Object affordance driven inverse reinforcement learning through conceptual abstraction and advice

Abstract: Within human Intent Recognition (IR), a popular approach to learning from demonstration is Inverse Reinforcement Learning (IRL). IRL extracts an unknown reward function from samples of observed behaviour. Traditional IRL systems require large datasets to recover the underlying reward function. Object affordances have been used for IR. Existing literature on recognizing intents through object affordances fall short of utilizing its true potential. In this paper, we seek to develop an IRL system which drives human intent recognition along with the capability to handle high dimensional demonstrations exploiting the capability of object affordances. An architecture for recognizing human intent is presented which consists of an extended Maximum Likelihood Inverse Reinforcement Learning agent. Inclusion of Symbolic Conceptual Abstraction Engine (SCAE) along with an advisor allows the agent to work on Conceptually Abstracted Markov Decision Process. The agent recovers object affordance based reward function from high dimensional demonstrations. This function drives a Human Intent Recognizer through identification of probable intents. Performance of the resulting system on the standard CAD-120 dataset shows encouraging result.

Keywords: inverse reinforcement learning, object affordance, human intent recognition, MDP

1 Introduction

Within human Intent Recognition (IR), a popular approach to learning from demonstration is Inverse Reinforcement Learning (IRL) which extracts an unknown reward function from samples of observed behaviour. A basic assumption in IRL is that expert’s intent is embedded within their trajectories which could be expressed through the reward function [1]. However, very large datasets are required to recover the reward function. Further, traditional IRL systems possess shortcomings in tackling high dimensional demonstrations such as a video stream. Suppose, an IRL system has been trained on demonstrations shown by a left-handed person performing a particular task. It may so happen that the same algorithm may behave differently if it is being trained by a right-handed person for the same task. Multiple experts introduce different ways to perform an activity. However, abstraction is a key capability to human cognition [2] and generating high level context dependent abstract concepts allows us to generalize to new situations from very few examples. Several approaches ranging from classical symbolic AI, qualitative representations and relational learning enrich the traditional IRL systems for handling high dimensional state space. These approaches exploit statistical regularities present in the training data by operating at higher level of abstraction [3]. However, such approaches have not been employed in the IRL literature for tackling high dimensional demonstrations. Our aim is to recognize human intents from videos. Major focus is to highlight the role of semantic object affordance in arriving at a human intention. Algorithms for IRL are developed for extracting an unknown reward function from samples of observed behaviour.

Varieties of definition for object affordance proliferates the literature. Here, object affordance refers to “properties of an object that determine what actions a human can perform on them” [4]. Our motivation of considering object affordance is two-fold. First, in a particular demonstration, object affordances can help the IRL system in...
tracking the optimal user preferences. These preferences in turn would help recognize human intents. Use of object affordance helps to reduce agent environment interactions [5]. Next, we would like to testify the robustness of this claim in high dimensional demonstrations. Overcoming brittleness in generalizing various experts (persons) and generating state abstraction technique for reducing state space is of prime importance. To learn skills over various tasks involving different objects, a common understanding (between robot and human) for associated actions with various other objects would be beneficial. Introducing object affordances within IRL framework could address this issue.

The research question being addressed here is: How does object affordances induce high level intentions from the visual input of human demonstrators? We bring together high level reasoning and inverse planning through IRL and semantic object affordance reasoning. Object affordance could be a relevant transferable knowledge from the expert demonstrations which links to the expert’s reward function and subsequent intentions. This makes fusion of object affordances within an IRL framework interesting. Towards this end, an architecture is presented, which consists of a novel pipeline for the IRL agent. The IRL agent drives the Human Intent Recognizer (HIR). It works on top of the extended version of Maximum Likelihood IRL (MLIRL) [1]. Instantiation of the architecture has been shown on the standard CAD-120 dataset [6].

2 Related work

It is challenging for existing IRL techniques to work on high dimensional demonstrations involving high level human behaviours [7]. Exploring the capability of IRL to recognize human intents from high dimensional demonstrations such as video data is a promising research direction. There are typically infinite number of reward functions that yield a given behaviour [8]. One would definitely like to examine those reward function parameters which: a) adapts to different human behaviour, b) minimizes agent environment interactions without losing expressiveness and c) tightly couples the action and perception. Object affordance is a familiar term within developmental robotics [9, 10] which possesses these attributes [5, 9]. We attempt the human IR problem within IRL by considering two aspects: a) the usefulness of having abstract high level information to the IRL framework and b) testify object affordance’s credibility as reward function parameters in adapting to the changing human intentions. Enabling abstraction through high level perceptual knowledge will help the IRL frameworks to deal with high dimensional states. This kind of knowledge helps them to better understand the similarity of previously encountered states. Apart from this, the presence of an advisor will help the IRL agent to tackle the less training data for this learning from demonstration (LfD) application scenario.

Object affordances from video data: The concept of object affordance has evolved since Gibson [4] proposed the elementary idea [11]. Work on object affordances involving video data has been reported [5, 6, 12–14]. Research group from Cornell [6, 12] have pushed vision based object affordance research. They have provided a universal benchmark through CAD-120 dataset. New research on grasping affordances or grasp dependent tool affordances from videos have also been reported [15]. Affordance works reported in [16, 17] are based on static images.

Object affordance research in RL: There is hardly any work which combines object affordance within an IRL framework for studying human-object interactions involving common household environment. Nevertheless, affordance has been used in the context of imitation learning [18–20]. Lopes et al. [20] demonstrates the work in “real environment”; however, it involves very less number of simple object affordances in activities like grasp, tap and touch. The importance of object affordance in inferring the task from expert’s demonstration has not been reflected. Further, there is no provision for handling multiple objects in a video frame or understanding demonstrations involving multiple experts.

