Discriminatively Learning Inverse Optimal Control Models for Predicting Human Intentions

Robotics Track

Sanket Gaurav
University of Illinois at Chicago
Chicago, Illinois
sgaura2@uic.edu

Brian Ziebart
University of Illinois at Chicago
Chicago, Illinois
bziebart@uic.edu

ABSTRACT

More accurately inferring human intentions/goals can help robots complete collaborative human-robot tasks more safely and efficiently. Bayesian reasoning has become a popular approach for predicting the intention or goal of a partial sequence of actions/controls using a trajectory likelihood model. However, the mismatch between the training objective for these models (maximizing trajectory likelihood) and the application objective (maximizing intention likelihood) can be detrimental. In this paper, we seek to improve the goal prediction of maximum entropy inverse reinforcement learning (MaxEnt IRL) models by training to maximize goal likelihood. We demonstrate the benefits of our method on pointing task goal prediction with multiple possible goals and predicting goal based activities in the Cornell Activity Dataset (CAD-120).

KEYWORDS

Goal Prediction; Intent Prediction; Inverse Reinforcement Learning; Maximum Likelihood Estimation

ACM Reference Format:

Sanket Gaurav and Brian Ziebart. 2019. Discriminatively Learning Inverse Optimal Control Models for Predicting Human Intentions. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

Humans and robots work in close collaboration for many tasks [2, 9, 26–28, 33] or simultaneously pursue separate tasks in shared workspaces [4, 6, 21, 36]. To enable effective task completion in either setting, robots should be able to anticipate human intentions prior to the completion of the pursued task. Doing so enables a robot to plan compatible actions ahead of time that are more productive for collaborative tasks or with fewer resource conflicts in separate tasks. For example, self-driving vehicles that can predict pedestrians' intentions and behaviors can navigate more safely and efficiently at intersections. However, improved methods for predicting human intentions are needed to support these examples of more synergistic decision making in autonomous systems.

Bayesian reasoning has been predominantly used to address the goal prediction task. Under this perspective, a predictive model of the trajectory of decisions given the goal is employed—along with a prior distribution over goals—to obtain the posterior distribution over goals. Numerous methods for the trajectory likelihood model have been employed [5, 10, 18, 23, 24, 40], ranging from simple goal-conditioned Markov models [13, 31] to inverse planning [5] and imitation learning methods [40]. Central to all of these methods is that the trajectory likelihood models are designed and optimized with sole consideration to trajectory prediction rather than goal prediction. While Bayes theorem holds for the true distributions of goal posteriors and trajectory likelihoods, it can produce error-prone goal posteriors when the likelihood model is noisily estimated from limited amounts of available data.

In this paper, we investigate training maximum entropy inverse reinforcement learning models [39] to maximize goal prediction likelihoods rather than trajectory likelihoods. In section 3, we develop our method for calculating the gradient for the likelihood of the final goal. By experimenting with an object reaching task with trained reward function from our new approach, we realize an average probability for the true goal given approximately 50% of the trajectory traveled that is not realized until 70% of the trajectory is traveled using the trajectory-based likelihood method [23]. Also, we also evaluate our method on the Cornell CAD-120 dataset [18].

The paper is organized as follows: we start with a summary of background information on decision processes, previous work on predicting human intention, the inverse optimal control formulation for imitation learning, and goal prediction using an inverse linear-quadratic regulation (LQR) formulation. Next, we describe in detail our algorithm for obtaining goal predictions from the MaxEnt IRL model trained using goal likelihood maximization rather than trajectory likelihood maximization. Next, we explain the experimental setup used to evaluate our proposed method. The result section summarizes the results obtained by our goal likelihood method versus the trajectory likelihood method and other baselines. Lastly, we provide conclusions and propose future work.

2 BACKGROUND AND RELATED WORK

2.1 Decision Processes and Goal Prediction

A wide variety of tasks can be represented using sequential decision process formulations. A Markov Decision Process (MDP) is defined[1] as a tuple $(\mathcal{S}, \mathcal{A}, \tau, \mathrm{R})$, where:

- state S is from a finite set of states $s \in \mathcal{S}$;
- action A is from a finite set of actions $a \in \mathcal{A}$;
- τ is the state transition probability from state s under action a;
- $\mathrm{R}(s_t)$ is the reward or cost received by visiting state s_t.

[1]We denote random variables with uppercase letters, fixed variables in lowercase, and matrices in boldface uppercase.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

A sequence of states and actions, $s_1, a_1, s_2, a_2, s_3, \ldots, s_T$, is produced by applying a decision policy $\pi(a_t|s_t)$ to the state transition dynamics of the decision process, $\tau(s_{t+1}|s_t, a_t)$.

In many domains, the decision processes for similar tasks differ only in small ways. We consider these differences being parameterized by a goal state g (where $g \in \mathcal{G}$, is set of all possible goals in the environment) that indicates the successful accomplishment of the goal when it is reached at final time step t_f (i.e., $s_{t_f} = g$). In contrast, if the goal state is not reached (i.e., $s_{t_f} \neq g$), a large cost (or negative reward) is incurred. More generally, the reward function can be parameterized by the goal g as: $R_g(s)$.

In this paper, we also consider continuous-valued states and actions that can be modeled using a linear-quadratic regulation (LQR) formulation. In LQR, the dynamics of a system being investigated are represented by a linear relationship,

$$s_{t+1} = \mathbf{A}s_t + \mathbf{B}a_t + \epsilon_t, \quad (1)$$

where s_t denotes the state of the system at time t, a_t denotes the action at time t, ϵ_t denotes some zero mean Gaussian noise, and $\mathbf{A}$ and $\mathbf{B}$ define the system dynamics. The state-action cost function,

$$cost(s_t, a_t) = \begin{bmatrix} a_t \\ s_t \end{bmatrix}^T \mathbf{M} \begin{bmatrix} a_t \\ s_t \end{bmatrix}, t < t_f, \quad (2)$$

is a quadratic function that penalizes the dynamics of the system/control at each time step. We also incorporate a final state quadratic cost that penalizes the final state, s_{t_f}, from deviating far from the desired goal g characterized by state s_g,

$$cost(s_{t_f}) = (s_{t_f} - s_g)^T \mathbf{M_f} (s_{t_f} - s_g), \quad (3)$$

where $\mathbf{M}$ and $\mathbf{M_f}$ are cost parameters. Similar to the MDP setting, the time-invariant state-action cost function and the final state cost can vary depending on the goal being pursued.

For the discrete MDP setting and the continuous LQR setting, the goal prediction task is defined as follows.

