2019-Jaques

Social and Affective Machine Learning

- **Date**: 2018-10-21
- **Link**
- **Authors:**
  - Jaques, Natasha
- **Cites:**
- **Cited by:**
- **Keywords:** #social-learning #reinforcement-learning
- **Collections:**
- **Status:** #in-progress

0 - Abstract

- *Social Learning* - agents communicating knowledge to other agents. This includes communication through non-verbal cues such as facial expressions.
- Novel reinforcement-learning ideas that improve social and affective learning:
  - Agents can learn from the affect of their actions on other agents, increasing coordination and communication.
  - Agents can learn from humans through non-verbal cues such as facial expressions.
  - Agents must learn effectively using sparse, off-policy data in environments without the ability to explore, given human safety considerations.
  - Agents must be able to interpret human facial cues.

1 - Introduction

- Motivation for studying social learning:
  - ML suffers from a lack of social intelligence.
    - *Example* 1- a chat bot's performance could be improved by measuring the mood of its partner.
    - *Example 2* - a recommender system's performance could be improved if it could sense its user's mood over time.
  - ML suffers from a lack of ability to learn from other agents or generalize to unfamiliar tasks.
    - These are abilities that we know humans learn by using social intelligence.
  - ML suffers from a need for large, human-curated data sets, particularly reinforcement learning.
    - The solution to this problem is to create agents with intrinsic motivations that generalize across environments.
    - One such "intrinsic motivation" could be the drive to socialize, as we know humans possess.
- Parts of the dissertation:
  - First - AI agents can learn from one another.
  - Second - AI agents can learn from humans using implicit social cues.
  - Third - AI agents can sense social cues.

1.1 - Part I - Social Learning from Other AI Agents

- RL requires a well curated environment with which to interact, including a fully specified reward function.
  - Real-world applications are often more complex and do not come with fully specified reward functions.
  - Some environments are subjective.
  - Hand specified reward functions for complex environments are time consuming.
  - Reward functions can be trivially exploited.

- Social rewards can help to overcome these problems.
  - Social rewards are dynamic, as one agent improves her policy, the other agent must adjust too.
  - Social rewards are not trivially exploitable. If one agent attempts a trivial strategy, then another agent can adjust to it.
- MARL research has operated under the assumption that agents can share private information.
  - Examples of private information: rewards, observations, gradients at training time.
  - Problems:
    - Not realistic for environments with competitive agents.
    - Doesn't make sense for agents with internal intrinsic motivations.
    - Doesn't work for agents developed at different institutions.
    - Does not generalize to human learning.
- New type of intrinsic social motivation for MAL where agents are motivated to have causal influence over the actions of other agents. (**Chapter 3**)
  - 2019-Jaques-2 - *Social influence as intrinsic motivation for multi-agent deep reinforcement learning*
  - Rewards the mutual information shared between agents' actions, enabling complex coordination and cooperative problem solving.
    - Coordination without the sharing of private information!
    - Requires agent's to predict other agent's actions.
      - "What would this agent do if I had taken this action?"

1.2 - Part II - Social and Affective Learning from Humans

- Train ML algorithms to directly optimize for human preferences.
- Optimizing for human preferences has required manual human feedback.
- To avoid this, have the agent passively sense the user's emotional state and train the agent to be intrinsically motivated to produce positive responses.
  - "Human in-the-loop training" without direct human input.
- Human data are sparse, so agents must be efficient with limited data.
  - First pretrain a model with readily available data, then fine-tune with respect to human feedback.
- Learning from implicit cues in human conversation. (**Chapter 4**).
- Restrict the model from exploring when online, for safety.
- Use facial expressions of humans viewing the output of generated sketches as model input. **(Chapter 5)**

1.3 - Part III - Detecting Social and Affective States

- AI agents should be able to interpret non-verbal human signals.
- AI agents should also be able to interpret the broader context in which these cues are sent.
- Predicting internal human affective states **(Chapter 6)**:
  - Example: predicting whether two people having a conversation are bonding.
- Difficulties collecting and working with human data, and the tools to help **(Chapter 7)**:
  - Example: method for learning to reliably predict affective outcomes even when missing a data source (i.e. missing physiological sensor or censored location data).
- Predicting a person's wellbeing **(Chapter 8)**:
  - Example: predicting the next day's happiness, stress, and health.
  - Found that people react differently to the same stimuli, this therefore requires personalized models based on the individual.