IRL and video understanding: Extending IRL framework to high-dimensional systems (like video understanding) and unrestricted reward representations is extremely challenging [21]. Initial work in this area was done by Kitani et al. [22]. Progress has been made in improving IRL systems by addressing new mechanisms to handle noise and hidden data which is common in real world applications [23, 24]. Bogert et al. [24] use a motion capture system coupled with a video camera to perform a ball sorting task. Their approach addresses the problem of hidden data in the expert’s trajectories. Deep learning revolution has lead to substantial work in this area [21, 25, 26]. Work reported by Sermanet et al. [21] claims to be the first vision-based reward learning method that can learn a complex robotic manipulation task. Nonfort et al. [27] present a continuous inverse optimal control (IOC) approach using Linear-Quadratic Regulation (LQR) for intention recognition and evaluates their approach on
CAD-120 data set. However, they have considered different sub-activities as different intentions. In contrast, we seek to develop IRL system which work on high level intents. **Incorporating advises within IRL:** Incorporation of advises within IRL has already been studied [28, 29]. Most of the algorithms within the IRL literature assume access to optimal number of trajectories; this makes learning of the expert’s reward function somewhat easier. Kunapuli et al. [28] relaxes the “optimality of the trajectories” assumption by considering the presence of a human expert’s advice. Advices were included as preferences over states and actions. Their work was inspired from the preference elicitation frameworks [30]. Kunapuli et al. assume the presence of a domain expert to furnish advices as preferences in action, state or reward spaces [28]. Odom and Natarajan [29] introduce active advice seeking by combining active learning and advice based learning; wherein domain experts are replaced by machine learning thereby eliminating the requirement of listing all the advices, a priori. **Object affordance and human IR:** The relation between object affordance and human intent has been explored by some researchers in [6, 13, 31–35]. Few existing works on object affordance from video data have considered temporal relations [6] in their models. Temporal aspect of object affordances has also been studied by Koppula and Saxena [12]. However, temporal sequence of activated object affordances in understanding high level behaviour has not been studied properly.

3 Architecture for human IR

Figure 1 shows the architecture for recognizing human intents from video data. Within the architecture, a novel pipeline is proposed for the IRL agent which drives the HIR. The IRL agent receives sensory input from the environment and return the affordance based reward function. This reward function is being utilized by HIR for identifying probable human intents corresponding to a particular demonstration.

3.1 Inverse reinforcement learning agent

This IRL agent is an object affordance driven IRL agent. Extensions to the original MLIRL [1] is achieved through the novel pipeline. Our motivation to include object affordance has already been provided in the section 1. The pipeline presented within the architecture (Figure 1) consists of a traditional Markov decision process (MDP) solver module. The other two components namely: a) SCAE and b) advisor help to extend the original MLIRL [1]. This pipeline further allows us to address the research question in an organized manner. Intuition behind SCAE and advisor is discussed below.

3.1.1 SCAE: Symbolic Conceptual Abstraction Engine

Representation of environment is crucial for designing robust algorithms in learning from demonstration or learning from observation. Employing quantitative representations (using real numbers for co-ordinates, positions, orientations etc.) will enable one to design precise state representations [36]. However, IRL system over such representation formalism fails to identify conceptually similar states which triggers exponential increase in state space. SCAE is introduced to eliminate this problem so that IRL agent can identify conceptually similar states from high dimensional demonstrations. Mapping high dimensional raw input from video frames requires us to continuously generate symbolic context dependent summaries. These would help one in tracking probable states and actions as well as tracing their similarities. **Qualitative abstraction:** Qualitative approach for spatial reasoning, Qualitative Spatial Reasoning (QSR) [37] helps to abstract away the precise metric information of a scene involving human-object interactions. Qualitative Distance Calculus (QDC) proposed by Clementini et al. [38] is a qualitative relational calculi which expresses the qualitative Euclidean distance between two points depending on defined region boundaries [39]. A set of QDC relations between human-object and object-object can represent a scene and changes in these relations can be used to explain relative motion. Generating QDC based qualitative symbolic representations from high dimensional demonstrations allows one to reduce the state space. These state abstraction mechanism allows IRL agent to improve comprehensibility of similar states. Variations introduced by multiple experts can also be reduced with the help of it. **Context dependent summary:** IRL agent is endowed with symbolic context dependent summary generation module. Human decision-making is influenced through intuitive cognition. Linking ongoing demonstration to previously acquired background knowledge helps one to generalize to new situations perfectly. Endowing IRL agent with such cognitive capability in sensory input processing phase would help to curb the curse of dimensionality. However, we are interested in employing...
background knowledge corresponding to human object usage in performing various activities. High dimensional demonstrations in our case involves human-object interactions. This module of SCAE generates a symbolic context dependent summary which involves: a) activated qualitative distance relations and b) properties of objects for each frame of the video. Properties of objects have been inferred from Object Property Ontology (O-PrO) [11]. This summary is being utilized by Markov Logic Network (MLN) [40] which is a Statistical Relational Learning (SRL) scheme. It can also be considered as a knowledge representation language which can combine symbolic information from household domain (background) knowledge and information generated by processing the videos. This summary will remain consistent whenever similar objects are present in the scene and qualitative relations between object-object and human-object do not change.

Apart from this, it also helps us to identify probable actions in a particular state. Activated object affordances can be inferred from this module which helps one to identify probable actions as well. Instead of considering velocity of human hand or positions of objects in the scene to identify action selection; this scheme is less burdensome as the number of object affordances is less. MLN uses symbolic causal rules to infer object affordances from sub-activities. Employing symbolic causal rules enables conceptual abstraction as well as analogical reasoning. This is found to be helpful in reinforcement learning [3].

3.1.2 Advisor

Research works in the field of learning from demonstrations involves learning complex manipulation tasks from real life scenarios and it is characterized by access to less training data. This direction will help to emulate human learning capability within IRL. Several algorithms in the IRL literature assume that they have access to optimal number of the expert trajectories. This makes the learning of the expert’s reward function easier. High dimensional demonstrations in our case must consist of multiple experts as well as object affordance annotations for achieving our aim.

