Goal Recognition with Real World Data

Felipe Meneguzzi†

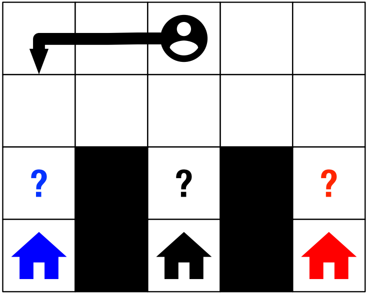
†Pontifical Catholic University of Rio Grande do Sul, Brazil
felipe.meneguzzi@pucrs.br

Porto Alegre, October, 2020

Table of Contents

What is it?

- **Goal Recognition** is the task of recognizing agents' goal that 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
- Roughly two types of approach:
  - Plan-library based (*classical* plan recognition)
  - Domain-theory based (plan recognition as planning, or PRAP)



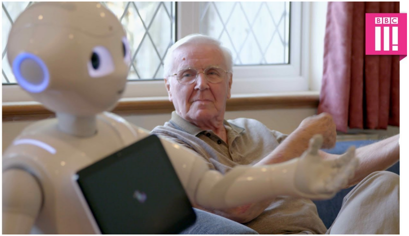
Why do we need goal recognition?

- Recognizing plans and goals of others is critical for meaningful interaction:
  - important for humans/agents working in the same environment
  - increasingly important as we build more intelligent systems

- Overall area of Plan, Activity and Intent Recognition
  - Activity recognition: recognizing meaningful activities from low-level sensor data
  - Plan/Intent/Goal recognition: recognizing intentional higher-level sequences of activities

Why do we need goal recognition?

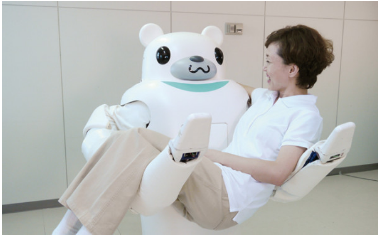
- Recognizing plans and goals of others is critical for meaningful interaction:
  - important for humans/agents working in the same environment
  - increasingly important as we build more intelligent systems



- Overall area of Plan, Activity and Intent Recognition
  - Activity recognition: recognizing meaningful activities from low-level sensor data
  - Plan/Intent/Goal recognition: recognizing intentional higher-level sequences of activities

Why do we need goal recognition?

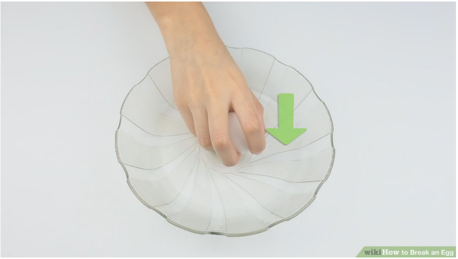
- Recognizing plans and goals of others is critical for meaningful interaction:
  - important for humans/agents working in the same environment
  - increasingly important as we build more intelligent systems



- Overall area of Plan, Activity and Intent Recognition
  - Activity recognition: recognizing meaningful activities from low-level sensor data
  - Plan/Intent/Goal recognition: recognizing intentional higher-level sequences of activities

An example of Activity Recognition

An example of Activity Recognition



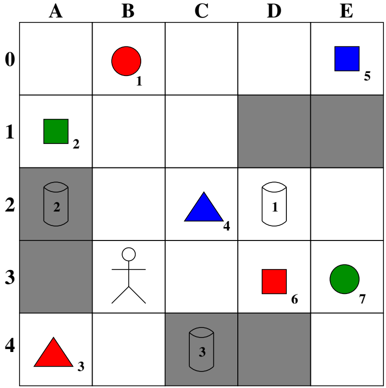
An example of Activity Recognition

An example of Activity Recognition

breaking egg

An Example of Goal/Plan Recognition

from Miquel Ramirez's thesis



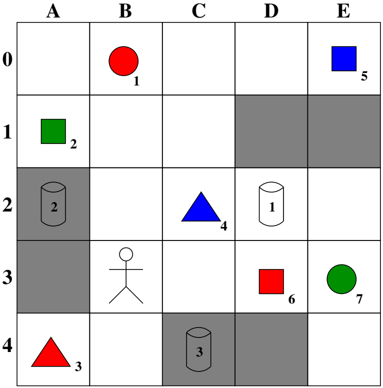
Wooden pieces $p_1, p_2, \ldots p_n$
Pieces have shapes and colors
Bins $b_1, b_2, \ldots, b_n$

The possible **goals** the trainer expected to pursue:

1. Store all triangles in b_1
2. Store all spheres in b_2
3. Store all cubes in b_3
4. Store red objects in b_2
5. Store green objects in b_3
6. Store blue objects in b_1

An Example of Goal/Plan Recognition

from Miquel Ramirez's thesis



Wooden pieces $p_1, p_2, \ldots p_n$

Pieces have shapes and colors

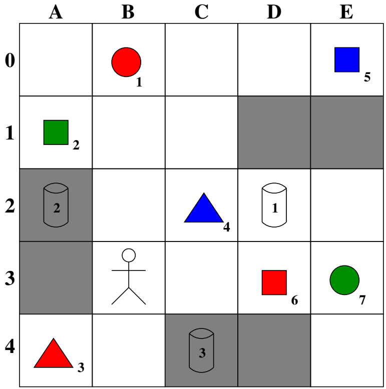
Bins $b_1, b_2, \ldots, b_n$

One possible *plan* for the trainer to achieve goal #1 (store all triangles in b_1):

1. Walk from B3 into A4
2. Pick p_3 up
3. Walk from A4 into B3
4. Walk from B3 into C2
5. Pick p_4 up
6. Throw p_3 into b_1
7. Throw p_4 into b_1

An Example of Goal/Plan Recognition

from Miquel Ramirez's thesis



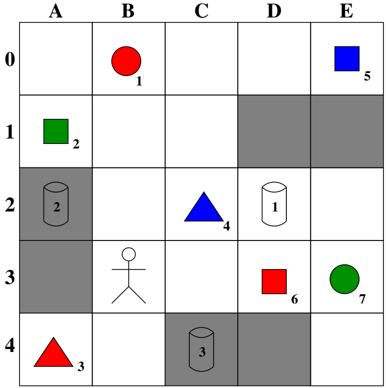
Wooden pieces $p_1, p_2, \ldots p_n$
Pieces have shapes and colors
Bins $b_1, b_2, \ldots, b_n$

If sensors miss 70% of *walk* actions and half *pick* and *drop* actions, we may only see:

1. Pick p_3 up
2. Walk from A4 into B3

An Example of Goal/Plan Recognition

from Miquel Ramirez's thesis



Wooden pieces $p_1, p_2, \ldots p_n$
Pieces have shapes and colors
Bins $b_1, b_2, \ldots, b_n$

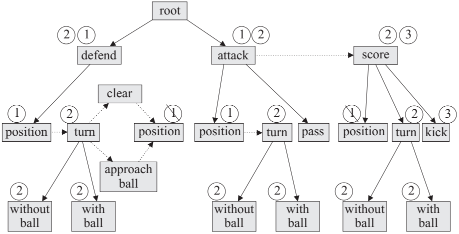
If sensors miss 70% of *walk* actions and half *pick* and *drop* actions, we may only see:

1. Pick p_3 up
2. Walk from A4 into B3

Here, we could deduce either goal #1 or #4 (store all red objects in b_2), as other tasks are less *likely*.

