Narrative Planning for Belief and Intention Recognition

Rachelyn Farrell, Stephen G. Ware. *Proceedings of the Sixteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (2020)

Presenter: Cameron Egbert

Belief and Intention Recognition

- Planning - find a sequence of actions for an agent to achieve a goal
- Plan recognition - given a partial sequence of actions we have observed an agent taking, infer the unobserved actions.

- Similar to goal recognition

- “What beliefs and intentions would be required to explain the actions we observe agents taking?”

Plan Recognition as Planning (PRP)

- Ramírez and Geffner (2009)
- Plan and goal recognition by constraining a planner to include all the agent’s observed actions in its solutions.
- demonstrates unobservable qualities (an agent’s goals and plans), can be inferred through in a plan-based environment
- designed for rational agents and assumes full observability

- Farrell and Ware are interested in human-like agents with limited knowledge

Narrative Planning Framework – $\langle P, A, O, C, s_0, g \rangle$

P - Propositional Fluents

A - Action Templates

Pre(a), Eff(a), Act(a), Obs(a)

O - Objects

C - Constants representing characters with beliefs and intentions

s_0 - Initial State

g - Goal

Beliefs and Intentions

modal predicates:

- $b(c, p)$ = character c believes proposition p.
- $i(c, p)$ = c intends p.
- $b(c, b(d, p))$ = “c believes that character d believes p.”
- $b(c, i(d, b(c, p))$ = “c believes that d intends for c to believe p.”
- $\neg b(c, p)$ is equivalent to $b(c, \neg p)$
- $\neg i(c, p)$ is NOT equivalent to $i(c, \neg p)$

Actions

Let α(a, s) to refer to the state that results from applying action a to state s

Let β(c, s) to refer to the set of beliefs for character c in state s

When an action a occurs:

- $\forall c \in \text{OBS}(a) \colon \beta(c, \alpha(a, s)) = \alpha(a, \beta(c, s))$
- $\forall c \notin \text{OBS}(a) \colon \beta(c, \alpha(a, s)) = \beta(c, s)$
- $\forall c \in C : \exists b(c, p) \in \text{EFF}(a) \implies p \in \beta(c, \alpha(a, s))$

Explained Actions

Definition 1. An action a is explained iff $\forall c \in \text{ACT}(a)$: a is *explained for* c in the state before a.

Definition 2. An action a is *explained for* $c \in \text{ACT}(a)$ in state s iff there exists a sequence of actions π that starts with a and meets the following criteria when taken from $\beta(c, s)$:

1. There exists a proposition p that holds at the end of π and $\forall a' \in \pi : \neg p \wedge i(c, p)$ holds in the state before a'.
2. $\forall a' \in \pi : \text{PRE}(a')$ holds in the state before a'.
3. $\forall a' \neq a \in \pi : \forall c' \in \text{ACT}(a') : a'$ is explained for c' in the state before a'.
4. π contains no sub-sequence that also meets these criteria. (This enforces π's causal coherency; it cannot be used to explain a if any step is redundant or unnecessary.)[1]

Belief and Intention Recognition - Method

Begin with: narrative planning problem, a set of possible candidates, and an observation sequence

Step 1: Transform the problem

- For each observed action in the sequence add a fully grounded action to the domain
- Add unique effects to actions in the sequence
- Add those effects to the preconditions of subsequent actions and to the goal
    - This preserves order

Belief and Intention Recognition - Method

Candidates contains both a set of beliefs and a set of intentions, each possibly empty

Step 2 - Produce a new problem for each candidate.

- Create a new problem with each candidate:
  add its beliefs and intentions to the initial state of the transformed problem.

Step 3 - Generate classical solutions but track explanations

Step 4 - Identify valid candidates

- Candidates with solutions having a maximal amount of explained *observed actions* with minimal *unexplained actions* given all candidates and solutions.

Theoretical Evaluation - Environment

Each agent:

- wants to have one item
- loves one other agent
- wants the agent they love to have their item

Each agent can:

- put items down/pick items up
- give items to one another
- trade items
- tell another what item they want (can be a lie)

Theoretical Evaluation - Dataset

- Generated 20 different initial states - randomized infatuation and locations

- Generate 10 candidates for each initial state - random wants and beliefs
- Compute a valid solution for each candidate
  - Goal: 2 randomly selected agents achieve their goal
  - Solutions are limited to 5 steps
  - Candidates with no solutions are resampled

- Each of the 20 * 10 = 200 sequences are ablated in 5 ways for 1000 sequences total
  - First 33%, First 67%, First 100%, Single Room Observations, Single Agent Observations
  - Empty Sequences are Discarded

