Lazy is Efficient (in plan recognition)

Gal A. Kaminka

The MAVERICK Group
Computer Science Department
(Gonda Brain Research Center, Nanotech Center)

Bar Ilan University, Israel
Abstract Mind Architecture (in artificial intelligence)

Two Processes

**Perception/World Modeling:**
- Vision, sensor processing, sensor fusion, ...

**Action Selection/Decision Making:**
- Planning, plan execution, goal deliberation, reacting...
My Scientific Problem:
The Nature of the Social Mind

The computational mechanisms that allow the mind to reason about, and interact with, others?

(differently than *about* or *with* inanimate objects)
Abstract Mind Architecture (in artificial intelligence)

Two Processes

**Perception/World Modeling:**

**Theory of Mind:**
Intents, plans of other minds
(in AI: Plan Recognition)

**Action Selection/Decision Making:**

**Coordination:**
Manipulating, acting w.r.t other minds

This talk
Since 1995....

- Tracking teammates (RESL, w/ Tambe)
- Overhearing teams (YOYO, YOYO*, w/ Tambe, Pynadath)
- Overhearing using Colored Petri-nets (w/ Gutnik)
- **Symbolic Recognition** (SBR, w/ Avrahami-Zilberbrandt)
- Decision-theoretic rec. (UPR, w/ Avrahami-Zilberbrandt)
- Intent detection (w/ Bochek-Dokow)
- **Mirroring** (w/ Vered)

See: www.cs.biu.ac.il/~galk/publications/class_rescat.html
Dynamic, Continuous, Multi-Agent
Recognition Challenges

• **Online:** Observations incrementally received
• **Complex state observations**
  – Observe state, not agent actions
  – State is factored (composed of multiple features)
• **Situated (reactive)**
  – Observed agents react, deviate from own plans
  – What goal/plan is true now
• **Observations stem from continuous world**
  – Sensitivity to discretization (loss of information)
SBR: Symbolic Behavior Recognition
[IJCAI 2005, MOO 2004-2005]

• Online, keyhole recognition, discrete observations
• Applied to vision tracker data (e.g., airport security)
SBR: Symbolic Behavior Recognition
[IJCAI 2005, MOO 2004-2005]

• Online, keyhole recognition, discrete observations
• Applied to vision tracker data (e.g., airport security)

What is agent doing?
Is its behavior anomalous?
SBR Plan Library: Layered, Durative Actions

- Directed acyclic connected graph
- Vertices denote plan steps (actions)
- Edges
  - Vertical (decomposition) edges
  - Horizontal (sequential) edges
- Every vertex may generate obs, have duration (cycles)
- “Or”-graph (edges denote ordered choices)
Example: Plans and Hypotheses

- entrance
- root
- security
- look
- X-ray
- board
- coffee
- gate
- without bag
- with bag
- Shop
- move
- without bag
- with bag
Example: Plans and Hypotheses

A Currently Executing Plan
Example: Plans and Hypotheses

Hypotheses matching obs “look”
Example: Plans and Hypotheses

Hypotheses NOT matching obs “look”
SBR Key Ideas

• Sacrifice memory to gain speed
  – Entire grounded library in memory, in advance
  – Auxiliary data structures to support fast queries

• Distinguish recognition queries
  – Given last observation, what might be true now
  – Given history of observations, what might have been true
SBR Key Ideas

• Sacrifice memory to gain speed
  – Entire **grounded** library in memory, in advance
  – Auxiliary data structures to support fast queries

• **Distinguish recognition queries**
  – Given *last observation*, what might be true now
  – Given *history of observations*, what might have been true

This allows computing only what is needed!
Current State Query

What is True

• Generating plan-path hypotheses:
  – Tag matching plans by observation time
  – Propagate tags up/down according to temporal consistency

• Plan P is temporally consistent in time-stamp t if:
  – No incoming sequential edges
  – OR: Exist previous plan tagged with t-1
  – OR: Tagged with t-1

