GOAL RECOGNITION IN LATENT SPACE

Leonardo Amado, Ramon Fraga Pereira, **João Paulo Aires**,
Mauricio Magnaguagno, Roger Granada and Felipe Meneguzzi

July 2018

PUCRS

INTRODUCTION

- Goal recognition is the task of inferring the intended goal of an agent by observing the actions of such agent.
- Current approaches of goal recognition assume that there is a domain expert capable of building complete and correct domain knowledge.

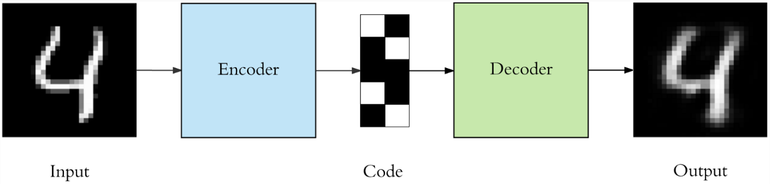
- This is too strong for most real-world applications.
- To overcome these limitations, we combine goal recognition techniques from automated planning and deep autoencoders to automatic generate PDDL domains and use them to perform goal recognition

BACKGROUND

A goal recognition problem is a tuple $\mathcal{P}_{GR} = \langle \mathcal{D}, \mathcal{F}, \mathcal{I}, \mathcal{G}, O \rangle$, where:

- $\mathcal{D}$ is a planning domain;
- $\mathcal{F}$ is the set of facts;
- $\mathcal{I} \subseteq \mathcal{F}$ is an initial state;
- $\mathcal{G}$ is the set of possible goals, which include a correct hidden goal G^* ($G^* \in \mathcal{G}$);
- and $O = \langle o_1, o_2, ..., o_n \rangle$ is an observation sequence of executed actions, with each observation $o_i \in \mathcal{A}$, and the corresponding action being part of a valid plan π that sequentially transforms $\mathcal{I}$ into G^*.

- Using autoencoders it is possible to encode an image to a binary representation (equiv. to logic fluents)
- To perform the encoding of complex images , a complex autoencoder can be used, using the **Gumbel Softmax**.
- The encoded representation is called *latent space*.



Source: https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

PLANNING IN LATENT SPACE

- Taking advantage of such autoencoders, LatPlan [Asai and Fukunaga, 2017] generates plans using only images of the initial and goal states.
- The initial state image and goal images are encoded in a binary representation.
- LatPlan uses traditional planning algorithms to plan using only the *latent-space*
    - LatPlan shows that many classical heuristics remain valid and effective even in latent space

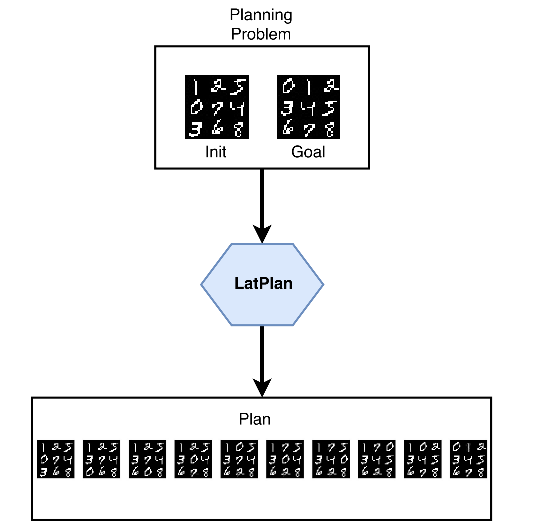


Figure: Latplanner.

GOAL RECOGNITION IN LATENT SPACE

- We propose an approach capable of recognizing goals in image based domains.
- We use the same tuple as planning goal recognition, but our states are now images.

