2020 29th IEEE International Conference on
Robot and Human Interactive Communication (RO-MAN)
August 31 - September 4, 2020 - Virtual Conference

Inferring Non-Stationary Human Preferences for Human-Agent Teams

Dana Hughes[1], Akshat Agarwal[2], Yue Guo[1], Katia Sycara[1]

Abstract— **One main challenge to robot decision making in human-robot teams involves predicting the intents of a human team member through observations of the human's behavior. Inverse Reinforcement Learning (IRL) is one approach to predicting human intent, however, such approaches typically assume that the human's intent is stationary. Furthermore, there are few approaches that identify when the human's intent changes *during* observations. Modeling human decision making as a Markov decision process, we address these two limitations by maintaining a belief over the reward parameters of the model (representing the human's preference for tasks or goals), and updating the parameters using IRL estimates from short windows of observations. We posit that a human's preferences can change with time, due to gradual drift of preference and/or discrete, step-wise changes of intent. Our approach maintains an estimate of the human's preferences under such conditions, and is able to identify changes of intent based on the divergence between subsequent belief updates. We demonstrate that our approach can effectively track dynamic reward parameters and identify changes of intent in a simulated environment, and that this approach can be leveraged by a robot team member to improve team performance.**

I. INTRODUCTION

Understanding the intents and goals of team members is an important aspect of high-performance human teams. Humans naturally exhibit the ability to infer these aspects in others from observed behaviors using Theory of Mind [1]. In effective teams, such abilities help to form Shared Mental Models, simplifying team synchronization and improving team performance [2]. Providing robots with similar intent inference is of great interest in human-agent teaming and human robot interaction [3]. To this end, several computational models have been developed to describe human behavior. Markov Decision Processes (MDPs) has been proposed as a particularly useful class of models, as they can be used as models of rational, utility-maximizing behavior in humans, as well as for decision making in artificial agents.

The process of inferring a human's goal from observed behaviors under an MDP is referred to as Inverse Reinforcement Learning (IRL). From the human-modeling perspective, IRL has been suggested as a computational model for Theory of Mind [4], and has been used to reason about human goals from observed actions [5], [6], human beliefs about the world [1], and human knowledge [7]. In robotics, several IRL algorithms have been used in the context of apprenticeship learning, where reward functions are learned from expert human demonstration for use by a robot [8], [9], [10], [11], [12], [13]. Additionally, IRL is used by robots during human-robot interactions to infer some aspect of the human's decision making process, such as goals or preferences [14], model of the world dynamics [15], or rationality [16].

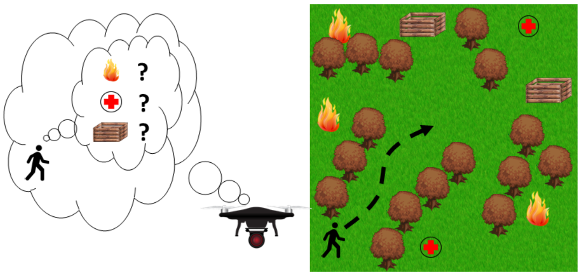

Fig. 1. A human-agent team in an emergency response scenario, used as a running example and for experimental evaluation. The human performs tasks in the environment (putting out fires, triaging victims, and collecting supplies), while the agent (here, a UAV) searches the environment for new tasks. The agent estimates the human's task preferences, and can alter its sensor configurations to make detecting preferred tasks more likely.

We consider the case where a human has preferences for performing specific tasks in an environment, captured as a parameterized reward function in an MDP model. Specifically, we are interested in scenarios where the human's preference may change over time, either through gradual drift or as a discrete, step-wise intent change. Our desire is to have an agent team mate (e.g., robot) maintain an estimate of the human's preferences over time, and adjust its behavior to best assist the human perform her preferred tasks. We assume that, while the *reason* for the preference change may be extrinsic or intrinsic, it is unknown to the agent.

Figure 1 shows an emergency response scenario that will be used as a running example throughout the paper. The human is tasked with putting out fires, triaging victims, and collecting supplies within the environment. As an example, a human may initially show a preference for triaging victims over putting out fires. However, while providing medical assistance, existing fires may start to spread, and she would shift her preference towards putting out fires (reflected as a drift in reward parameters). When she notices her supplies running low, she may then immediately opt to collect supplies (reflected as a step-wise change of intent). A UAV flying over the environment detects tasks and communicates their locations to the human. The UAV may select different sensing configurations that are more or less sensitive to each task type; in order to be an effective team mate, the UAV must maintain an estimate of the human's preferences, and

This work has been supported by AFOSR/AFRL FA9550-18-1-0251 and ARL W911NF1920146

[1] The authors are with the School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. {danahugh,yueguo,sycara}@andrew.cmu.edu

[2] The author is with Nuro, Montain View, California, USA. agarwalaks30@gmail.com

978-1-7281-6075-7/20/$31.00 ©2020 IEEE

select configurations to support those preferences.

The main contribution of this paper is the development of a model used to infer human reward preferences that is cognizant of temporal changes of such preferences, and to demonstrate that incorporating the model into an agent improves the performance of human-agent teams. The remainder of the paper is structured as follows: We discuss related works in Section II as context for this work, and provide a background on MDPs and IRL in Section III. We formulate our approach in Section IV. Section V provides experimental results for simulations of the previously described emergence response scenario. Section VI concludes the paper and discussed potential avenues of future work.

II. RELATED WORK

Our approach utilizes IRL to infer the human's preferences (encoded as a reward function in an MDP) from observed trajectories. Several algorithms have been developed over the last two decades to solve this problem using a variety of techniques to find the optimal reward function, including max-margin [8], [9], Bayesian [10], gradient-based [11], and max-entropy [12] approaches, as well as using nonparameteric models [17] and deep neural networks [18] to learn non-linear reward functions. In general, these approaches incorporate some heuristic to avoid degenerate solutions [8].