Definition 2.1. The **goal prediction task** seeks a probability distribution over potential goals given a partial sequence of states: $P(s_{t_f} = g|s_1, \ldots, s_t)$ for discrete decision processes and $P(G = g|s_1, \ldots, s_t)$ for continuous control processes. In the discrete setting, the exact goal state is reached whereas the final state need only be sufficiently close to state g in the continuous setting.

2.2 Existing Goal Prediction Methods

Many goal prediction methods approach the goal prediction task using Bayesian reasoning. Given a generative, goal-conditioned model of the state sequence, $P(s_{1:t}|g)$, the goal posterior is obtained using Bayes theorem:

$$P_\theta(g_i|s_{1:t_i}) = \frac{P(s_{1:t_i}|g_i)P(g_i)}{\sum_{g' \in \mathcal{G}} P(s_{1:t_i}|g')P(g')}, \quad (4)$$

where $s_{1:t_i}$ is the partial trajectory of states from time step 1 to time step t_i, g_i is the inferred goal, and g' of a goal from the set of pre-defined goals ($\mathcal{G}$) in the environment.

A simple Bayesian approach for the discrete setting is the goal-conditioned Markov model [13, 31]. It estimates the next state given the current state and goal based on the empirical frequency,

$$P(s_{t+1}|s_t, g) = \frac{\text{count}(s_{t+1}, s_t, g) + \alpha_{s_{t+1}, s_t, g}}{\text{count}(s_t, g) + \alpha_{s_t, g}},$$

where count$(\cdot)$ is the number of occurrences in the training dataset and α provide a set of optional pseudo-count values. The state trajectory likelihood of Eq. (4) is:

$$P(s_{1:t}|g) = P(s_1) \prod_{t=1}^{t_f - 1} P(s_{t+1}|s_t, g).$$

Predestination [20] uses Bayes theorem to infer destinations from driving routes. It uses a history of driver destinations and driving behaviors to predict where the driver is heading (final destination). Similarly, comMotion [22] uses a set of previously visited destinations to predict a person's destination using a Bayes classifier. More sophisticated trajectory likelihood modeling approaches treat the prediction tasks as the "inverse" of a planning process [5, 10, 14, 37]. For example, Baker, Tenenbaum & Saxe [5] use inverse planning, which assigns a probability distribution to different plans, to compute goal inferences. They investigated three different settings for goal prediction: single underlying goal, complex goals, and changing goals. In this paper, we consider the single underlying goal setting and leave extensions to the other settings as future work.

Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) [39], which we describe in more detail in the next subsection, uses a trajectory likelihood model to predict driver destinations given partial driving trajectories. It has also been used to predict user intent for robotic teleoperation with application to brain-computer interface (BCI) manipulation tasks [24]. In another extension of MaxEnt IRL, the notion of legibility and predictability [10] are used to interpret action analogy, and by Holladay et al [14] to generate pointing configurations that make the goal object legible.

All of the these works use generative models of the trajectory distribution to enable goal prediction using Bayesian reasoning. Though less prevalent, there is some work on discriminative approaches for goal prediction given partial trajectory [3, 35]. The Delphian Desktop [3] predicts user intentions in a desktop environment given the cursor trajectories using simple linear regression based on features like peak velocity and distance to the target. Logistic regression [15] has been used to predict the goal given partial trajectory [8] based on features like cosine of the angle, distance using peak velocity, and curve fitting for predicting intended goal. Additionally, the anticipatory temporal conditional random field (ATCRF) [19] and object affordances [18] is used to anticipate human activities. They have produced the Cornell Activity Dataset (CAD-120) [18] for their experiments. We use the same CAD-120 dataset for evaluating our method and compare against this approach.

2.3 Maximum Entropy Inverse Optimal Control

Inverse optimal control (also known as inverse reinforcement learning) [1, 17, 25] considers a Markov decision process *without* a reward function and learns the reward function that rationalizes demonstrated decision sequences [1]. Assumption a reward function linear in the state feature vectors parameterized by reward parameter θ, $R(s_t) = \theta \cdot \phi(s_t)$, Abbeel & Ng [1] propose the apprenticeship learning approach based on Inverse Reinforcement

Learning [25]. They devise a strategy of matching feature expectations between expert's policy (π_E) and learner's policy ($\tilde{\pi}$):

$$\left\| \mathbb{E}\left[\sum_{t=1}^{t_f} \phi(S_t) \middle| \pi_E\right] - \mathbb{E}\left[\sum_{t=1}^{t_f} \phi(S_t) \middle| \tilde{\pi}\right] \right\|_\infty \leq \epsilon, \tag{5}$$

where ϵ is the largest error allowed when approximately matching feature vectors. While useful for prescriptive behavior in imitation learning tasks, this approach is not as useful for prediction due to the ambiguities arising from many different mixtures of deterministic policies producing the same feature counts.

Ziebart et al. [39] employed the principle of maximum entropy [16] to resolve the ambiguity of mixing policies to match feature counts by selecting a probability distribution:

$$P_\theta(s_{1:t_f}) = \frac{e^{\sum_{t=1}^{t_f} \theta^T \phi(s_t)}}{Z^\theta} \tag{6}$$

where $Z^\theta = \sum_{s'_{1:t_f}} e^{\sum_t \theta^T \cdot \phi(s'_t)}$ is the partition function and $s_{1:t_f}$ is the trajectory or path traveled from time step 1 through t_f. The parameters θ that maximize the trajectory log likelihood,

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{s_{1:t_f} \in \Xi} \log P(s_{1:t_f}|\theta), \tag{7}$$

are employed by the model. Further, the gradient of Z^θ (partition function) is established in Lemma 2.2.

Lemma 2.2. *The gradient of the partition function, Z_θ, is:*

$$\nabla_\theta \log Z^\theta_{s_{a\to b}} = -\mathbb{E}\Big[\sum_{t=1}^{t_f} \phi(S_t)|S_1 = a, S_{t_f} = b\Big] = -\mathbb{E}\Big[\phi(S_{a\to b})\Big].$$

Proof. Using the definition of Z^θ from equation 6 we have,

$$\begin{aligned}
\nabla_\theta \log Z^\theta_{s_{a\to b}} &= \frac{1}{Z^\theta_{s_{a\to b}}} \sum_{s_{1:t_f}: s_1 = a, s_{t_f} = b} e^{-cost_\theta(s_{a\to b})}(-\phi(s_{a\to b})) \\
&= -\sum_{s_{1:t_f}: s_1 = a, s_{t_f} = b} P(s_{a\to b})\phi(s_{a\to b}) \\
&= -\mathbb{E}\left[\sum_{t=1}^{t_f} \phi(S_t)|S_1 = a, S_{t_f} = b\right] = -\mathbb{E}\Big[\phi(S_{a\to b})\Big].
\end{aligned}$$

□

Following from Lemma 2.2, the gradient of the trajectory log likelihood function for a set of trajectories and corresponding goals, denoted by Ξ, is:

$$\nabla_\theta \log \prod_{(s_{1:t_f}, g) \in \Xi} P(s_{1:t_f}|\theta, g) = \mathbb{E}\left[\phi(S_{1:t_f})|g\right] - \phi(s_{1:t_f}). \tag{8}$$

Thus, when maximized, this gradient is zero and the expected feature counts must match the training data feature counts.

