# Intent-aware Multi-agent Reinforcement Learning

Siyuan Qi and Song-Chun Zhu

2018 IEEE International Conference on Robotics and Automation(ICRA)

**Soutrik Bandyopadhyay**

2020EEZ8451

Term Paper Presentation for ELV700

Apr 10, 2023

# Table of Contents

# Introduction



- In the course we saw how to predict "**intent**"
- After figuring out the intent, how can we go about considering complex tasks ?
  1. Cooperative tasks
  2. Safety critical tasks
  3. Competitive tasks

# On Utility

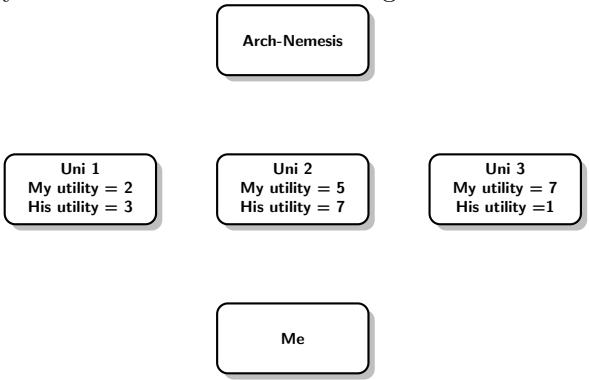


We need a way to encode the inter-relationship between different agents

- How good a particular goal is for a given agent
- Expected long term reward for given agent pursuing the given goal

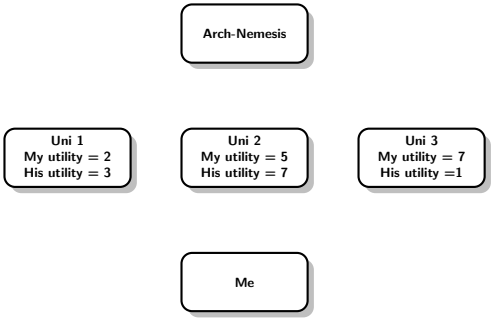
# Example

You and your arch-nemesis are selecting universities to study in




If both of you choose the same uni , a penalty of 10 is doled out

# Example




| My Action | My belief of other agent | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 1 | -10 | 2 | 2 |
| 2 | 5 | -10 | 5 |
| 3 | 7 | 7 | -10 |

# Intent aware multi-agent RL



- Intrinsic value for the different goals unknown
- Can we learn the intrinsic value from reward data considering the intent of other agents

# Table of Contents

# Problem Statement



We consider the multi-agent problem with

- $n$ agents
- $m$ goals
- Each agent has an inner-loop controller that brings the agent to the desired goal location
- Objective is to choose the desired goal location based on the intent of the other agents.

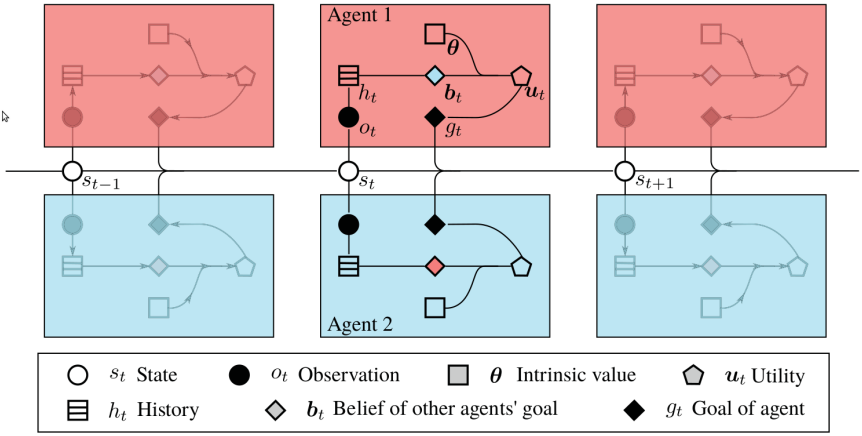
# Mathematical Notation



At each time step $t$

- Agent makes an observation $o_t$ from overall state $s_t$(unknown)
- Past history of observations $h_t$
- Based on history agent infers intent of other agents ($b_t$, probability distribution over agents)
- Compute the desired goal location ($g$) for each agent based on the intent of other agents.