Many IRL approaches are concerned with learning reward functions that are defined by multiple intentions. For instance, in [13], expectation-maximization is used to cluster trajectories and calculate the reward parameters for a predefined number of intents. Nonparametric priors over intents allows for an arbitrary number of intents to be learned [19], [20], [21]. These approaches typically require multiple trajectories to learn reward functions, and learn discrete, static reward parameters, while our approach identifies different intents *online*, and assume dynamic reward parameters.

Traditionally, IRL has been used as an approach to apprenticeship learning, where an agent attempts to learn a reward function based on expert demonstrations in order to mimic the expert's behavior. Recently, online approaches to IRL have been explored for applications where monitoring and predicting the behavior of an agent is of interest, including approaches based on gradient-based updates [22] or an online extension of max-entropy IRL [23]. DARKO [24] is an online approach to incrementally learning an MDP which handles multiple intents / subgoals using a stopping heuristic. Our approach is similar to these in that we perform online updates of the estimated reward function, however, our approach differs in that it can handle dynamic reward functions, and does not use a heuristic to identify change in intents.

IRL has been used to learn models of various aspects of a human's mental state from observed behavior, especially for the purposes of human-robot interaction. One of the main applications has been to infer the goals or intentions of a human [5], [6], [1]. The ability to predict a human's goals in such a manner has been demonstrated to improve performance of human-robot teams [14].

More recently, inferring other aspects of human decision making has been explored. Modeling human navigation through IRL has enabled robots to navigate an environment in the presence of humans [25]. In [15], a human's internal model of the dynamics of an environment is learned by an agent for the purpose of shared autonomy. In [16], the *rationality* of a human is modeled, allowing a robot to modify its behavior gracefully when it observes unexpected behavior.

III. PRELIMINARIES

A. Markov Decision Processes

A Markov Decision Process (MDP) is a mathematical model of systems where agents are allowed to make decisions in an environment with stochastic dynamics, and is described by the 5-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, r, \gamma)$, with each element defined as

- a set of system states, $s \in \mathcal{S}$,
- a set of actions, $a \in \mathcal{A}$,
- a transition function, $T(s, a, s')$, defining the probability transitioning from state s to state s' when action a is performed, $T(s, a, s') = p(s'|s, a)$,
- a reward function, $r(s)$, indicating the immediate reward for entering state s,
- a discount factor, γ, which describes the tradeoff between immediate and future rewards.

A policy, $\pi(s, a)$, is defined as a function describing the probability of performing action a when in state s, $\pi(s, a) = p(a|s)$. Under a given policy, the value of each state is given by the *value function*, $V^\pi(s)$, and is defined as the expected total discounted reward obtained under the policy

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0 = s, \pi\right] \tag{1}$$

Similarly, the Q-function (i.e., *action-value function*), $Q^\pi(s, a)$, is the value of performing action a in state s under the given policy

$$Q^\pi(s, a) = \mathbb{E}\left[R(s) + \gamma \sum_{s' \in \mathcal{S}} T(s, a, s') V^\pi(s')\right] \tag{2}$$

An *optimal policy* is one which maximizes the expected total discounted reward in all states, denoted as

$$\begin{aligned} V^*(s) &= R(s) + \gamma \sum_{s'} T(s, \pi^*(s), s') V^*(s') \\ Q^*(s, a) &= R(s) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q^*(s', a') \end{aligned} \tag{3}$$

and an optimal policy is one that simply selects the action which produces the best expected return

$$\pi^*(s, a) = \arg\max_a Q^*(s, a) \tag{4}$$

B. Inverse Reinforcement Learning

The task of inverse reinforcement learning is to attempt to infer a suitable reward function that describes a given a policy or a set of observed trajectories. This may be viewed as the inverse of reinforcement learning—while the task of RL is to learn an optimal policy given a reward function, the task of IRL attempts to learn a reward function given observations of a (presumed) optimal policy.

The dynamics of the observed agent and environment are formalized in an MDP without reward (MDP\r). The goal of IRL is to find a state-dependent reward function, $R(s)$, that maximizes the likelihood of a provided policy or observed trajectory. In other words, if an observed trajectory τ was generated by a policy that is optimal given R, then an IRL algorithm should produce an R that maximizes the likelihood of the observation, i.e.,

$$R = \arg\max_{R} P(\tau|R) \tag{5}$$

Unfortunately, several degenerate solutions exist for R, including simply assigning R to zero [8]. To account for this, IRL approaches in the literature typically introduce a heuristic that accounts for such degeneracy, such as maximizing the margin between the best and next-best reward solutions [9], selecting rewards which maximizes the entropy of the policy [12], or assumes a prior over rewards [10].

A second consideration is that learning the value of a reward function for each state is difficult for large state spaces. To account for this, parameterized reward functions are often used in order to simplify the task. The results of an IRL algorithm is then to produce the parameter set that maximizes the likelihood of observations. While approaches to nonlinear parameterizations of reward functions have been explored in the literature [17], a common tactic is to define the reward function as a linear parameterization of predefined state features, $\psi(s)$, i.e., $R(s;\boldsymbol{\theta}) = \boldsymbol{\theta}^T\psi(s)$ [9].

IV. APPROACH

A. Problem Specification

The goal of our approach is to enable an agent to use IRL to infer the reward preferences of an observed human team member, represented as parameters of a linear reward model, where the human's preferences may vary over time. Specifically, we are interested in scenarios where the human's reward preferences can exhibit 1) gradual drift over time, and 2) discrete step changes (referred to as *change of intent*). In order to achieve this, the agent maintains a *belief* over the human's reward parameters, and updates these beliefs as it observes human behavior. In the case where the human is modeled using an MDP, it is computationally impractical to perform such an update over the entire reward parameter space, as the human's policy would need to be calculated for each possible reward parameter.