In this paper, we extend MaxEnt IRL to predict human intentions given partial trajectory by maximizing the true goal likelihood instead of the trajectory likelihood.

2.4 Inverse Linear-Quadratic-Regulation

Maximum entropy inverse reinforcement learning methods for MDPs have been extended to linear-quadratic regulation (LQR) settings to learn the $\mathbf{M}$ and $\mathbf{M_f}$ coefficient matrices (reward parameters) from demonstrated behaviors using the principle of maximum causal entropy [38]. Under this model, computing the features ϕ_{g_i} of the partial trajectory ($s_{1:t_i}$) given the goal (g_i),

$$\phi_{g_i}(s_{1:t_i}) = \sum_{t=0}^{t_i - 1} \begin{bmatrix} a_t \\ s_t \end{bmatrix} \begin{bmatrix} a_t \\ s_t \end{bmatrix}^T \tag{9}$$

the expectation of the features $\phi_{g_i}(s_{t_i\to g_i})$ of the remaining trajectory ($s_{t_i\to g_i}$) from the current position (t_i) to the goal (g_i),

$$\mathbb{E}[\phi_{g_i}(S_{t_i\to g_i})|g_i] = \sum_{t=t_i}^{t_f - 1} \left(\mu_{a_t s_t}\mu^T_{a_t s_t} + \Sigma_{a_t s_t}\right), \tag{10}$$

the expectation of the features $\phi_{g_i}(s_{1\to g_i})$ of the complete trajectory ($S_{1\to g_i}$) from the starting point to the goal (g_i),

$$\mathbb{E}[\phi_{g_i}(S_{1\to g_i})|g_i] = \sum_{t=1}^{t_f - 1} \left(\mu_{a_t s_t}\mu^T_{a_t s_t} + \Sigma_{a_t s_t}\right), \tag{11}$$

can be achieved efficiently based on the fact that all marginal state probabilities are multivariate Gaussians with analytical expressions mean ($\mu_{a_t s_t}$) and variance ($\Sigma_{a_t s_t}$) for these expectations. Finally, the probability of the true goal (g_i) given the partial trajectory ($s_{1:t_i}$) is obtained using Bayes theorem as;

$$P(g_i|s_{1:t_i}) \propto P(g_i|s_t) \prod_{i=1}^{t_f} \pi(a_t|s_t, g_i). \tag{12}$$

This predictive linear-quadratic regulator [23] for inverse optimal control is used to predict human intentions and trajectory forecasting. Promising results have been demonstrated on the Cornell Activity Dataset (CAD-120) [18]. In this paper, we have extended the technique used in [23] by training the MaxEnt IRL model by maximizing true goal likelihood.

3 APPROACH

Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) is a widely used method to infer the true goal or intentions of a sequential decision maker given a partial trajectory by employing Bayesian reasoning. The reward parameters in the MaxEnt IRL setting are trained via maximizing the trajectory likelihood as shown in Equation 8. The trajectory likelihood models are designed and optimized solely with consideration to trajectory prediction rather than goal predictions. While Bayes theorem, the foundation of Bayesian reasoning, holds correctly for the true distributions of goal posteriors and trajectory likelihoods, it can produce error-prone goal posteriors when the likelihood model noisily estimated from limited amounts of available data. To address this problem, in this section, we develop our approach for training the MaxEnt IRL model for goal prediction using goal likelihood maximization in place of the traditional trajectory likelihood maximization approach.

3.1 Goal Likelihood Maximization Formulation

To derive our optimization procedure, we first establish Lemma 3.1 for computing the gradient (∇_θ) of the log likelihood of a partial trajectory given a goal ($P_\theta(s_{1:t_i}|g_i)$) with respect to the reward parameter (θ).

Lemma 3.1. *The gradient for computing the probability of a partial trajectory ($s_{1:t_i}$) given the goal (g_i) can be separated into the sum of expectations and the feature vector,*

$$\nabla_\theta \log P_\theta(s_{1:t_i}|g_i) = -\,\phi_{g_i}(s_{1:t_i}) - \mathbb{E}\Big[\phi_{g_i}(S_{t_i \to g_i})|g_i\Big] + \mathbb{E}\Big[\phi_{g_i}(S_{1 \to g_i})|g_i\Big].$$

Proof. Using the definition from equation 6, we have:

$$\log P(s_{1:t_i}|g_i) = -cost_\theta(s_{1:t_i}) + \log(Z^\theta_{s_{t_i} \to g_i}) - \log(Z^\theta_{s_1 \to g_i})$$

[Since, $\log \frac{m}{n} = \log m - \log n$]. Taking the gradient with respect to the reward parameter θ and simplifying after using Lemma 2.2 proves Lemma 3.1. □

Next, using Lemma 3.1, we establish the maximum goal likelihood gradient for MaxEnt IRL given a partial sequence of states.

Theorem 3.2. *The gradient for MaxEnt IRL for maximum goal likelihood given a partial trajectory decomposes into a sum of expectations, features and probabilities,*

$$\begin{aligned}\nabla_\theta \log P_\theta(g_i|s_{1:t_i}) = -\phi_{g_i}(s_{1:t_i}) - \mathbb{E}\Big[\phi_{g_i}(S_{t_i \to g_i})|g_i\Big] \\ + \mathbb{E}\Big[\phi_{g_i}(S_{1 \to g_i})|g_i\Big] + \sum_{g' \in \mathcal{G}} P(g'|s_{1:t_i})\Big(\phi_{g'}(s_{1:t_i}) \\ + \mathbb{E}\Big[\phi_{g'}(S_{t_i \to g'})|g'\Big] - \mathbb{E}\Big[\phi_{g'}(S_{1 \to g'})|g'\Big]\Big),\end{aligned}$$

where: $s_{1:t_i}$ is the partial trajectory from time step 1 to t_i, g_i is the true goal and g' are the possible goals ($\mathcal{G}$) in the environment.

