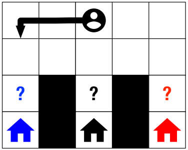
Goal Recognition as Reinforcement Learning

Leonardo Amado[1], Reuth Mirsky, [2,3], Felipe Meneguzzi [4]
[1] Pontifícia Universidade Católica do Rio Grande do Sul, Brazil
[2] Bar Ilan University, Israel
[3] The University of Texas at Austin, USA
[4] University of Aberdeen, Scotland

March 16, 2022

- Goal recognition (GR) is the task of inferring the goal of an *actor* based on a sequence of observations.
  i.e., the goal that best explains a sequence of observations of its actions
  - Related to plan recognition, i.e. recognizing a *top-level* action
  - A specific form of the problem of abduction



- Most GR approaches rely on specifications of the environment dynamics
- There are several limitations to this process:
  - Cost of Domain Description.
  - Susceptibility to Noise.
  - Online Costs.

Approach

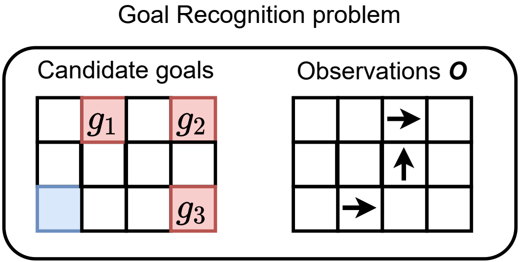
- We develop a set of RL-based approaches to address these limitations
- We replace manually crafted representations with model-free Reinforcement Learning (RL) techniques.
- The resulting approaches perform efficient and noise-resistant GR without the need to craft a domain model.

- Our contributions are threefold:
    1. We revisit the GR problem definition to accommodate RL-based domains;
    2. A first instance of the formulation of GR as RL;
    3. We evaluate the resulting techniques on domains with partial and noisy observability.

Definition (Goal recognition problem)

Given a domain theory $\mathbb{T}_Q(\mathcal{G})$ or $\mathbb{T}_\pi(\mathcal{G})$ and a sequence of observations $\boldsymbol{O}$, output a goal $g \in \mathcal{G}$ that **explains** $\boldsymbol{O}$.[a]

[a] Ramírez and Geffner, "Plan recognition as planning".

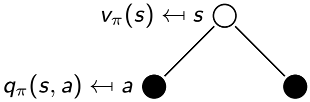


- Traditional goal recognition often assumes a deterministic environment
- Nevertheless, some approaches do allow for stochastic environments (MDPs)
  - Much harder to model stochastic environments by hand
- Reinforcement learning algorithms allow us to build informative functions describing a agent's preferences

	Input	Output
GR	$\mathbb{T}$, $\mathcal{G}$, $\boldsymbol{O}$,	$g \in \mathcal{G}$
Solve MDP	$M = (\mathcal{S}, \mathcal{A}, p, r)$	$\pi(a \mid s)$
Model-free RL	$\mathcal{S}, \mathcal{A}$	$\mathcal{Q}, Q(s, a)$

- Traditional goal recognition often assumes a deterministic environment
- Nevertheless, some approaches do allow for stochastic environments (MDPs)
  - Much harder to model stochastic environments by hand
- Reinforcement learning algorithms allow us to build informative functions describing a agent's preferences

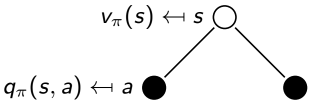
	Input	Output
GR	$\mathbb{T}$, $\mathcal{G}$, $\boldsymbol{O}$,	$g \in \mathcal{G}$
Solve MDP	$M = (\mathcal{S}, \mathcal{A}, p, r)$	$\pi(a \mid s)$
Model-free RL	$\mathcal{S}, \mathcal{A}$	$\mathcal{Q}$, $Q(s, a)$



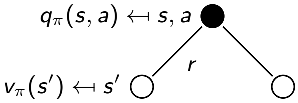
$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) q_\pi(s, a)$$

- Traditional goal recognition often assumes a deterministic environment
- Nevertheless, some approaches do allow for stochastic environments (MDPs)
  - Much harder to model stochastic environments by hand
- Reinforcement learning algorithms allow us to build informative functions describing a agent's preferences