In our case, one advisor is going to communicate with the SCAE and MDP solver module. Through this advisor, we have incorporated advice within the MLIRL formalism. SCAE needs to communicate with the advisor regarding qualitative relations and background knowledge. On the other hand, MDP solver needs to communicate with the advisor for action/state/reward spaces over MDP. The advisor can be thought of as a domain expert while communicating with the SCAE. The same can be considered a machine learning expert while communicating with the MDP solver. It can also act as a service provider for the MDP solver with the help of SCAE.

3.1.3 MDP solver

In our case, the MDP solver present within the IRL agent needs to solve forward reinforcement learning (RL) problem repeatedly. Conceptually Abstracted Markov Decision Process (CAMDP) allows the IRL agent to work upon finite and abstract state and action space from the demonstrated trajectories. However, it avails the service offered by SCAE and advisor to solve CAMDP.

The domain upon which the architecture is instantiated consists of finite number of objects and human joints. Predicates of various types are defined over these enti-
ties using the syntax of first order logic (FOL). Under this knowledge representation (KR) scheme, grounded QSR relations provide useful information about the state of the environment. Similarly, logical rules are grounded to provide us substantial clue about the possible actions in a particular state. In grounding the FOL constructs for state representation, IRL agent needs to communicate with SCAE and advisor. In grounding the FOL rules for action identification, IRL agent queries the advisor after receiving context dependent summary. To answer the query, advisor acts as a machine learning agent (specifically speaking statistical relational learning agent) for identifying highly probable action in that specific state.

3.2 Human Intent Recognizer (HIR)

Figure‡ 2 is presented to briefly discuss the problem addressed in this work. Our aim is to recognize human intents from video data. It is basically a three step process shown in Figure 2. Step 2 and step 3 of HIR is driven by the novel pipeline introduced in Figure 1. IRL agent processes the environmental data (video frames) and learns a object affordance driven reward function. Suppose, person shown in Figure 2 demonstrates a high level activity called “microwaving food”. This high dimensional demonstration is processed by IRL agent to learn the reward function. Activated object affordances from this set of demonstrations can be identified from this reward function.

Underlying criteria or preferences in a demonstrated behaviour could be a cue to demonstrator’s intent. Optimally finding these preferences in the form of activated affordances could drive the ranking policy of various human intents. IRL agent help us to retrieve these multiple criteria or preferences. Analytical Hierarchy Process (AHP) [41] is designed to solve multi-criteria decision problems. Traditional AHP is utilized by HIR in step 3 (Figure 2) to rank various human intents.

4 Problem definition

In the problem of inferring human intents (high level intentions); we assume that there are $J$ or fewer intents present in the RGBD video dataset, $D$. Each of such intent can be represented by weight, $\psi_I$. The reward weights are nothing but the semantic object affordance weights ($W_t$ in Figure 2). The $\psi_I$ is to be determined from the dotted rectangle encapsulating the sequence of $W_t$s. The IRL agent, $I_a$ is provided with a set of trajectories, $T > I$; where $D = \{E_1, E_2, \ldots, E_T\}$ generated by the human (experts). Each of such trajectories is generated by the expert during the course of executing certain intents given by $I$. The goal of $I_a$ is to learn a reward function from the set of trajectories corresponding to a particular intent $J$. This reward function is about optimally finding $W_t$s which corresponds to various object affordances. Goal of HIR $H_a$ is to extract a $\psi^*$ from observed trajectory $E_x \in D$ which best explains the underlying intent, $J_x \in J$. $H_a$ is dependent on $I_a$ for successful completion of its assigned task.

5 Methodology

Intuition behind each component of the architecture has already been described in the previous section. This section is more about the methods required to instantiate the novel pipeline as well as the architecture.

5.1 Modeling

This is related to the initial setup of the IRL agent. Its main task is to find out the right way to feed the high dimensional raw information about the environment to the IRL agent.

Video (RGBD) frames corresponding to a single person performing various activities have been provided. In this type of human-object interactions, two kinds of data are available to be utilized. Firstly, skeletal data consists of position and orientation of 15 joints of the person. Secondly, co-ordinates corresponding to various objects present in the scene are available to be utilized by the IRL agent. State representation technique based on these kind of highly precise metric information increases the brittleness in IRL systems.

5.2 Extended MLIRL

Our work of human IR is based on extending the original MLIRL algorithm proposed in [1]. This extension primarily focuses on including advisor and SCAE into the original MLIRL formulation. This section consists of two main components namely: a) advisor and b) environmental observa-

‡ The source of the Robot picture corresponding to Figure 2 is: https://www.keycdn.com/support/what-is-a-robots-txt-file/
tion. The formalization of the advisor and the discussion related to SCAE help one to understand the conversion of sensorimotor level continuous input from video data to low dimensional symbol level output. Second phase discusses the episode generation procedure. The success of this procedure will depend on the active communication between advisor and SCAE.

5.2.1 Formalization of the advisor

“Advice” provided by the advisor is utilized by the other two components of pipeline shown in Figure 1. In our setting, the expert and the advisor are two completely different entities and their roles can not be replaced by one another.

Advice is defined to be 4-tuple $< A_s, A_a, A_r, A_i >$ where $A_s$ denotes State advice, $A_a$ denotes Action advice, $A_r$ denotes Reward advice and $A_i$ denotes Intent advice.