# Pictorial Representation



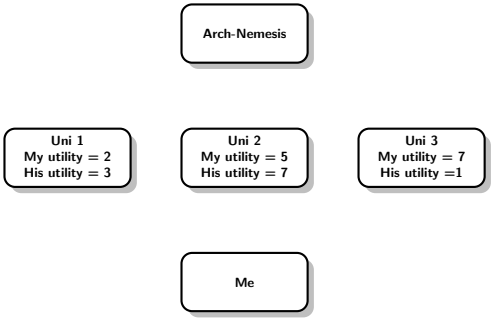
# Utility of the agents



$$g_i = \underset{g}{\operatorname{argmax}} \; u(g|b_{-i}; h, \theta) \tag{1}$$

- $u$ denotes the utility function
- $g_i$ denotes the chosen goal for the current agent
- $b_{-i}$ denotes belief of other agents
- $h$ denotes history of observations
- $\theta$ denotes the intrinsic value

# Example




| My Action | My belief of other agent | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 1 | -10 | 2 | 2 |
| 2 | 5 | -10 | 5 |
| 3 | 7 | 7 | -10 |

# Generalizability of the framework



- Safety problem (Avoid the same goal)
- Cooperation for the same goal
- Cooperation for two different goals

# Q-function

$$u(g_{ik}|b_{-i};h,\theta) = \sum_j \sum_l \theta_{ik,jl} p(g_{jl}|h) \tag{2}$$

where $\theta = (\theta_{ik,jl}) \in \mathbb{R}^{m\times n\times m}$ is the intrinsic utility (Agent $i$ going to goal $k$ and Agent $j$ going to goal $l$)

The Q function is defined as

$$Q(h, g_{ik}) = u(g_{ik}|b_{-i};h,\theta) =< \theta, \phi_{ik}(h) > \tag{3}$$

where $\phi_{jl}(h) = p(g_{jl}|h)$ and $p(g_{ik}|h) = 1$ and $p(g_{iq}|h) = 0$ for $q \neq k$

# How to compute $p(g|h)$?

$$p(g|h) = \frac{p(g)p(h|g)}{p(h)} \propto p(g)p(h|g) \quad (4)$$

where $p(h|g)$ is given by

$$p(h|g) = \frac{1}{z}\exp(-\beta d(\Gamma_o, \Gamma_p)) \quad (5)$$

- $\Gamma_o$ is observed trajectory from history $h$
- $\Gamma_p$ is predicted trajectory assuming agent is heading for $g$

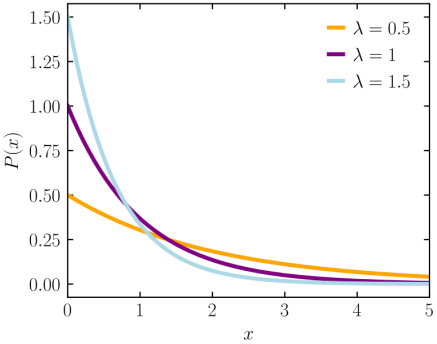



Figure: Boltzmann Distribution

# Dynamic Time Warping algorithm

How to compute distance between vectors of unequal lengths ?[1]

