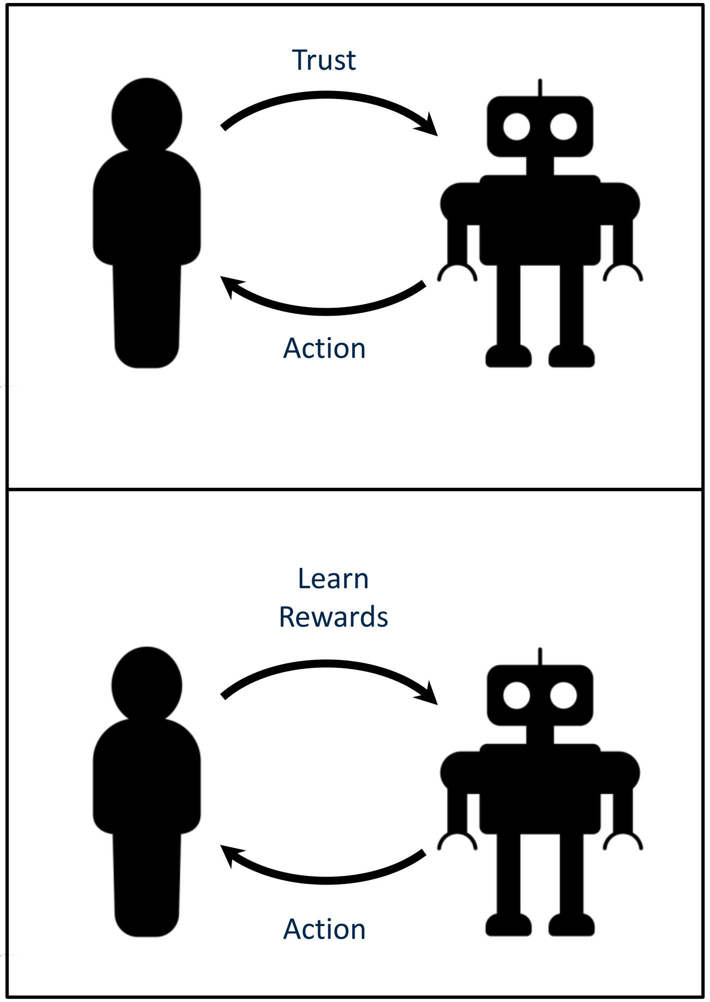
Effects of Learning State Dependence of Reward Weights on Trust and Team Performance in a Human-Robot Sequential Decision-Making Task

SHREYAS BHAT, JOSEPH B. LYONS, CONG SHI AND X. JESSIE YANG

Introduction

- Trust is a key factor to facilitate effective collaboration [1]
- Trust has been used to drive the decision-making of robots in human-robot teams [2, 3]
- However, most prior research makes an important assumption - The human-robot team has a reward function independent of the state of the team [3]
- In this work, we try to remove this assumption



Trust-Aware Markov Decision Process

Item	Description
States	Trust, Contextual Information
Actions	Actions recommended by the robot and implemented by the human
Transition Function	Trust Update Model, Contextual Information Updates
Reward Function	Rewards obtained for choosing actions in specific states
Human Behavior Model	Probabilities of the human choosing each action given the recommendation

Table 1 - Components of the Trust-Aware MDP

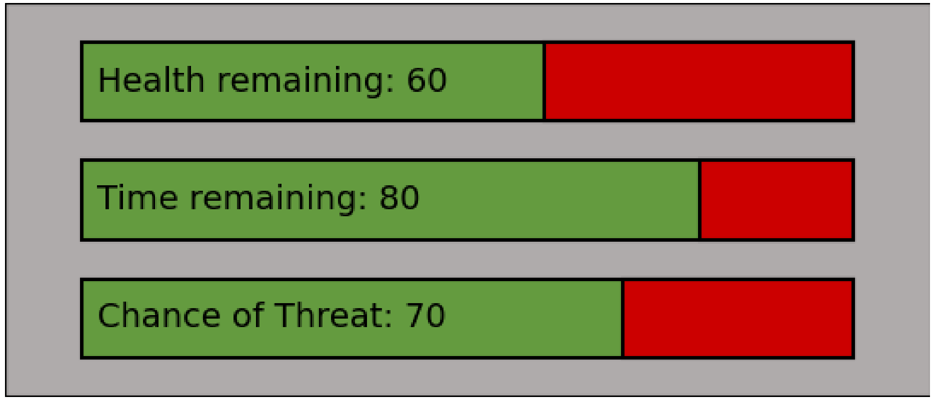
- Major assumption in previous work [1]
  - The reward function is independent of the state

