Coactive Learning with a Human Expert for Robotic Monitoring

Thane Somers and Geoffrey A. Hollinger

Abstract— **We present a coactive learning algorithm to solve the problem of learning a human expert's preferences in planning trajectories for robotic monitoring. The algorithm learns these preferences by iteratively presenting solutions to the expert and updating an estimated utility function based on the expert's improvements. We applied these algorithms in the context of underwater exploration using a pair of risk and reward maps. In simulated trials, the algorithm successfully learns the underlying weighting behind a utility map used by a human planning trajectories. This work shows it is possible to create algorithms for autonomous navigation with reward functions that mimic a human planner's preferences.**

I. Introduction

When robotic vehicles collaborate with humans, true autonomy relies on the robot having a clear understanding of its goals and the tradeoffs it faces when making decisions. When a robot is assisting a human, the robot's goals must often mimic those of the human. One example of this is in planning trajectories for underwater robots performing scientific monitoring. The robot must autonomously navigate the environment while maintaining the same goals as a human scientist.

When planning trajectories for underwater gliders during such robotic monitoring, a scientist implicitly balances several environmental variables, such as risk of collision, uncertainty in ocean currents, and the location of points of interest. While current planning algorithms can account for all of these variables, it is difficult to learn the correct tradeoffs between them [3]. In this work, we study applying a coactive learning algorithm to learn a human path planner's weighting of the variables involved in choosing a trajectory. In this way, we can create an autonomous system that generalizes to different problems while still capturing the scientist's expert knowledge and experience.

Most previous work on coactive learning algorithms has studied problems where both the expert and the learner are computer programs which solve and improve the solution using different methods [1]. In our work, the solver attempts to learn a human expert's preference in the area of robotic path planning, modeled after planning for underwater scientific exploration. Specifically, the algorithm attempts to learn the expert's judgment of the utility of a particular path.

II. Coactive Learning Algorithm

Our proposed coactive learning algorithms attempt to learn an expert's utility function, $U(\langle x, y\rangle) \to \mathbb{R}$, for judging a candidate solution y for a given problem x (as in [1]). We assume that the expert's utility function can be approximated as a weighted linear function of the features of the candidate solution: $\hat{U}(\langle x, y\rangle) = \vec{w}^\top \vec{\phi}(\langle x, y\rangle)$. The ultimate goal of the algorithm is to learn the parameters $\vec{w}$ that match the expert's method for judging the utility of a solution.

On each update of the coactive learning algorithm, the algorithm creates a candidate solution y_t based on its current estimate $\hat{U}$ of the expert's utility function and presents that solution to the expert. The expert has a set of operators, $\mathbb{O}$, that can be applied to the solution to improve it: $\mathbb{O}_i \in \mathbb{O} : \langle x, y\rangle \to \langle x, y'\rangle$. In path planning, these operators might involve altering the trajectory. The cost for the update C_t is equal to the number of operators the expert applies to improve the solution. The learning algorithm then adjusts $\hat{U}$ based on the difference in parameters between y_t and y'.

Algorithm 1: CoactiveLearningUpdate (problem x_t, learning algorithm's solution y_t, improved solution y', cost C_t)

```
**if** $C_t > 0$ **then**
    $\vec{\Delta}_t := \vec{\phi}(\langle x_t, y'\rangle) - \vec{\phi}(\langle x_t, y_t\rangle)$
    $\vec{w}_{t+1}^\top = \vec{w}_t^\top + \lambda_t * \vec{\Delta}_t$
**end**
```

Algorithm 1 shows how the weights w are updated. If the expert has improved the proposed solution, the difference in parameters $\vec{\Delta}$ between the proposed and improved solutions is calculated. This difference is then scaled by the learning rate and added to the previous estimated weights to find the new estimated weights.

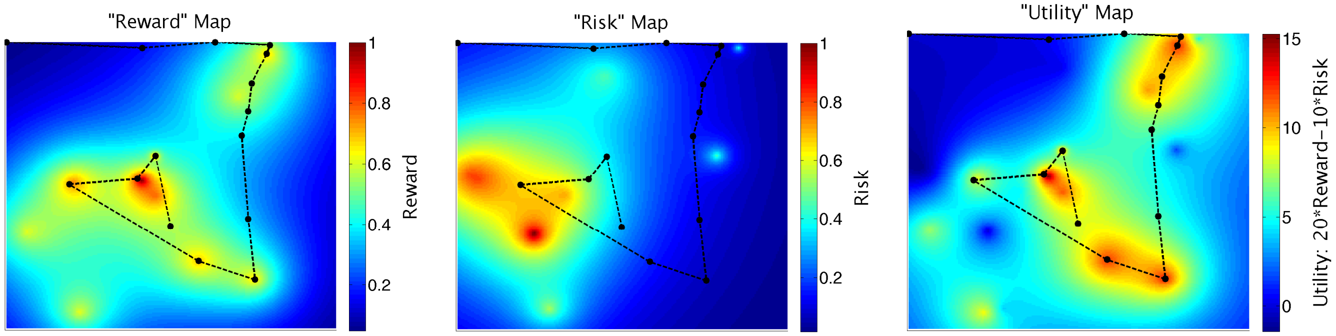
Several variations of the coactive learning algorithm are created by adjusting the learning rate λ. Two of the most commonly used are perceptron with a constant learning rate and passive aggressive, which adjusts lambda to ensure the most recent mistake is corrected [1].