**Dynamic Time Warping algorithm**

Let $\Gamma_o$ have length $n$ and $\Gamma_p$ have length $m$

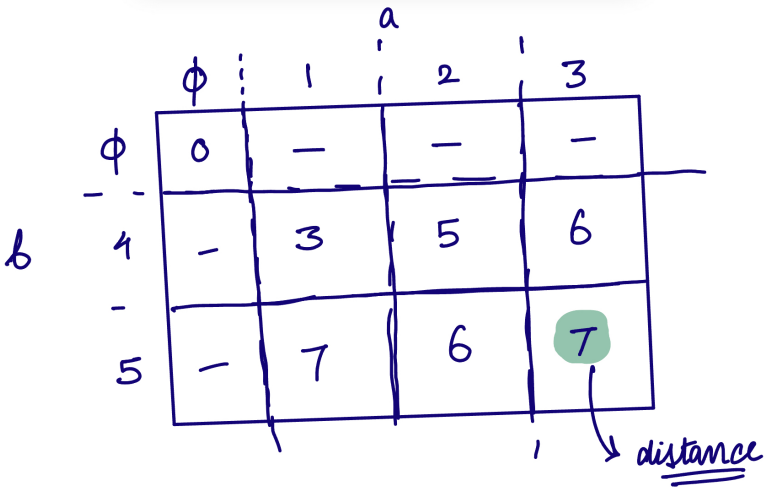
$$d(\Gamma_o[:i], \Gamma_p[:j]) = d(\Gamma_o[i], \Gamma_p[j]) + \min(d(\Gamma_o[:i-1], \Gamma_p[:j]), d(\Gamma_o[:i], \Gamma_p[:j-1]), d(\Gamma_o[:i-1], \Gamma_p[:j-1]))$$

corresponding to insertion, deletion, match.

[1] "Dynamic time warping," *Information Retrieval for Music and Motion*, pp. 69–84, 2007. DOI: 10.1007/978-3-540-74048-3_4.

# Example

Computing distance between $a = [1, 2, 3]$ and $b = [4, 5]$



# Learning the unknown $\theta$

Temporal Difference(TD) error

$$\delta = r_{t+1} - \overline{r}_t + Q(h_{t+1}, g_{t+1}; \theta) - Q(h_t, g_t; \theta) \tag{6}$$

where

$$\theta_{t+1} = \theta_t + \alpha\delta\nabla Q(h_t, g_t; \theta_t) \tag{7}$$

$$\overline{r}_{t+1} = \overline{r}_t + \beta\delta \tag{8}$$

**Algorithm 1:** Intent-aware learning in continuing tasks

**Input** : Linear approximation function $Q(h, g; \boldsymbol{\theta}) =< \boldsymbol{\theta}, \boldsymbol{\phi}(h) >$

**Parameters:** Learning rate $\alpha, \beta > 0$
Goal update frequency $f$

Initialize $\boldsymbol{\theta}, \bar{r}$ arbitrarily (e.g., $\boldsymbol{\theta} = \mathbf{0}, \bar{r} = 0$)
Initialize obervation $o$ and goal $g$
Initialize history $h$ as an empty stack
**while** *True* **do**
  Plan for $g$, observe $r$, $o$
  Push $o$ into $h$: $h' \leftarrow [h, o]$
  **if** $r == 0$ *and* $random() < \frac{1}{f}$ **then**
    `// Keep goal with randomness`
    **continue**
  **else**
    Compute $\boldsymbol{\phi}(h, o)$ by inferring intent of other agents
    Choose goal $g'$ as a function of $Q(h, \cdot; \boldsymbol{\theta})$ (e.g., $\epsilon$-greedy)
    **if** $r! = 0$ **then**
      $\delta \leftarrow r - \bar{r} + Q(h', g'; \boldsymbol{\theta}) - Q(h, g; \boldsymbol{\theta})$
      $\bar{r} \leftarrow \bar{r} + \beta\delta$
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\delta\nabla Q(h, g; \boldsymbol{\theta})$
    **end**
    $o \leftarrow o', g \leftarrow g'$
  **end**
**end**