Proof. Taking the gradient with respect to θ of the goal log likelihood, after expanding using Equation 4:

$$\begin{aligned}&\nabla_\theta\Big(\log P_\theta(s_{1:t_i}|g_i) + \log P(g_i) - \log \sum_{g' \in \mathcal{G}} P_\theta(s_{1:t_i}|g')P(g')\Big) \\ &\overset{(a)}{=} \nabla_\theta \log P_\theta(s_{1:t_i}|g_i) + \nabla_\theta \log P(g_i) \\ &\quad - \log \sum_{g' \in \mathcal{G}} P_\theta(g'|s_{1:t_i}) \nabla_\theta P_\theta(s_{1:t_i}|g')P(g') \\ &\overset{(b)}{=} -\phi_{g_i}(s_{1:t_i}) - \mathbb{E}\Big[\phi_{g_i}(S_{t_i \to g_i})|g_i\Big] + \mathbb{E}\Big[\phi_{g_i}(S_{1 \to g_i})|g_i\Big] \\ &- \sum_{g' \in \mathcal{G}} P(g'|s_{1:t_i})\Big(-\phi_{g'}(s_{1:t_i}) - \mathbb{E}\Big[\phi_{g'}(S_{t_i \to g'})|g'\Big] + \mathbb{E}\Big[\phi_{g'}(S_{1 \to g'})|g'\Big]\Big),\end{aligned}$$

where: (a) follows from properties of the gradient applied to logarithms and the definition of the goal posterior and (b) is obtained after employing Lemma 3.1. □

This gradient trivially equals zero when the goal predictions are perfectly correct (i.e., $P(t_f = g_i|s_{1:t_i}) = 1$). However, this is often difficult to achieve when training from noisy data. In general, optimizing the reward parameter to maximize goal likelihood is non-concave. However, we can obtain a reasonable local maxima by changing the starting conditions and other factors. For example, initializing the reward parameter optimization at the maximum trajectory likelihood parameters guarantees no worse parameters than the trajectory-based approach.

Algorithm 1 Learning IOC model for goal prediction

Input: The reward parameter θ; Set of training trajectories reaching goals Ξ; Set of Goals $\mathcal{G}$
Output: The optimized/learned reward parameter θ

1: **for** (s, g_i, t_i) $\in \Xi$ **do**
2: Extract partial trajectory $s_{1:t_i}$
3: $\nabla_\theta \leftarrow -\phi_{g_i}(s_{1:t_i}) - \mathbb{E}\Big[\phi_{g_i}(S_{t_i \to g_i})|g_i\Big] + \mathbb{E}\Big[\phi_{g_i}(S_{1 \to g_i})|g_i\Big]$
4: **for** g'$\in \mathcal{G}$ **do**
5: $\nabla_{g'} \leftarrow \phi_{g'}(s_{1:t_i}) + \mathbb{E}\Big[\phi_{g'}(S_{t_i \to g'})|g'\Big] - \mathbb{E}\Big[\phi_{g'}(S_{1 \to g'})|g'\Big]$
6: Compute $P(g'|s_{1:t_i})$
7: $\nabla_\theta \leftarrow \nabla_\theta + P(g'|s_{1:t_i})\nabla_{g'}$
8: **end for**
9: $\theta \leftarrow \theta + \eta \nabla_\theta$
10: **end for**
11: **return** θ

The learning procedure (Algorithm 1) takes as input an initial reward parameter, a set of training trajectories, and a set of possible goals in the space. It iterates over randomly selected training trajectories, extracting the partial trajectory from the selected trajectory, i.e., $s_{1:t_i}$, and then constructs the full gradient from its components in step 3, step 5, and step 7. Step 3 computes the difference in expected features for the true goal. Step 5 computes the same differences for each possible goal and then Step 7 weights these by the goal probabilities. Lastly, Step 9 applies a gradient step weighted by η to improve towards locally optimal reward parameters θ^* using expectations computed for the true goal and all other goals. In practice, more sophisticated gradient-based updates [11, 34] can be employed. The algorithm repeats steps 2 through 8 (with decreasing learning weights) for all of the training trajectories until approximately converging to a locally optimal point.

We note the contrast from previous goal prediction methods using MaxEnt IRL trained by maximizing the likelihood over the trajectory to train the reward parameter as explained in Equation (8). Critically, the likelihood of the correct goal given a partial sequence of actions is inferred using Bayesian reasoning. This produces a mismatch between the training and application objective and can produce error-prone goal likelihoods. Thus, the most significant advantage of training using the proposed method (maximum goal likelihood) is that we maximize the likelihood over the true goal, which correctly matches the application objective.

3.2 Extension to Linear-Quadratic Regulation

Algorithm 1 provides a general algorithm for MaxEnt IRL trained to maximize goal prediction for the case of discrete state/action decision processes. We can extend this general method to other settings/controllers to match other real-life scenarios. In this paper, we use inverse LQR to conduct our experiments and we have the

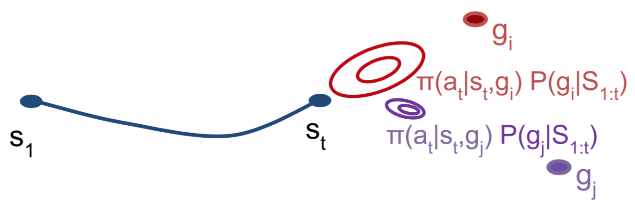

Figure 1: A partial trajectory ($S_{1:t_i}$) and distribution for two goals in the space at a trajectory point S_t. Inverse LQR with the trained reward parameters using the algorithm 1 are used to calculate the distributions for both goals.

cost function $\mathbf{M}$ and $\mathbf{M_f}$ to train as mentioned in section 2. In algorithm 1, for our inverse LQR setting we replace the reward parameter θ with $\mathbf{M}$ and $\mathbf{M_f}$. The computation of terms of Algorithm 1 in inverse LQR formulations can be referred from Equations (9), (10), (11), and (12) from section 2.4.

Figure 1 depicts the scenario of goal prediction based on the partial trajectory traveled in a real-time situation. There is an agent who starts from the starting point s_1 and travels to point s_t. The goal set $\mathcal{G}$ consists of two goals: g_i and g_j. At trajectory point s_t, we can compute goal distributions for both goals in the space based on the partial trajectory $s_{1:t}$ covered. The color contours represent the corresponding probability distribution (likelihood) of the goal. The corresponding mathematical expressions for each action conditioned on goal in the figure provide the goal probability in the inverse LQR setting, as part of Equation 12. These distributions are calculated using the trained reward parameter from Algorithm 1. The most probable goal can be obtained from the posterior goal distribution. In Figure 1, for example, goal g_i is the most probable. Thus, in this way we predict the goal given the partial trajectory in real-time.