Flavors of Recognition Formalism

Plan Library



Domain Theory (PRAP)

```
(define (domain grid)
(:requirements :strips :typing)
(:types place shape key)
(:predicates (conn ?x ?y — place)
             (key—shape ?k — key ?s — shape)
             (lock—shape ?x — place ?s — shape)
             (at ?r — key ?x — place )
             (at—robot ?x — place)
             (locked ?x — place)
             (carrying ?k — key)
             (open ?x — place)
)


(:action unlock
:parameters (?curpos ?lockpos — place ?key — key ?shape — shape)
:precondition (and (conn ?curpos ?lockpos) (key—shape ?key ?shape)
                   (lock—shape ?lockpos ?shape) (at—robot ?curpos)
                   (locked ?lockpos) (carrying ?key))
:effect (and  (open ?lockpos) (not (locked ?lockpos)))
)

(:action move
:parameters (?curpos ?nextpos — place)
:precondition (and (at—robot ?curpos) (conn ?curpos ?nextpos) (open ?n
:effect (and (at—robot ?nextpos) (not (at—robot ?curpos)))
)

(:action pickup
:parameters (?curpos — place ?key — key)
:precondition (and (at—robot ?curpos) (at ?key ?curpos))
:effect (and (carrying ?key)
   (not (at ?key ?curpos)))
)

)
```

Table of Contents

Automated Planning

Definition (Planning)

A planning instance is represented by a triple $\Pi = \langle \Xi, \mathcal{I}, G \rangle$, in which:

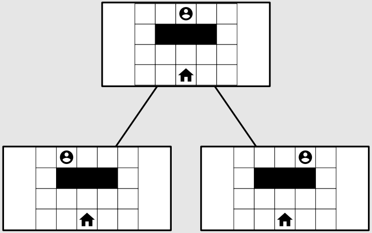
- $\Xi = \langle \Sigma, \mathcal{A} \rangle$ is the **domain definition**, and consists of a finite set of **facts** Σ and a finite set of **actions** $\mathcal{A}$ (action costs typically 1);
- $\mathcal{I} \subseteq \Sigma$ and $G \subseteq \Sigma$ represent the **planning problem**, in which $\mathcal{I} \subseteq \Sigma$ is the **initial state**, and $G \subseteq \Sigma$ is the **goal state**.

- Actions $a \in \mathcal{A}$ are tuples $a = \langle pre(a), eff(a), cost(a) \rangle$
- Facts Σ can be modeled in a variety of ways:
  - As a logic language (restricted FOL): states are truth assignments
  - As a set of variables $\mathcal{V}$ with finite domains: states are variable assignments

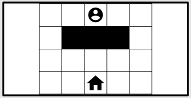
Automated Planning - Less boring

Planning problems have three key ingredients

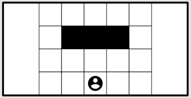
Domain Description



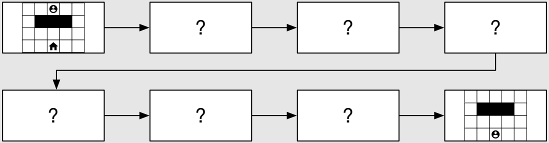
Initial State



Goal State



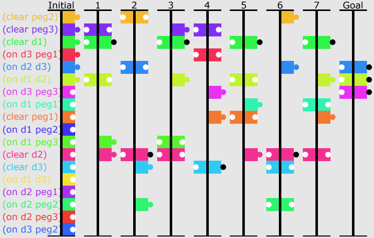
Solution



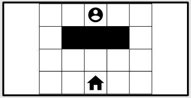
Automated Planning - Less boring

Planning problems have three key ingredients

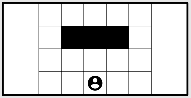
Domain Description



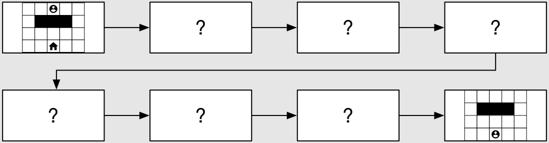
Initial State



Goal State



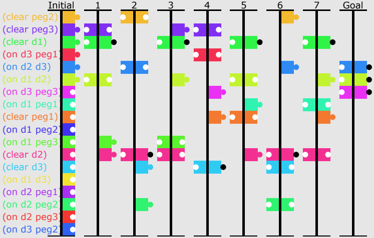
Solution



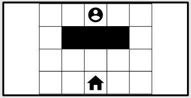
Automated Planning - Less boring

Planning problems have three key ingredients

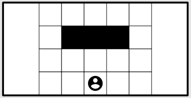
Domain Description



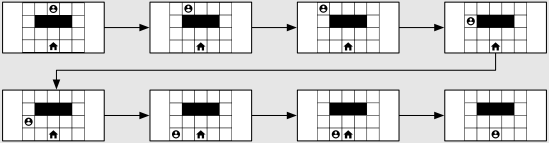
Initial State



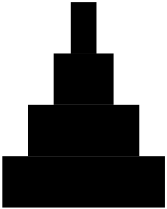
Goal State



Solution



Planning Heuristics

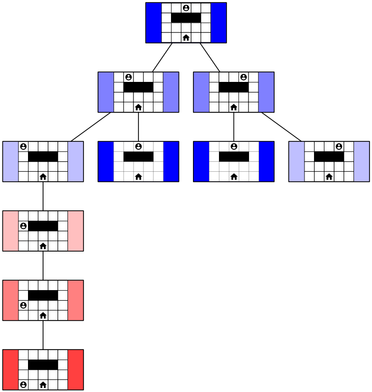


Most modern planners rely on heuristics to efficiently search the state-space.

Two key challenges in research on novel heuristics

- Informativeness of the heuristic
- Computational efficiency

Planning Heuristics



Most modern planners rely on heuristics to efficiently search the state-space.

Two key challenges in research on novel heuristics

- Informativeness of the heuristic
- Computational efficiency

Goal Recognition Problem

Definition (Goal Recognition Problem)

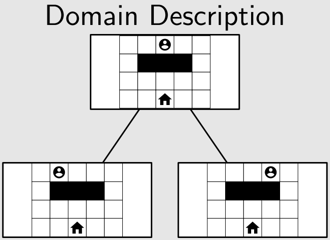
A goal recognition problem is a tuple $P = \langle \Xi, \mathcal{I}, \mathcal{G}, \mathrm{O} \rangle$, where:

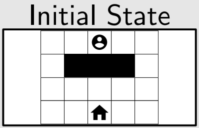
- $\Xi = \langle \Sigma, \mathcal{A} \rangle$ is the domain definition (facts and actions) ;
- $\mathcal{I} \subseteq \Sigma$ is the initial state;
- $\mathcal{G}$ s.t. $\forall G \in \mathcal{G}, G \subseteq \Sigma$ is a set of candidate goals (with an assumed hidden goal G); and
- O is a sequence $\langle o_1, \ldots o_n \rangle$ of observations, where $o_i \in \mathcal{A}$

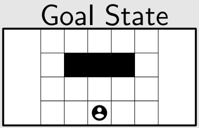
- The solution for a goal recognition problem is the hidden goal $G \in \mathcal{G}$ that is most consistent with observation sequence O.
- Caveat: we may have other representations for the observations
- This is what I will refer to as PRAP