- State advice $A_s$: To generate qualitative relations from sensory inputs, range corresponding to QDC has to be specified beforehand. This advice corresponds to range specification by the advisor. It indirectly helps in state abstraction.
- Action advice $A_a$: High dimensional demonstration corresponding to a high level intent such as “microwaving food” consists of various base level actions such as move, reach, place etc. This advice is utilized by the IRL agent for tracing base level actions in a state.
- Reward advice $A_r$: This corresponds to one shot initial advice. The weight of various object affordance parameters are initialized from the first trajectory of the expert demonstration. This advice is related to the “automated inferencing agent”. This agent is an one shot learning agent which can infer weights from handful of training examples. The service provided by this agent is fully based on SCAE (O-PrO+QDC+MLN). This agent works on sequence of temporal segments of a particular trajectory to generate this reward advice. Thus, the advisor acts as a service provider to MDP solver through SCAE.
- Intent advice $A_i$: This advice is required by the HIR (Figure 1) for identifying probable intents. $A_i$ is about quantifying the importance of a particular criterion corresponding to available alternatives.

5.2.2 Environmental observation

High dimensional input provided in the format (mentioned in the sub-section 5.1) is the only input to the IRL agent. Usefulness of knowledge representation is made available to the IRL system through SCAE.
Figure 3: Grounded symbolic snapshot corresponding to the video frame (video frame is taken from CAD-120 [6]).

Phase 1: Context dependent grounded symbol generation

Generation of low level grounded symbols from a particular video frame is achieved through SCAE. Figure 3 depicts the output of SCAE for a particular video frame. Grounded symbol generation consists of two modules namely: a) video processing module and b) ontology processing module. Video processig module helps the IRL agent to process the metric information and generate qualitative information for state space abstraction. This abstraction will generate uniform representations irrespective of variations introduced by different experts. SCAE also consists of a ontology (O-PrO [11]) processing module which generates grounded symbols corresponding to various object properties.

Phase 2: Episode generation

For proper functioning of the IRL agent, sensory input has to be in the form of <state, action> pairs. Each high level demonstration corresponding to each high level intent has to be in this format only. Episode generation is entitled to perform this pre-processing task.

1. State generation: State generation internally involves two processes: a) symbol generation and b) grounding classifier. Symbols corresponding to the state space of CAMDP will be provided by the advisor (domain expert). Grounding classifier will first generate grounded symbols through video processing and ontology processing module. Afterwards, it has to assign a particular state label corresponding to a sequence of video frames. Advisor has to track the change of snapshots generated across the sequence of frames. Unchanged qualitative relations over a period of time can signify a particular state. Suppose, another expert demonstrates the same high level intent consisting of same objects. Since, we are not using absolute position of objects and skeletal joints; linking the ongoing demonstration to previously encountered state is easier.

2. Action generation: Similar to state generation, action generation procedure internally involves symbol generation and a grounding classifier. Symbols corresponding to the action space of CAMDP will be provided by the advisor (domain expert). Grounding classifier has to assign a particular action label corresponding to a particular symbolic state. Advisor provides the action advice \( A_a \) through this grounding classifier which is based on “automated inferencing agent”. This agent is the statistical relational learning scheme called Markov Logic Network (MLN) [40]. This can also be considered as a knowledge representation language which can combine symbolic information from household domain knowledge and information generated by processing the videos. It helps to map the high dimensional input to low dimensional context dependent snapshot. It can inform activated object affordances within an unknown base level action by using causal symbolic rules. Key object affordance helps to detect the action during a particular symbolic state (temporal segment).

Thus, advisor along with SCAE help us to guide the episode generation process.

5.3 Extended MLIRL agent in action

Algorithms designed for IRL often takes a MDP without a reward function as input. \( MDP = (S, A, T, \gamma) \) where \( S \) denotes state space, \( A \) denotes action space, \( T \) denotes transition function \( T : S \times A \times S \rightarrow [0, 1] \) and \( \gamma \in [0, 1) \) denotes discount factor. These algorithms also have access to the observed behaviour of the ex-
pert in the form of a sequence of state–action pairs. Thus, standard IRL agent have access to these trajectories \( \{E_1, E_2, \ldots, E_T\} \) where expert takes action in the \( \mathcal{MDP}_r \). This behaviour is assumed to be (nearly) optimal in the MDP with respect to an unknown reward function [1].

IRL agent, \( I_0 \) works on top of MLIRL algorithm [1] which has been extended in two directions. Firstly, \( I_0 \) works on Conceptually Abstracted–\( \mathcal{MDP} \) (\( \mathcal{CA} – \mathcal{MDP} \)) rather than standard \( \mathcal{MDP} \). The standard formulation for an \( \mathcal{MDP} \) consists of two types of ingredients: (a) qualitative information (related to the state space, set of available decisions (action space), the feedback that can be received by the agent, etc.) and (b) quantitative information (for example: transition probability) [42]. \( \mathcal{CA} – \mathcal{MDP} \) aspire to work on the qualitative information for creating conceptually abstract state space. IRL agent, \( I_0 \) in our setting works on these states; where each state describe the world environment at a discrete time step. Similarly, action space of \( \mathcal{CA} – \mathcal{MDP} \) is different from standard \( \mathcal{MDP} \) in the sense that \( I_0 \) works on base level actions (such as move, reach, place etc.). In our case, \( I_0 \) has access to high dimensional demonstrations in the form of trajectories \( \{E_1, E_2, \ldots, E_T\} \) where expert takes action in the \( \mathcal{CA} – \mathcal{MDP} \). Usefulness of this extension and the procedure to realize it have already been discussed.

Other extension is related to addition of advisor within the MLIRL algorithm. Inclusion of advisor helps MLIRL formalism to interact with the expert and guide \( I_0 \) for generating better behaviour. Role of advisor for generating state and action space within \( \mathcal{CA} – \mathcal{MDP} \) has already been discussed.

