Observation Decoding with Sensor Models: Recognition Tasks via Classical Planning

Diego Aineto, Sergio Jimenez, Eva Onaindia

October 16, 2020

Universitat Politècnica de València

What is decoding?

What is decoding?

Decoding: finding the most likely explanation to some evidence.

What is decoding?

Decoding: finding the most likely explanation to some evidence.

Basic reasoning tool in:

- "Plan recognition as Planning" (Ramirez and Geffner, 2009).
- "Diagnosis as Planning Revisited" (Sohrabi et al., 2010).
- "Counterplanning using Goal Recognition and Landmarks" (Pozanco et al. 2018)
- "Learning action models with minimal observability" (Aineto et al., 2019)

What is decoding?

Decoding: finding the most likely explanation to some evidence.

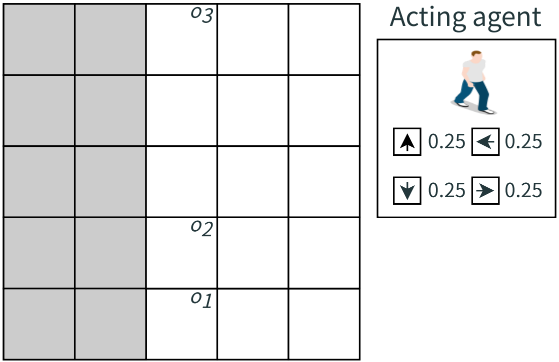
Basic reasoning tool in:

- "Plan recognition as Planning" (Ramirez and Geffner, 2009).
- "Diagnosis as Planning Revisited" (Sohrabi et al., 2010).
- "Counterplanning using Goal Recognition and Landmarks" (Pozanco et al. 2018)
- "Learning action models with minimal observability" (Aineto et al., 2019)

Contributions:

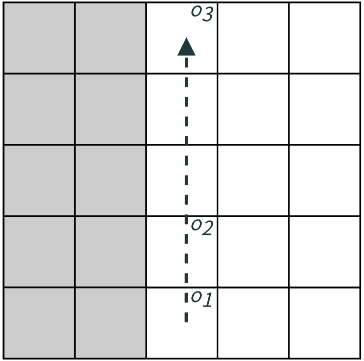
- Formalization of the decoding problem within a probabilistic framework.
- Extension of decoding to support sensor models.

Motivating example

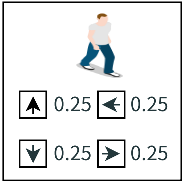


$$O = (\langle loc = (0,2) \rangle, \langle loc = (1,2) \rangle, \langle loc = (4,2) \rangle$$

Motivating example



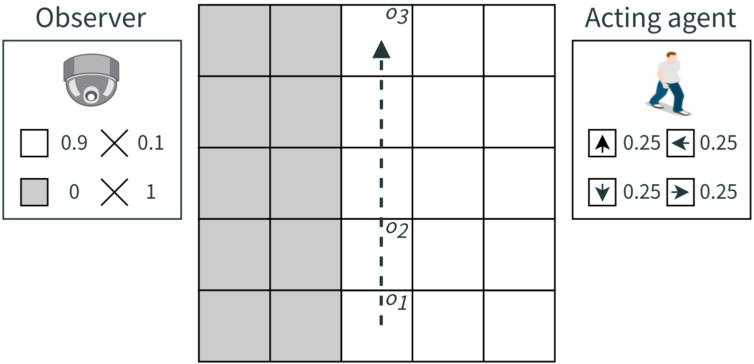
Acting agent



$$O = (\langle loc = (0,2)\rangle, \langle loc = (1,2)\rangle, \langle loc = (4,2)\rangle$$

$$\tau_1 = (\langle (at\ x0\ y2)\rangle, \langle (at\ x1\ y2)\rangle, \langle (at\ x2\ y2)\rangle, \langle (at\ x3\ y2)\rangle, \langle (at\ x4\ y2)\rangle)$$

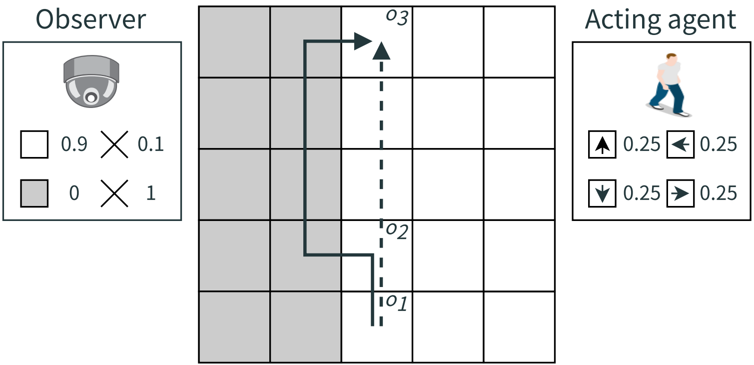
Motivating example



$$O = (\langle loc = (0,2)\rangle, \langle loc = (1,2)\rangle, \langle loc = (4,2)\rangle$$

$$\tau_1 = (\langle (at\, x0\, y2)\rangle, \langle (at\, x1\, y2)\rangle, \langle (at\, x2\, y2)\rangle, \langle (at\, x3\, y2)\rangle, \langle (at\, x4\, y2)\rangle)$$

