COPS RL reading grp

Topic: Transfer Learning in MARL

3rd Meet

10th and 13th Feb

Topics covered:

Transfer Learning

- Simultaneously Learning and Advising in Multiagent Reinforcement Learning by Silva, Felipe Leno da; Glatt, Ruben; and Costa, Anna Helena Reali. AAMAS, 2017.
- Accelerating Multiagent Reinforcement Learning through Transfer Learning by Silva, Felipe Leno da; and Costa, Anna Helena Reali. AAAI, 2017.
- Accelerating multi-agent reinforcement learning with dynamic co-learning by Garant D, da Silva B C, Lesser V, et al. Technical report, 2015
- Transfer learning in multi-agent systems through parallel transfer by Taylor, Adam, et al. ICML, 2013.
- Transfer learning in multi-agent reinforcement learning domains by Boutsioukis, Georgios, Ioannis Partalas, and Ioannis Vlahavas. European Workshop on Reinforcement Learning, 2011.
- Transfer Learning for Multi-agent Coordination by Vrancx, Peter, Yann-Michaël De Hauwere, and Ann Nowé. ICAART, 2011.

Yash

Simultaneous Learning and Advising in Multi-Agent Reinforcement Learning

Teacher Student Framework

- When an agent (student) is not confident about what action to take while learning, when in a particular state, the teacher suggests an action, if the teacher is confident about his own policy in that state.
- Initially, in single agent RL, the teacher was an expert in the task, (could be agent or human).
- Also bound by an advice budget b.
- Hence, necessary to choose, when to give advice.
- Teacher’s policy is fixed. (generally)

Importance Advising Strategy

Provide advice, only when the Importance Metric I(s) is greater than a threshold.

$$I(s) = \max_a Q_{teacher}(s, a) - \min_a Q_{teacher}(s, a).$$

Basically, the difference in the best and worst action, so if a wrong action is taken by student, it might lead to bad results, hence advise. Budget might not be used properly here.

Mistake Correcting Advice

- Provide advice, only when the student’s intended action is wrong according to the teacher’s policy.
- Formulated to include multiple teachers.
- Agent can refuse their suggestions.

JOINTLY INITIATED FRAMEWORK

Both student and teacher need to agree that advice needs to be given.

Advising Strategy in MAS

- Every agent is learning. Most of the agents might have suboptimal policy.
- Hence, we have ad hoc advisor-advisee relations between agents.
- These relations are established for a single step.
- Relations are based on the confidence of each agent in its own policy in that state.

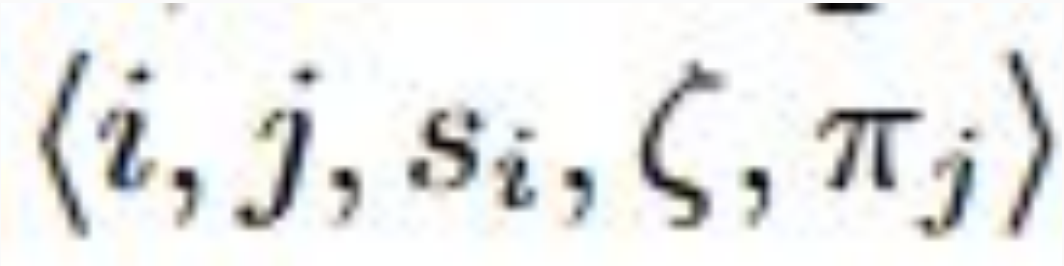
Each agent has a tuple:

- Probability of asking for advice (should decrease with time)
- Probability of giving advice(increase)
- Asking Budget
- Giving Budget
- Set of Reachable Agents
- Function to combine received advice

$$\langle P_{ask}, P_{give}, b_{ask}, b_{give}, G, \Gamma \rangle$$

Advisor Advisee Relation

- Advisee (with P-ask)
- Advisor (with P-give)
- State from the perspective of the advisee
- Function to translate state such that advisor is able to interpret
- Advisor’s policy



How does an advisee select action?