• Efficient: O(M log L)
  – M number of matching plan steps
  – L size of library
Current State Query: Example

- root
- security
- entrance
- board
- look
- X-ray
- move
- gate
- without bag
- with bag
- Shop
- coffee
- look
- move
Current State Query: Example

Time 1: Observation is "look"
Current State Query: Example

Time 1: Propagate
Current State Query: Example

Time 2: Observation is "moving"
Current State Query: Example

Time 2: Propagate
Current State Query: Example

Time 2: Propagate
Current State Query: Example

Time 3: Observation is “gate”
Current State Query: Example

Time 3: Propagate
History of States Query

What was True

- CSQ is situated: only hypothesizes as to current state
- Does not explain complete sequences

All possible sequences (1,2,3)? No.
History of States

security

board

look

X-ray

1

2

3

entrance

root

look

move

without bag

with bag

coffee

Shop

look

look

look

with bag

without bag

x-ray completes security

Security terminates at time 2
History of States

Security must start at time 1
History of States

- entrance
- coffee
- move
- Shop
- look
- without bag
- with bag
- look
- X-ray
- without bag
- with bag
- look
- move
- gate

Finally
History of States Query

- Generate (or incrementally build) hypotheses graph
  - Vertices: denoting hypotheses at time $t$
  - Edges: connect valid continuations from $t$ to $t+1$

Time

1
- Entrance → look
- Entrance → move → with bag
- Entrance → move → without bag
- Security → look
- Security → xray → with bag
- Security → xray → without bag
- Board → move

2

3
- Board → gate
History of States Query

• Generate (or incrementally build) hypotheses graph
  – Vertices: denoting hypotheses at time $t$
  – Edges: connect valid continuations from $t$ to $t+1$

• Valid hypotheses: paths from vertices in time $t$ to time 1

Time

1

Entrance → look

Entrance → move → with bag

Entrance → move → without bag

Security → look

Security → xray → with bag

Security → xray → without bag

Board → move

Board → gate
SBR Highlights
[IJCAI 2005, MOO 2004-2005]

• **Highly efficient, Complete:**
  – Match observations to plan library vertices in $O(1)$
  – CSQ is $O(M \log L)$, $M$ # of matches, $L$ size of library
  – History query is polynomial:
    • Graph construction is $O(M^2)$ for each observation
    • Hypotheses extraction is $O(TM^2)$
• Extensions for interleaving, interrupting, ...
• Limitation: grounded library, must fit in memory
Lazy Commitment in SBR

• Book keeping allows delaying inference
• **Compute only if queried**
  - e.g., hypotheses graph can be built only on history query
• No commitment to ranking
  - Probabilistic or decision-theoretic ranking is separate
  - SBR as filter [AAAI 2007]; PHATT, SLIM use similar approach
Mirroring
[ACS 2016, IJCAI 2017, AAAI 2018]

Observations and Plans are in Continuous Space

gestures, motions, goal locations: trajectories
Tracked Trajectories
Discretization (e.g., using grid)
Discretization (e.g., using grid)
Discretization (e.g., using grid)
Resulting Observations

1
2
3
4
5
6

1
2-3
4-6
Early Commitment to Discretization

**Theorem 1:**
For any discretization, can find case where it fails

**What is the goal of the agent?**
A or B?

**Continuous Version**

**Discrete Version**
Late Commitment to Discretization

**Theorem 2:**
For any case, can find discretization where it succeeds

What is the goal of the agent?
A or B?

Continuous Version  
Discrete Version
Challenge: Allow Late Commitment to Discretization

- Plan recognition libraries require early commitment
  - Hierarchical structures, grammars, HMMs and variants

- Plan recognition by planning (PRP) has potential
  - Generates hypotheses ad-hoc, after getting observations
  - Expensive for online recognition
PRP in Continuous Domains

- Observations are of continuous actions?
  - Requires domain theory describing continuous domains

- Requires planner that can work in continuous domains
  - Lots of these in OMPL (Open Motion Planning Lib)
  - But cannot compute path that “deviates from O”
Mirroring [IJCAI 17, AAAI 18]

• Revised procedure for ranking hypotheses
  – Optimal plan (plan(G)) vs observed plan (plan(G+O))
  – Closer to formulation in [R&G 2009] (abandoned?)