- In this work, we remove this assumption

1. BHAT ET AL (2024)

Human-Robot Team Task

- The human-robot team performs a reconnaissance mission
- They sequentially search through a town to look for threats
- At each search site, there are two actions –
  - USE the armored robot
  - NOT USE the armored robot
- Using the armored robot takes time but gets no loss of health
- Not using the armored robot is faster but risky, as the human will lose health if a threat is encountered without protection from the armored robot



Their objective is to minimize the loss of time and health

Reward Function

- Thus, the reward function is a weighted sum of costs for health loss and time loss

Health Action Reward Weight Time Loss Cost

$$R(H, C, a, D) = -w(H, C)\mathcal{H}(a, D) - (1 - w(H, C))\mathcal{C}(a).$$

Reward Time Threat Health Loss Cost

- Our previous studies [X], [Y] did not consider the state dependence of the reward weight and assumed it to be constant throughout the interaction
- However, this may not be true – humans may be more risky when health is high and time is low and more conservative otherwise

Study 1 – Learning State Dependence of Rewards

The Critical Chance of Threat Presence - d^*

- Taking the expectation of the reward function over the chance of threat presence, we see that at a certain chance of threat presence, the two actions result in the same expected reward
- At a chance below d^*, NOT USING the armored robot is better on average
- At a chance above d^*, USING the armored robot is better on average

$$d^*(H,C) = \frac{(1 - w(H,C))c}{w(H,C)h}$$

$$\Updownarrow$$

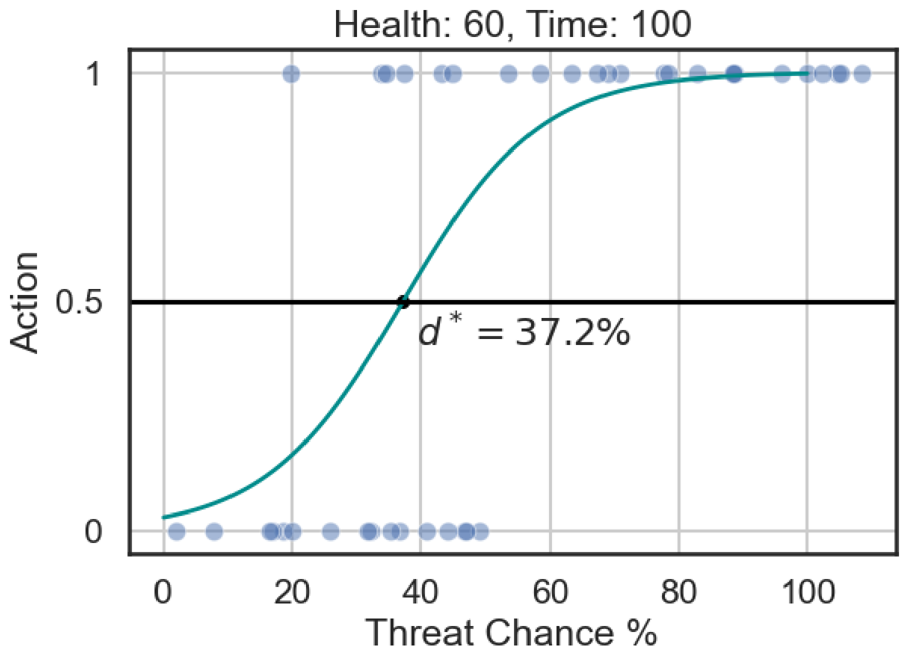
$$w(H,C) = \frac{c}{\underset{\text{Time Loss Cost}}{c} + \underset{\text{Health Loss Cost}}{hd^*(H,C)}}$$