	Input	Output
GR	$\mathbb{T}$, $\mathcal{G}$, $\boldsymbol{O}$,	$g \in \mathcal{G}$
Solve MDP	$M = (\mathcal{S}, \mathcal{A}, p, r)$	$\pi(a \mid s)$
Model-free RL	$\mathcal{S}, \mathcal{A}$	$\mathcal{Q}$, $Q(s, a)$



$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) q_\pi(s, a)$$



$$q_\pi(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')$$

Definition (Utility-based Domain Theory)

A utility-based domain theory $\mathbb{T}_Q(\mathcal{G})$ is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{Q})$ such that $\mathcal{Q}$ is a set of Q-functions $\{Q_g\}_{g \in \mathcal{G}}$.

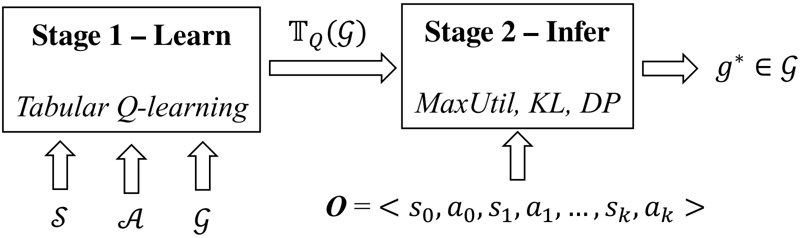
Definition (Policy-based Domain Theory)

A policy-based domain theory $\mathbb{T}_\pi(\mathcal{G})$ is a tuple $(\mathcal{S}, \mathcal{A}, \Pi)$ such that Π is a set of policies $\{\pi_g\}_{g \in \mathcal{G}}$.

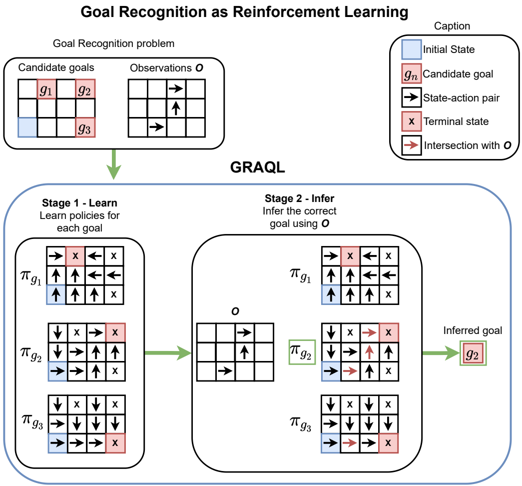
Goal Recognition Problem (new)

Definition (Goal Recognition Problem)

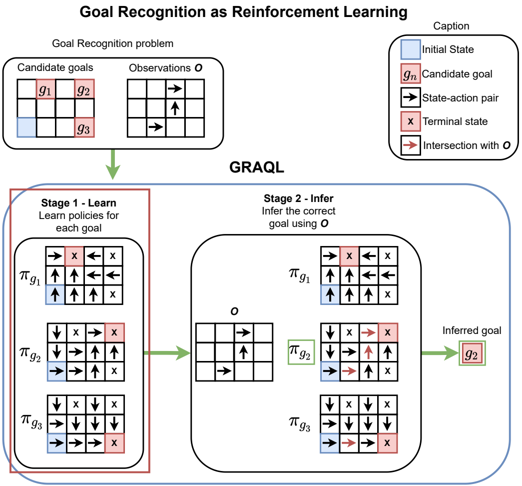
Given a domain theory $\mathbb{T}_Q(\mathcal{G})$ or $\mathbb{T}_\pi(\mathcal{G})$ and a sequence of observations $\boldsymbol{O}$, output a goal $g \in \mathcal{G}$ that **explains** $\boldsymbol{O}$.