3.3 Complexity Analysis

The time complexity of the discrete case proposed in Theorem I is $O(|\mathcal{G}||\mathcal{S}||\mathcal{A}|T)$, where $\mathcal{G}$ is the set of potential goals, $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of action and T is the total time steps in the trajectory (i.e., trajectory length). In this paper, we have implemented the above general algorithm for the inverse LQR setting which is an example of the continuous case. So, the time complexity of the proposed algorithm for inverse LQR setting requires $O(T)$ matrix updates.

The most significant advantage of using this approach is that the matrix updates only need to be computed once when performing inference over sequences sharing the same time horizon and goal positions. Further, to improve the efficiency of our computation we used the Armadillo C++ linear algebra library for fast linear computations [29].

4 EXPERIMENTAL SETUP

In this section, we explain our experimental setup used for evaluating our proposed Algorithm 1 from Section 3 for the inverse LQR setting. We have used two real-life datasets to evaluate our proposed method.

4.1 Goal Pointing Task Data

For our first set of experiments, we have used an existing dataset of pointing tasks [30]. The data was collected using a Baxter robot from Rethink Robotics and a Microsoft Kinect camera. For the training data, 10 balls were hung from the ceiling (5 on both sides of the Baxter robot), and a teleoperator was asked to stand in front of the Kinect Camera (input sensor). The teleoperator was asked to reach the displayed ball number on Baxter's head-mounted display from a neutral position. Another operator moved Baxter's corresponding arm in zero gravity mode from a neutral position to the displayed goal in synchronization with the human arm motion. This Kinect-Baxter correspondence data was used to train a linear regression correspondence model for robotic teleoperation. Further, we used the training sequence to extract states and actions for inverse LQR system and trained cost functions $\mathbf{M}$ and $\mathbf{M_f}$.

The 10 hanging balls from the ceiling were then shuffled to new positions (different from the training set-up) for the testing phase. The 18 teleoperators were asked to teleoperate the Baxter robot's arm by standing in front of the Kinect camera from a neutral position to reach the goal that was displayed on the Baxter head-mounted screen. This process was repeated for each goal and three different control assistance method ((i) Sigmoid assist, (ii) Step assist, and (iii) No assist), for details please refer to [30]. The three control assistance methods were also repeated twice in random order to maintain consistency. Thus, each person performed 60 trajectory sequences of reaching the displayed goal.

In total, the dataset consisted of 1080 goal reaching trajectories. Figure 2 explains the steps of test data collection. The dataset contains the Kinect skeleton values, the Baxter end-effector position while the volunteer was teleoperating the Baxter robot and the probability distribution across all five goals along the trajectory. In this paper, we use the Baxter end-effector positions as the trajectory points (states) for training and testing of the inverse LQR model.

4.2 Cornell Activity Dataset (CAD-120)

For our second set of experiments, we employed our Algorithm 1 to train reward parameters on the publicly available Cornell Activity Dataset (CAD-120) to strengthen our claim. This dataset consists of 120 depth camera video of daily activities. There are ten high-level activities: making cereal, taking medicine, stacking objects, unstacking objects, microwaving food, picking objects, cleaning objects, carrying food, organizing objects and eating a meal. These activities are further divided into ten sub-activities: reaching, moving, pouring, eating, drinking, opening, placing, closing, cleaning and null. For example, the task of making cereal can be broken down: reaching (cereal box), moving (cereal box on top of bowl), pouring (from cereal box to a bowl), moving (cereal box to the previous position) and null (moving the hand back).

In this study, we have divided the trajectories based on the above 10 sub-activities. We disregarded null sub-activity as it has an undefined goal or intention. First, we extracted goals for each of the trajectory in the sub-activity. Second, we trained the cost functions $\mathbf{M}$ and $\mathbf{M_f}$ for each of these sub-activities separately. We withheld

a. Starting Neutral Position b. Teleoperate Arm Towards Goal c. At Goal d. Results Displayed

Figure 2: The steps of a task in our testing sequence from a pointing dataset [30] include starting from the robot's neutral position (a) and then teleoperating the arm of the robot (b) to the goal location (c) at which point confirmation is displayed on the robot's screen (d).

10% of each sub-activity dataset for testing and used the rest 90% to train the reward parameters (i.e., $\mathbf{M}$ and $\mathbf{M_f}$). Similar to the previous experiment, trajectory points were used as states and final trajectory point as goal state.

4.3 Estimating the Reward Parameters

The inverse LQR model used in this paper has two separate reward/cost parameter matrices $\mathbf{M}$ and $\mathbf{M_f}$ to train. To provide the strongest guarantees, we first train the reward parameters using the maximum trajectory likelihood method as explained in Equation (8) on the training data for both datasets. Then we use these trained reward parameters to initialize Algorithm 1 to learn using our proposed method for maximizing goal likelihood on the training data.

We have used accelerated stochastic gradient descent with an adaptive learning rate [11, 34] and L1 regularization on both parameters simultaneously. This regularized approach prevents overfitting over the demonstrated trajectories of the datasets used in this paper. In the next section, we would describe goal predictions using inverse LQR controller on the test data for both datasets.

4.4 Goal Prediction via Inverse LQR

Following the existing formulations employed for maximum trajectory likelihood methods [30], a goal is defined as a location in x_g, y_g, z_g translational space that we want the robot arm end-effector to approximately reach. The end-effector is the endpoint of the robot arm, which is calculated using forward kinematics [32]. The end-effector consists of x_t, y_t, z_t translational and x_r, y_r, z_r, w_r quaternion angles as rotational dimensions referenced from the associated robot's coordinate frame. We have considered only translational dimensions for goal positions.

Following the approach outlined for the inverse LQR setting [23], the authors of [30] assume the linear dynamics of Equation (1), in which the state of the end-effector is defined as,

$$s_t = [x_t, y_t, z_t, \dot{x_t}, \dot{y_t}, \dot{z_t}, \ddot{x_t}, \ddot{y_t}, \ddot{z_t}, 1]^T, \tag{13}$$

and end-effector actions as

$$a_t = [\dot{x_t}, \dot{y_t}, \dot{z_t}]^T, \tag{14}$$

where $(\dot{x}_t, \dot{y}_t, \dot{z}_t)$ are velocities, $(\ddot{x}_t, \ddot{y}_t, \ddot{z}_t)$ are accelerations, and a constant of 1 is added to the state representation to incorporate linear features into the quadratic cost function in Equation (2). Additionally, goal state i of the end-effector is represented using only the goal's translational position,

$$g_i = [x_{g_i}, y_{g_i}, z_{g_i}, 0, 0, 0, 0, 0, 0, 0]^T. \tag{15}$$

To compute goal predictions along the test trajectories, we train the reward parameters $\mathbf{M}$ and $\mathbf{M_f}$ using our proposed method (maximum goal likelihood) as described in Algorithm 1 on the training data. From these trained cost matrices, the probabilities of different possible goal states are inferred given the observed partial trajectory of the end-effector in real time. The process is clearly depicted in Figure 1 and Equation (12). These goal state probabilities are $P(g_i|s_{1:t_i})$ and the probability of the most likely intended goal of the partial trajectory, I, is,

$$I = \max_i P(g_i|s_{1:t_i}). \tag{16}$$

4.5 Prior Distribution

The inverse LQR goal prediction method is a Bayesian inference method that benefits significantly from a prior distribution over the possible goals [23]. In the previous trajectory likelihood maximization experiment [30], they used a distance prior similar to the one used in previous work [23],

$$P(g_i|s_t) \propto e^{-\beta dist(s_t, g_i)}, \tag{17}$$

where $dist(s_t, g_i)$ is a function that computes the Euclidean distance between the spatial coordinates of s_t and g_i, and β is an adjustable coefficient that increases the importance of distance on