Goal Recognition Problem - Less boring

Goal/Plan Recognition problems have three key ingredients

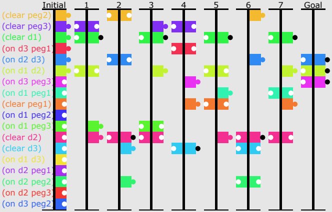




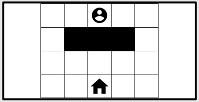


Goal/Plan Recognition problems have **four** key ingredients

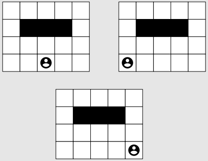
Domain Description



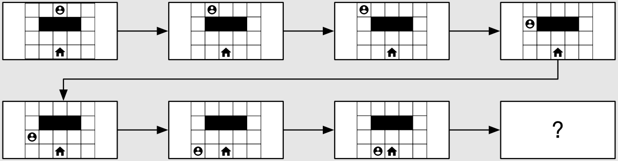
Initial State



Goal **Hypotheses**



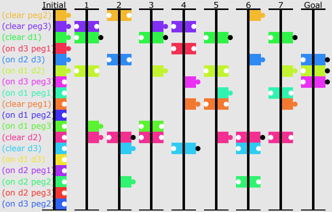
Observations



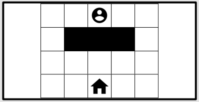
Goal Recognition Problem - Less boring

Goal/Plan Recognition problems have **four** key ingredients

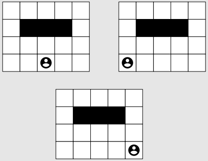
Domain Description



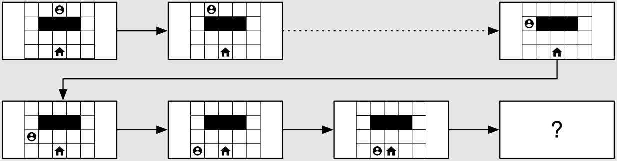
Initial State



Goal **Hypotheses**



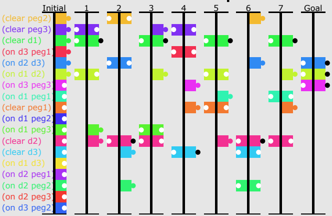
Observations



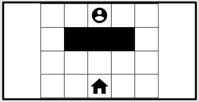
Goal Recognition Problem - Less boring

Goal/Plan Recognition problems have **four** key ingredients

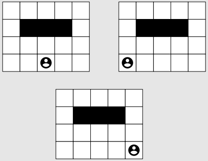
Domain Description



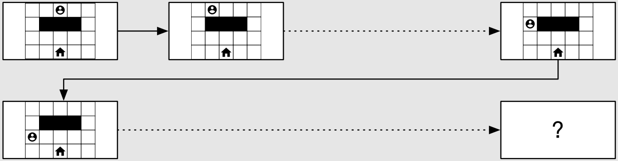
Initial State



Goal **Hypotheses**



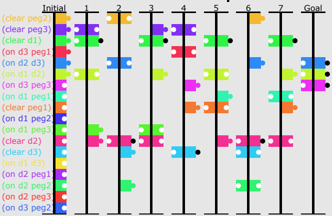
Observations



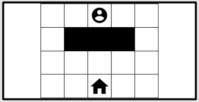
Goal Recognition Problem - Less boring

Goal/Plan Recognition problems have **four** key ingredients

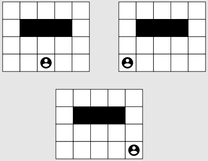
Domain Description



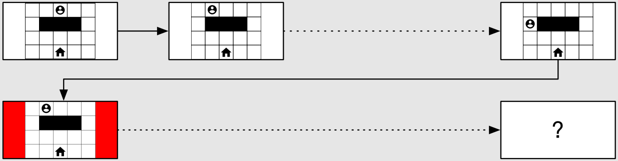
Initial State



Goal **Hypotheses**

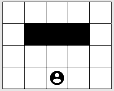


Observations



Solution

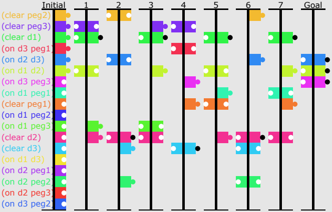
Correct Goal



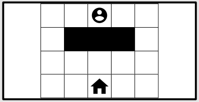
Goal Recognition Problem - Less boring

Goal/Plan Recognition problems have **four** key ingredients

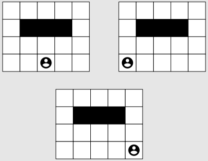
Domain Description



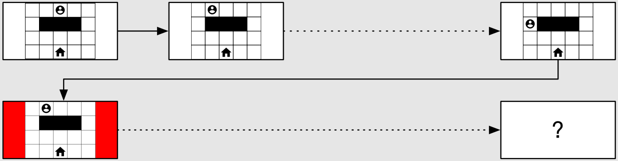
Initial State



Goal **Hypotheses**



Observations



Solution

Probability Distribution

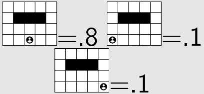


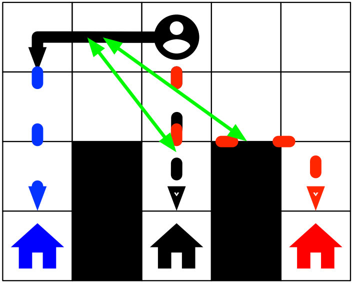
Table of Contents

Goal Recognition using Planning Domains I

Ramirez and Geffner (2009 and 2010)

- First approaches to goal recognition: Plan Recognition as Planning (PRAP)
- Probabilistic model aims to compute $P(G \mid O)$
- Following Bayes Rule $P(G \mid O) = \alpha P(O \mid G) P(G)$
- Given $P(G)$ as a prior, key bottleneck is computing $P(O \mid G)$

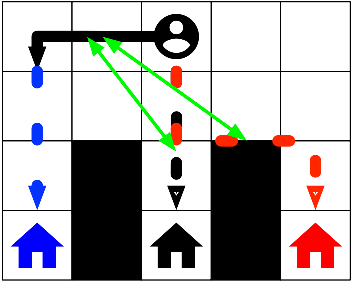
- Compute $P(O \mid G)$ in terms of a cost difference $c(G, O) - c(G, \bar{O})$
- Costs **two planner calls per goal hypothesis**



Sohrabi et al. (2016)

- Conceptually similar to Ramirez and Geffner: aims to compute $P(G \mid O)$ via $\alpha P(O \mid G)P(G)$
- Compilation of plan recognition problem into **multiple planning** problems (one for each G)

- Compute Top-k or diverse plans π to approximate $P(O \mid G) = \sum_{\pi} P(O \mid \pi) \cdot P(\pi \mid G)$
- Compensate noisy observations by imposing a cost on dropped Observations



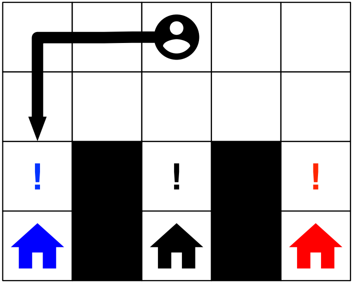
Goal Recognition using Planning Heuristics

Pereira, Oren and Meneguzzi (2017):

- **Obviate the need to execute a planner multiple times** for recognizing goals; and
- Novel goal recognition heuristics that use **planning landmarks**.
- **More accurate** and **orders of magnitude faster** than all previous approaches.

Planning Landmarks:

- Are **necessary conditions** for any valid plan
- Theoretical cost of computation is the same as planning



Goal Recognition using Operator-Counting Constraints

Meneguzzi, Pereira and Pereira (2020):

- Use **operator counting** heuristic information for recognizing goals; and
- Operator counts and LP constraints cope explicitly with noisy observations.

Key advantages:

- **More accurate** than all previous approaches; and
- Provides an **extensible framework** for further goal recognition work.

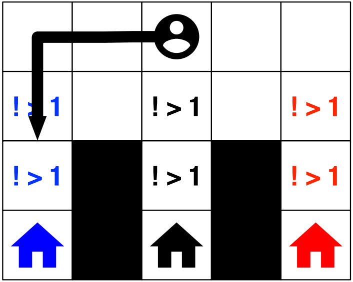
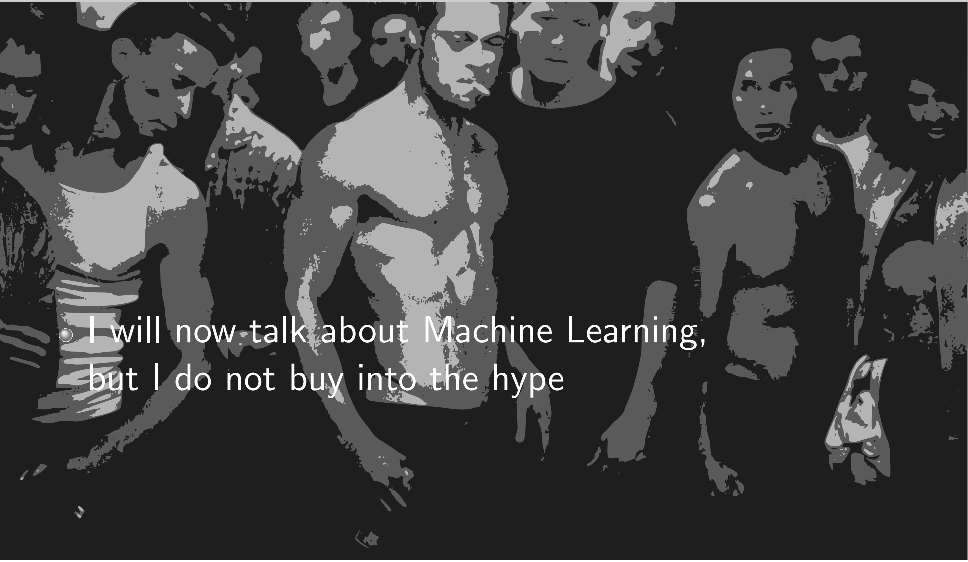