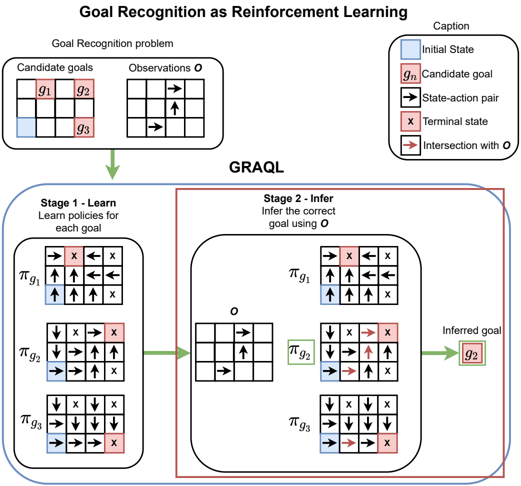

Stage 1 – Learn
Tabular Q-learning
$\mathbb{T}_Q(\mathcal{G})$
Stage 2 – Infer
MaxUtil, KL, DP
$g^* \in \mathcal{G}$
$\mathcal{S}$
$\mathcal{A}$
$\mathcal{G}$
$\boldsymbol{O} = < s_0, a_0, s_1, a_1, \ldots, s_k, a_k >$

GR as RL example



GR as RL example




Goal Recognition as Reinforcement Learning
Goal Recognition problem
Candidate goals
Observations O
Caption
Initial State
Candidate goal
State-action pair
Terminal state
Intersection with O
GRAQL
Stage 1 - Learn
Learn policies for each goal
Stage 2 - Infer
Infer the correct goal using O
Inferred goal

Here we provide a first implementation for this framework, called GRAQL.

- We use a standard tabular Q-learning algorithm
- Our goal is to learn informative domain theory with minimal effort.
- Reward for reaching the goal is 100, and 0 otherwise, and the discount factor is 0.9.
- Exploration is ϵ-greedy with linearly decaying values.

Shaping the initial policy can speed up the learning process: for each goal g, an optimal planner generates a single trajectory to the goal.

- Positive values for state-action pairs that are part of its goal's optimal path p_g.
- Similar to the original formulation of planning-based GR of Ramirez and Geffner.
- We don't use reward shaping for the results of this work.

Three distinct *distance* metrics inspired by three common RL measures:

1. MaxUtil,
2. KL-divergence,
3. Divergence Point.

Using these metrics, goal recognition reduces to the finding the minimal distance between actual observations Ω and the observations expected from the value/policy functions of each goal.

$$\mathrm{g}^{*} \leftarrow \operatorname*{arg\,min}_{g \in \mathcal{G}} \textsc{Distance}(Q_g, \boldsymbol{O})$$

MaxUtil is an accumulation of the utilities collected from the observed trajectory.

$$MaxUtil(Q_g, \boldsymbol{O}) = \sum_{i \in |\boldsymbol{O}|} Q_g(s_i, a_i) \tag{1}$$

KL-Divergence is a measure for the distance between two distributions, so we construct two policies, π_g and $\pi_{\boldsymbol{O}}$ for Q_g and $\boldsymbol{O}$ respectively.

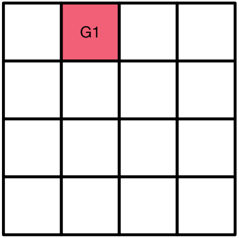
$$KL(Q_g, \boldsymbol{O}) = \mathrm{D}_{\mathrm{KL}}(\pi_g \parallel \pi_{\boldsymbol{O}}) = \sum_{i \in |\boldsymbol{O}|} \pi_g(a_i \mid s_i) \log \frac{\pi_g(a_i \mid s_i)}{\pi_{\boldsymbol{O}}(a_i \mid s_i)} \tag{2}$$

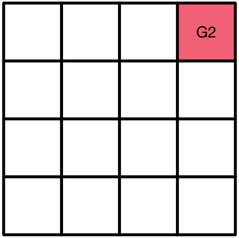
Divergence Point (DP) is a measure[1] of, given a trajectory $\boldsymbol{O}$ and a policy π, what is the minimal point in time in which the action taken by $\boldsymbol{O}$ has zero probability to be chosen by π.

$$DP(Q_g, \boldsymbol{O}) = -min\{t \mid \pi_g(a_{t-1} \mid s_{t-1}) \leq \delta\} \tag{3}$$