Theoretical Evaluation - Results

242 out of the 855 sequences yielded only the candidate used for generation

22 out of the 855 sequences yielded valid sets with all 10 candidates

- accuracy (small valid candidate sets) is not necessarily desirable

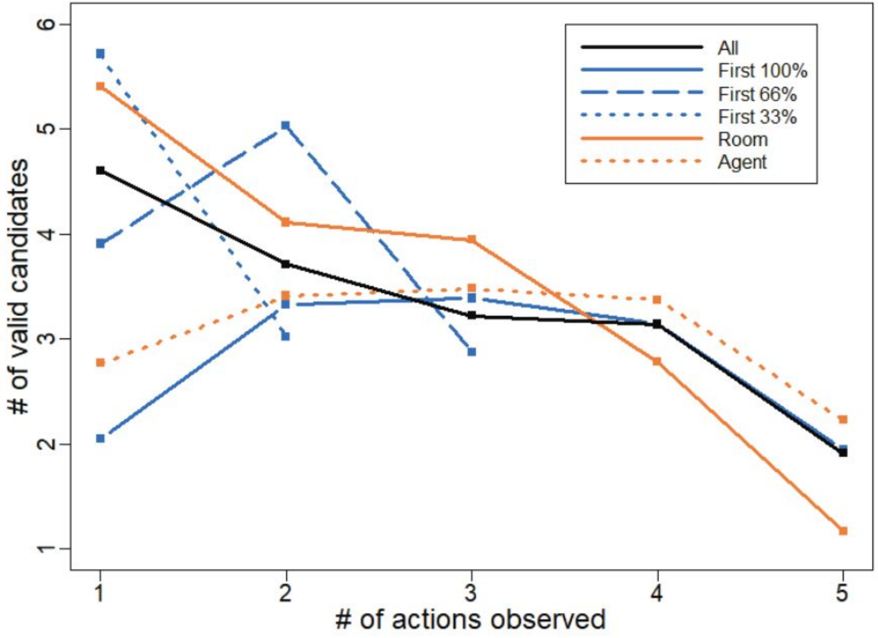

Figure 1: Valid candidates by number of observations

Practical Evaluation - Experimental Setup

- 33 different sequences generated by humans playing a short simulation
- trained raters identify the player's beliefs and intentions.
- Theses responses are compared to those produced by the algorithm.

- Beliefs: *cover, distance, point*
- Intentions: *surrender, threatened, kill*

- 8 * 6 = 48 candidates

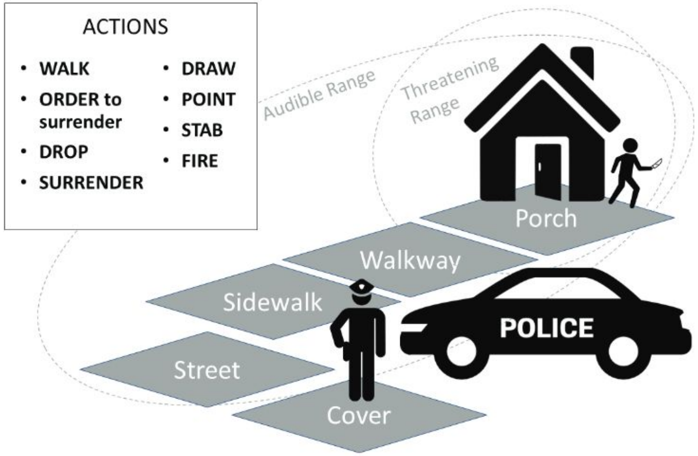

Figure 2: Police Use of Force Domain

Practical Evaluation - Evaluation and Results

- 33 sequences with 8 or fewer steps
- Classical solutions were generated with depth-limited BFS

- Moderate agreement between the three raters (Krippendorff's $\alpha = 0.5460$)

Evaluation 1:

- Used only the sequences for which at least 2 raters completely agree (12/33)
- The ratings were contained in the yielded valid candidates (8/12) times

Practical Evaluation - Evaluation and Results

Evaluation 2:

- Used the set of candidates deemed plausible by the raters
- For each feature the majority agreed on, candidates that disagreed were discarded
- The average size of these plausible candidate sets was 4
- The algorithm is considered correct if the majority of the candidates in this set are among those it returns
- This happened (20/33) times
- A random selection has a 1.2% chance of success

Conclusion

Encouraging results

Results are likely affected negatively by experimental setup.

- i.e. clear feedback when
- Much of the error may be due to the small number of raters, and the algorithm failing to identify features that were only accidentally agreed upon
- The authors believe they would have achieved higher accuracy had they searched one depth higher than the maximum sequence length

Fin.