Reward functions in MLIRL setting are parameterized by a vector of reward weights \( \mathcal{W} \) applied to a feature vector for each state–action pair \( \phi(s, a) \) [1]. Thus, a reward function can be written as \( r_{\phi}(s, a) = \mathcal{W}^T \phi(s, a) \). Suppose, expert’s reward function is \( \mathcal{W}_E \). Now, \( I_0 \) has to use information from observed trajectories to infer parameters regarding \( \mathcal{W}_E \). Expert may be considering various features and assign weights to them differently which give rise to a particular trajectory. \( I_0 \) can only hypothesize about these weights (say, \( \mathcal{W}_A \) ) of various features which maximize the likelihood of the demonstrated trajectories. For detailed understanding of the extended MLIRL agent, it is assumed that the readers are well aware of value iteration, Q-learning, linear function approximation and maximum likelihood estimation. We briefly cover the relation among these entities so that the readers can understand the working principle of extended MLIRL agent reported in this work.

Equation 1 is presented to show that the value function of a state can be represented through a linear function of state features. For Q-values, we have equation 2. To solve a MDP, we need to find a policy and it is related with the Q-values through equation 3.

\[
V(s) = \mathcal{W}_0.1 + \mathcal{W}_1.\phi_1(s) + \mathcal{W}_2.\phi_2(s) + ... + \mathcal{W}_n.\phi_n(s) 
\]

\[
Q(s, a) = \mathcal{W}_0.1 + \mathcal{W}_1.\phi_1(s, a) + \mathcal{W}_2.\phi_2(s, a) + ... + \mathcal{W}_n.\phi_n(s, a) 
\]

\[
\pi(s) = \arg\max_a Q(s, a) 
\]

In finding the policy through linear function approximation technique, we need to find out the updates in terms of reward weights \( \mathcal{W} \) (see equation 2). Equation 4 is related to the standard Q-Learning update [43]. The updates will be in terms of parameters of the reward function. Equation 4 can be re-written as equation 5 where the “max” operator has been replaced via Boltzmann exploration [44]. This step allows one to make the likelihood differentiable [1]. The weight of the reward function parameters at each step will help one to find out an estimate of the Q function. This can be used to solve the \( \mathcal{CA} – \mathcal{MDP} \) through equation 6.

\[
Q(s, a) = \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma \max_a Q(s', a')] 
\]

\[
Q_{\mathcal{W}_A}(s, a) = \mathcal{W}_A^T \phi(s, a) + \gamma \sum_{s'} T(s, a, s') \mathcal{W}_A(s', a') 
\]

\[
\pi_{\mathcal{W}_A}(s, a) = e^{\beta Q_{\mathcal{W}_A}(s, a)} / \sum_a e^{\beta Q_{\mathcal{W}_A}(s, a)} 
\]

Next, we shift our focus to maximum likelihood estimation whose goal is to find set of parameters given by equation 7.

\[
\tilde{\mathcal{W}} = \arg\max_\theta P(D|M, f, \mathcal{W}) 
\]

Thus, \( \tilde{\mathcal{W}} \) is an estimate of experts reward weight, \( \mathcal{W}_E \). Within extended MLIRL setting, \( f(\mathcal{W}) \) is expert’s reward function. We are only able to find an estimate of the expert’s reward function, \( f(\mathcal{W}) \). This estimate is related to the process of finding optimal policy in our \( \mathcal{CA} – \mathcal{MDP} \). From equation 7, it is visible that we want to find reward function parameters that fits the given trajectories \( D \). After finding \( Q_{\mathcal{W}_A} \) and \( \pi_{\mathcal{W}_A} \) at each step, we can compute the log likelihood of a particular trajectory. Afterwards, \( \tilde{\mathcal{W}} \) can be determined by employing gradient ascent to maximize the probability of the given trajectories.
From the above discussion, one can observe that the main principle behind MLIRL is kept intact in our setting. However, introduction of $\mathcal{A} - \mathcal{MDP}\tau$ and advisor along with object affordance leads to an extension to the original MLIRL algorithm [1]. One key difference with standard MLIRL algorithm is that advisor provides one shot initial advice $A_0$. Thus, $I_a$ does not start with random weights corresponding to the reward vector. Reward shaping through $A_0$ facilitates faster convergence rates. Apart from this, for a robot to learn new diverse set of tasks, one needs to think of features which could adapt to various demonstrations involving different complex tasks. We believe that object affordance could be a good candidate for this purpose. Linking object affordance in the IRL paradigm could reveal the true power of object affordances.

Control loop of extended MLIRL agent $I_a$ over the proposed pipeline (Figure 1) is written in the following abstract steps. Here, $I_a$ has to assign weights of object affordance based parameters in such a way that the likelihood of the observed trajectory reaches the maximum.

**Step 1.** Acquire $A_s$ from the advisor.

**Step 2.** Generate trajectories ($E_1, E_2, \ldots, E_T$) in the form of <state, action> pairs with the help of SCAE and advisor through $A_s$ and $A_d$). This procedure adheres to $\mathcal{A} - \mathcal{MDP}\tau$ formalism.

**Step 3.** Acquire set of trajectories ($E_1, E_2, \ldots, E_N$) corresponding to high level intent, $I_i$ (where $i \leq 10$).

**Step 4.** Initialize video domain with respect to $\mathcal{A} - \mathcal{MDP}\tau$ (transition probability matrix) and object affordance features $\phi$.

**Step 5.** Initialize reward weights through the advisor input (one shot initial advice, $A_s$).

**Step 6.** For each <state, action> pair corresponding to each trajectory ($E_j$ where $j \leq N$); MDP Solver (Figure 1): a) solves $\mathcal{A} - \mathcal{MDP}$, b) compute log likelihood of trajectory ($E_j$) and c) perform gradient ascent procedure.

To successfully complete each of these sub-steps, MDP Solver communicates with SCAE and advisor frequently.

**Step 7.** Store learnt weights corresponding to the parameters of the reward function, $\mathcal{W}_A$. Thus, $I_a$ generates $\mathcal{W}_A$ with respect to unknown $W_E$. HIR $H_a$ utilizes $\mathcal{W}_A$ for its own purpose.