Learning State Dependence of Rewards

- For a set of states $\{H_i, C_i\}_{i \in N}$ get responses from participants about their choice of action for a range of threat levels $d_k \in [0, 100\%]$

- Train logistic regressions for each i
  - The threat level d^* is the threat level at which the classifier gives an equal probability for both actions for the state H_i, C_i

- Data collected via Amazon Mechanical Turk
  - 396 queries (6 health * 6 time * 11 threat levels)
  - 124 workers
  - 4092 responses

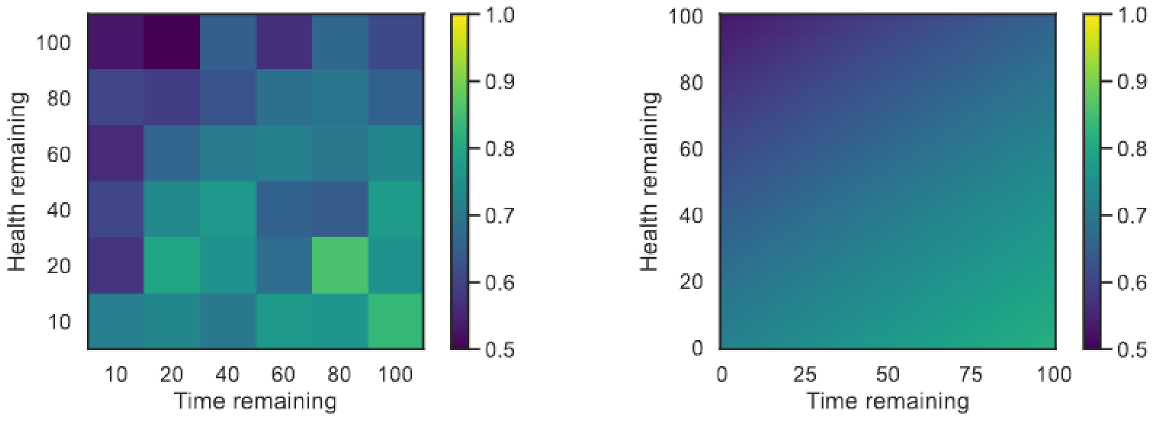


State Dependent Reward Function

- The raw data of learned reward weights is then smoothed by fitting a logistic regression model

- We use forward selection using the Akaike Information Criterion (AIC) for selecting features for the final model

$$w(H, C) = \frac{1}{1 + \exp(0.26H - 0.17C - 0.79)}$$



(a) Raw data (b) Smoothed model

Fig. 3: Heatmaps showing (a) raw data of learned health reward weights at each queried state and (b) the smoothed function for the state dependence of reward weights

Study 2 – Effects on Team Performance and Trust

Simulation Setup

- We compare two interaction strategies for the recommender robot
  - One uses the **state dependent** reward function for generating the recommendations
  - The other uses a **constant** reward weight of **0.81** for losing health