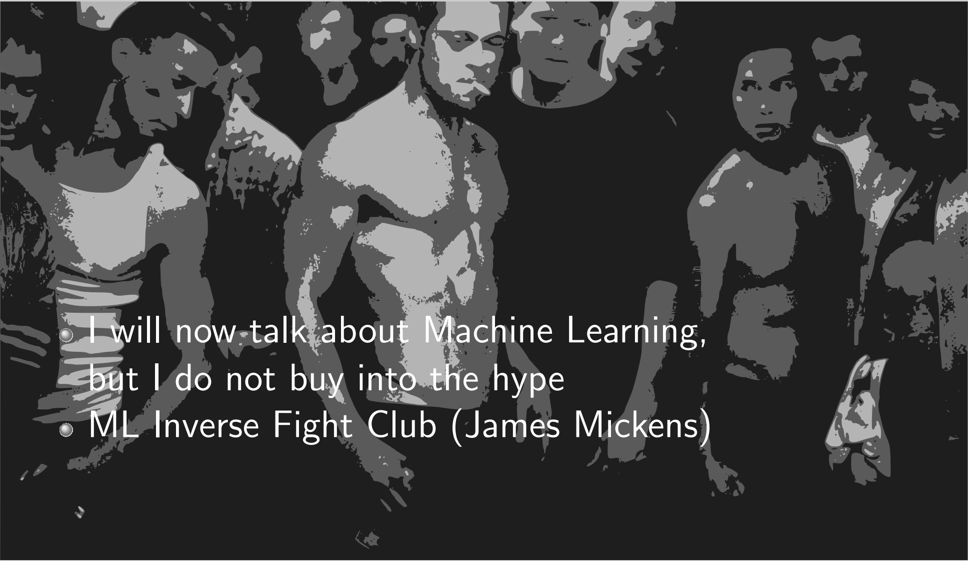
Table of Contents

Warning



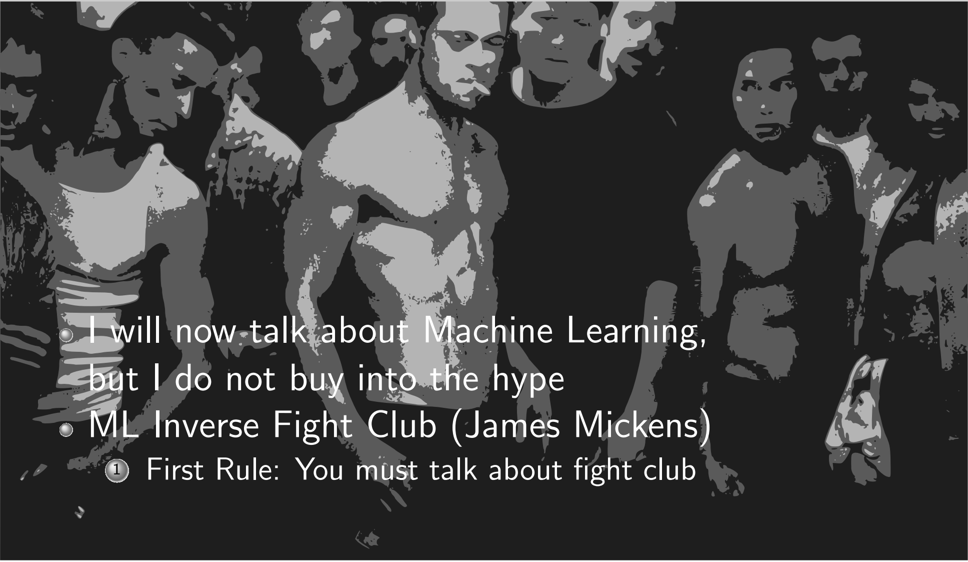
- I will now talk about Machine Learning, but I do not buy into the hype

Warning



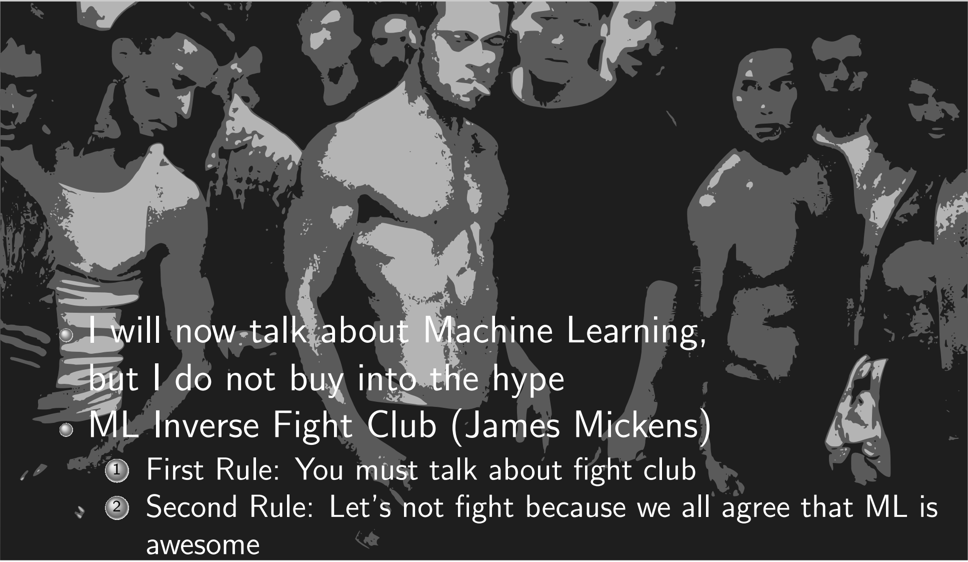
- I will now talk about Machine Learning, but I do not buy into the hype
- ML Inverse Fight Club (James Mickens)

Warning



- I will now talk about Machine Learning, but I do not buy into the hype
- ML Inverse Fight Club (James Mickens)
  1. First Rule: You must talk about fight club

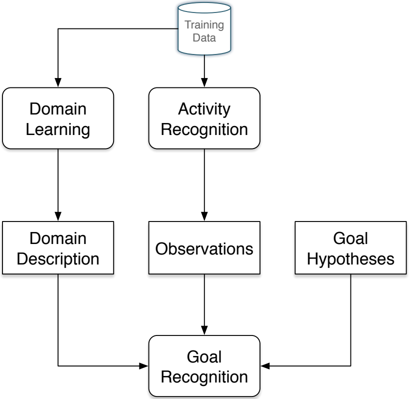
Warning



- I will now talk about Machine Learning, but I do not buy into the hype
- ML Inverse Fight Club (James Mickens)
  1. First Rule: You must talk about fight club
  2. Second Rule: Let's not fight because we all agree that ML is awesome

Where can we use real-world data?

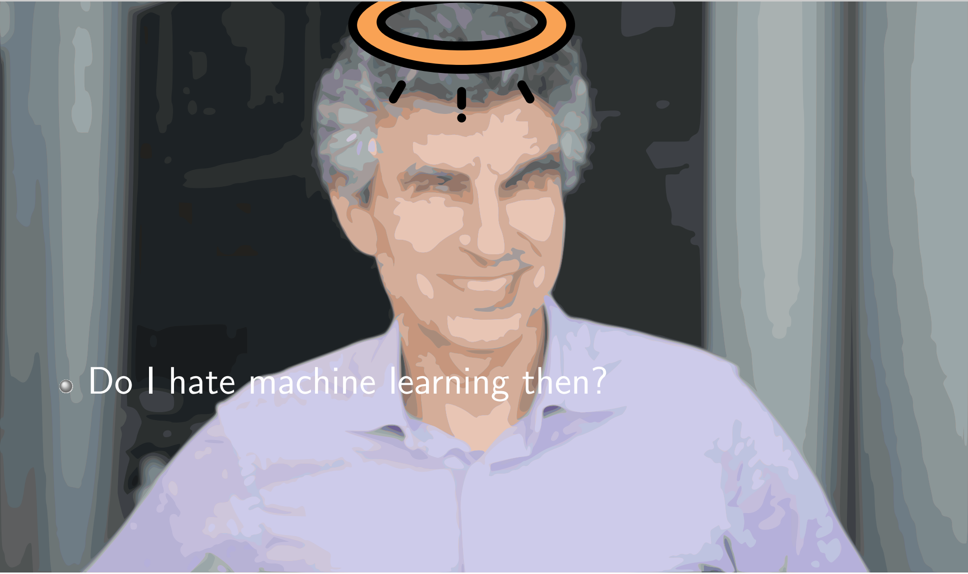
- Domain description:
  What we want to recognize?
  - Environment domain
  - Subject preferences
- Goal Recognition:
  How do we deal with the observations?
  - Generate observations from raw data
  - Cope with noise from observations