**Step 8.** Goto step 3 for demonstrations corresponding to high level intent, $I_{i+1}$.

### 5.4 AHP based HIR

AHP [41] has been widely used in multi-attribute decision making. HIR $H_a$ is based on AHP to find out the probable intents from the set of available intents. Multiple attributes in our setting are the object affordances. It is assumed that human assigns various weights to these attributes to execute a particular intent.

HIR $H_a$ has to work on the learnt reward function to find out the probable human intents. More specifically, various $W_A$s are present for different demonstrations corresponding to various high level intents. These are obtained from extended MLIRL agent $I_a$ by instantiating the novel pipeline (Figure 1). Advisor provides $A_i$ to the $H_a$. This helps HIR $H_a$ for pairwise comparisons between attributes (preferences / criteria) and alternatives (intents). Apart from this advice, annotations corresponding to high level intents have not been utilized by $I_a$.

Following equation gives the overall procedure followed to rank various intents to identify probable intents

$$R_i = a_{ij}w_j$$

where, $R_i$ is the rank of the $i$th intent (alternatives), $a_{ij}$ is the actual value of the $i$th intent (alternative) in terms of the the $j$th criterion (object affordance) by considering other intents as well, and $w_j$ is the weight (importance) of the $j$th object affordance. Please note that $w_j$ is obtained dynamically from $\mathcal{W}_A$ (extended MLIRL agent $I_a$) corresponding to $I_i$. On the other hand, $a_{ij}$ is obtained by the first two steps of the standard procedure involved with respect to AHP. $R_i$ corresponds to $\psi^\ast$ in the problem formulation.

### 6 Experimental evaluation

#### 6.1 Dataset description

The experiments are conducted in the CAD-120 dataset [6]. To the best of our knowledge, this is the only publicly available benchmark dataset for object affordance analysis from video. Ground-truth temporal segmentation is assumed to be available for our approach. Higher level activity labels are being considered as the human intents. No “intent” label is provided a priori to our computational approach. Few more information about this dataset are
given below:

- 4 experts (two female, two male, one left-handed) demonstrates high level activities.
- 10 high level activities are present in CAD-120. These are: arranging objects, cleaning objects, having a meal, making cereal, microwaving food, picking objects, stacking objects, taking food, taking medicine, unstacking objects.
- 9 base level actions are considered: reach, move, pour, eat, drink, open, place, close, scrub.
- 11 object affordance labels: reachable, movable, pourable, pourto, containable, drinkable, openable, placeable, closable, scrubbable, scrubber.

6.2 Results

Before we present the results, a note on certain assumptions underlying the experimental evaluation is prudent. During the dataset description, we have mentioned that 11 object affordance labels are present. However, \( I_a \) provides information regarding 9 object affordances only.

Assumption 1: “Pourable” and “scrubbable” have been eliminated as presence of “pourto” and “scrubber” can nullify their presence.

Stand-alone SCAE is able to infer object affordances from video data. It can produce 100% accuracies for 5 affordance labels over CAD-120 dataset. Following assumptions have been put forward to maintain uniformity of the experiment.

Assumption 2: We assume that SCAE can also recognize remaining 4 object affordances with 100% accuracy.

Assumption 3: SCAE is assumed to detect key object affordances from the set of activated object affordances corresponding to a particular temporal segment.

Following assumption is related to handling of long sequence of high level activities.

Assumption 4: Ground truth temporal segmentation is being used from the dataset for generating the trajectories for \( I_a \).

SCAE is composed of three components: a) ontology processing module b) video processing module and c) MLN. Ontology processing module has been implemented in Java using OWL API 3.4.3 and Hermit 1.3.8.1 (reasoner). The video processing module has also been implemented in Java. Alchemy\(^1\) has been utilized to get the functionality of MLN. Advisor and MDP Solver components presented in the pipeline have been implemented in Java. Traditional BURLAP [45] has been modified to realize the extended MLIRL agent, \( I_a \).

Let us start with the first set of demonstrations corresponding to high level intent \( I_i \), where \( i = 1, i \leq 10 \) fed to our architecture. Figure 4a and Figure 4b corresponds to the output produced by the extended MLIRL agent \( I_a \) (grey colored graph) and HIR \( H_a \), respectively.

Thus, \( H_a \) is not able to clearly distinguish between three high level intents namely: a) arranging object, b) stacking object and c) unstacking object. However, \( I_a \) recovers the reward function parameters where it assigns positive weights (higher preferences) to three object affordances namely: a) movable, b) reachable and c) placeable.

Consider the coupled graphs shown in Figure 4c and Figure 4d. These two correspond to second set of demonstrations corresponding to high level intent \( I_i \), where \( i = 2, i \leq 10 \) fed to our architecture. It can be observed from Figure 4d, \( H_a \) is able to clearly recognize the high level intent, “cleaning object intention”. \( I_a \) also recovers the reward function parameters. From Figure 4c, one can see that \( I_a \) assigns positive weights (higher preferences) to five object affordances namely: a) reachable, b) scrubber, c) openable, d) closable and e) placeable.

Refer to Figure 4e and Figure 4f. These correspond to third set of demonstrations corresponding to high level intent \( I_i \), where \( i = 3, i \leq 10 \) fed to our architecture. Thus, \( H_a \) is able to clearly recognize the high level intent, “having meal intention”. According to Figure 4e, \( I_a \) recovers the reward function parameters where it assigns positive weights (higher preferences) to four object affordances namely: a) movable, b) reachable, c) placeable and d) drinkable.

Figure 5a and Figure 5b are obtained by feeding the fourth set of demonstrations to our architecture. This demonstration corresponds to high level intent \( I_i \), where \( i = 4, i \leq 10 \). Thus, it can be seen from Figure 5b that \( H_a \) is able to clearly recognize the high level intent, “making cereal intention”. \( I_a \) recovers the reward function parameters where it assigns positive weights (higher preferences) to four object affordances namely: a) movable, b) pourto and d) placeable.

