy, 2016.

[14] Rensis Likert. A Technique for the Measurement of Attitudes. *Archives of Psychology*, 22(140):55, 1932.

[15] Manuel Lopes, Francisco Melo, and Luis Montesano. Active Learning for Reward Estimation in Inverse Reinforcement Learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 31–46, 2009.

[16] Roman Marchant and Fabio Ramos. Bayesian optimisation for Intelligent Environmental Monitoring. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 2242–2249. IEEE, 2012.

[17] Jens Brehm Nielsen and Computer Science. Pairwise Judgements and Absolute Ratings with Gaussian Process Priors. Technical Report SVN:1336, Technical University of Denmark, 2014.

[18] Ken Perlin. An image synthesizer. *ACM SIGGRAPH Computer Graphics*, 19(3):287–296, 1985.

[19] Pannaga Shivaswamy and Thorsten Joachims. Online structured prediction via coactive learning. *Proceedings of the International Conference on Machine Learning*, pages 1–8, 2012.

[20] Thane Somers and Geoffrey A. Hollinger. Human-robot planning and learning for marine data collection. *Autonomous Robots*, 40(7):1123–1137, 2016.