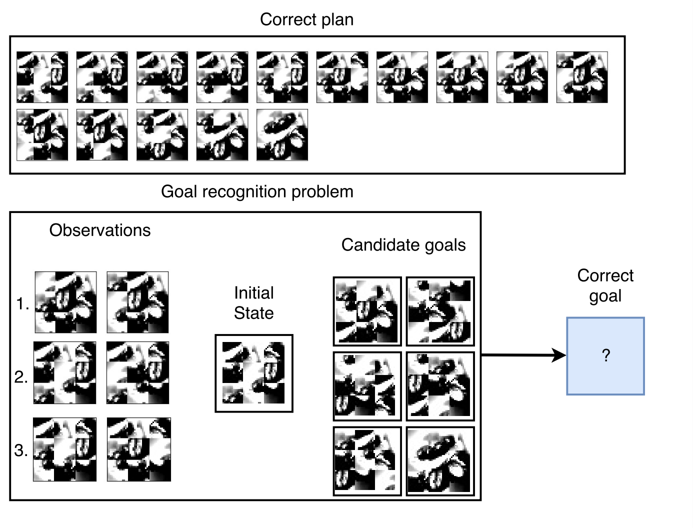
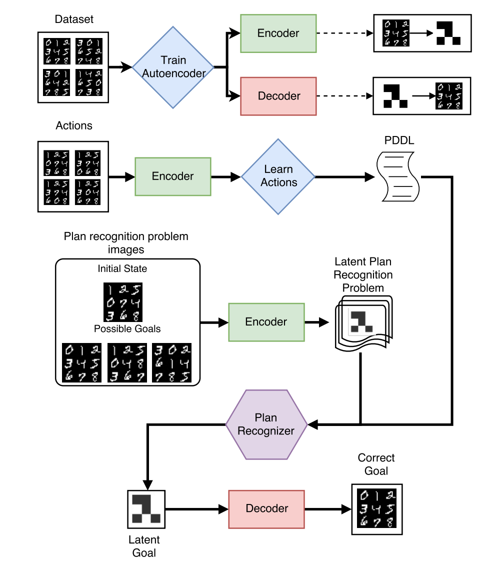


Figure: Goal Recognizer.

To recognize goals in image based domains, there are 4 milestones we must achieve.

1. First, we must train an autoencoder capable of creating a latent representation to a state of such image domain.
2. Second, we derive a PDDL domain, by extracting the transitions of such domain when encoded in latent space, obtaining a domain $\mathcal{D}$.
3. Third, we must convert to a latent representation a set of images representing, the initial state $\mathcal{I}$, the set of facts $\mathcal{F}$ and a set of possible goals $\mathcal{G}$, where the hidden goal G^* is included.
4. Finally, we can apply goal recognition techniques using the computed tuple $\langle \mathcal{D}, \mathcal{F}, \mathcal{I}, \mathcal{G}, O \rangle$

- Use a dataset with 20000 states to train the autoencoder.
- Use a dataset with all the state transitions to extract a PDDL.
- Convert the GR problem to latent space using the autoencoder.
- With the domain PDDL and the encoded PR problem, recognize a plan in latent space.



We use the autoencoder with the following structure, using 36 bits for the latent representation:

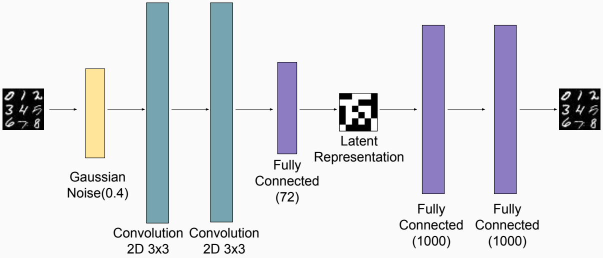


Figure: Autoencoder structure.

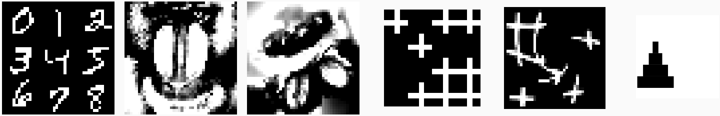
To derive a domain PDDL from raw data, we use the following method.

1. We encode every single transition using the autoencoder.
2. We then group up transitions that have the same effect.
3. We then derive a precondition by comparing which bits do not change between each transition of each group of effects.
4. Having both a precondition and an effect, we derive a PDDL action.

EXPERIMENTS

Domains

To test our approach, we use 6 domains from 3 distinct games.



(a) MNIST (b) Mandrill (c) Spider (d) LO Digital (e) LO Twist (f) Hanoi

Figure: Sample state for each domain.

First, we analyze the quality of the PDDL domain and the accuracy of the autoencoder.

Table: PDDL generation performance for each domain.

