Introduction

We aim to develop artificial agents which can
• Learn interpretable objectives (through language and behavior) [1]
• Behave competently with respect to these objectives, even when they conflict [2]
• Explain their behaviors to human teammates in terms of these objectives (and correct objectives or world models if needed)

We have developed a system which constructs explanations for the behavior of a multi-objective linear temporal logic (LTL) planning agent operating in a Markov decision process (MDP).

Test scenario: ShopWorld

• Agent is a robot sent to go shopping for its user in a store selling a watch
• User wants the watch, but gives the robot insufficient money to afford it

Linear temporal logic (LTL) [5]

A simple propositional logic encoding time
\[ \phi ::= p \mid \neg \phi_1 \mid \phi_1 \lor \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid X\phi_1 \mid G\phi_1 \mid F\phi_1 \]
where \( p \) a proposition, \( \phi_1, \phi_2 \) LTL statements.

\[ \begin{align*}
X\phi_1 & : \text{in the next time step, } \phi_1 \\
G\phi_1 & : \text{in all present and future time steps, } \phi_1 \\
F\phi_1 & : \text{in some present or future time step, } \phi_1 \\
\phi_1 U\phi_2 & : \text{true will be until } \phi_2 \text{ becomes true}
\end{align*} \]

LTL objectives in ShopWorld

“Leave the store while holding the watch”
\( F(\text{leaveStore} \land \text{holding}) \)

“Do not leave the store while holding anything you have not bought”
\( G(\neg \text{leaveStore} \land \neg \text{holding} \land \neg \text{bought}) \)

Multi-objective LTL planning

We define a multi-objective LTL planning problem as a tuple
\[ \mathcal{P} = (\mathcal{M}, \phi, w, z) \]
where
\[ \mathcal{M} = (S, A, P, R, \gamma) \] is a Markov decision process
\[ \phi = \phi_1, \ldots, \phi_n \] a set of (syntactically safe/co-safe) LTL objectives
\[ w, z \in \mathbb{R}^n \] contain the weight \( w_i \) and priority \( z_i \) respectively of \( \phi_i \), specifying preferences among objectives.

Objectives with the same priority are traded off using weights.
Objectives with different priorities take precedence (lexicographic ordering).

Basic solution approach:
• Compile each objective \( \phi_i \) into a finite state machine (FSM)
• Construct a new “product” MDP \( M^F \) whose state space is \( S \times Q^{\phi_1} \times \cdots \times Q^{\phi_n} \) which accepts only on “good prefixes” of \( \phi \)
• Construct a new “product” MDP \( M^F \) whose state space is \( S \times Q^{\phi_1} \times \cdots \times Q^{\phi_n} \) which accepts only on “good prefixes” of \( \phi \)
• Construct a new “product” MDP \( M^F \) whose state space is \( S \times Q^{\phi_1} \times \cdots \times Q^{\phi_n} \) which accepts only on “good prefixes” of \( \phi \)
• Construct a new “product” MDP \( M^F \) whose state space is \( S \times Q^{\phi_1} \times \cdots \times Q^{\phi_n} \) which accepts only on “good prefixes” of \( \phi \)
• Construct a new “product” MDP \( M^F \) whose state space is \( S \times Q^{\phi_1} \times \cdots \times Q^{\phi_n} \) which accepts only on “good prefixes” of \( \phi \)

Planning in ShopWorld

Given the LTL objectives above (and assuming the anti-shoplifting objective has higher priority), the agent performs only the single action \( \text{leaveStore} \).

Explanation structures

Consider an agent who has acted according to trajectory \( \tau \). We wish to answer questions of the form “Why \( \phi \)?”, where \( \phi \) is an arbitrary (safe/co-safe) LTL formula.

In response to such a question, we construct an explanation structure \( (\mathcal{G}, E, \tau', E') \) where
• \( \mathcal{G} \in \{ \text{QUERYFALSE}, \text{NEQQUERYIMPOSSIBLE}, \text{ALTVQUERY} \} \)
• \( \tau' \) is either a trajectory, or the empty set
• \( E \) contains one or more pairs \( (\phi, \text{EVIDENCE}(\tau, \phi)) \) where \( \phi \) is an LTL statement
• \( \text{EVIDENCE}(\tau, \phi) = \min \{|E|: E \subseteq \{0, \ldots, T\} \times I(\mathcal{G})\} \)
for all \( \tau' \) s.t. \( \tau' \models \tau, \tau' \not\models \text{Traj}(\phi) \)

In ShopWorld,
\[ \text{EVIDENCE}(\tau, \text{F(leaveStore } \land \text{holding)}) = \{0, \neg(\text{holding}), \neg(\text{holding})\} \]

\( E' = E \) as \( E \), but for \( \tau' \)

Future work

• Incorporating explicit causal models (esp. in \( \text{NEQQUERYIMPOSSIBLE} \))
• Tailoring explanations to interactant knowledge
• Adapting to stochastic environments
• Need to represent multiple trajectories or probability distribution
• Improving efficiency of planner
• Impropractical for nontrivial domains
• Dropping assumption that agent has perfect knowledge of transition dynamics

Acknowledgments

This project was supported in part by NSF IIS grant 43520050 and by NASA grant C17-2D00-TU.

References