Motivating example

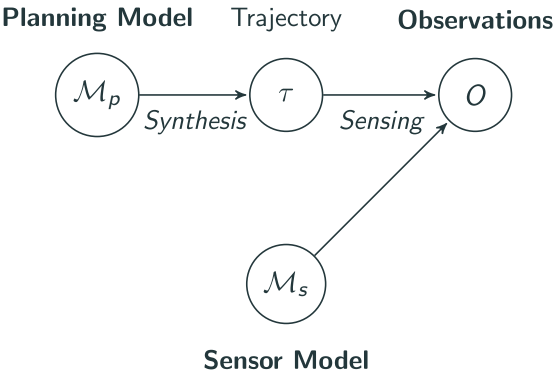


$$O = (\langle loc = (0,2)\rangle, \langle loc = (1,2)\rangle, \langle loc = (4,2)\rangle$$

$$\tau_2 = (\langle (at\, x0\, y2)\rangle, \langle (at\, x1\, y2)\rangle, \langle (at\, x1\, y1)\rangle, \langle (at\, x2\, y1)\rangle, \langle (at\, x3\, y1)\rangle, \langle (at\, x4\, y1)\rangle, \langle (at\, x4\, y2)\rangle)$$

Problem Definition

Probabilistic Framework



Sensor Model and Observations

A **sensor model** $\mathcal{M}_s = \langle X, Y, \Phi \rangle$

- X are the state variables.
- Y are the observable variables.
- Φ is the set of sensing functions $f_i : C_i \times Y_i \rightarrow [0, 1]$
  - *exhaustive* ($\bigcup_{c \in C_i} S_c = S$), and
  - *exclusive* ($S_c \cap S_{c'} = \emptyset, \forall c, c' \in C_i$)

Sensor Model and Observations

A **sensor model** $\mathcal{M}_s = \langle X, Y, \Phi \rangle$

- X are the state variables.
- Y are the observable variables.
- Φ is the set of sensing functions $f_i : C_i \times Y_i \to [0, 1]$
  - *exhaustive* ($\bigcup_{c \in C_i} S_c = S$), and
  - *exclusive* ($S_c \cap S_{c'} = \emptyset, \forall c, c' \in C_i$)

Blindspots example

Clear tile ($x \geq 2$): $f_{loc}(at_{x,y}, loc = (x, y)) = 0.9$, $f_{loc}(at_{x,y}, loc = \epsilon) = 0.1$

Blindspot tile ($x \leq 1$): $f_{loc}(at_{x,y}, loc = \epsilon) = 1$

Sensor Model and Observations

A **sensor model** $\mathcal{M}_s = \langle X, Y, \Phi \rangle$

- X are the state variables.
- Y are the observable variables.
- Φ is the set of sensing functions $f_i : C_i \times Y_i \to [0, 1]$
  - *exhaustive* ($\bigcup_{c \in C_i} S_c = S$), and
  - *exclusive* ($S_c \cap S_{c'} = \emptyset, \forall c, c' \in C_i$)

An **observation** $o = \langle Y_1 = w_1, \ldots, Y_{|Y|} = w_{|Y|} \rangle$ is a full assignment of Y.

The Observation Decoding Problem

An observation decoding problem is a triplet $D = \langle \mathcal{M}_p, \mathcal{M}_s, O \rangle$ where:

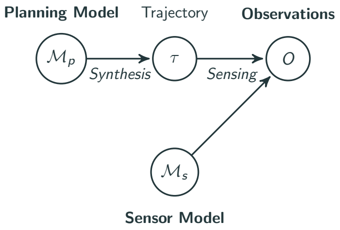
- $\mathcal{M}_p = \langle X, A \rangle$ is a planning model,
- $\mathcal{M}_s = \langle X, Y, \Phi \rangle$ is a sensor model, and
- $O = \langle o_0, o_1, \ldots, o_m \rangle$ is an input observation sequence.