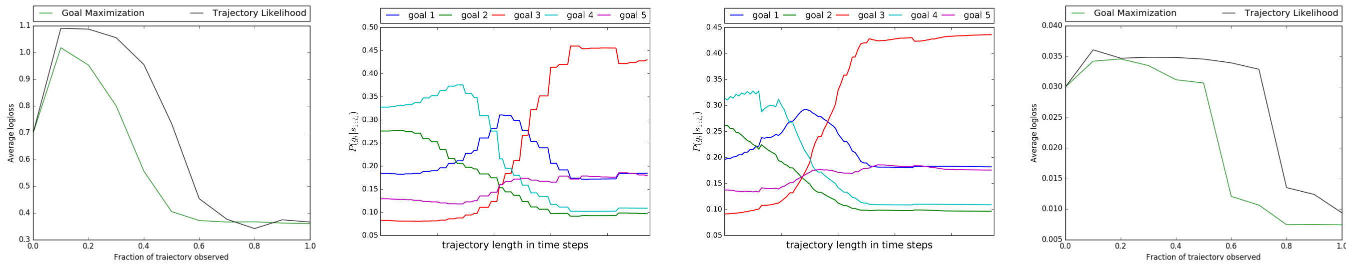

Figure 3: (a) Plot showing comparison of logloss by our goal likelihood maximization method to the trajectory likelihood maximization model on goal pointing task data; (b) Change of probability distribution over goals across a trajectory of reaching goal #3 using trajectory likelihood maximization model; (c) Change of probability distribution over goals across a trajectory of reaching goal #3 using goal likelihood maximization method; (d) Plot showing comparison of logloss by goal likelihood to trajectory likelihood on CAD-120.

the distribution. As $dist(s_t, g_i)$ decreases, $P(g_i|s_t)$ increases effectively making closer targets more probable. We have used the same formulation for most intended goal prediction for both experiments in this paper.

4.6 Baselines

To compare our goal likelihood method on goal pointing task data from the two datasets, we use the nearest target (predicting the nearest goal as the true goal along the trajectory points) prediction. It is the simplest baseline for goal prediction, and all methods should be expected to perform better than it. We additionally use logistic regression [8] as the discriminative method comparison baseline. We also compare with the previous approach of constructing a model using trajectory likelihood maximization [23]. For CAD-120 dataset, in addition to comparing to the trajectory likelihood method, we also compared our method with ATCRF [19].

4.7 Evaluation Metrics

To evaluate our proposed method against the existing trajectory maximum likelihood method, we use two evaluation metrics. First, we compute the logarithmic loss for true goal probability across the whole trajectory. The logarithm loss has been plotted for both methods at various fractions of the trajectory covered in Figure 3-a on pointing task dataset and Figure 3-d on CAD-120. Second, we compute the accuracy of our proposed method and other baselines across different fractions of the trajectory in predicting the true goal. We have also reported precision and recall for both methods. Tables 1 and 2 report the results for both datasets.

5 RESULTS AND DISCUSSION

The proposed optimization of the reward parameter to maximize goal likelihood involves maximizing a non-concave function. This prevents any guarantees of convergence to a global optimum. However, still, we can reach some local maximum that provides a better result than previous trajectory-based optimization methods. We have experimented with three different starting points to train the cost function $\mathbf{M}$ and $\mathbf{M_f}$: (1) initial values of all 0; (2) pre-trained initial values using the optimization objective of past work (i.e., trajectory likelihood maximization); and (3) randomized starting points. We find convergence to very similar parameters with all three of the different starting points, indicating that we can reach a stable local maxima without strong sensitivity to the initial values.

Table 1: A comparison of the trajectory likelihood model, the goal likelihood model, and the nearest goal baseline for the goal pointing task dataset evaluated using the accuracy, macro precision, and macro recall given various fractions of the trajectory.

		Fraction of the trajectory				
Method	Measure	20%	40%	60%	80%	100%
Nearest Goal	Accuracy	21.4	28.6	58.6	100	100
	Macro Prec.	50.0	50.0	50.0	100	100
	Macro Recall	10.7	14.2	39.3	100	100
Trajectory Likelihood	Accuracy	28.6	28.6	64.3	100	100
	Macro Prec.	50.0	50.0	50.0	100	100
	Macro Recall	14.3	14.3	32.2	100	100
Goal Likelihood	Accuracy	28.6	35.7	92.8	100	100
	Macro Prec.	50.0	50.0	50.0	100	100
	Macro Recall	14.3	17.9	46.5	100	100

Table 2: A comparison of the trajectory likelihood model, the goal likelihood model, and the ATCRF model for the CAD-120 dataset evaluated using accuracy, macro precision, and micro precision given various fractions of the trajectory.

		Fraction of the trajectory				
Method	Measure	20%	40%	60%	80%	100%
ATCRF [19]	Accuracy	-	-	-	-	86.0
	Macro Prec.	-	-	-	-	84.2
	Macro Recall	-	-	-	-	76.9
Trajectory Likelihood	Accuracy	80.9	82.5	84.1	90.4	100
	Macro Prec.	65.0	73.4	79.1	87.5	100
	Macro Recall	77.3	91.4	94.2	96.2	100
Goal Likelihood	Accuracy	81.8	86.4	90.1	100	100
	Macro Prec.	71.8	78.1	83.3	100	100
	Macro Recall	75.0	81.0	87.5	100	100

We have tested our method on two different real-life datasets involving human and robot goal-directed movements. Figure 3-a illustrates the logarithmic loss of the correct goal prediction given a partial trajectory computed across the fraction of the trajectory for the pointing task dataset. The black color represents the trajectory likelihood method and the green color represents the proposed goal likelihood approach. It is evident from Figure 3-a that the goal likelihood maximization method's logarithmic loss decreases faster and reaches the true goal probability in approximately 50% of the trajectory. On the other hand, the trajectory likelihood maximization method achieves the same performance at 70% of the trajectory. In both settings, we have used a distance prior, so the probability distribution rapidly increases from a uniform distribution as the true goal may be farther from the neutral position than other targets.