# Table of Contents

# My implementation



https://github.com/SoutrikBandyopadhyay/intent_aware_rl

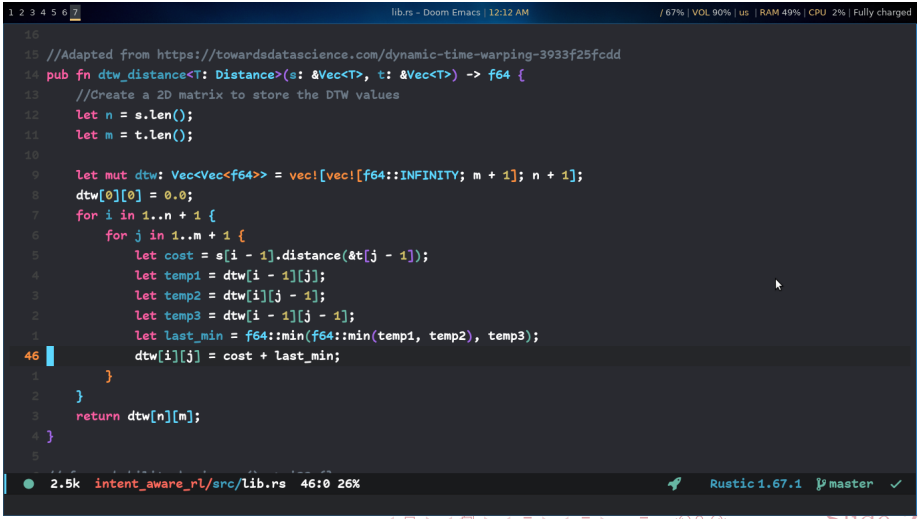
A library I developed for implementing the Q learning framework mentioned in this paper.

# A brief technical aside

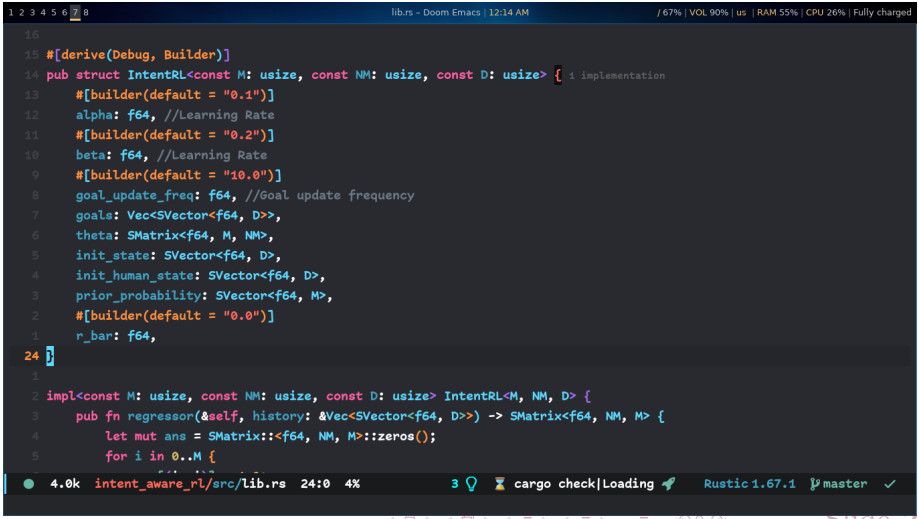


- Project written in Rust (programming language) from scratch
- Why Rust?
  1. Excellent performance (close to C)
  2. Memory Safe (reduces a significant number of memory related vulnerabilities)
  3. Friendly syntax