The solution to $D = \langle \mathcal{M}_p, \mathcal{M}_s, O \rangle$ is the **most likely trajectory** τ^* defined as

$$\tau^* = \arg\max_{\tau \in \mathcal{T}} P(O, \tau | \mathcal{M}_p, \mathcal{M}_s),$$

Synthesis and Sensing Probabilities

$$\tau^* = \arg\max_{\tau \in \mathcal{T}} P(O, \tau | \mathcal{M}_p, \mathcal{M}_s) = \arg\max_{\tau \in \mathcal{T}} P(\tau | \mathcal{M}_p) P(O | \tau, \mathcal{M}_s)$$



Synthesis probability

The probability of generating τ with $\mathcal{M}_p$:

$$P(\tau | \mathcal{M}_p) = P(s_0) \prod_{i=1}^{|\tau|} P(s_i | s_{i-1}, \mathcal{M}_p), \qquad (1)$$

Sensing probability

The probability of perceiving O from τ:

$$P(O | \tau, \mathcal{M}_s) = \prod_{i=1}^{|\tau|} P(o_i | s_i, \mathcal{M}_s), \qquad (2)$$

Observation decoding via Classical Planning

Compilation

From probability maximization to cost minimization:

$$\tau^* = \arg\max_{\tau \in \mathcal{T}} P(O, \tau | \mathcal{M}_p, \mathcal{M}_s) \rightarrow \tau^* = \arg\min_{\tau \in \mathcal{T}} - \log P(O, \tau | \mathcal{M}_p, \mathcal{M}_s)$$

Compile $D = \langle \mathcal{M}_p, \mathcal{M}_s, O \rangle$ to a planning problem $P' = \langle F', A', I', G' \rangle$ such that $A' = A_t \cup A_e$ where:

- transition actions A_t are the cost-normalized versions of A
- sensing actions A_e to process an observation

Compilation

From probability maximization to cost minimization:

$$\tau^* = \arg\max_{\tau \in \mathcal{T}} P(O, \tau | \mathcal{M}_p, \mathcal{M}_s) \rightarrow \tau^* = \arg\min_{\tau \in \mathcal{T}} - \log P(O, \tau | \mathcal{M}_p, \mathcal{M}_s)$$

Compile $D = \langle \mathcal{M}_p, \mathcal{M}_s, O \rangle$ to a planning problem $P' = \langle F', A', I', G' \rangle$

If π is a solution plan for P' then:

- $cost(\pi_t) = -\log P(\tau^\pi | \mathcal{M}_p)$.
- $cost(\pi_e) = -\log P(O | \tau^\pi, \mathcal{M}_p)$.
- $cost(\pi) = -\log P(O, \tau^\pi | \mathcal{M}_p, \mathcal{M}_s)$.

Sensing Actions

A_e contains a sense_k action for each observation $o_k \in O$

- Implement an acceptor automaton for trajectories that satisfy the observation.
- Accumulate $-\log P(O|\tau, \mathcal{M}_s)$

$$guard(\text{sense}_\text{k}) := P(o_k|s_i, M_s) > 0$$

$$reset(\text{sense}_\text{k}) := x^+ = x - \log P(o_k|s_i, M_s)$$

Sensing Actions

$$O = (\langle loc = (0,2)\rangle, \langle \boldsymbol{loc} = (1,2)\rangle, \langle loc = (4,2)\rangle)$$

$$\begin{array}{ll}
\text{pre}(\text{sense}_2) & \mathit{sensed}_1 \\
\text{eff}(\text{sense}_2) & \mathit{sensed}_2 \wedge \\
& \text{when } (\mathit{at}\ x1\ y2) \\
& \quad \text{increase } \mathit{total_cost}\ - \mathit{log}\ (0.9) \\
& \text{when } (\mathit{not}\ (\mathit{at}\ x1\ y2)) \\
& \quad (\mathit{deadend})
\end{array}$$

Experimental Evaluation

Setup

Evaluate the effectiveness of using a sensor model for decoding.

- OD_N: optimal plan that satisfies the observation.
- OD_S: the approach presented here.

Metric: plan diversity[1]

$$\delta_\alpha(\pi_i, \pi_j) = \frac{|S_i - S_j|}{|S_i| + |S_j|} + \frac{|S_j - S_i|}{|S_i| + |S_j|}$$

[1] "Domain independent approaches for finding diverse plans" (Srivastava et al., 2007).

Results

Domain	H	L	OD_S	OD_N
Blindspots	100	0	0.03	0.18
	80	20	0.08	0.20
	60	40	0.11	0.17
Intrusion	100	0	0	0.58
	80	20	0.07	0.18
	60	40	0.13	0.14
Blocks 2h	100	0	0	0.34
	80	20	0.05	0.27
	60	40	0.07	0.26
Office	100	0	0	0.58
	80	20	0.23	0.38
	60	40	0.16	0.23

H: Observability of the high observability region
L: Observability of the low observability region
OD_S : $\delta_\alpha(\pi, \pi_S)$
OD_N : $\delta_\alpha(\pi, \pi_N)$

Conclusions

Conclusions

- Formalization of the decoding problem within a probabilistic framework.
- Extension of decoding to support sensor models.
- Unifying probabilistic framework (future work).