3 - Multi-agent Social Learning Via Causal Influence

- Propose a unified mechanism for achieving coordination and communication in MARL through rewarding agents for having causal influence over other agents' actions.
- Proven to enhance coordination and communication.
- Agents learn a model of other agents using deep NNs.
    - Allows for decentralized computation of influence rewards.

3.1 - Introduction

- *Intrinsic Motivation* - reward functions that allow agents to learn useful behaviors across a variety of tasks and environments (sometimes without environmental reward).
- Achieve coordination and communication by giving agents an intrinsic reward for having *causal influence* on other agent's actions.
- *Causal Influence* - the impact of the agent on another agent's actions.
- Causal influence is determined by counterfactual reasoning.
    - At each timestep, for each action that could have been taken, the agent estimates the effect of this action on the other agent's behavior.
    - More influential actions are rewarded more.
- This is related to maximizing the mutual information between agent's actions.
- Hypothesize that this drives cooperative behavior.
- Experiments in this chapter:
    - Studied the influence reward in the Sequential Social Dilemma (SSD) environment.
        - Influence reward allows agents to learn to coordinate more effectively in SSDs.
        - Social agents outperform the baseline deep RL agents.
    - Studied the influence reward with an explicit communication channel.
        - Produced better collective outcomes.
        - Correlation between being influenced through communication messages and obtaining higher individual reward.
        - Influence reward is essential to agent coordination.
    - Trained agents independently, with each agent using a *Model of Other Agents* (MOA).
        - *Model of Other Agents (MOA)* - deep neural network that predicts other agent's actions, used to determine how an agents actions could've effected other agent's behavior.
        - Influence reward dramatically enhances coordination.

3.2 - Sequential Social Dilemmas

- *Sequential Social Dilemmas (SSDs)* - partially observable, spatially and temporally extended multi-agent games with a game-theoretic payoff structure.
- Agents can choose to defect or not:
    - If an agent defects, then she gains a higher short term reward.
    - If all agents choose not to defect, then they all receive the highest reward.
- Cooperation guarantees the highest collective reward.
- Two SSDs used:
    - *Harvest*
        - Public pool resource game where agents collect apples.
        - Apples respawn at a rate proportional to the amount of nearby apples.
        - If apples are picked too quickly, they will not respawn.
        - A greedy agent who quickly consumes all the apples will hurt everyone.
        - Agents must spread out, two agents consuming apples too closely could drain the supply.

- *Cleanup*
  - Public goods game where agents must clean a river before apples can grow, but are not able to harvest apples while cleaning.
  - Agents cleaning the river must be given the chance to consume apples.
  - Agents that only consume apples will harm the agents who clean the river, disincentivizing river cleaning hurts everyone, as the apples will not grow.
- *Schelling diagram* - diagram that shows individual payout (defector or cooperator or average of the two) as a function of the number of cooperating agents.
- Schelling diagrams for *Cleanup* and *Harvest* show that while it is advantageous for an individual to defect, the group as a whole does significantly better when more agents cooperate.
- Traditional RL agents struggle to solve this problem.