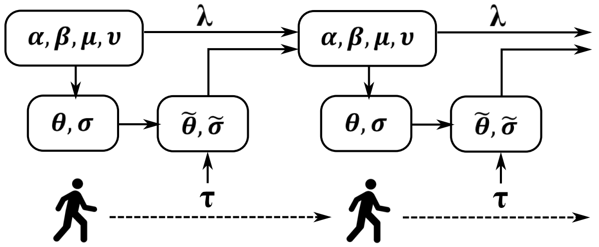
$\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{\nu}$
$\boldsymbol{\lambda}$
$\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{\nu}$
$\boldsymbol{\lambda}$
$\boldsymbol{\theta},\boldsymbol{\sigma}$
$\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\sigma}}$
$\boldsymbol{\theta},\boldsymbol{\sigma}$
$\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{\sigma}}$
$\boldsymbol{\tau}$
$\boldsymbol{\tau}$

Fig. 2. Graphical model representing the method used to update the belief over reward parameters (θ, σ) from an observed human trajectory (τ). Parameters of the conjugate distribution over the belief (α, β, μ, ν) are decayed (λ) and updated with a reward parameter pseudo-estimate ($\tilde{\theta}$, $\tilde{\sigma}$).

To address this, we model the agent's belief over time with a Gaussian distribution with mean $\boldsymbol{\mu}_t$ and covariance matrix $\boldsymbol{\Sigma}_t$, i.e., $\boldsymbol{\theta}_t = (\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$; a conjugate distribution is used to model the distribution over these parameters. Figure 2 provides a graphical model demonstrating an update of the agent's belief: the agent's belief is calculated as the posterior predictive distribution of the conjugate; after observing a short trajectory of human actions, a pseudo-estimate of the reward parameters is calculated using the agent's belief as a prior and the trajectory as evidence; the conjugate distribution hyperparameters are decayed to reduce the influence of previous observations, and updated using the pseudo-estimate.

Pseudocode for our approach is given in Algorithm 1. Details on estimating reward parameters from observed trajectories, updating the reward parameter belief, and identifying change of intent is given in Subsections IV-B–IV-E.

Algorithm 1 Reward Parameter Estimate Update

```
Initialize conjugate distribution parameters (Section IV-D)
Initialize trajectory buffer, $\tau = \{\ \}$
Initialize set of intents, $\boldsymbol{\Theta} = \emptyset$
Set decay rate, $\lambda$;
Set change of intent threshold, $\epsilon_{intent}$

repeat
  Observe current state and agent action, $(s, a)$
  Append $(s, a)$ to $\tau$
  if $\tau$ full then
    Calculate $\boldsymbol{\theta}_t$
    Calculate $\tilde{\boldsymbol{\theta}}$ (Algorithm 2)
    Update conjugate distribution parameters
    Set $\tau = \{\}$
    if $KL(\boldsymbol{\theta}_t \| \boldsymbol{\theta}_{t+k}) > \epsilon_{intent}$ then
      if $\exists \boldsymbol{\theta} \in \boldsymbol{\Theta} \mid KL(\boldsymbol{\theta} \| \boldsymbol{\theta}_{t+k}) < \epsilon_{intent}$ then
        Set $\boldsymbol{\theta}_{t+k} = \boldsymbol{\theta}$
      else
        Set $\boldsymbol{\theta}_{t+k} = \tilde{\boldsymbol{\theta}}$
        $\boldsymbol{\Theta} = \boldsymbol{\Theta} \cup \boldsymbol{\theta}_t$
      end if
      Initialize conjugate distribution parameters;
    end if
  end if
until forever
```

B. Reward Parameter Pseudo-Estimate

Similar to other formulations [16], [13], the agent models the human using a noisy-rational (i.e., Boltzmann) decision making policy, parameterized by the reward parameters $\boldsymbol{\theta}$,

$$\pi(s,a;\boldsymbol{\theta}) = \frac{e^{bQ(s,a;\boldsymbol{\theta})}}{\sum_{a'\in\mathcal{A}} e^{bQ(s,a';\boldsymbol{\theta})}} \tag{6}$$

where b represents the rationality of the human—a high value of b results in selecting more actions with high Q value.

Given a set of k steps of the human's trajectory from time steps t to $t+k$, $\tau = \{(s_1,a_1)\dots(s_k,a_k)\}$, the likelihood of the trajectory given reward parameters is

$$P(\tau \mid \boldsymbol{\theta}) = \prod_{(s,a)\in\tau} \pi(s,a;\boldsymbol{\theta}) \tag{7}$$

Additionally, the belief over the reward parameters at time t provides a prior over $\boldsymbol{\theta}$,

$$P(\boldsymbol{\theta}|\boldsymbol{\theta}_t) = \frac{1}{\sqrt{2\pi|\boldsymbol{\Sigma}_t|}} e^{-0.5\left[(\boldsymbol{\theta}-\boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_t)\right]} \tag{8}$$

Using Bayes rule, the posterior over $\boldsymbol{\theta}$ given τ and $\boldsymbol{\theta}_t$ is

$$P(\boldsymbol{\theta} \mid \tau;\boldsymbol{\theta}_t) = \frac{P(\tau \mid \boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\theta}_t)}{P(\tau)} \tag{9}$$

The pseudo-estimate of the reward parameter is calculated as *maximum a posteriori* (MAP) estimate of $\boldsymbol{\theta}$,

$$\tilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} logP(\boldsymbol{\theta}|\tau;\boldsymbol{\theta}_t) \tag{10}$$

Combining (7)–(10), the log-likelihood of $P(\boldsymbol{\theta}|\tau;\boldsymbol{\theta}_t)$ is

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) = & \sum_{(s,a)\in\tau} log\pi(s,a;\boldsymbol{\theta}) \\ & - 0.5\left[(\boldsymbol{\theta}-\boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_t)\right] \\ & - log\sqrt{2\pi|\boldsymbol{\Sigma}_t|} - logP(\tau) \end{aligned} \tag{11}$$

The first term in the log-likelihood equation is identical to several other IRL formulations, The second term stems from the fact that we use the belief over the reward parameters at time t as a prior over the pseudo-estimate—in effect, the likelihood of the parameter decreases as a function of its Mahalanobis distance from belief at time t.