Limitations of previous approaches

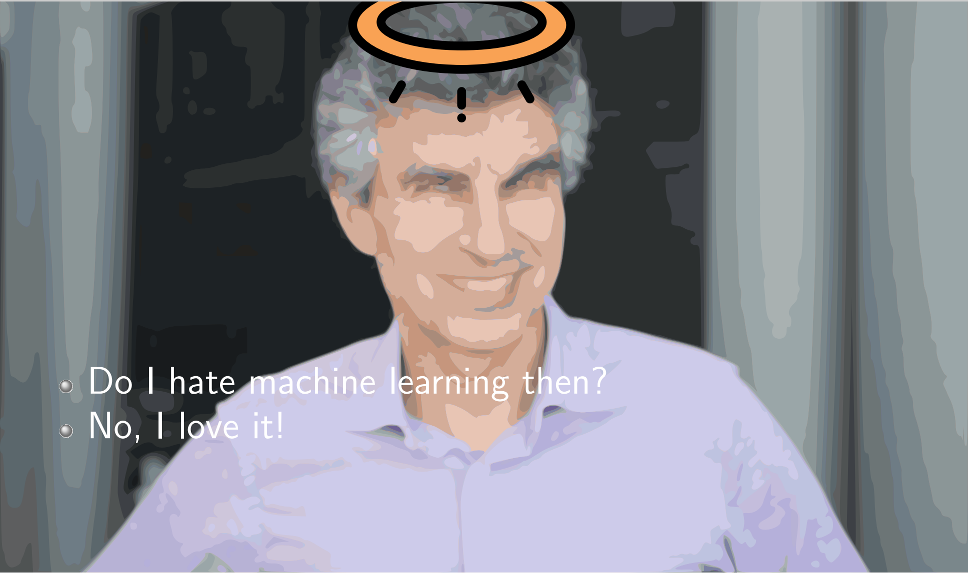
- Domain Knowledge:
  - Must be engineered by humans
  - Must be **perfect**
- Observations:
  - Must be “well-behaved” in some sense
  - Do not use raw, real-world data

Machine Learning Again



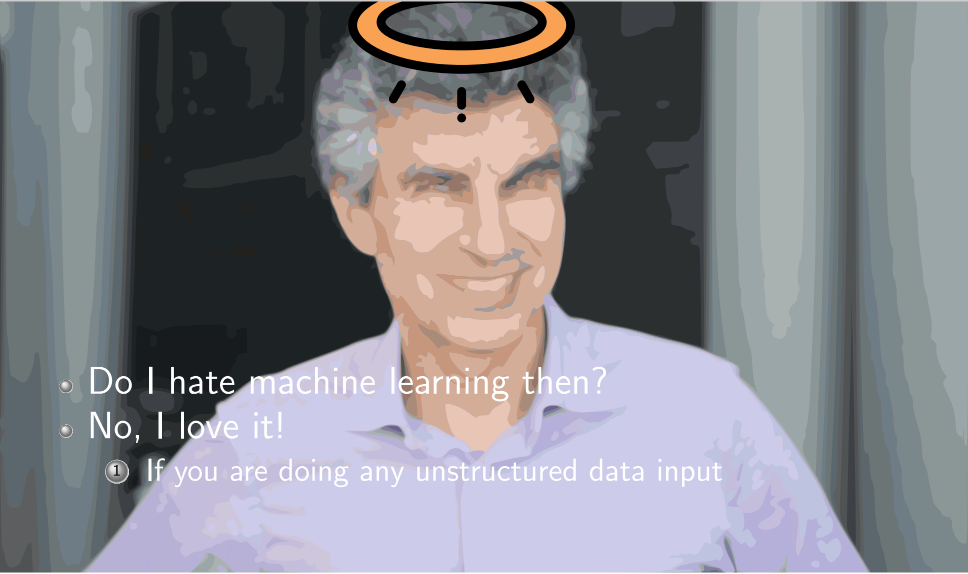
- Do I hate machine learning then?

Machine Learning Again



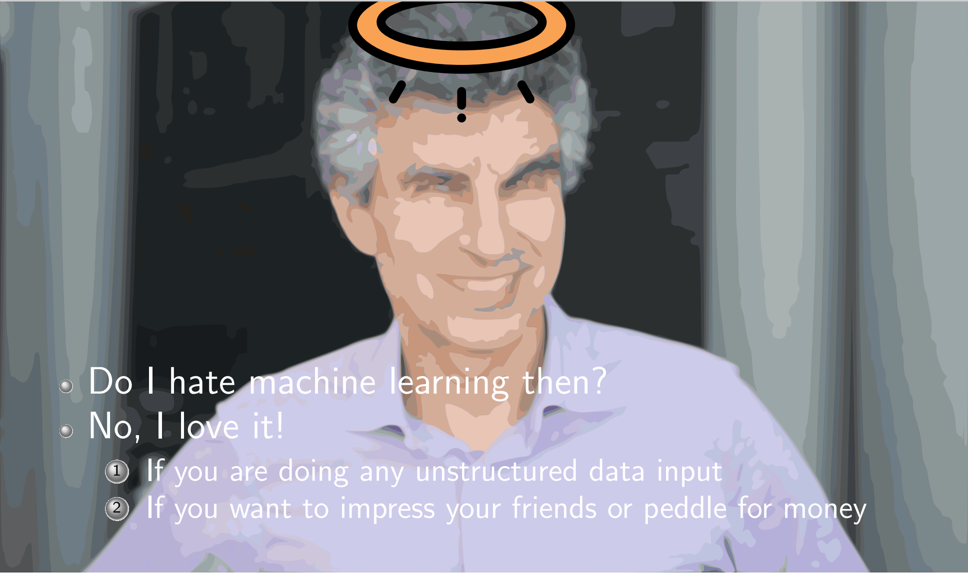
- Do I hate machine learning then?
- No, I love it!

Machine Learning Again



- Do I hate machine learning then?
- No, I love it!
  1. If you are doing any unstructured data input

Machine Learning Again



- Do I hate machine learning then?
- No, I love it!
  1. If you are doing any unstructured data input
  2. If you want to impress your friends or peddle for money

How do we try to solve this?

- To Generate Symbolic Observations:
  - ML to map raw data into recognition algorithm
  - ML algorithms to generate **symbolic observations**
- Obtain Domain Knowledge:
  - Cope with expected noisy observations relaxing the domain model
  - Learn PDDL representations from image data
  - Learn **Nominal Models** from raw data
- To work on both problems simultaneously
  - Hybrid engineering/learning of PDDL representations

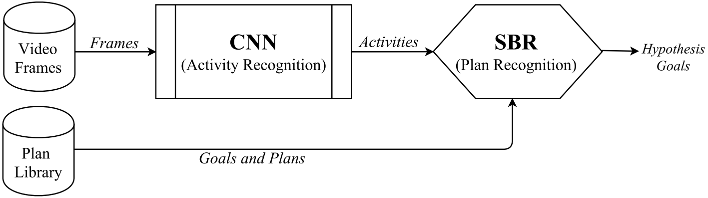
Plan Recognition using Video Data

Plan Recognition using Video Data

- Most research focuses on activity and plan recognition separately;
- We develop a hybrid approach that comprises both;
- The approach infers, from a set of candidate plans, which plan a human subject is pursuing based exclusively on fixed-camera video.

A Hybrid Architecture for Activity and Plan Recognition

- **Conceptually divided in two main parts**
  - CNN-based activity recognition (CNN)[1]
  - CNN-backed symbolic plan recognition (SBR)[2]



[1] That's us!

[2] Not our work: Avrahami-Zilberbrand and Kaminka. Fast and Complete Symbolic Plan Recognition. IJCAI 2005

How are we doing so far?

- To Generate Symbolic Observations:
  - ML to map raw data into a recognition algorithm ✓
  - ML algorithms to generate **symbolic observations**
- Obtain Domain Knowledge:
  - Cope with expected noisy observations relaxing the domain model
  - Learn PDDL representations from image data
  - Learn **Nominal Models** from raw data
- To work on both problem simultaneously
  - Hybrid engineering/learning of PDDL representations

Goal Recognition in Incomplete Domains

What is an Incomplete Domain?

In a nutshell:
It is a STRIPS/PDDL domain that allows me to state that some preconditions/effects **may or may not** be there!

Why use Incomplete Domains?

- A **step forward** to **more realistic settings**; and
- The **lack of domain knowledge**, human-error, and etc.