III. Robotic Monitoring Results

The problem we examine consists of several components: a planned trajectory of waypoints, a "risk" map that represents the cost of traveling in a given region, a "reward" map that represents the quality and value of information gained by traveling in a given area [4], and a target $\vec{w}^\top$ of risk and reward weightings for the learning algorithm to learn.

In order to make the problem tenable for use with a human expert, we make a number of modifications to the general coactive learning algorithm. We assume that the expert's utility function is linearly composed of two features: the risk the robot incurs, and the information it gains during its tour [2]. The total risk and total information for a path are

T. Somers and G. Hollinger are with the School of Mechanical, Industrial & Manufacturing Engineering, Oregon State University, Corvallis, OR 97330 USA, (e-mail: {somersth,geoff.hollinger}@oregonstate.edu).



(a) Reward map representing the value of traveling in a particular area.

(b) Risk map showing the risk of traveling in a region.

(c) Utility map generated from a weighted sum of the risk and reward maps. Here, the target weights of risk and reward are -10 and 20 respectively.

Fig. 1: An example path and utility field generated using the proposed algorithm after one trial. Only the utility map shown in (1c) is presented to the expert during trials. The black line represents the robot's path through the environment after the expert has made local improvements to it. The proposed algorithm learns the underlying weighting between risk and reward using coactive learning.

found by integrating each respective feature map along the path. At each update, the human expert improves the learning algorithm's proposed path by moving one point. We use the perceptron variation of the algorithm with an exponentially decaying λ.

To test the algorithm's ability to learn a human expert's weighting, the expert is presented with a randomly generated path overlaid on a map of the utility at each location in a region, as shown in in Fig. 1. Maps of risk and reward are generated as a random sum of Gaussians. The utility map is generated by weighting these risk and reward maps by their respective target weights and summing them. Since the human expert is optimizing the path based on a map of utility calculated using the target weights, we can test how effectively the learning algorithm finds the target weights.

At each update, the expert improves the path by moving one of the points of the path. The change in information and risk are calculated and used in the coactive learning update to update the learning algorithm's estimate of the expert's utility function. Using the new estimate of utility, the algorithm then runs a locally optimal Travelling Salesman Solver [5] on the updated set of points to generate a new guess at the optimal path through the map. Each trial consists of 10 updates on one randomly generated map.

IV. Conclusion and Future Directions

We successfully applied a coactive learning algorithm to path planning using a human expert, showing that the algorithm can learn and mimic a human expert's priorities. Over several trials using a set of target weights we found that the algorithm's estimated weights would converge on the target weights in a reasonable amount of time for use with a human expert. An example trial is shown in Figure 2. However, the learning algorithm is susceptible to being misled by imperfect improvements by the human expert. Additional quantitative results will be presented in the final workshop paper.

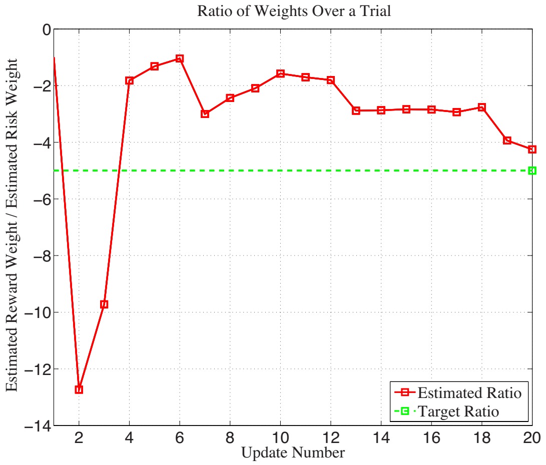


Fig. 2: An example plot of the ratio between the risk and reward weights over a trial. An exponentially decreasing learning rate over time was used. While the estimated weights converge on the target weights, note that imperfections in the human expert's improvements slows the convergence.

Further work is needed to make the algorithm usable in a real-world situation. Perceptron, cost-sensitive, and passive-aggressive learning rates should be studied to select which learns quickly while also being resistant to the imperfections of the human operator. Additionally, other path parameters could be included in order to more closely match the human's intentions. Ultimately, we hope to be able to learn a human's preferences in trajectory planning without complete knowledge of the underlying parameters used.

Acknowledgment

The authors thank Robby Goetschalckx, Alan Fern, and Prasad Tadepalli from Oregon State University for their insightful comments.

References

[1] R. Goetschalckx, A. Fern and P. Tadepalli, "Coactive Learning for Locally Optimal Problem Solving", AAAI, 2014, to be published.

[2] G. Hollinger and G. Sukhatme, "Trajectory Learning for Human-robot Scientific Exploration", ICRA, 2014, to be published.

[3] D. Silver, J. A. Bagnell, and A. Stentz. "Learning from demonstration for autonomous navigation in complex unstructured terrain," Int. J.Robotics Research, 29(1):15651592, 2010.

[4] A. Singh, A. Krause, C. Guestrin, and W. Kaiser. "Efficient informative sensing using multiple robots," J. Artificial Intelligence Research, 34:707755, 2009.

[5] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, "The Traveling Salesman Problem: A Computational Study," Princeton Univ. Press, 2006.