Similar to other approaches, we calculate the optimal value of $\boldsymbol{\theta}$ using gradient ascent. Noting that the third and fourth terms in the log-likelihood equation are not functions of $\boldsymbol{\theta}$, the gradient of the likelihood with respect to $\boldsymbol{\theta}$ is

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) = \left[\sum_{(s,a)\in\mathcal{D}} \nabla_{\boldsymbol{\theta}} log\pi(s,a;\boldsymbol{\theta})\right] - \boldsymbol{\Sigma}_t^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_t) \tag{12}$$

The gradient of the log of the Boltzmann policy (6) is

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} log\ \pi(s,a;\boldsymbol{\theta}) = & b\nabla_{\boldsymbol{\theta}} Q(s,a;\boldsymbol{\theta}) \\ & - b\sum_{a'\in\mathcal{A}} \pi(s,a';\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} Q(s,a';\boldsymbol{\theta}) \end{aligned} \tag{13}$$

Algorithm 2 Reward Parameter Pseudo-Estimate

```
Input: Observed Trajectory, $\tau$;
       MDP\r, $\mathcal{M}$;
       Prior Reward Parameter Estimate, $\boldsymbol{\theta}_t = (\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$
Parameter: Learning Rate $\alpha$
Output: Reward Parameter Pseudo-Estimate $\tilde{\boldsymbol{\theta}}$

Set $\tilde{\boldsymbol{\theta}} = \boldsymbol{\mu}_t$
repeat
    Compute $Q(s,a;\tilde{\boldsymbol{\theta}})$
    Compute $\nabla_{\boldsymbol{\theta}} Q(s,a;\tilde{\boldsymbol{\theta}})$
    Evaluate $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\tilde{\boldsymbol{\theta}})$ using Equation 12
    Set $\tilde{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}} + \alpha\mathcal{L}(\tilde{\boldsymbol{\theta}})$
until convergence
return $\tilde{\boldsymbol{\theta}}$
```

The $\max$ operator in the Q-function makes the Q-function non-differentiable with respect to $\boldsymbol{\theta}$. However, the gradient of Q can be calculated through Bellman Gradient Iteration [26] by approximating the max operator with a softmax operator (we set the parameter p to 25),

$$max(x_0 \dots x_n) \approx \frac{\sum_{i=0}^{n} log\left(e^{px_i}\right)}{p} \tag{14}$$

Algorithm 2 summarizes the calculation of a pseudo-estimate, $\tilde{\boldsymbol{\theta}}$. At each iteration, the Q function is calculated using value iteration, and the gradient of the Q function is calculated using Bellman Gradient Iteration [26]. While *de novo* calculations of the value function generally require several iterations to converge, small changes in reward parameters (such as due to drift) results in only requiring a few iterations to update the value function from the previous calculation.

C. Pseudo-Estimate Covariance

Section IV-B calculates a point estimate of the reward parameters that best explains the observed trajectory. As the trajectories are expected to be short (or even single actions), there may be a large variation in reward parameters that could explain the trajectory. Thus, to make suitable updates to the reward parameter belief, the covariance of the pseudo-estimate is also needed. This can be calculated using importance sampling. A set of reward parameters, $\{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \dots \boldsymbol{\theta_m}\}$ are sampled from a uniform distribution over the range of possible reward parameters. The weight of each sample, w_i, is calculated as the likelihood ratio of the sample and observed reward parameters,

$$w_i = e^{\mathcal{L}(\boldsymbol{\theta}_i) - \mathcal{L}(\tilde{\boldsymbol{\theta}})} \tag{15}$$

The covariance of the pseudo-estimate is calculated empirically from the weighted samples

$$\tilde{\boldsymbol{\Sigma}} = \frac{1}{\sum_{i=1}^{m} w_i} \sum_{i=1}^{m} w_i \left(\boldsymbol{\theta}_i - \tilde{\boldsymbol{\theta}}\right)^2 \tag{16}$$

D. Belief Update

The primary purpose of our approach is to update the belief over the reward parameter as pseudo-estimates are made. The reward parameter belief at time $t+k$ is calculated using a Bayesian update, using the belief at time t as a prior and the pseudo-estimate of the reward as an observation. In the general case, performing such an update can be difficult, as it requires normalizing over the probability of the psuedo-estimate. However, as we model each reward parameter belief using a Gaussian distribution, a conjugate distribution can be utilized to simplify the Bayesian update [27].