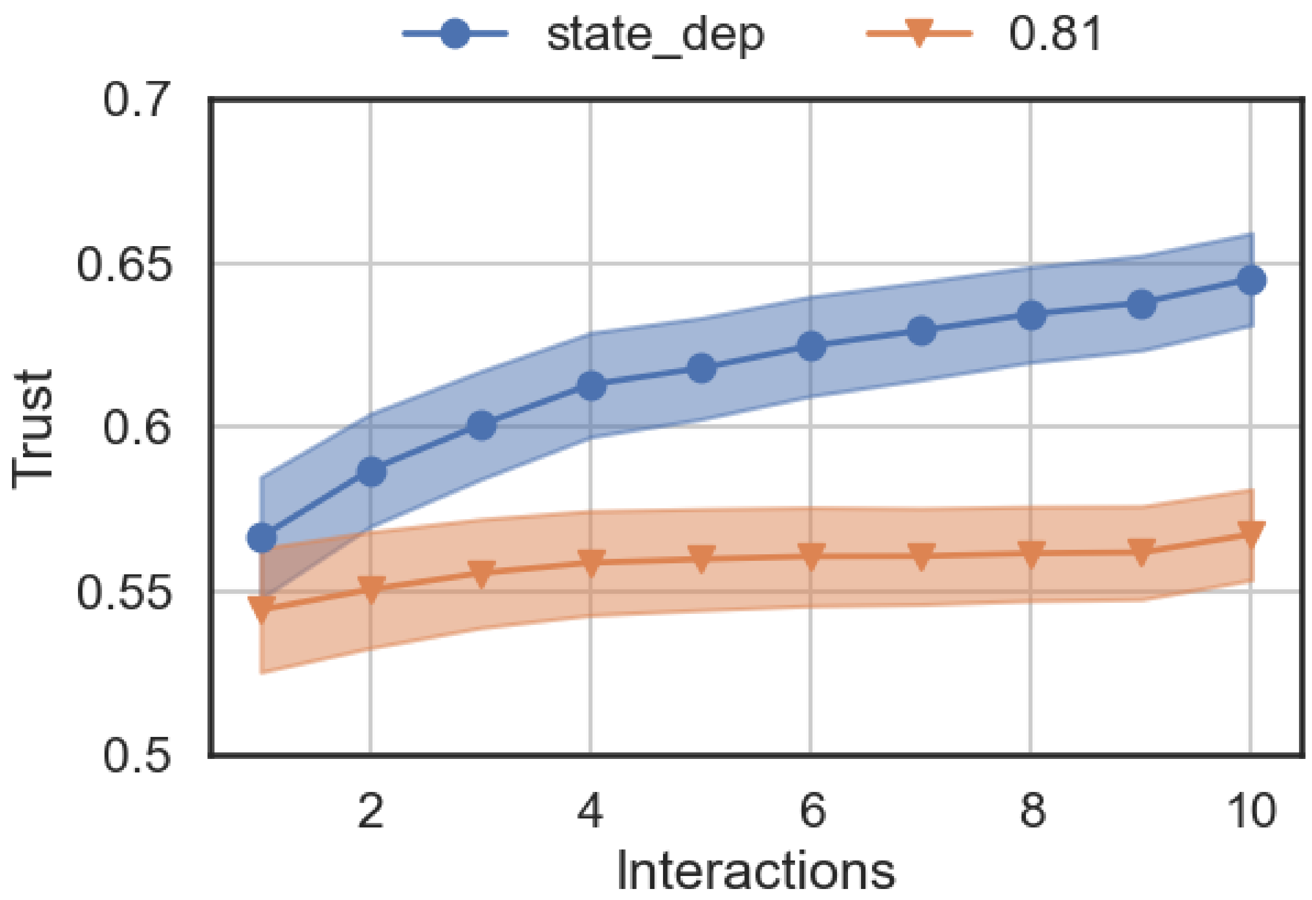
- Simulating the human
  - We use the human behavior model to simulate the action choices of the human
  - Trust parameters are sampled from values obtained from an earlier study

- Setting threats and threat levels
  - With 50% probability, threats are set with a probability of 0.7
  - With 50% probability, threats are chosen “actively” to induce a difference between the two robot strategies*