Algorithm 1 Action selection for a potential advisee i

Require: advising probability function P_{ask}, budget b_{ask}, action picker function Γ, confidence function Υ.

for all training steps **do**
  Observe current state s_i.
  if $b_{ask} > 0$ **then**
    $p_{s_i} \leftarrow P_{ask}(s_i, \Upsilon)$
    With probability p_{s_i} **do**
      Define reachable agents $G(s_i)$.
      $\Pi \leftarrow \emptyset$
      for $\forall z \in G(s_i)$ **do**
        $\Pi \leftarrow \Pi \cup z.advice(s_i)$
      if $\Pi \neq \emptyset$ **then**
        $b_{ask} \leftarrow b_{ask} - 1$
        $a \leftarrow \Gamma(\Pi)$
        Execute a.
  if no action was executed in this step **then**
    Perform usual exploration strategy.

How does an advisor give advice?

Algorithm 2 Response to advice requirement.

Require: advising probability function P_{give}, budget b_{give}, advisor policy π, advisee state s_i, state translation function ζ, confidence function Ψ.

1: **if** $b_{give} > 0$ **then**
2: $\quad p_{s_j} \leftarrow P_{give}(s_i, \Psi)$
3: $\quad$ **With** probability p_{s_j} **do**
4: $\quad\quad b_{give} \leftarrow b_{give} - 1$
5: $\quad\quad$ **return** $\pi(\zeta(s_i))$
6: **return** $\emptyset$

Deciding When to Give Advice

New importance metric, taking into account, how many times did the agent explore the given state

$$P_{ask}(s, \Upsilon) = (1 + v_a)^{-\Upsilon(s)}$$

$$P_{give}(s, \Psi) = 1 - (1 + v_g)^{-\Psi(s)}$$

$$\Upsilon_{visit}(s) = \sqrt{n_{visits}(s)},$$

$$\Psi_{visit}(s) = log_2 n_{visits}.$$

TD Based Confidence Function

$$\Psi_{TD}(s) = \Upsilon_{visit}(s)\ |max_a Q(s,a) - min_a Q(s,a)|$$

- Visit Based Advising
- Temporal Difference Based Advising

Tharun

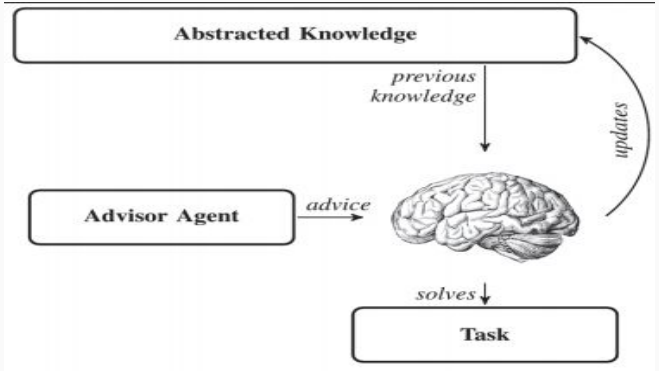
Accelerating Multiagent Reinforcement Learning through Transfer Learning

Overview

- Multi-agents suffer from curse of dimensionality and has scalability issues.
- Transfer learning by reusing previous knowledge, both from past solutions and advising between agents.
- Ex. an agent learning to play football can use skills acquired by agent which was trained to run and navigate.

Methods

- Transfer learning framework proposed :
  - Agent has an **Abstract Knowledge Base** from which it can extract previous solutions of tasks that are similar to the new one.
  - An **Advisor Agent** may provide advice helping the agent to achieve its goals.



Advisor Agent : Teacher student framework

- A teacher agent observes a student agent and suggests actions for states.

Drawbacks

- Requires a teacher able to perform the task with expertise when the learning starts.
- The teacher and student cannot learn together (the teacher must be pretrained on a similar task)

Alternative : Advisor-advisee relations

- Instead of having a fixed teacher, the advisee evaluates its confidence in the current state, and broadcasts an advice request for all the reachable friendly agents in case its confidence is not high enough.
- In case the advisor’s confidence is high enough, the action is taken.
- All the agents learn together.