For a multivariate Gaussian distribution, an Inverse Wishart distribution is conjugate with respect to the covariance matrix of the Gaussian distribution; as we are assuming the covariance matrix is diagonal, we use an Inverse Gamma distribution for each individual variance. A Gaussian is conjugate with respect to the mean of the distribution. As mentioned previously, the reward parameters are assumed to be independent (i.e., $\mathbf{\Sigma}_t$ is diagonal). In this configuration, a Gaussian and an Inverse Gamma distribution are conjugate with respect to the mean and variance of the individual reward parameter beliefs, respectively. These distributions are parameterized by $\mu_{cp}, \nu, \alpha, \beta$, and are updated by

$$\begin{aligned}
\alpha^{(i)} &\leftarrow \alpha^{(i)} + k/2 \\
\beta^{(i)} &\leftarrow \beta^{(i)} + \frac{k\sigma^{(i)2}}{2} + \frac{k\nu}{k+\nu}\frac{\left(\tilde{\theta}^{(i)} - \mu_{cp}^{(i)}\right)^2}{2} \\
\mu_{cp}^{(i)} &\leftarrow \frac{\nu\mu_{cp}^{(i)} + k\tilde{\theta}^{(i)}}{\nu + k} \\
\nu &\leftarrow \nu + k
\end{aligned} \tag{17}$$

with (i) indicating that the parameters are for the i^{th} element of the reward parameter, and the observed trajectory has length k. The updated reward parameter is given as

$$\begin{aligned}
\mu_{t+1}^{(i)} &= \mu_{cp}^{(i)} \\
\sigma_{t+1}^{2}{}^{(i)} &= \frac{\beta^{(i)}}{\alpha^{(i)} - 1}
\end{aligned} \tag{18}$$

1) Hyperparameter Decay: For stationary reward parameters, the parameters of the conjugate distribution would converge to mean and covariance of the sample estimates. Naïvely applying this approach to non-stationary reward parameters would result in an estimate that converges to the average reward parameter over time and a large covariance. On the other hand, ignoring previous estimates of the reward parameters would bias the estimate to recent observations only, resulting in noisy estimates over time. To balance these two cases, we decay the parameters of the conjugate distributions so that the effect of previous parameter estimates is reduced over time. For the Normal Inverse Gamma distribution, the parameters are discounted by

$$\begin{aligned}
\nu &\leftarrow \lambda\nu \\
\alpha &\leftarrow \lambda(\alpha - 1) + 1 \\
\beta &\leftarrow \lambda\beta
\end{aligned} \tag{19}$$

The discount factor λ is defined as the amount to discount the parameters for a single state-action observation; for a trajectory buffer with k elements, the parameters are discounted by λ^k after calculating the associated pseudo-observation. Discounting the conjugate distribution parameters in this manner ensures that the reward parameter belief remains unchanged as time elapses, and that previous observations have less influence. Combining (17) and (19), ν has a fixed point $\nu = \lambda^k k/(1-\lambda^k)$, which reflects the limit of the effect of prior evidence relative to the current pseudo-estimate.

E. Change of Intent

In the context of our approach, the dynamics of the reward parameters for a given intent are assumed to drift slowly over time, which is reflected by a gradual change of behavior on the agent's part. A change of intent, however, involves a more sudden jump in reward parameters, reflected by a new behavior on the human's part that deviates significantly from its recent behavior. In addition to tracking dynamic reward parameters, we are interested in identifying when the underlying intent of the human changes.

We use the Kullback-Liebler (KL) divergence between reward parameter beliefs at two subsequent updates as a quantitative measure of change of intent. The KL divergence from the reward parameters at time t to time $t+k$, $KL(\boldsymbol{\theta}_t || \boldsymbol{\theta}_{t+1})$, reflects how much the updated reward parameter belief deviates from the previous belief. A change of intent is indicated by the divergence exceeding a predefined threshold, ϵ_{intent}.

V. EVALUATION

A. Test Environment

We evaluate our approach using an simulated environment representative of the emergency response scenario described in Figure 1. The environment consists of a 20 × 20 grid in which tasks (putting out fires, triaging victims, and collecting supplies) are positioned in random locations. We denote the set of tasks as $\mathcal{J} = \{fire, triage, supply\}$, and an individual task type from the set as $j \in \mathcal{J}$. Certain areas of the grid are blocked off, representing untraversable regions of the environment. We consider a human-UAV dyad, where the human seeks out tasks to perform, while the UAV discovers and communicates the locations of tasks to the human.

Each time step, the human or UAV can move in one of the cardinal directions. The human is assumed to be Boltzmann rational, moving in directions proportional to the expected return of the action; to simulate disorientation, human actions will result in moving in a random direction 30% of the time. When entering a grid cell with a task, the human is assumed to perform the task, resulting in its removal from the cell. UAVs can move over untraversable regions.

For our experiments, we simulated human decision making for 1000 time steps using a Boltzmann policy (see Section IV-B) with $b = 20$ and a discount factor of $\gamma = 0.9$. The human's desires are represented as a linear reward function, $\boldsymbol{r}^H(s; \boldsymbol{\theta}^H) = \boldsymbol{\theta}^H \boldsymbol{\psi}(s)$, with time-varying parameters representing the preference for performing each type of task. $\boldsymbol{\psi}(s)$

is a function that indicates if tasks of each type are present in the given cell. The reward parameters are defined by

$$\theta_{fire}(t) = \begin{cases} 1 + sin\left(\frac{\pi t}{200}\right) & 0 \leq t < 400 \\ -1.0 & 400 \leq t < 600 \\ 1 + cos\left(\frac{\pi(t-600)}{200}\right) & 600 \leq t < 1000 \end{cases}$$

$$\theta_{triage}(t) = \begin{cases} -cos\left(\frac{\pi t}{400}\right) & 0 \leq t < 400 \\ -1.0 & 400 \leq t < 600 \\ cos\left(\frac{\pi(t-600)}{400}\right) & 600 \leq t < 1000 \end{cases}$$

$$\theta_{supply}(t) = \begin{cases} -2 & 0 \leq t < 400 \\ 2 & 400 \leq t < 600 \\ -2 & 600 \leq t < 1000 \end{cases} \tag{20}$$

Using sinusoidal functions for the reward ensures that the rate of change is not consistent over time. This reward function contains intent changes at $t = 400$ and $t = 600$.