Definition (**Incomplete STRIPS Domain Model**[a])

[a] *Weber and Bryce, Planning and Acting in Incomplete Domain Models. ICAPS, 2011.*

An incomplete STRIPS domain model is a tuple $\widetilde{\mathcal{D}} = \langle \mathcal{R}, \widetilde{\mathcal{O}} \rangle$, where:

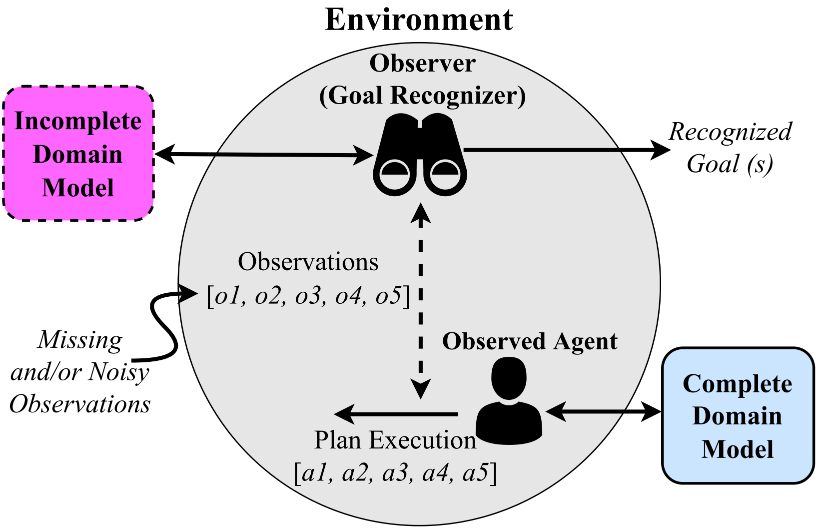
- $\mathcal{R}$ is a set of predicates with typed variables;
- $\widetilde{\mathcal{O}}$ is a set of incomplete operators. An operator $\widetilde{op} \in \widetilde{\mathcal{O}}$ defines:
  - $pre(\widetilde{op}) \subseteq \mathcal{R}$ as a set of known preconditions;
  - $add(\widetilde{op}) \subseteq \mathcal{R}$ as a set of known add effects;
  - $del(\widetilde{op}) \subseteq \mathcal{R}$ as a set of known delete effects;

Definition (**Incomplete STRIPS Domain Model**[a])

[a] *Weber and Bryce, Planning and Acting in Incomplete Domain Models. ICAPS, 2011.*

An incomplete STRIPS domain model is a tuple $\widetilde{\mathcal{D}} = \langle \mathcal{R}, \widetilde{\mathcal{O}} \rangle$, where:

- $\mathcal{R}$ is a set of predicates with typed variables;
- $\widetilde{\mathcal{O}}$ is a set of incomplete operators. An operator $\widetilde{op} \in \widetilde{\mathcal{O}}$ defines:
  - $pre(\widetilde{op}) \subseteq \mathcal{R}$ as a set of known preconditions;
  - $\widetilde{pre}(\widetilde{op}) \subseteq \mathcal{R}$ as a set of **possible preconditions**;
  - $add(\widetilde{op}) \subseteq \mathcal{R}$ as a set of known add effects;
  - $\widetilde{add}(\widetilde{op}) \subseteq \mathcal{R}$ as a set of **possible add effects**;
  - $del(\widetilde{op}) \subseteq \mathcal{R}$ as a set of known delete effects;
  - $\widetilde{del}(\widetilde{op}) \subseteq \mathcal{R}$ as a set of **possible delete effects**;


Environment
Observer
(Goal Recognizer)
Incomplete
Domain
Model
Recognized
Goal (s)
Observations
[o1, o2, o3, o4, o5]
Missing
and/or Noisy
Observations
Observed Agent
Complete
Domain
Model
Plan Execution
[a1, a2, a3, a4, a5]

How are we doing so far?

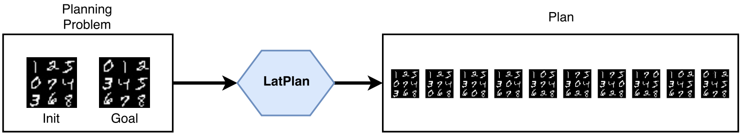
- To Generate Symbolic Observations:
  - ML to map raw data into a recognition algorithm ✓
  - ML algorithms to generate **symbolic observations**
- Obtain Domain Knowledge:
  - Cope with expected noisy observations relaxing the domain model ✓
  - Learn PDDL representations from image data
  - Learn **Nominal Models** from raw data
- To work on both problem simultaneously
  - Hybrid engineering/learning of PDDL representations

Plan Recognition in Latent Space

Motivation

- Goal and Plan Recognition depend on high-quality domain engineering
  - PDDL domain theory for PRAP
  - Plan Library for traditional PR
- We would like to obviate the need for most domain engineering
  - Learn domain models directly from raw data
  - Recognize goals using raw data as observations

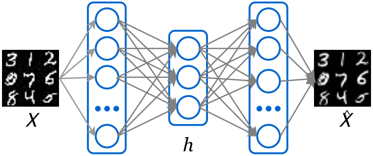
Inspiration: LatPlanner[3]



[3] Not our Work: Asai and Fukunaga. Classical Planning in Deep Latent Space: Bridging the Subsymbolic-Symbolic Boundary, AAAI, 2018

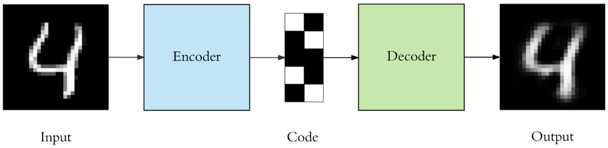
Variational Autoencoders

- Autoencoders are a specific type of Neural Network intended to learn representations
- Consists of three key parts
  - Encoder network
  - Latent layer (the middle layer)
  - Decoder network
- Trained in an unsupervised manner
- Variational Autoencoders force the the encoder network to generate latent vectors following some distribution
  (typically a unit-gaussian one)



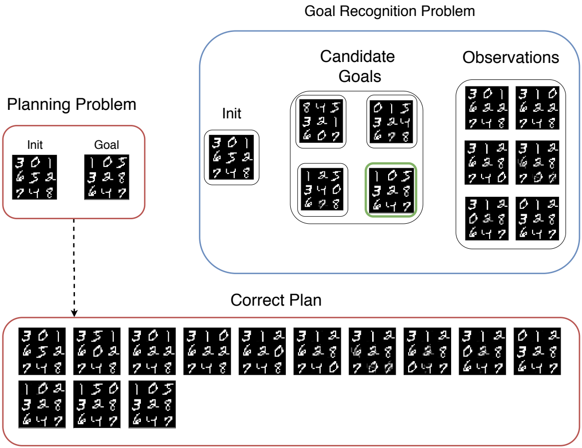
Gumbel-softmax autoencoders and planning

- Planners view states as sets of logical propositions (i.e. binary vectors)
- We would like to be able to learn state representations from raw data
- We force an autoencoder to have a categorial distribution in the latent layer:
  - Gumbel-softmax activation can be annealed into a categorical distribution
  - Latent layer now correspond to **logic bits**
  - Can learn a PDDL transition function from pairs of states



Goal Recognition using Raw Data

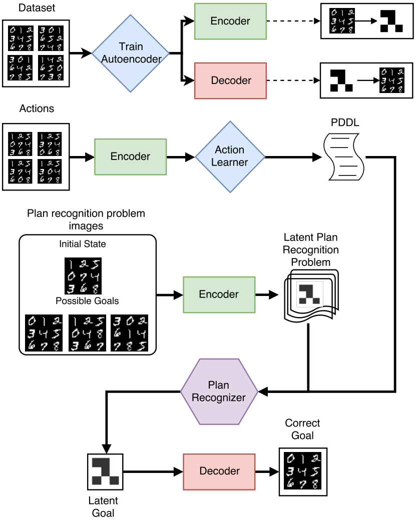
- Once we learn the internal representation, we can recognize plans as sequences of images, but using symbolic goal recognition algorithms



Goal Recognition in Latent Space

Key steps in the approach

- Train autoencoder using an image dataset (binary representation)
- Infer transition function from pairs of encoded images (states)
- Convert Goal Recognition problem to latent space using autoencoder
- Recognize goals using PRAP algorithms (ours)



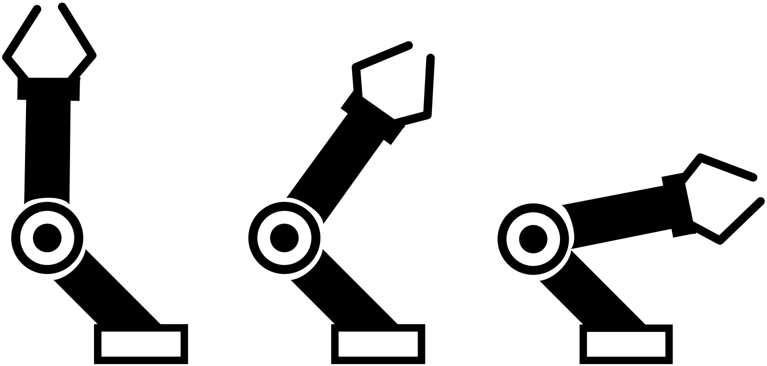
How are we doing so far?