Domain	Total Transitions	Encoded Transitions	SAE Accuracy %	Computed Actions	Ground Actions	PDDL Redundancy
MNIST	967680	963795	99.6%	4946	192	25.76
Mandrill	967680	967680	100.0%	495	192	2.578
Spider	967680	967680	100.0%	763	192	3.974
LO Digital	1048576	1048576	100.0%	5940	1392	4.267
LO Twisted	1048576	1048576	100.0%	12669	1392	9.101
Hanoi	237	237	100.0%	211	38	5.552

Second, we show the results obtained by goal recognition techniques using hand-made PDDL domains.

- We consider different levels of observability: 10, 30, 50, 70, and 100%
- We evaluate Time, Accuracy, and Spread over the three games
- We use three different standard Goal Recognizers

Sample of the obtained results

				POM (h_{uniq})			RG		
Domain	$\|\mathcal{G}\|$	(%) Obs	$\|O\|$	Time (s) θ (0 / 10)	Accuracy % θ (0 / 10)	Spread in $\mathcal{G}$ θ (0 / 10)	Time (s)	Accuracy %	Spread in $\mathcal{G}$
8-Puzzle	6.0	10	1.0	0.074 / 0.080	33.3% / 33.3%	2.6 / 2.6	0.179	100.0%	4.8
		30	3.0	0.079 / 0.085	83.3% / 83.3%	1.0 / 2.5	0.188	100.0%	1.3
		50	4.0	0.088 / 0.091	100.0% / 100.0%	1.1 / 1.6	0.191	100.0%	1.3
		70	5.3	0.092 / 0.100	100.0% / 100.0%	1.0 / 1.0	0.210	100.0%	1.0
		100	7.3	0.108 / 0.110	100.0% / 100.0%	1.0 / 1.0	0.246	83.3%	1.1

Comparing hand-made and automatic generated PDDL domains.

				POM (h_{uniq})			RG		
Domain	$\lvert\mathcal{G}\rvert$	(%) Obs	$\lvert O\rvert$	Time (s) θ (0 / 10)	Accuracy % θ (0 / 10)	Spread in $\mathcal{G}$ θ (0 / 10)	Time (s)	Accuracy %	Spread in $\mathcal{G}$
8-Puzzle	6.0	10	1.0	**0.074** / **0.080**	33.3% / 33.3%	2.6 / **2.6**	**0.179**	**100.0%**	**4.8**
		30	3.0	**0.079** / **0.085**	**83.3%** / **83.3%**	**1.0** / **2.5**	**0.188**	**100.0%**	**1.3**
		50	4.0	**0.088** / **0.091**	**100.0%** / **100.0%**	**1.1** / **1.6**	**0.191**	**100.0%**	**1.3**
		70	5.3	**0.092** / **0.100**	**100.0%** / **100.0%**	**1.0** / **1.0**	**0.210**	**100.0%**	**1.0**
		100	7.3	**0.108** / **0.110**	**100.0%** / **100.0%**	**1.0** / **1.0**	**0.246**	83.3%	**1.1**
MNIST	6.0	10	1.2	0.555 / 0.562	**40.0%** / **60.0%**	**1.6** / 3.2	21.25	**100.0%**	6.0
		30	3.0	0.587 / 0.599	20.0% / 80.0%	1.4 / 3.0	22.26	**100.0%**	4.8
		50	4.0	0.609 / 0.628	60.0% / 80.0%	2.2 / 2.8	22.48	**100.0%**	4.8
		70	5.8	0.631 / 0.654	60.0% / **100.0%**	2.4 / 3.6	23.53	**100.0%**	3.2
		100	7.8	0.676 / 0.681	80.0% / **100.0%**	2.4 / 3.0	26.34	**100.0%**	3.4

CONCLUSION AND FUTURE WORK

Conclusion

- We developed an approach for goal recognition capable of obviating the need for human engineering to create a task for goal recognition.
- Empirical results shows that our approach comes close to standard goal recognition techniques.
- Regardless, our approach allows breakthroughs in goal recognition techniques.
- Our current approach has two main limitations:
  - we need all possible transitions of the domain;
  - we currently use relatively small images as input.

Future work

- For future work we aim to improve pruning of redundant actions in the domain inference process.
- Furthermore, we would like to develop plan recognition algorithms for incomplete domain models.
- Finally, we aim to develop an approach that applies goal recognition over video streams.

Thank you!

leonardo.amado@acad.pucrs.br

joao.aires.001@acad.pucrs.br