Figure 5c and Figure 5d correspond to fifth set of demonstrations corresponding to high level intent \( I_i \), where \( i = 5, i \leq 10 \) fed to our architecture. Thus, \( H_a \) is able to

---

\(^1\) [https://alchemy.cs.washington.edu/](https://alchemy.cs.washington.edu/)
Figure 4: This figure consists of three pair of graphs originated from three categories of demonstrations of CAD-120 dataset. Each pair corresponds to a particular category of demonstration. The first graph (dark one) of each pair is obtained due to $I_a$ and it shows the learnt reward function. The second graph of each such pair is present in the right hand side of this graph (dark one). It is obtained due to $H_a$. It shows the probable intentions discovered by our approach. Various legends have been used for representing the attribute values of both the graphs of each pair. In Figure 4a and Figure 4b, the meaning of the legends (markers) have been provided in the figure itself. The legends in Figure 4c and Figure 4e carries the same meaning as annotated in Figure 4a. Similarly, the meaning of the legends in Figure 4d and Figure 4f carries the same meaning as annotated in Figure 4b.
Figure 5: This figure is similar to Figure 4. It consists of three pair of graphs originated from three categories of demonstrations of CAD-120 dataset. However, these three categories are different from the Figure 4. In Figure 4a and Figure 4b, the meaning of the legends (markers) have been provided in the figure itself. The legends in Figure 5a, Figure 5c and Figure 5e carries the same meaning as annotated in Figure 4a. Similarly, the meaning of the legends in Figure 5b, Figure 5d and Figure 5f carries the same meaning as annotated in Figure 4b.
Figure 6: This figure is similar to Figure 4. It consists of three pairs of graphs originated from three categories of demonstrations of CAD-120 dataset. However, these three categories are different from Figure 4 and Figure 5. In Figure 4a and Figure 4b, the meaning of the legends (markers) have been provided in the figure itself. The legends in Figure 6a, Figure 6c and Figure 6e carries the same meaning as annotated in Figure 4a. Similarly, the meaning of the legends in Figure 6b, Figure 6d and Figure 6f carries the same meaning as annotated in Figure 4b.
clearly recognize the high level intent, “microwaving food intention”. \(I_a\) recovers the reward function parameters where it assigns positive weights (higher preferences) to five object affordances namely: a) movable, b) reachable, c) openable, d) closable and e) placeable. There is a close contest between “microwaving food intention” and “taking food intention” in this set of demonstrations.

Figure 5e and Figure 5f are obtained after feeding sixth set of demonstrations to our architecture. Both of them correspond to high level intent \(I_i\), where \(i = 6, i \leq 10\). Thus, \(H_a\) is able to clearly recognize the high level intent, “picking object intention”. \(I_a\) recovers the reward function parameters where it assigns positive weights (higher preferences) to two object affordances namely: a) movable and b) reachable.

Refer to Figure 6a and Figure 6b. These two figures correspond to seventh set of demonstrations fed to our architecture. It correspond to high level intent \(I_i\), where \(i = 7, i \leq 10\). From Figure 6b, it can be observed that \(H_a\) is not able to clearly recognize the high level intent, “stacking object intention”. \(H_a\) assumes “picking object intention” is the underlying intent behind this set of demonstrations. \(I_a\) recovers the reward function parameters where it assigns positive weights (higher preferences) to four object affordances namely: a) movable, b) openable, c) placeable and d) closable. Video frames involved in this demonstrations consists of a person taking out his / her food from the microwave. Thus, conceptually similar scene triggers this recognition process.

Figure 6e and Figure 6f correspond to ninth set of demonstrations corresponding to high level intent \(I_i\), where \(i = 9, i \leq 10\) fed to our architecture. Figure 6e and Figure 6f are itself self-explanatory to track the performance of \(I_a\) and \(H_a\) respectively. Thus, \(H_a\) is able to clearly recognize the high level intent, “taking medicine intention”. \(I_a\) recovers the reward function parameters where it assigns positive weights (higher preferences) to two object affordances namely: a) movable, b) openable, c) placeable, d) drinkable and e) containable.

The coupled graph (shown in Figure 7) corresponds to tenth set of demonstrations corresponding to high level intent \(I_i\), where \(i = 10, i \leq 10\) fed to our architecture. Observing the right hand side of Figure 7, we see that \(H_a\) is not able to clearly recognize this high level intent (unstacking object intention). It assumes the underlying intent behind this demonstrations is “picking object intention”. On the other hand, \(I_a\) (see the left hand side of Figure 7) recovers the reward parameters where it
Table 1: Correctly recognized and incorrectly recognized intents based on proposed approach.

<table>
<thead>
<tr>
<th>Correctly recognized intents</th>
<th>Incorrectly recognized intents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cleaning object intention</td>
<td>Arranging object intention</td>
</tr>
<tr>
<td>Having meal intention</td>
<td>Stacking object intention</td>
</tr>
<tr>
<td>Making cereal intention</td>
<td>Taking food intention</td>
</tr>
<tr>
<td>Microwaving food intention</td>
<td>Un-stacking object intention</td>
</tr>
<tr>
<td>Picking object intention</td>
<td></td>
</tr>
<tr>
<td>Taking medicine intention</td>
<td></td>
</tr>
</tbody>
</table>

assigns positive weights (higher preferences) to three object affordances namely: a) movable and b) reachable.