Abstract Knowledge Base

- Agent extract previous solutions of tasks that are similar to the current one.
- The agents must be able to compute task similarities in order to select the most similar task from the knowledge base and reuse its solution.
- Ex : Comparing the attributes of the objects in the environment.

Example

- Multiagent ObjectOriented MDP (MOO-MDP) (Silva, Glatt, and Costa 2016), a relational approach in which the state space is described through classes of objects.
- Ex: Goldmine , Class c = {gold, wall,miner} and attributes = {x,y}.

Drawback

- Provide hand-coded task mappings, autonomously computing task similarities is hard.

Vikhyath

TRANSFER LEARNING IN MULTI-AGENT SYSTEMS THROUGH PARALLEL TRANSFER

Introduction

- Paper from 2013
- Basically a paper on communication
- Given the name “Parallel Transfer Learning”(PTL)
- Contrasted with Sequential TL
- Seq TL means first one task is conducted, then the second
- PTL has tasks happening all at once

TL in Multi Agent Systems

- Earlier paper(Boutsioukis et al. (2012)) shows that TL is beneficial in MAS, but using a multi-agent source task has no benefit over a single-agent source task.
- Another limitation is that no guarantee good info is transferred, negative TL
- 3 metrics to measure performance(better early performance, better final performance and speed of learning).
- PTL works for MAS as the agents are likely to learn similar things

Parallel Transfer Learning

“At each time step, if an agent decides it has useful information to share with another agent, it sends the information to that agent. It then checks its own communications buffer to see if any information has been received from other agents in the system. If it has, it decides whether to accept this information or discard it. Through repeating this procedure, over time learned information will propagate through the system.”

When to Transfer

- Everytime value of a state changes significantly. Consumes too much bandwidth and resources. Chance of it being true value less as it may have been sparsely sampled
- Only after it appears to have converged. Very infrequent transfer.
- Ideal would be somewhere in between depending on case