To illustrate the behavior of the goal prediction methods, we select a trajectory from pointing task test data and plot the probability distribution across five goals along the trajectory length in Figure 3-b and c. The plot of the resulting distribution in Figure 3-b corresponds to the trajectory likelihood method and Figure 3-c corresponds to our proposed goal likelihood maximization method. We can see that our goal likelihood maximization method performs better than the trajectory likelihood maximization method. Our proposed method realizes a high probability prediction for the true much earlier than the previous trajectory likelihood maximization method with a smoother transition across different goal probabilities.

Figure 3-d shows the logarithmic loss of goal prediction along the trajectory for reaching a goal from the CAD-120 dataset. The trajectory likelihood maximization method is represented by the black color and our proposed goal likelihood maximization method is shown in green. The plot clearly shows that our goal likelihood maximization method predicts the true goal (approximately 60%) much earlier in the trajectory than the trajectory likelihood method (approximately 80%).

In Table 1, we report the accuracy, precision, and recall for goal prediction for three methods, i.e., the nearest goal predictor, trajectory likelihood maximization model, and the goal likelihood maximization model. Both the previous (trajectory likelihood) and proposed (goal likelihood) models perform significantly better than the simplest baseline method, i.e., the nearest goal baseline. At 40% and 60% of the trajectory, our proposed goal likelihood-based method outperforms the trajectory likelihood-based method by a noticeable margin. The result also matches with our log loss metrics as shown in Figure 3-a. We also compare our results with a logistic regression model [8] as the generative method baseline. The reported goal prediction accuracy of 57.9% is obtained from a partial trajectory of length 60 time-steps. The average range of the trajectories of pointing task dataset is 110 time-step. So, at 60% of the trajectory length, we found that our proposed method predicts the true goal with an accuracy of 92.8%, which is significantly better than logistic regression.

Table 2 shows the performance results of the experiment conducted on the CAD-120 dataset. We compared the performance of our proposed goal-based method with other baselines based on trajectory likelihood maximization and the ATCRF model (only result for 100% is available). We can see that the trajectory likelihood method achieves comparable accuracy at 40% of the trajectory what is not realized until 100% of the sequence is observed using the ATCRF model. The result of the ATCRF method is on the unmodified CAD-120 dataset, which consists of null sub-activities, which prevents it from achieving 100% accuracy even when observing the complete trajectory. From the beginning of the trajectory, our proposed goal-based method outperforms the trajectory-based method by a considerable margin, which matches the log loss results shown in Figure 3-d and achieves 100% accuracy in prediction at 80% of the trajectory.

Thus, these experiments strongly support our claim that by re-training the MaxEnt IRL approach using goal likelihood maximization for goal predictions, we can achieve better and faster goal prediction than existing methods—specifically those based on trajectory likelihood maximization. As this is an important subproblem for planning symbiotic robot behavior, we believe these improvements will help increase the productivity of human-robot collaborative tasks when used appropriately.

6 CONCLUSION AND FUTURE WORK

In this paper, we have proposed training inverse reinforcement learning models that were initially designed for policy estimation, to instead be optimized for goal prediction. We derived the gradient for optimizing goal likelihoods under the general discrete maximum entropy inverse reinforcement learning (MaxEnt IRL) setting and under the continuous inverse linear-quadratic regulation (LQR) setting. We demonstrated that our goal likelihood maximization method provides significant improvements for goal prediction compared to previous methods based on trajectory likelihood maximization in practice. Thus, with our new approach, we can more accurately infer intended goals farther in advance than previous approaches, enabling robots to know human intentions to make more compatible decisions.

As future work, we will test our method on real-world human-robot tasks like assisting robotic teleoperation [30]. These tasks often involve additional complications that should also be modeled to improve goal prediction. For example, though we have assumed that the robot's workspace is free of obstacles in this paper, many real-world robotic workspaces contain numerous obstacles. We plan to extend our goal prediction optimization approach to the hybrid, two-level imitation learning method [7] that incorporates discrete waypoints at the top level and employed LQR predictions conditioned on the waypoints at the bottom level. We believe that through arm motion demonstrations of obstacle avoidance during training, the cost function can be learned to reason about arm movements around obstacles in testing environments. Further, in this paper, we have assumed that the goals are static in the environment. We will relax this assumption by allowing goals to change over time without being reached [5, 12].

ACKNOWLEDGMENTS

We thank Mathew Monfort for sharing his linear quadratic regulator for inverse optimal control code with us for modification. This research is supported as part of the Future of Life Institute (futureoflife.org) FLI-RFP-AI1 program, grant #2016-158710 and NSF grant #1652530.

REFERENCES

[1] Pieter Abbeel and Andrew Y. Ng. 2004. Apprenticeship Learning via Inverse Reinforcement Learning. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML '04)*. ACM, New York, NY, USA, 1–. https://doi.org/10.1145/1015330.1015430

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

[3] T. Asano, E. Sharlin, Y. Kitamura, K. Takashima, and F. Kishino. 2005. Predictive interaction using the delphian desktop. In *Proceedings of the Annual ACM symposium on User Interface Software and Technology*. ACM, Proceedings of the Annual ACM symposium on User Interface Software and Technology, 141.

[4] Andrea Bajcsy, Dylan P. Losey, Marcia K. OâĂŹMalley, and Anca D. Dragan. 2017. Learning Robot Objectives from Physical Human Interaction. In *Proceedings of the 1st Annual Conference on Robot Learning (Proceedings of Machine Learning Research)*, Sergey Levine, Vincent Vanhoucke, and Ken Goldberg (Eds.), Vol. 78. PMLR, 217–226. http://proceedings.mlr.press/v78/bajcsy17a.html

[5] Chris L Baker, Joshua B Tenenbaum, and Rebecca R Saxe. 2007. Goal inference as inverse planning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 29.

[6] Siddhartha Banerjee and Sonia Chernova. 2017. Temporal Models for Robot Classification of Human Interruptibility. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS '17)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1350–1359. http://dl.acm.org/citation.cfm?id=3091282.3091312

[7] Arunkumar Byravan, Mathew Montfort, Brian Ziebart, Byron Boots, and Dieter Fox. 2014. Layered hybrid inverse optimal control for learning robot manipulation from demonstration. In *NIPS workshop on autonomous learning robots*. Citeseer.