Figure 8 pictorially depicts the performance of our HIR over 10 categories of demonstrations. Hidden intents present within those input demonstrations are shown in the X axis. For a particular demonstration presented in Figure 8,

![Figure 8: Performance over 10 categories of demonstrations.](image)

the vertical width of each color segment is associated with the activation of the corresponding human intent. In the positive direction of Y axis, the more the vertical width of a particular color segment, the more will be its chance to be called as the detected intention. On the negative direction of Y axis, the lesser the vertical width, the more will be its chance to be called as the detected intention. HIR is able to correctly recognize human intents hidden in six different categories of demonstrations. Intentions such as cleaning object, having meal, making cereal, microwaving food, picking object and taking medicine can be correctly inferred from demonstration 2, demonstration 3, demonstration 4, demonstration 5, demonstration 6 and demonstration 9 respectively. Six intentions have been recognized correctly. However, HIR is not able to recognize four intents hidden inside four categories of demonstrations (demonstration 1, demonstration 7, demonstration 8 and demonstration 10). For example, the output shown by our proposed approach corresponding to input demonstration 1 can be visually observed from Figure 8. Stacking object, unstacking object and arranging objects have equal vertical width in the positive direction. This concludes that our proposed approach is not able to infer the hidden intent from input demonstration 1. Table 1 shows correctly and incorrectly recognized intents over CAD-120 [6].

6.3 Discussion

Our work is towards providing an end-to-end inverse reinforcement learning pipeline from high dimensional demonstrations. Appropriateness of constructing conceptual abstraction through symbolic representation has been tested with respect to IRL systems working with high dimensional continuous domain. Symbolic RL has been explored in high dimensional demonstrations by Garnelo et al. [3]. However, our problem formulation along with proposed solution is unique in the sense that symbolic IRL has not been studied in high dimensional demonstrations. The recovered object affordance based reward function can rightly point to high level intent with 60% accuracy. This is when we have not utilized high level intent labels a priori. It shows that the effectiveness of the extended MLIRL agent.

To evaluate our presented approach, we aim to answer the following three basic questions.

Q1: How does the presented knowledge representation scheme avoids including all possible qualitative relations (QDC based) between objects and human-objects which could result in a very large state space?

The knowledge representation scheme exploits the conceptual neighbourhood of the QDC and only includes possible qualitative relations which are conceptual neighbours of a given relation instead of all the QDC relations.

Q2: Are the advices provided by the advisor effective?

There is no specific quantitative measure whose value can signify the effectiveness of adding advisor. During experimentation, it is found that inclusion of $A_d$ helps us in convergence. Presence of $A_d$ and $A_s$ definitely allows the extended MLIRL agent $I_a$ to work on less number of states. The number of states in an MDP has big impact on
Q3: How does the available ground truths and assumptions made in the reported work impacts the effectiveness of the overall proposed solution?

First three assumptions are related to object affordances. We strongly believe that these assumptions can be relaxed. In this reported work, readers might observe that intricate details about SCAE has been avoided. The focus is to understand the importance of object affordance in extending MLIRL to high dimensional states. Unrolling the sub-modules of SCAE would force us to explain each module separately and the results of SCAE need to be evaluated further. To avoid clutter, we make these three assumptions. It allows the readers to focus on the extended MLIRL agent in a better way. It also helps us to address the research question in an organized manner.

The last assumption is about ground truth temporal segmentation. QSR has been used widely in handling noise in various applications involving indoor environments (household domain [47]) as well as outdoor environments (airport domain [48]) which consists of real world videos. Although, we have assumed ground truth temporal segmentation in our reported work; we argue that the set of qualitative distance relations will help to discriminate the temporal segmentations of input videos. Work reported by Tayyub et al. [47] achieves automatic temporal segmentation of sub-activities in CAD-120 data set by adopting QSR approach.

Apart from these three questions, there are two more general questions which may help the readers to judge the presented work.

Q4: Why is Inverse Reinforcement Learning chosen instead of going for supervised or unsupervised learning paradigms for recognizing intents?

Human learn to perform a task through interacting with the environment. Human intent recognition problem can be casted as a classification problem and study either in the supervised learning paradigm or unsupervised learning paradigm. However, human learning capability inspires our approach and lead us to study it within the IRL paradigm. This approach further will inspire the IRL researchers to build robots which can learn complex manipulation tasks.

Q5: Why is AHP utilized to rank various human intents instead of MLIRL agent handling this task as well?

Our interest to build an object affordance driven IRL system allows us to scrutinize the role of sequence of activated object affordances in visualizing an expert trajectory. It further helps us to find out conceptually similar intents such as “taking food” and “microwaving food”. Decoupled solution in finding out multiple intentions is also exercised in [1]. Utilization of AHP in our approach may further turn on the discussion that multi-objectivization of human intent recognition problem is necessary?

7 Conclusion

Before learning the complex manipulation skills for performing similar activity, household robot has to interpret human instructions conveyed through visual input. Robots should also be capable enough to understand human’s conceptual abstraction ability in handling high dimensional visual input. In this work, traditional MLIRL algorithm proposed by Babes et al. [1] is extended through a novel pipeline. This object affordance driven novel pipeline allows us to: a) show that symbolic conceptual abstraction helps to curb curse of dimensionality issue present in high dimensional demonstration, b) prove that the sequence of object affordance can directly link to higher level behaviours such as intents, c) include advisor within MLIRL formalism which in turn helps to facilitate interaction between the learning algorithm and the expert and, d) show the importance of object affordance as a transferable knowledge from the demonstrations. One weakness of this approach is that deterministic transition has been considered here. This work assumes that there is an automatic and correctly working grounding classifier for the IRL agent. The working of this classifier has not been demonstrated explicitly for state and action identification. In the future work, we plan to extend this work by describing the symbolic planner in detail for a generic IRL agent in relational domain.

Acknowledgement: The authors would also like to thank Zubin Bhuyan, University of Massachusetts Lowell, for the discussion regarding IRL.

References


[34] D. Xie, T. Shu, S. Todorovic, S.-C. Zhu, Modeling and inferring human intents and latent functional objects for trajectory prediction, 2016, arXiv1606.07827
[37] A. G. Cohn, S. M. Hazari, K. Arulkumaran, Qualitative spatial representation and reasoning An overview, Fundamenta Informaticae, 2001,
[38] E. Clementini, P. Di Felice, D. Hernández, Qualitative representation of positional information, Artificial intelligence, 1997, 95(2), 317–356