• Generalize planning domain theories
  – Actions generate trajectories, not single final state

• Online recognition
  – 2T planner calls, each goal  → (T+1) planner calls each
  – Heuristics can improve performance (guaranteed)
Plan Recognition Problems

- Given $R = \langle W, I, G, O \rangle$
  - $W$ domain; $I$ initial state; $G$ set of possible goals; $O$ observations
- Find plans in $W$, from $I$ to a goal in $G$, that match $O$
Plans in Domain W

Plan: sequence of actions

- States defined using fluents (numerical values allowed)
  - e.g., on(A,B)=true, fuel-remaining(robot)=50.33, pos-x(r)=4.5

- Actions: trajectories of state-changes
  - $\delta(s_{BEG},a) = (s_{BEG},...,s_{END})$
  - $s_i = (s_{i-1} \setminus \text{DEL}_a(s_i)) \oplus \text{ADD}_a(s_i)$
  - Proper generalization of STRIPS actions
Finding good plan hypotheses

• Want: plan hypothesis $\pi_R = \arg\max \ P(\pi \mid O)$
  - Intuitively – “best matches the observations”

• We want matching that maximizes
  
  \[
  P(\pi \mid O) = \beta \ P(O \mid \pi) \ P(\pi) \\
  = \beta \ P(O \mid \pi) \ P(\pi \mid g) \ P(g)
  \]

  Focus: Maximize $P(O \mid \pi)$, $P(\pi \mid g)$
How to maximize $P(O|\pi)$, $P(\pi|g)$:

Two principles:

• $P(O|\pi)$ [obs. given plan]: prefer plans matching obs.
  – Minimize $\text{Error}(\pi, O)$: Accumulated distance between $\pi$ and $O$
  – $P(O|\pi) = 1 / (1+\text{Error}(\pi, O))$

• $P(\pi|g)$ [plan given goal]: prefer optimal plans
  – Assume rationality of observed
  – Higher $P(\pi|g)$ when $\pi$ closer to ideal plan $\pi^*$
  – Defined as (normalized) ratio between costs of $\pi$ and $\pi^*$
How to maximize $P(O|\pi)$, $P(\pi|g)$: Shortcut

Two principles:

- $P(O|\pi)$ [obs. given plan]: prefer plans matching obs.
  - Minimize $Error(\pi, O)$: Accumulated distance between $\pi$ and $O$
  - $P(O|\pi) = \frac{1}{1+Error(\pi, O)}$
    - = 1 when perfect = plan that goes through obs.

- $P(\pi|g)$ [plan given goal]: prefer optimal plans
  - Assume rationality of observed
  - Higher $P(\pi|g)$ when $\pi$ closer to ideal plan $\pi^*$
  - Defined as (normalized) ratio between costs of $\pi$ and $\pi^*$
Domains

• Motion planners in OMPL [Şucan et al. 2012]
• Polygon drawing planner [Vered et al. 2016]
• ROS MoveBase standard navigation package
Summary: Lazy is Good

• Late commitment to discretization: a MUST
  – Remember the theorems (and shown experimentally)
  – Mirroring: a novel form of PRP allowed this

• Late commitment to computing queries: Efficient

• Promising: distinguish queries in PRP
  – Ignore (most of) the past [Masters & Sardina 2017]
  – Pre-computing [Marting et al. 2015, Pereira et al. 2016, 2017]

• Thanks: Friendly organizers and atmosphere at PAIR

galk@cs.biu.ac.il
http://www.cs.biu.ac.il/~galk/