[8] Aditya Chaudhary. 2017. Discriminative Predictive Analysis for Goal Prediction (MS Thesis). (2017). http://hdl.handle.net/10027/22163

[9] Yiannis Demiris. 2007. Prediction of intent in robotics and multi-agent systems. *Cognitive Processing* 8, 3 (01 Sep 2007), 151–158. https://doi.org/10.1007/s10339-007-0168-9

[10] Anca D. Dragan, Kenton T. Lee, and Siddhartha S. Srinivasa. 2013. Legibility and predictability of robot motion. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 301–308.

[11] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.* 12 (July 2011), 2121–2159. http://dl.acm.org/citation.cfm?id=1953048.2021068

[12] Sanket Gaurav. 2017. Goal Predictive Robot Teleoperation using Predictive filtering and Goal Change Modeling (MS Thesis). (2017). http://hdl.handle.net/10027/21840

[13] Zoubin Ghahramani. 2001. An introduction to hidden Markov models and Bayesian networks. *International journal of pattern recognition and artificial intelligence* 15, 01 (2001), 9–42.

[14] Rachel M Holladay, Anca D Dragan, and Siddhartha S Srinivasa. 2014. Legible robot pointing. In *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*. IEEE, 217–223.

[15] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An introduction to statistical learning*. Vol. 112. Springer.

[16] Edwin T. Jaynes. 1957. Information theory and statistical mechanics, II. *Physical review* 108, 2 (1957), 171–190.

[17] R. Kalman. 1964. When is a linear control system optimal? *Trans. ASME, J. Basic Engrg.* 86 (1964), 51–60.

[18] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. 2013. Learning human activities and object affordances from RGB-D videos. *The International Journal of Robotics Research* 32, 8 (2013), 951–970. https://doi.org/10.1177/0278364913478446 arXiv:https://doi.org/10.1177/0278364913478446

[19] H. S. Koppula and A. Saxena. 2016. Anticipating Human Activities Using Object Affordances for Reactive Robotic Response. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 1 (Jan 2016), 14–29. https://doi.org/10.1109/TPAMI.2015.2430335

[20] John Krumm and Eric Horvitz. 2006. Predestination: Inferring Destinations from Partial Trajectories. In *Proc. Ubicomp*. 243–260.

[21] Jangwon Lee. 2017. A survey of robot learning from demonstrations for Human-Robot Collaboration. *arXiv preprint arXiv:1710.08789* (2017).

[22] Natalia Marmasse and Chris Schmandt. 2002. A User-Centered Location Model. *Personal and Ubiquitous Computing* 6, 5 (01 Dec 2002), 318–321. https://doi.org/10.1007/s007790200035

[23] Mathew Monfort, Anqi Liu, and Brian D. Ziebart. 2015. Intent Prediction and Trajectory Forecasting via Predictive Inverse Linear-Quadratic Regulation. In *Proceedings of The Twenty-Ninth AAAI Conference on Artificial Intelligence*, Vol. 5. 3672–3678.

[24] Katharina Muelling, Arun Venkatraman, Jean-Sebastien Valois, John Downey, Jeffrey Weiss, Shervin Javdani, Martial Hebert, Andrew B. Schwartz, Jennifer L. Collinger, and J. Andrew (Drew) Bagnell. 2015. Autonomy Infused Teleoperation with Application to BCI Manipulation. In *Proceedings of Robotics: Science and Systems*.

[25] Andrew Y. Ng and Stuart J. Russell. 2000. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 663–670. http://dl.acm.org/citation.cfm?id=645529.657801

[26] Xinlei Pan and Yilin Shen. 2018. Human-Interactive Subgoal Supervision for Efficient Inverse Reinforcement Learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1380–1387. http://dl.acm.org/citation.cfm?id=3237383.3237906

[27] Siddharth Reddy, Anca Dragan, and Sergey Levine. 2018. Shared Autonomy via Deep Reinforcement Learning. In *Proceedings of Robotics: Science and Systems*. Pittsburgh, Pennsylvania. https://doi.org/10.15607/RSS.2018.XIV.005

[28] Leonel Rozo, Heni Ben Amor, Sylvain Calinon, Anca Dragan, and Dongheui Lee. 2018. Special issue on learning for human–robot collaboration. *Autonomous Robots* 42, 5 (01 Jun 2018), 953–956. https://doi.org/10.1007/s10514-018-9756-z

[29] Conrad Sanderson. 2010. Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments. (2010).

[30] Christopher Schultz, Sanket Gaurav, Mathew Monfort, Lingfei Zhang, and Brian D Ziebart. 2017. Goal-predictive robotic teleoperation from noisy sensors. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 5377–5383.

[31] R. Simmons, B. Browning, Yilu Zhang, and V. Sadekar. 2006. Learning to Predict Driver Route and Destination Intent. In *2006 IEEE Intelligent Transportation Systems Conference*. 127–132. https://doi.org/10.1109/ITSC.2006.1706730

[32] Mark W Spong and Mathukumalli Vidyasagar. 2008. *Robot dynamics and control*. John Wiley & Sons.

[33] Kyle Strabala, Min Kyung Lee, Anca Dragan, Jodi Forlizzi, Siddhartha Srinivasa, Maya Cakmak, and Vincenzo Micelli. 2013. Towards Seamless Human-Robot Handovers. *Journal of Human-Robot Interaction* (January 2013).

[34] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Sanjoy Dasgupta and David McAllester (Eds.), Vol. 28. PMLR, Atlanta, Georgia, USA, 1139–1147. http://proceedings.mlr.press/v28/sutskever13.html

[35] Ryo Takagi, Yoshifumi Kitamura, Satoshi Naito, and Fumio Kishino. 2002. A fundamental study on error-corrective feedback movement in a positioning task. In *Proc. of Asian Pacific Computer Human Interaction*. 160–172.

[36] J. Gregory Trafton, Laura M. Hiatt, Anthony M. Harrison, Franklin P. Tamborello, II, Sangeet S. Khemlani, and Alan C. Schultz. 2013. ACT-R/E: An Embodied Cognitive Architecture for Human-robot Interaction. *J. Hum.-Robot Interact.* 2, 1 (Feb. 2013), 30–55. https://doi.org/10.5898/JHRI.2.1.Trafton

[37] B. D. Ziebart. 2010. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. Ph.D. Dissertation. Carnegie Mellon University.

[38] Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. 2013. The principle of maximum causal entropy for estimating interacting processes. *IEEE Transactions on Information Theory* 59, 4 (2013), 1966–1980.

[39] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum Entropy Inverse Reinforcement Learning.. In *AAAI*. 1433–1438.

[40] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2009. Human Behavior Modeling with Maximum Entropy Inverse Optimal Control.. In *Association for the Advancement of Artificial Intelligence Spring Symposium: Human Behavior Modeling*.