* - PLEASE REFER TO THE COMPLETE PAPER FOR THE ACTIVE THREAT SELECTION STRATEGY

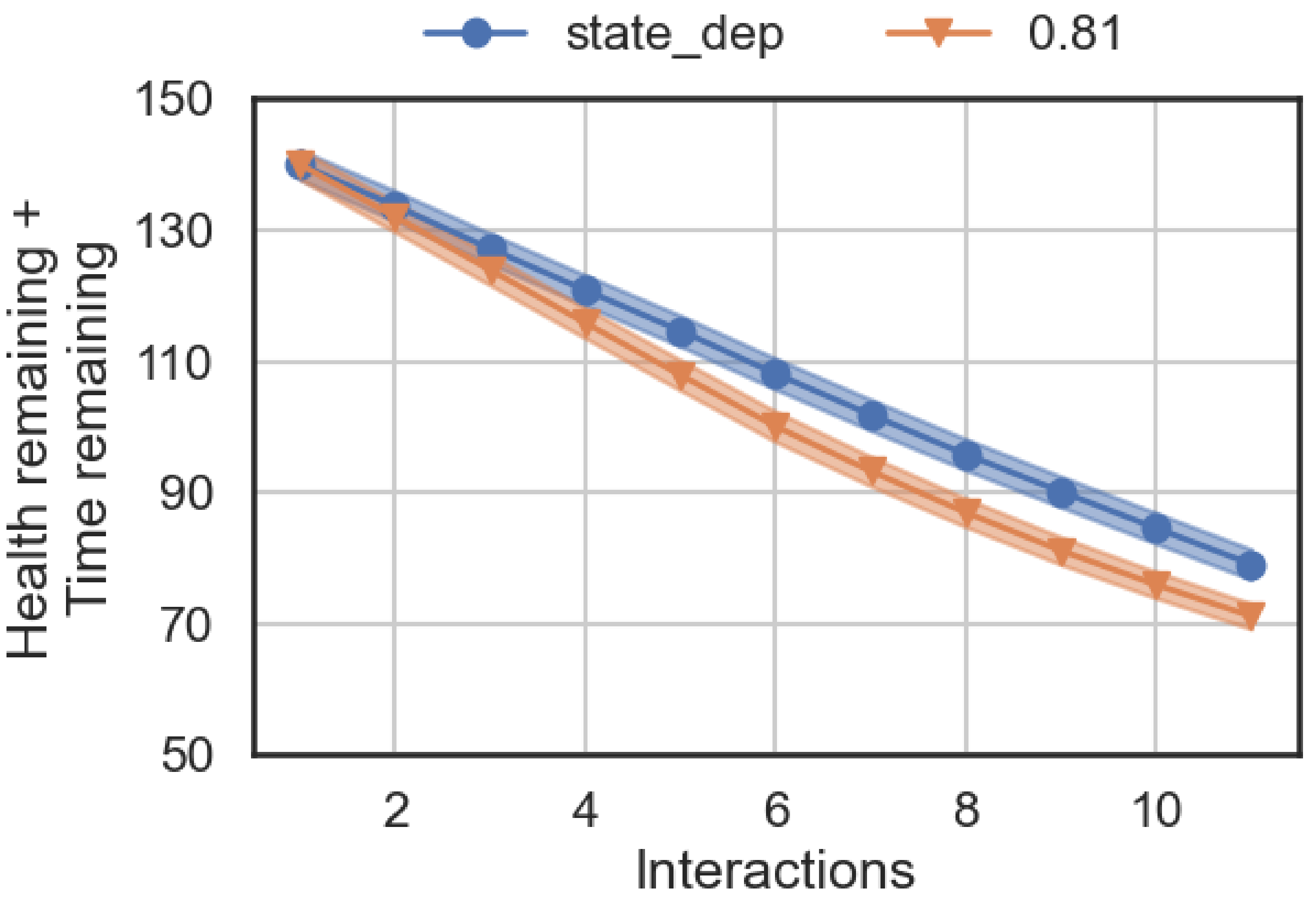
Trust Dynamics

- We ran 100 independent simulations each with a starting health and time chosen from the set {100, 70, 40}, resulting in 900 total simulations
- Each simulation had 10 interactions with the robot
- The state dependent strategy was rated **higher in trust**



Team Performance

- We ran 100 independent simulations each with a starting health and time chosen from the set {100, 70, 40}, resulting in 900 total simulations
- Each simulation had 10 interactions with the robot
- The state dependent strategy resulted in **better team performance**



Limitations and Future Work

- The state-dependent rewards learning framework is demonstrated in a very specific scenario of reconnaissance missions
  - However, it can easily be translated to other situations where there are two conflicting objectives

- The comparison results are only in simulation at this point and may not necessarily translate well into real life
  - We are working towards validating these results through a human-subjects study

Summary

- We proposed a framework for learning state-dependent rewards in a situation with two conflicting objectives
  - We demonstrated the framework in the context of reconnaissance missions through a study done via Amazon Mechanical Turk

- We compared two robot interaction strategies in the reconnaissance mission context through simulations
  - Results indicate that a strategy using the state-dependent rewards results in higher trust and better team performance

- In the future, we will try to validate these simulation results through a human-subjects study

- I am currently on the job market
- Looking for roles: Robotics Engineer/Software Engineer
- Contact me – shreyasb@umich.edu

Thank You

Questions?

Personal Website