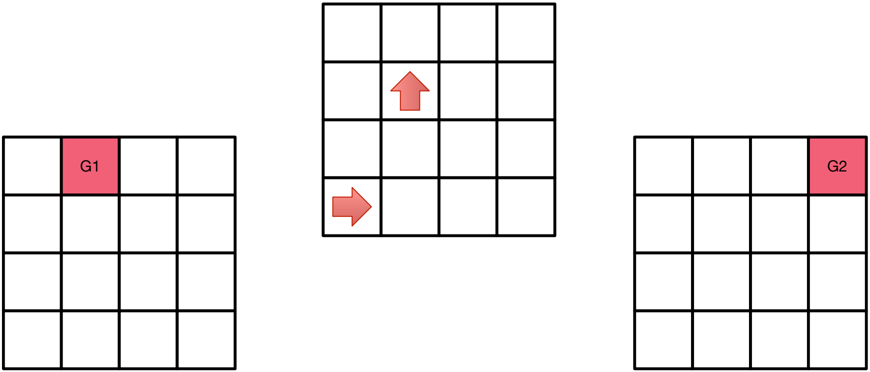
[1] Adapted from (Macke, Mirsky, and Stone, "Expected Value of Communication for Planning in Ad Hoc Teamwork")

MaxUtil Example

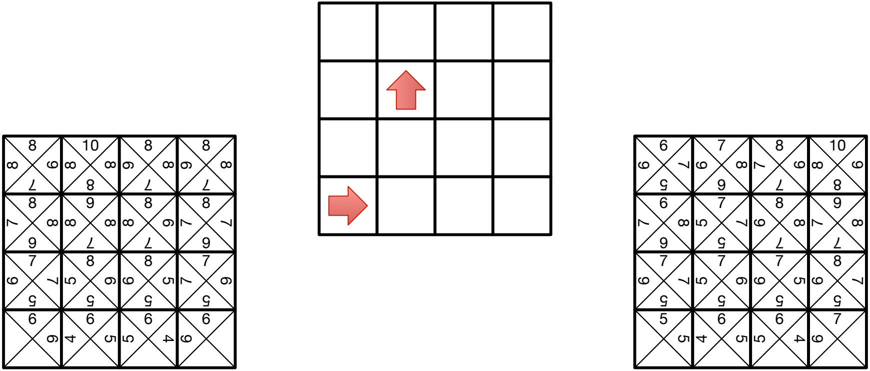




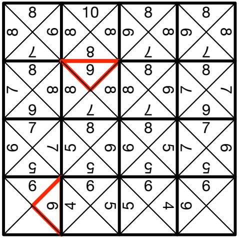
MaxUtil Example



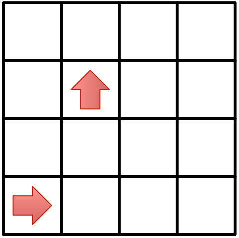
MaxUtil Example

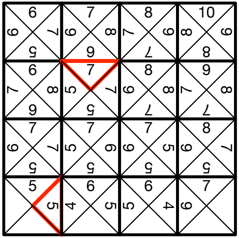


MaxUtil Example



$MaxUtil(Q_{g1}, \boldsymbol{O}) = 15$





$MaxUtil(Q_{g2}, \boldsymbol{O}) = 12$

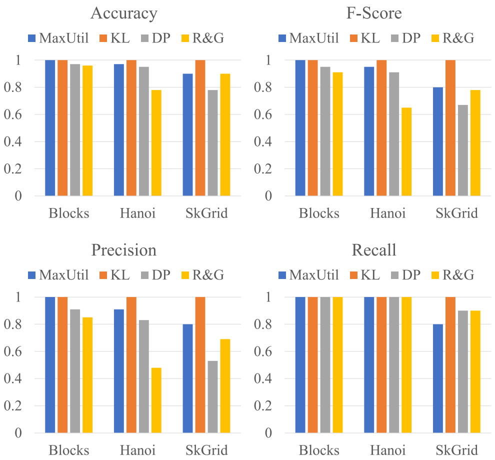
We use three domains from the PDDLGym library for their similarity with commonly used GR evaluation domains:

1. Blocks,
2. Hanoi,
3. SkGrid (highly resembles common GR navigation domains with obstacles)

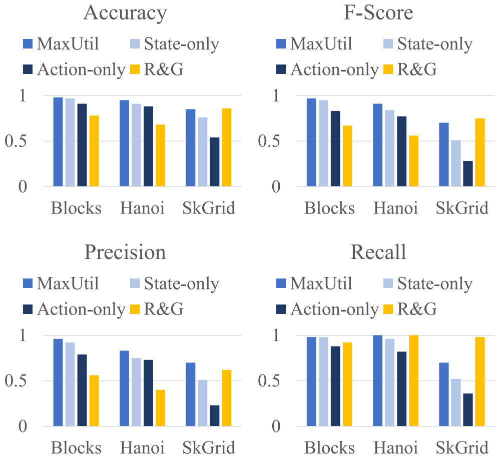
- For each domain, we generate **10** GR problems with **4** candidate goals. We manually choose ambiguous goals.
- Each problem has 7 variants, including partial and noisy observations. We have 5 variants with varying degrees of observability (10%, 30%, 50%, 70%, and full observability), and 2 variants that include noise observations with varying degrees of observability (50% and full observability).
- Our test set includes 210 GR problems, which we compare with R&G[2].

[2] Ramírez and Geffner, "Plan recognition as planning".

Results regarding full observability



Results with different types of observations



Snapshot of noisy results.

		Accuracy				Precision			
OBS	Domain	MaxUtil	KL Div	DP	R&G	MaxUtil	KL Div	DP	R&G
50%	Blocks	**0.95**	0.62	0.93	0.84	**0.95**	0.33	0.77	0.56
	Hanoi	**0.97**	0.90	0.93	0.68	**0.91**	0.80	0.77	0.38
	SkGrid	0.75	0.75	0.57	**0.88**	0.50	0.50	0.35	**0.64**
100%	Blocks	**1.00**	**1.00**	0.95	0.96	**1.00**	**1.00**	0.83	0.83
	Hanoi	**1.00**	**0.95**	0.90	0.78	**1.00**	0.90	0.71	0.48
	SkGrid	0.85	**0.95**	0.65	0.90	0.70	**0.90**	0.40	0.69
Avg	Blocks	**0.97**	0.81	0.94	0.90	**0.97**	0.60	0.80	0.70
	Hanoi	**0.99**	0.93	0.91	0.73	**0.95**	0.85	0.74	0.43
	SkGrid	0.80	0.85	0.61	**0.89**	0.60	**0.70**	0.37	0.67

- Learning action models from data: **Amir and Chang 2008[3]; Amado et al. 2019[4]; Asai and Muise 2020[5]; Juba, Le, and Stern 2021[6]**
- Inverse reinforcement learning (IRL): **Zeng et al 2018[7]**.
- Other metric-based GR: **Masters and Sardina 2017[8]; Mirsky et al. 2019[9]**

[3] Amir and Chang, "Learning partially observable deterministic action models".
[4] Amado et al., "Goal recognition in latent space".
[5] Asai and Muise, "Learning Neural-Symbolic Descriptive Planning Models via Cube-Space Priors: The Voyage Home (to STRIPS)".
[6] Juba, Le, and Stern, "Safe Learning of Lifted Action Models".
[7] Zeng et al., "Inverse Reinforcement Learning Based Human Behavior Modeling for Goal Recognition in Dynamic Local Network Interdiction."
[8] Masters and Sardina, "Cost-based goal recognition for path-planning".
[9] Mirsky et al., "New goal recognition algorithms using attack graphs".

- Our work paves the way for a new class of GR approaches based on model-free reinforcement learning.
- Future work: more robust distance measures; function approximation models e.g., neural networks).
  Note that all operations in the distance metrics apply to function approximation models
- While our work is theoretically compatible with non-tabular representations of the value functions, we chose to focus our experiments on domains that are translatable to PDDL.
- We plan to extend this work to image-based domains rather than PDDL-based ones.

Thank you!
Questions?