The UAV infers the human's reward parameter over time using Algorithm 2. The initial parameters for the conjugate distributions were $\nu^{(i)} = 5$, $\alpha^{(i)} = 2$, $\beta^{(i)} = 4$, and $\mu^{(i)} = 0$. The decay factor used was $\lambda = 0.985$. The trajectory buffer used consisted of 20 state-action pairs, and pseudo-estimates and belief updates were performed either when the buffer was full, or when a tasks was performed or discovered. These initial values of $\alpha^{(i)}$, $\beta^{(i)}$, and $\mu^{(i)}$ reflect an initial estimate of reward parameters of 0 with high uncertainty ($\sigma^{(i)} = 2$), while the low initial value of $\nu^{(i)}$ ensures that this initial estimate has little weight during future updates. The value of λ is such that ν has a fixed point ~ 56.67; in other words, prior evidence has weight of ~ 2.83 compared to a pseudo-estimate of 20 times steps. The intent change threshold, ϵ_{intent}, was empirically set to 0.05.

B. Human Monitoring

We first evaluate the ability of the algorithm to infer the human's reward function over time and correctly identify changes of intent. For this evaluation, the UAV can observe and provide the human with a complete view of the environment and tasks within it. A total of 30 tasks of random type (fire, victim, supplies) are randomly located throughout the environment; when a task is completed by the human a new random task is discovered at a random location.

1) Reward Function Recovery: We first evaluate the estimated reward parameters over 100 runs of the simulation. The estimated reward function is first standardized to have the same mean and standard deviation over time as the true reward function; as mentioned in [28], [29], the policy generated by each reward is invariant under this transformation. Figure 3 compares the average normalized reward parameter estimates to the true reward parameters given in (20).

To quantify the difference between the estimated and true reward parameters, we calculate the Inverse Learning Error (ILE) over time of the estimated parameters [28]. The Inverse Learning Error is defined as the L2-norm of the value functions calculated using the human's policy and the policy induced by the estimated reward parameters, under the reward function defined by the true parameters:

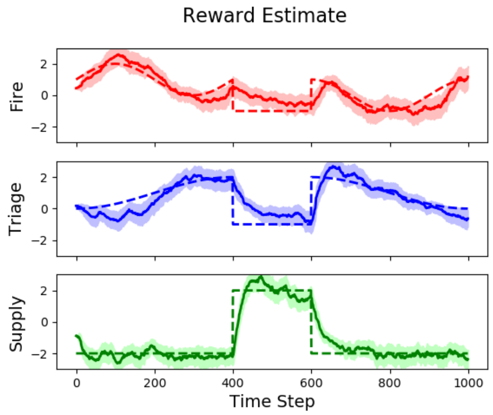

Fig. 3. Mean and standard deviation of reward parameter estimates over time. Dotted lines indicate true reward parameters given by (20)

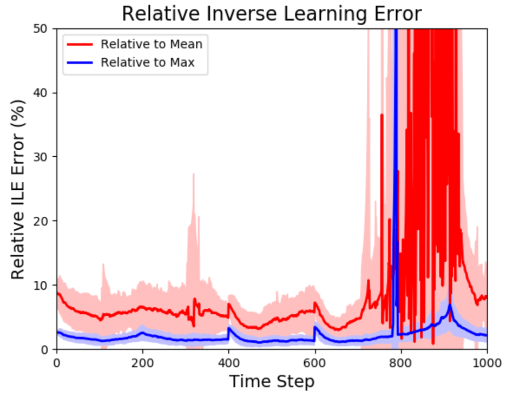

Fig. 4. Inverse Learning Error over time, relative to the mean and maximum values of the value function under the human's policy.

$$ILE = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \left\| V^H\left(s; \boldsymbol{r}^H\right) - V^\theta\left(s; \boldsymbol{r}^H\right) \right\|_2 \tag{21}$$

In effect, this calculates the decrease of expected return averaged over all states when using the policy generated by the estimated reward parameters instead of the true ones.

Figure 4 shows the average ILE, scaled to the mean and maximum of value function of the human's policy at each time step. For a large majority of the time, the ILE is within 5% of the maximum of the value function, and 10% of the mean of the value function. Between time steps ∼800 and ∼950, the mean ILE is excessively large. This can be attributed to the reward function during this time. The reward parameters are small for the fire and triage task (both are zero at $t = 800$), and negative for the supply task; this effective amplifies the ILE in these regions compared to other regions. Additionally, the reward for both fire and triage switch signs in this region; the reward parameter belief may not have been updated to reflect this switch in signs, thus the inferred policy will seek tasks that the human wishes to avoid.

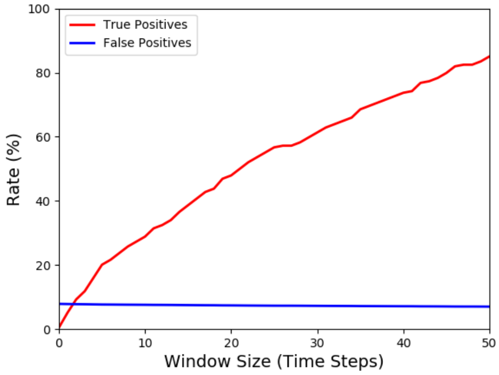

Fig. 5. True and false positive rates of detection of intent change as a function of window size.

2) Intent Inference: We next evaluate the ability of our approach to effectively determine changed of intent. As the algorithm may perform a reward estimate update (and subsequently, check for possible change of intent) at different times, the actual intent change detection time will lag the true intent change time. Additionally, a change of intent may not be reasonably detected if specific types of tasks are not located near the human at the time of intent change. Thus, we calculate the true positive and false positive detection rate when an predicted intent change is within a given time window after the true intent change.