- To Generate Symbolic Observations:
  - ML to map raw data into a recognition algorithm ✓
  - ML algorithms to generate **symbolic observations**
- Obtain Domain Knowledge:
  - Cope with expected noisy observations relaxing the domain model ✓
  - Learn PDDL representations from image data ✓
  - Learn **Nominal Models** from raw data
- To work on both problem simultaneously
  - Hybrid engineering/learning of PDDL representations

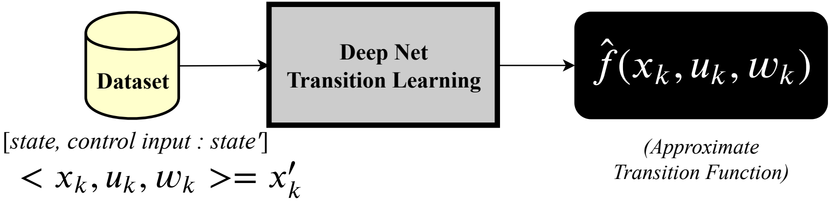
Goal Recognition Using Nominal Models

Motivation

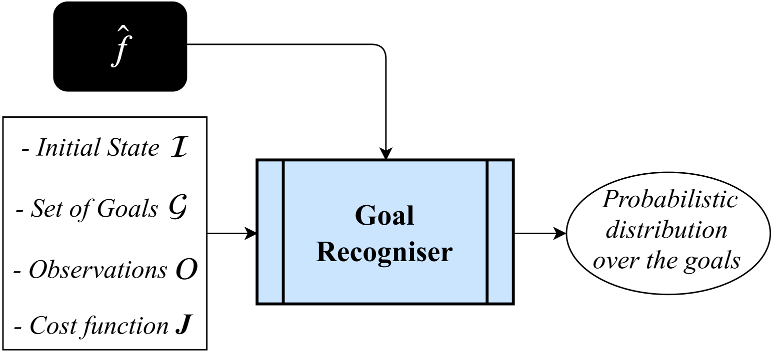
- Existing goal recognition approaches **rely on complete models** with **known system dynamics**;
- We **drop the assumption** that the transition function is given and well defined, using **Nominal (approximate) models**



Deep Neural Networks as Nominal Models



- We acquire **nominal models** by training a DNN
- Trained DNN becomes the **transition function**
- *Nominal models* support **continuous** action and state spaces;



- We define the observations O as **trajectory of states** induced by a policy π that minimises J, and **achieve a hidden goal** $G^* \in \mathcal{G}$.

Probabilistic Goal Recognition over Nominal Models

We adopt the probabilistic interpretation of Ramírez and Geffner (2010)[4]:

- $P(G|O) = \alpha P(O|G)P(G)$
  - $P(G)$ is a *prior* probability to a goal G;
  - $P(O|G)$ is the probability of observing O when the goal is G;
  - α is a normalisation factor.

Here, since $P(G)$ is equal for every candidate goal, the question is:

- **How do we compute** $P(O|G)$?

[4] *Ramírez and Geffner, Probabilistic Plan Recognition using off-the-shelf Classical Planners. AAAI, 2011.*

Goal Recognition as Nominal Mirroring: ηMIRRORING

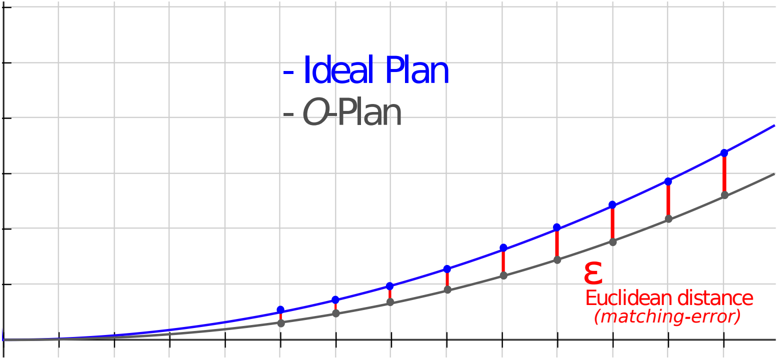
We develop our first approach using the concept of *Mirroring*[5] to **compare two plans for each of the candidate goals in** $\mathcal{G}$:

- **Ideal-plan** (π_G): a plan computed from $\mathcal{I}$ to every goal G in $\mathcal{G}$;
- O-**plan** ($\pi_{O,G}$): a plan computed for every pair $\mathcal{I}$, G, which must visit every state in O.

[5] *Vered et al., Online Goal Recognition through Mirroring: Humans and Agents. ACS, 2016.*

ηMirroring: *matching-error* ϵ

We compare the **Ideal-plan** and the *O*-**plan** using the *matching-error*[6] ϵ, i.e., the **Euclidean distance** between the trajectories.



[6] *Kaminka et al., Plan Recognition in Continuous Domains. AAAI, 2018.*

How are we doing so far?

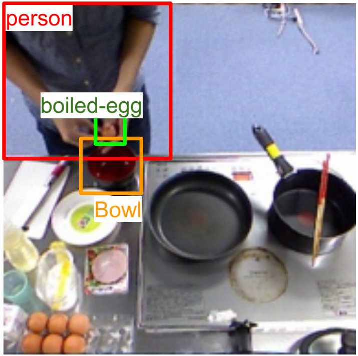
- To Generate Symbolic Observations:
  - ML to map raw data into recognition algorithm ✓
  - ML algorithms to generate **symbolic observations**
- Obtain Domain Knowledge:
  - Cope with expected noisy observations relaxing the domain model ✓
  - Learn PDDL representations from image data ✓
  - Learn **Nominal Models** from raw data ✓
- To work on both problem simultaneously
  - Hybrid engineering/learning of PDDL representations

Engineering GR Domains using ML

- Machine Learning models are the unchallenged state of the art for computer vision:

Machine Learning and Computer Vision

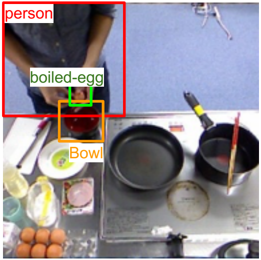
- Machine Learning models are the unchallenged state of the art for computer vision:
- Most computer vision datasets already contain **annotated semantic information** (and algorithms assume their existence):
  - Labels for **objects** and **relations**
- Why not use this semantic information to co-design GR domains around them?



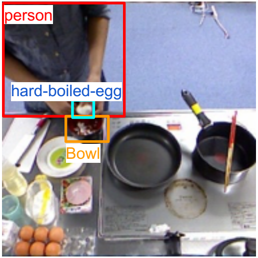
Relations:

`<person,holding,boiled-egg>`
`<boiled-egg,holding,bowl>`

Deriving PDDL from ML Algorithms



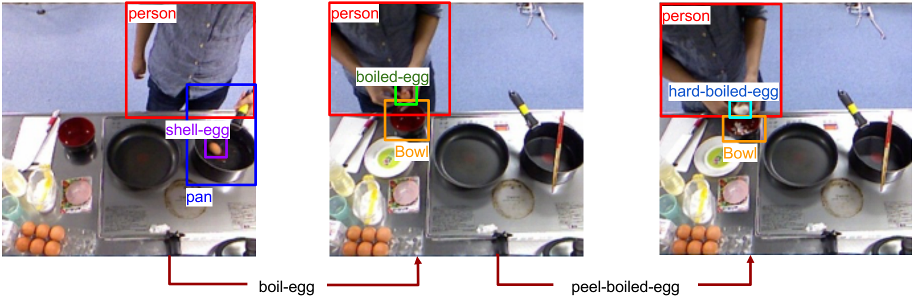
peel-boiled-egg →



Relations:
<person,holding,boiled-egg>
<boiled-egg,holding,bowl>

```
(:action peel−boiled−egg
    :parameters (?e − egg ?d − cooking−dish)
    :precondition (and (boiled−egg ?e)
                       (holding ?e) (on ?e ?d))
    :effect (and (hard−boiled−egg ?e)
                 (not (shell−egg ?e)))
)
```

Generating Semantically-meaningful Observations with ML



<person,holding,shell-egg>
<shell-egg,in,pan>
<person,holding,hashi>

<person,holding,boiled-egg>
<boiled-egg,on,bowl>

<person,holding,hard-boiled-egg>
<hard-boiled-egg,on,bowl>

How are we doing so far?