What to Transfer

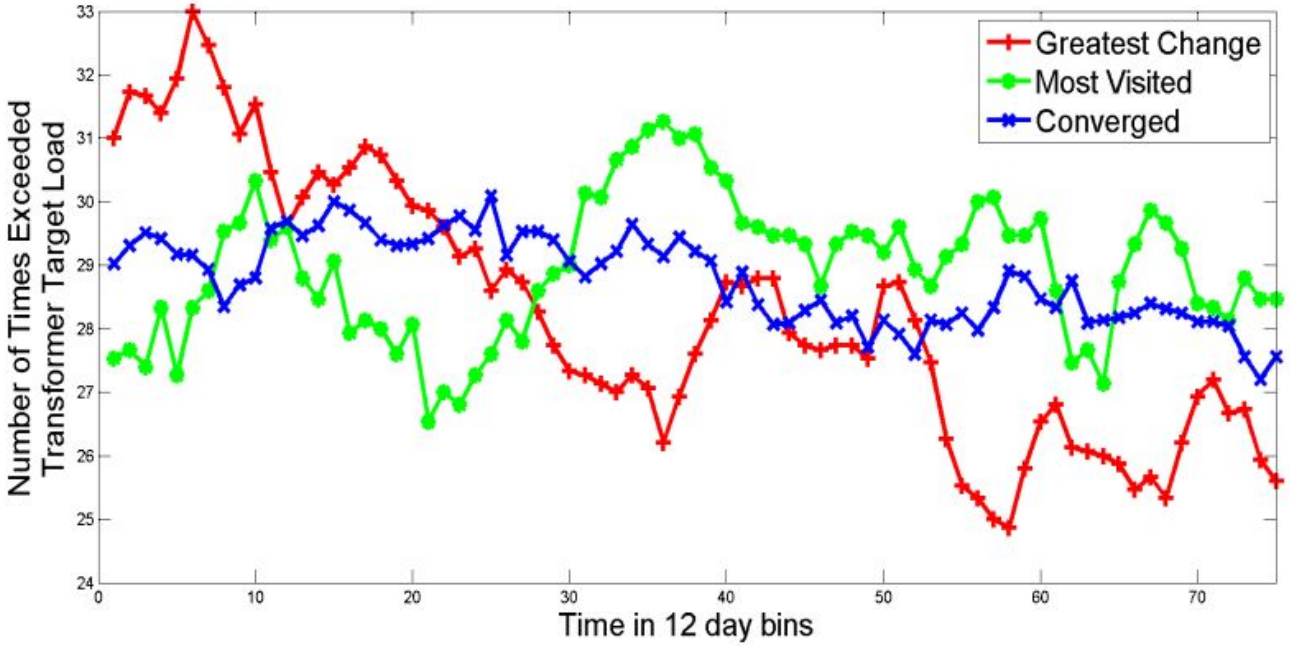


Figure 4. Moving average of number of times over desired transformer load using the Greatest Change, Most Visited and Converged methods.

Receiving Information

- Maintaining statistics about how often a state has been visited and comparing this to how often the information received was sampled and accepting the most sampled value is a possible approach => This fails if the env keeps changing
- Another method is if merging the new information can move a particular value towards its true value, then it should be done otherwise the received information should be discarded. This can be done by seeing the direction of movement of the value
- The steps towards the true value should get smaller the closer to convergence the value is, if the steps have already become small then it is probably not worth merging the received information.

Source and Target Selection

- If all agents have one-to-one one-way comms, makes N(N-1) channels. Infeasible
- An agent should preferentially share knowledge with agents exploring different parts of the environment than necessarily with its near neighbours.

Results

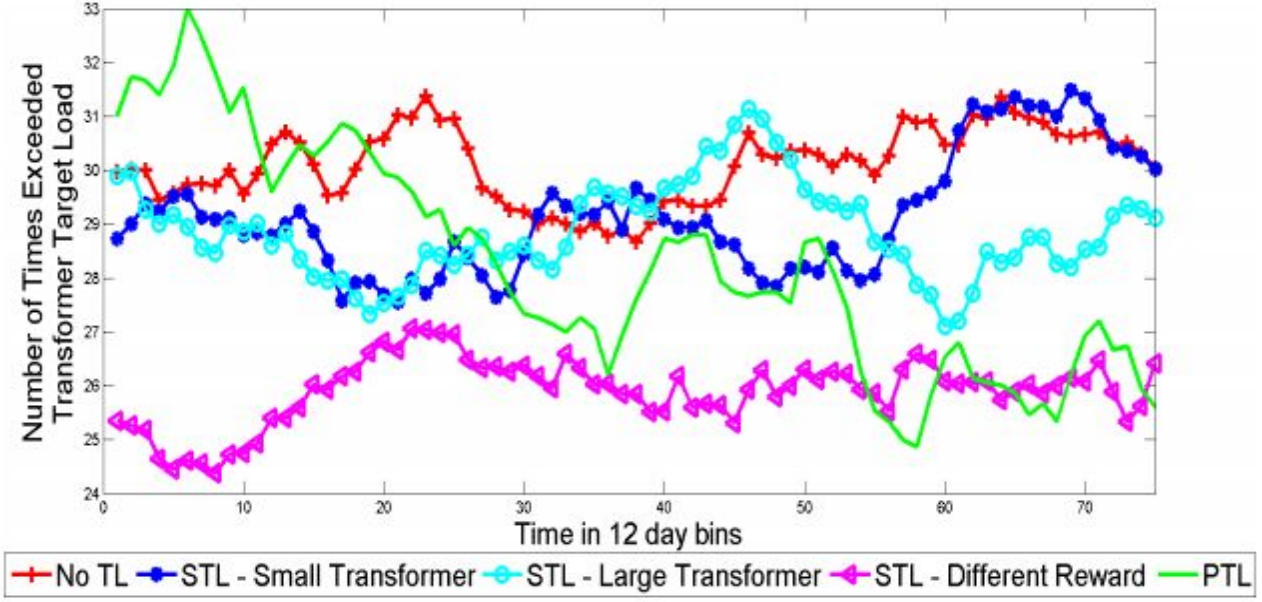


Figure 5. Moving average of number of times over desired transformer load using No Transfer Learning, Parallel Transfer Learning and Sequential Transfer Learning.