Figure 5 shows the percentage of true positive and false positive predicted intent changes for time windows ranging from 0 to 50 time steps. At 50 time steps, the true positive detection rate is 85.05%; the false positive rate ranges from 7.03% (50 time step window) to 7.88% (0 time step window).

C. UAV

Our second experiment evaluates the overall benefit of providing the UAV with the human intent tracking algorithm. In this experiment, there are initially no known tasks to the human or UAV in the environment. The UAV is able to observe a 5×5 region of the environment centered at its current position, s; the set of observed cells is denoted $x \in \mathcal{O}(s)$. Additionally, for each cell in the environment, the agent maintains the number of time steps since it observed the cell, referred to as the staleness of the cell, $S(x)$.

At each time step, the UAV selects one of the cardinal directions to move to proportional to the decrease in the total staleness after the movement, and potentially detects tasks in each of the observed cells in its updated position. After each observation, the staleness of each observed cell is set to 0. The UAV employs a policy that selects actions based on the total reduction of cell staleness after the next observation:

$$\pi_R(a|s) = \frac{\sum_{s' \in \mathcal{S}} T(s,a,s') \sum_{x \in \mathcal{O}(s')} S(x)}{\sum_{a' \in \mathbf{A}} \sum_{s' \in \mathcal{S}} T(s,a,s') \sum_{x \in \mathcal{O}(s')} S(x)} \quad (22)$$

When a cell is observed, a task is discovered with probability $p_{discover}$; if a task is discovered, the type of task is determined by the distribution $p_{type}(j)$. For our experiments, $p_{discover}$ was set to 0.01.

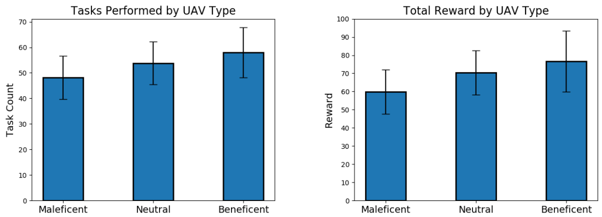

Fig. 6. Total number of tasks performed (left) and total human reward (right) of runs based on the type of the UAV team mate.

We consider three kinds UAVs, *neutral*, *beneficent*, and *maleficent*. The characteristics of each kind of UAV is reflected by the task discovery distribution:

- **Neutral:** The type of discovered task is uniformly selected from all possible types:
$$p_{type}(j) = 1/|\mathcal{J}| \quad (23)$$
- **Beneficent:** The type of discovered task is proportional to the UAV's current estimation of the corresponding reward parameter for the task:
$$p_{type}(j) = \frac{e^{\theta_t^{(j)}}}{\sum_{j' \in \mathcal{J}} e^{\theta_t^{(j')}}} \quad (24)$$
- **Maleficent:** The type of discovered task is inversely proportional to the UAV's current estimation of the corresponding reward parameter for the task:
$$p_{type}(j) = \frac{e^{-\theta_t^{(j)}}}{\sum_{j' \in \mathcal{J}} e^{-\theta_t^{(j')}}} \quad (25)$$

We hypothesize that the total number of tasks performed and total reward received by the human will be positively correlated to the beneficence of the UAV: As the beneficent UAV is more likely to detect what it believes is the human's preferred task, more tasks of that type will exist in the environment, and the human will need to travel less to perform the task; similarly, as the maleficent UAV is unlikely to detect preferred tasks, the human will spend more time traveling to complete preferred tasks.

We performed 100 runs of the simulation with each type of UAV described above. Figure 6 shows the total number of tasks performed by the human and the total reward received by the human for each UAV type. The performance of the human-robot team confirms our hypothesis: the UAV that utilizes its prediction of the human's intent, and selects sensor configurations more suited to the intent, results in increased number of tasks performed and accumulated reward.

VI. CONCLUSION AND FUTURE WORK

We presented an IRL-based approach to modeling human intents which accounts for time-varying reward functions and discrete changes of intent. Critical aspects of our approach is that it maintains a belief over possible reward parameters, performs Bayesian updates using a buffer of recent sequence of observed actions, and uses a distribution for the belief that allows for closed-form updates to a conjugate distribution. By discounting the parameter of the conjugate distributions, we can balance the influence of prior beliefs and recent observations, allowing our approach to track changes to the reward parameters over time while minimizing the effect of potentially large sample variances. Additionally, we utilize the KL-divergence of subsequent reward parameter estimates as a metric to indicate a change of intent.

Experimental results demonstrate that our approach produces reward parameters that match well with true reward parameters over time. Additionally, the KL metric used to identify new intents correctly identifies new intents 85.05% of the time, with a false positive rate of 7.03%.

Our approach contains similar aspects of Extended Kalman Filtering (EKF), namely maintaining a prediction of reward parameter estimates, and correcting the prediction based on observed behaviors. However, the MAP estimate of the reward parameters in (10), which is analogous to the observation function in an EKF formulation, is calculated using gradient ascent; the Jacobian of the observation function at this point is (near) zero, making direct application of EKF unsuitable. It may be possible to use the covariance estimate approach in Section IV-C in conjunction with EKF; we leave this exploration as potential future work.

There are several additional avenues for future work. Our approach is parameterized by a discount factor and threshold for new intents. Analyzing the effect of these parameters could provide better insight into selecting appropriate values for a given domain, potentially alleviating the need to manually select these. Modeling and updating the reward parameter belief is a second potential area of improvement. For instance, a Gaussian process model could be used in place of the Bayesian update method, allowing for limited prediction into future reward parameters, or a more expressive belief distribution (e.g., Gaussian mixture model) could be used to represent more complex beliefs over the human reward parameters. Finally, we are interested in extending this approach into joint decision-making models of human-robot collaboration, such as the one described in [30].