- To Generate Symbolic Observations:
  - ML to map raw data into recognition algorithm ✓
  - ML algorithms to generate **symbolic observations** ✓
- Obtain Domain Knowledge:
  - Cope with expected noisy observations relaxing the domain model ✓
  - Learn PDDL representations from image data ✓
  - Learn **Nominal Models** from raw data ✓
- To work on both problem simultaneously
  - Hybrid engineering/learning of PDDL representations ✓

Table of Contents

Summary

- We progressively drop assumptions used by goal recognition about:
  - Precision of domain knowledge

- Along the way, we showed how to perform goal recognition:
  - Using incomplete domain knowledge

Summary

- We progressively drop assumptions used by goal recognition about:
  - Precision of domain knowledge
  - Quality of observations

- Along the way, we showed how to perform goal recognition:
  - Using incomplete domain knowledge
  - Using real world video-data

Summary

- We progressively drop assumptions used by goal recognition about:
  - Precision of domain knowledge
  - Quality of observations
  - Exclusively discrete domains

- Along the way, we showed how to perform goal recognition:
  - Using incomplete domain knowledge
  - Using real world video-data
  - Using learned (nominal) models

Summary

- We progressively drop assumptions used by goal recognition about:
  - Precision of domain knowledge
  - Quality of observations
  - Exclusively discrete domains
  - Existence of domain knowledge
- Along the way, we showed how to perform goal recognition:
  - Using incomplete domain knowledge
  - Using real world video-data
  - Using learned (nominal) models
  - In Latent Space

Summary

- We progressively drop assumptions used by goal recognition about:
  - Precision of domain knowledge
  - Quality of observations
  - Exclusively discrete domains
  - Existence of domain knowledge
- Along the way, we showed how to perform goal recognition:
  - Using incomplete domain knowledge
  - Using real world video-data
  - Using learned (nominal) models
  - In Latent Space
  - Achieving lasting world peace

Summary

- We progressively drop assumptions used by goal recognition about:
  - Precision of domain knowledge
  - Quality of observations
  - Exclusively discrete domains
  - Existence of domain knowledge
- Along the way, we showed how to perform goal recognition:
  - Using incomplete domain knowledge
  - Using real world video-data
  - Using learned (nominal) models
  - In Latent Space
  - Achieving lasting world peace (Ok, maybe not)

Future Directions

- Plan Recognition with Domain Theories
  - Extend heuristics to temporal and non-uniform-cost; domains
  - Experiment with more advanced notions of numeric landmarks (e.g. Scala et al.); and
  - Automatically infer first-order logic literals.
- More effective GR techniques combining learning and symbolic reasoning.

A last bit of wisdom

Effective AI combines search (symbolic reasoning) and machine learning (sensing the noisy world)

A last bit of wisdom

Effective AI combines search (symbolic reasoning) and machine learning (sensing the noisy world)

Do not be naughty

Thanks and Acknowledgement

People involved in this research

- Ramon F. Pereira (ex-PhD, La Sapienza University of Rome, Italy)
- Maurício Magnaguagno (ex-PhD)
- Leonardo Amado (PhD Student)
- Juarez Monteiro (PhD Student)
- Roger Granada (Postdoc)
- Mor Vered (Monash University, Australia)
- Gal Kaminka (Bar Ilan University, Israel)
- Miquel Ramirez (University of Melbourne, Australia)
- Nir Oren (University of Aberdeen, Scotland)
- André Grahl Pereira (UFRGS)
- Rodrigo Barros (PUCRS colleague)
- Duncan Ruiz (PUCRS colleague)

The money guys

Institutions

- The Scottish Informatics and Computer Science Alliance (SICSA) Distinguished Visiting Fellowship (DVE)
- Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – PQ Fellowship

Meneguzzi Goal Recognition with Real World Data Porto Alegre, October, 2020

Papers reporting these results I

PEREIRA, Ramon. F.; PEREIRA, André G.; MENEGUZZI, Felipe. **Landmark-Enhanced Heuristics for Goal Recognition in Incomplete Domain Models.** ICAPS, 2019.

PEREIRA, Ramon. F.; VERED, Mor; MENEGUZZI, Felipe; RAMIREZ, Miquel. **Online Probabilistic Goal Recognition over Nominal Models.** IJCAI, 2019.

AMADO, Leonardo R.; AIRES, João Paulo; PEREIRA, Ramon F.; MAGNAGUAGNO, Maurício C.; GRANADA, Roger L.; MENEGUZZI, Felipe. **An LSTM-Based Approach for Goal Recognition in Latent Space.** PAIR@AAAI, 2019.

MENEGUZZI, Felipe; PEREIRA, André G.; PEREIRA, Ramon. F.. **Robust Goal Recognition with Operator-Counting Heuristics.** XAIP@ICAPS, 2019.

Papers reporting these results II

AMADO, Leonardo R.; PEREIRA, Ramon F.; AIRES, João Paulo; MAGNAGUAGNO, Maurício C.; GRANADA, Roger Leitzke; and MENEGUZZI, Felipe. **Goal Recognition in Latent Space.** IJCNN, 2018.

AMADO, Leonardo R.; PEREIRA, Ramon F.; AIRES, João Paulo; MAGNAGUAGNO, Maurício C.; GRANADA, Roger Leitzke; and MENEGUZZI, Felipe. **Goal Recognition in Latent Space.** IJCNN, 2018.

VERED, Mor; PEREIRA, Ramon F.; KAMINKA, Gal; and MENEGUZZI, Felipe. **Online Goal Recognition as Reasoning over Landmarks.** PAIR@AAAI, 2018.

VERED, Mor; PEREIRA, Ramon F.; KAMINKA, Gal; and MENEGUZZI, Felipe. **Towards Online Goal Recognition Combining Goal Mirroring and Landmarks.** AAMAS, 2018.

Papers reporting these results III

FRAGA PEREIRA, Ramon; MENEGUZZI, Felipe. **Landmark-based Plan Recognition.** ECAI, 2016.

PEREIRA, Ramon F.; OREN, Nir; and MENEGUZZI, Felipe. **Landmark-Based Heuristics for Goal Recognition**. AAAI, 2017.

PEREIRA, Ramon F.; OREN, Nir; and MENEGUZZI, Felipe. **Monitoring Plan Optimality using Landmarks and Domain-Independent Heuristics**. PAIR Workshop@AAAI, 2017.

GRANADA, Roger L.; PEREIRA, Ramon F.; MONTEIRO, Juarez; BARROS, Rodrigo; RUIZ, Duncan; and MENEGUZZI, Felipe. **Hybrid Activity and Plan Recognition for Video Streams**. PAIR Workshop@AAAI, 2017.

PEREIRA, Ramon F.; OREN, Nir; and MENEGUZZI, Felipe. **Detecting Commitment Abandonment by Monitoring Plan Execution**. AAMAS, 2017.

MONTEIRO, Juarez; GRANADA, Roger; BARROS, Rodrigo and MENEGUZZI, Felipe. **Deep Neural Networks for Kitchen Activity Recognition**. IJCNN, 2017.

VERED, Mor; PEREIRA, Ramon F.; MAGNAGUAGNO, Maurício C.; KAMINKA, Gal; and MENEGUZZI, Felipe. **Online Goal Recognition Combining Landmarks and Planning**. GRW@IJCAI, 2017.

Student Recruitment Plug

If this talk was interesting and you want to know more, talk to me:

MSc and PhD admissions

22nd November 2019

Areas of work and advantages:

- Automated Planning and Goal Recognition
- Machine Learning (within reason)
- Excellent Food

Thank you!
Questions?

PUCRS | ESCOLA POLITÉCNICA