Relational Forward Models for Multi-Agent Learning

Authors: Andrea Tacchetti, H. Francis Song, Pedro A. M. Mediano, Vinicius Zambaldi, Neil C. Rabinowitz, Thore Graepel, Matthew Botvinick and Peter W. Battaglia

Presented by: Oriol Corcoll
Relational reasoning
What is relational reasoning? Why is it important?

- Extracting meaningful patterns out of an stream of data
- Relations between objects, ideas, situations, ...
- Different types: analogy, anomaly, antimony, and antithesis
- Mainly controlled by the Prefrontal Cortex
The early emergence and puzzling decline of relational reasoning:
Effects of knowledge and search on inferring abstract concepts,
Gopnik et al. 2016
“Without it, humans would be trapped in a world filled with isolated stimuli, unable to connect objects perceived across time and space“, William James 1890
Relational Forward Models (RFM)
Reinforcement learning

https://becominghuman.ai/the-very-basics-of-reinforcement-learning-154f28a79071
RFM module - Architecture

- Based on Graph Networks (GN)
- Trained using supervised learning
- Used for action or reward prediction
RFM module - Graph Networks

$e'_k = \phi^e (e_k, v_{r_k}, v_{s_k}, u)$,  \hspace{1cm}  \hat{e}'_i = \rho^{e\rightarrow v} (E'_i)$,

$v'_i = \phi^v (\bar{e}'_i, v_i, u)$,  \hspace{1cm}  \bar{v}' = \rho^{v\rightarrow u} (V')$,

$u' = \phi^u (\bar{e}', \bar{v}', u)$,  \hspace{1cm}  \bar{e}' = \rho^{e\rightarrow u} (E')$
RFM module - Graph GRU

- Update functions ($\Phi$) implemented with Gated Recurrent Unit (GRU)
- Each GRU has a hidden state of 32 for each of vertices, edges, and globals
- Aggregation functions are summations
- Outputs an updated GN
RFM module - Encoder/Decoder

- **Encoder**: GN to encode raw input
- **Decoder**: produces action or reward predictions
- Update functions $\Phi$ are MLPs with 1 hidden layer and 64 units
- Aggregation functions are summations
What are RFMs trying to solve?

- Relational reasoning for reinforcement learning (RL)
- Decentralised multi-agent coordination
- Enable the analysis of multi-agent systems
- Independent module to augment RL agents
Experimental setup
Cooperative Navigation (Lowe et al., 2017)

- Two agents in a 6x6 arena
- Two target tiles
- **Goal**: cover both tiles
- Agents are **rewarded** if goal is met
Coin Game (Raileanu et al., 2018)

- Two agents in a 8x8 room with 12 coins
- Each coin can be one of 3 colors (4 coins per color)
- 2 colours carry positive reward and 1 color negative
- Agents have access to information about 1 good color
- Collective reward for positive and negative colors
Stag Hunt (Peysakhovich & Lerer, 2017b)

- Two (or four) agents in an arena with 3 static Stags
- Agents can pick apples for a reward of +5
- If two agents step into a square with a Stag at the same time they receive a reward of +10
- They get negative reward if only one agent steps into an Stag
Model training

- Graph network built with all the agents and entities (apples, stags, coins, ...)
- Vertex value is the position, type of entity/agent, available/collection, last action
- Each agent is connected to all entities and other agents
- 500K episodes for training and 2.5K episodes for testing
RFM for action prediction
Comparison against other models

- Neural Relational Inference (NRI)
- Vertex Attention Interaction Network (VAIN)
- Feedforward (no recurrent)
- No relational
- Encoder MLP + LSTM + Decoder MLP
Graph Network based on Variational Autoencoders (VAE)

Graph structure learnt in an unsupervised manner

Exactly the same as RFM but includes the learning of the graph

NRI

VAIN

Special case of Graph Networks

VAIN uses different edge update functions

Feedforward Graph
Action prediction
Analysis Tools
Influence over other agents

- The Euclidean norm of the edge value ($||e||$) as influence measurement
- Measures how entities or other agents can influence an agent
- Answer the question of: “Where will the agent move next?”
- It can also predict coordination between agents
Influence over other agents

(a) Edge activation magnitude is predictive of future behavior.
Influence over other agents

(b) Edge activation magnitude reveals changes in what drives agents' behavior over time.
Influence over other agents

(c) Edge activation magnitude discovers situations that alter agents’ social influence.
Usefulness of other agents

- Prune graph by removing the edge between the agent 1 and agent 2

- If $G_{pruned} > G_{full}$ then agent 2 is hindering agent 1

- If $G_{pruned} \leq G_{full}$ then agent 2 is helping agent 1
Usefulness of other agents
Agents augmented with RFM
Architecture

- RFM module trained alongside the policy
- Action predicted by the RFM module used to create a heatmap with the next position of the other agent(s)
- Base policy is a CNN+MLP+LSTM
Results
Coin Game results analysis

- The efficiency difference is due to the ability to discern teammate’s preferences
- Not as complex as the authors thought
- Other papers reporting on this game used poor performing baselines
- Strong baselines leaving little room for improvement
Deep diving into Coin Game results
Conclusions

- RFM can capture the social dynamics of multi-agent environments
- Agents learn to cooperate with one another faster when using RFM
- RFM allows researchers to answer new questions like “who/what is influencing the agent’s decisions?”
Thanks!

Q&A