References

[1] C. Baker, R. Saxe, and J. Tenenbaum, “Bayesian theory of mind: Modeling joint belief-desire attribution,” in *Proceedings of the annual meeting of the cognitive science society*, vol. 33, no. 33, 2011.

[2] J. E. Mathieu, T. S. Heffner, G. F. Goodwin, E. Salas, and J. A. Cannon-Bowers, “The influence of shared mental models on team process and performance.” *Journal of applied psychology*, vol. 85, no. 2, p. 273, 2000.

[3] R. Choudhury, G. Swamy, D. Hadfield-Menell, and A. D. Dragan, “On the utility of model learning in hri,” in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2019, pp. 317–325.

[4] J. Jara-Ettinger, “Theory of mind as inverse reinforcement learning,” *Current Opinion in Behavioral Sciences*, vol. 29, pp. 105–110, 2019.

[5] C. L. Baker, R. Saxe, and J. B. Tenenbaum, “Action understanding as inverse planning,” *Cognition*, vol. 113, no. 3, pp. 329–349, 2009.

[6] T. Ullman, C. Baker, O. Macindoe, O. Evans, N. Goodman, and J. B. Tenenbaum, “Help or hinder: Bayesian models of social goal inference,” in *Advances in neural information processing systems*, 2009, pp. 1874–1882.

[7] A. N. Rafferty, M. M. LaMar, and T. L. Griffiths, “Inferring learners’ knowledge from their actions,” *Cognitive Science*, vol. 39, no. 3, pp. 584–618, 2015.

[8] A. Y. Ng and S. J. Russell, “Algorithms for inverse reinforcement learning.” in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 663–670.

[9] P. Abbeel and A. Y. Ng, “Apprenticeship learning via inverse reinforcement learning,” in *Proceedings of the Twenty-First International Conference on Machine Learning*. ACM, 2004, pp. 1–8.

[10] D. Ramachandran and E. Amir, “Bayesian inverse reinforcement learning,” *Urbana*, vol. 51, no. 61801, pp. 1–4, 2007.

[11] G. Neu and C. Szepesvári, “Apprenticeship learning using inverse reinforcement learning and gradient methods,” in *Proceedings of the Twenty-Third Converence on Uncertainty in Artificial Intelligence*, 2007, pp. 295–302.

[12] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning.” in *AAAI*, vol. 8, 2008, pp. 1433–1438.

[13] M. Babes, V. Marivate, K. Subramanian, and M. L. Littman, “Apprenticeship learning about multiple intentions,” in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 897–904.

[14] C. Liu, J. B. Hamrick, J. F. Fisac, A. D. Dragan, J. K. Hedrick, S. S. Sastry, and T. L. Griffiths, “Goal inference improves objective and perceived performance in human-robot collaboration,” *arXiv preprint arXiv:1802.01780*, 2018.

[15] S. Reddy, A. Dragan, and S. Levine, “Where do you think you’re going?: Inferring beliefs about dynamics from behavior,” in *Advances in Neural Information Processing Systems*, 2018, pp. 1454–1465.

[16] D. Fridovich-Keil, A. Bajcsy, J. F. Fisac, S. L. Herbert, S. Wang, A. D. Dragan, and C. J. Tomlin, “Confidence-aware motion prediction for real-time collision avoidance,” *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 250–265, 2020.

[17] S. Levine, Z. Popovic, and V. Koltun, “Nonlinear inverse reinforcement learning with gaussian processes,” in *Advances in Neural Information Processing Systems*, 2011, pp. 19–27.

[18] C. Finn, S. Levine, and P. Abbeel, “Guided cost learning: Deep inverse optimal control via policy optimization,” in *International conference on machine learning*, 2016, pp. 49–58.

[19] J. Choi and K.-E. Kim, “Nonparametric bayesian inverse reinforcement learning for multiple reward functions,” in *Advances in Neural Information Processing Systems*, 2012, pp. 305–313.

[20] B. Michini, T. J. Walsh, A.-A. Agha-Mohammadi, and J. P. How, “Bayesian nonparametric reward learning from demonstration,” *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 369–386, 2015.

[21] A. Šošić, A. M. Zoubir, and H. Koeppl, “Inverse reinforcement learning via nonparametric subgoal modeling,” in *2018 AAAI Spring Symposium Series*, 2018.

[22] K. Li and J. W. Burdick, “Online inverse reinforcement learning via bellman gradient iteration,” *arXiv preprint arXiv:1707.09393*, 2017.

[23] S. Arora, P. Doshi, and B. Banerjee, “A framework and method for online inverse reinforcement learning,” *arXiv preprint arXiv:1805.07871*, 2018.

[24] N. Rhinehart and K. M. Kitani, “First-person activity forecasting with online inverse reinforcement learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3696–3705.

[25] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, “Socially compliant mobile robot navigation via inverse reinforcement learning,” *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 2016.

[26] K. Li, Y. Sui, and J. W. Burdick, “Bellman gradient iteration for inverse reinforcement learning,” *arXiv preprint arXiv:1707.07767*, 2017.

[27] D. Fink, “A compendium of conjugate priors,” http://www.people.cornell. edu/pages/df36/CONJINTRnew% 20TEX.pdf, 1997, accessed: 2019-03-07.

[28] S. Arora and P. Doshi, “A survey of inverse reinforcement learning: Challenges, methods and progress,” *arXiv preprint arXiv:1806.06877*, 2018.

[29] J. A. M. Mendez, S. Shivkumar, and E. Eaton, “Lifelong inverse reinforcement learning,” in *Advances in Neural Information Processing Systems*, 2018, pp. 4507–4518.

[30] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, “Cooperative inverse reinforcement learning,” in *Advances in Neural Information Processing Systems*, 2016, pp. 3909–3917.