# Code snippets



# Code snippets (contd.)



# Table of Contents

# Simulation Setup

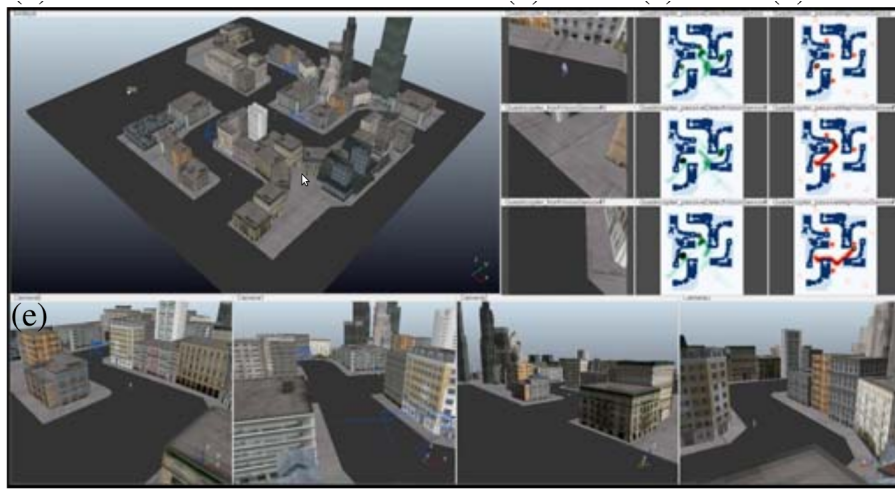


**Multi-robot aerial surveillance**

- There are 5 buildings, and a person is trying to sneak into the buildings.
- There are 3 aerial robots that are tasked to not let the human walk into a building undetected

Intent must be identified

1. Human - For suspicious activity
2. Other robots - For collaboration


(e)

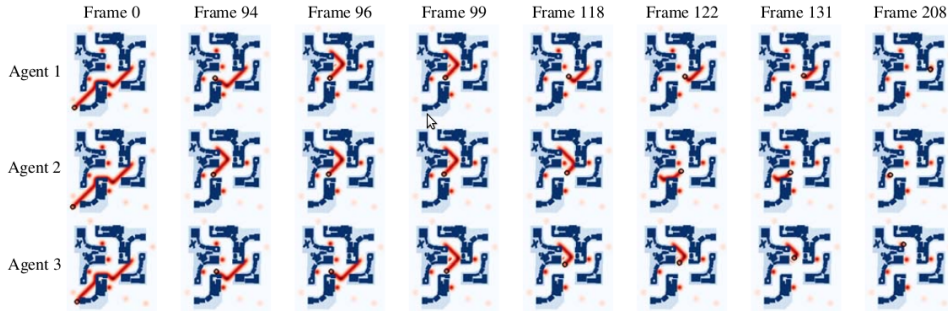



Fig. 8: Cooperative behavior of the learned model. Each row demonstrates the values of different goals (the higher the value the higher intensity of red), the chosen goal, and the planned path for a robot at different time frames. The values of monitoring buildings are higher than monitoring entrances or crossroad. At the beginning, all robot agents intended for the farthest building since they thought it is less likely for the other robots to monitor. During the process, the agents realized that the other agents might have chosen the same goal as themselves and chose another goal accordingly (e.g., agent 2 at frame 94). The agents chose different goals iteratively and finally reached an equilibrium that three robots were monitoring three different buildings. This dynamic process reflects how humans naturally interact with each other in a complex environment.

# Comparision



| | Testing Accuracy(%) | | | | | |
|---|---|---|---|---|---|---|
| | Training Scene | | | | | Testing Scene |
| Training Iterations | 100 | 200 | 300 | 500 | 1000 | — |
| random | 29.7 | 29.7 | 29.7 | 29.7 | 29.7 | 34.4 |
| greedy | **31.4** | **31.4** | 31.4 | 31.4 | 31.4 | 35.2 |
| RNN-POMDP | 17.3 | 22.9 | 20.8 | 21.0 | 23.6 | 14.3 |
| ours | 13.8 | 19.1 | **60.3** | **62.4** | **59.3** | **77.3** |

# Table of Contents

# Concluding Remarks



- Proposed an intent-aware multi-agent RL algorithm
- Generalizable framework
- Q function is linear (guaranteed convergence)

# Possible Extensions to the work



- Sophisticated intent estimators
- Sophisticated high-level planning algorithms

# References I

[1] "Dynamic time warping," *Information Retrieval for Music and Motion*, pp. 69–84, 2007. DOI: 10.1007/978-3-540-74048-3_4.

Thank you :)