3.3 - Multi-agent RL for SSDs

- RL set up:
  - MARL Markov game is defined by the tuple $\langle S,\, T,\, A,\, r \rangle$.
  - Each agent seeks to maximize her own reward.
  - At each timestep t :
    - Each agent, k, chooses an action, $a^k \in \mathcal{A}$.
    - Actions are combined into the vector, $a_t = [a_t^0,\, \ldots,\, a_t^N]$.
    - Yields a transition in the environment, $T(s_{t+1} \mid a_t,\, s_t)$.
    - Where T is the state transition distribution.
    - Each agent receives its own reward, $r^k(a_t,\, s_t)$.
  - Partially observable setting where agent k can only see a portion of the state, s_t^k .
  - Agents seek to maximize their expected return, $R^k = \sum_{i=0}^{\infty} \gamma^i r_{t+i}^k$.
  - Distributed asynchronous advantage actor-critic (A_3C) is used to train each agent's policy π^k.
- Network design:
  - Network layers:
    - Convolutional layer that takes the board game image as input.
    - Fully connected layers.
    - LSTM recurrent layer.
    - Linear layers.
  - Input: images.
  - Output: policy, π^k , and value function, $V^{\pi_k}(s)$.
  - Some networks consume additional inputs, and output communication policies or other agents' behavior.
  - Let u_t^k refer to the internal LSTM state of the k^{th} agent at timestep t .

3.4 - Basic Social Influence

- Social influence intrinsic motivation modifies an agent's immediate reward:

$$r_t^k = \alpha\, e_t^k \,+\, \beta\, c_t^k$$

  where e_t^k is the extrinsic (or environmental) reward and c_t^k is the causal influence reward.
- Computing causal reward:
  - Suppose two agents, k and j.
  - Allow Agent j to condition its policy on agent k's action at time t, a_t^k.
    - This yields a probability distribution over potential actions for j - $p(\, a_t^j \mid a_t^k,\, s_t^j\,)$.

- Agent k could then consider a counterfactual action, $\tilde{a}_t^k$, and ask how Agent j 's distribution would've differed, $p(\ a_t^j \mid \tilde{a}_t^k,\ s_t^j\ )$.
- Obtain the marginal policy of j by sampling several counterfactual actions and averaging the result:

$$p(\ a_t^j \mid s_t^j\ ) = \sum_{\tilde{a}_t^k} p(\ a_t^j \mid \tilde{a}_t^k,\ s_t^j\ )\ p(\ \tilde{a}_t^k\ s_t^j\ )$$

  a.k.a. - j 's policy if it did not consider k 's action.

> KEY IDEA - the discrepancy between the marginal policy of j and the condition policy of j given k 's action is a measure of the causal influence of k on j .

- Causal influence reward for agent k:

$$c_t^k = \sum_{j=0,\ j \neq k}^{N} \left[D_{\mathrm{KL}} \left[p(\ a_t^j \mid a_t^k,\ s_t^j\ ) \,\middle|\middle|\, p(\ a_t^j \mid s_t^j\ ) \right] \right]$$

- This reward is related to the mutual information (MI), $I(a^k; a^j|s)$, shared between the actions of the agents.
  - In expectation, the influence reward incentivizes agents to maximize the mutual information between their actions.
- Assumptions:
  - Used centralized training to compute c_t^k directly from the policy of agent j. (See Section 3.6 for an algorithm that removes this assumption).
  - Influence is unidirectional, those that are being influenced are not also influencing.
- Causal influence algorithm:

Algorithm 2: Computing basic influence reward for agent k

Require: τ, trajectory containing actions and observations for all agents. Let T be the trajectory length.
influence $= [0] * T$
for *timestep t in* $[0, T)$ **do**
  for *other agent j in N* **do**
    prob_aj $= 0$
    for *action* $\tilde{a}_t^k$ *in* $\mathcal{A}$ **do**
      Compute $p(\tilde{a}_t^k | s_t^k; \theta^k)$ using agent k's policy network
      Compute $p(a_t^j | \tilde{a}_t^k, s_t^j; \theta^j)$ using agent j's policy network
      prob_aj = prob_aj $+ p(a_t^j | \tilde{a}_t^k, s_t^j) p(\tilde{a}_t^k | s_t^k)$
    end
    $p(a_t^j | s_t^j) \leftarrow$ prob_aj
    influence$[t]$ = influence$[t] + D_{KL}[p(a_t^j | a_t^k, s_t^j) || p(a_t^j | s_t^j)]$
  end
end

3.4.1 - Experiment I: Basic Influence

- Comparison between three models:
  - *A3C baseline* - normal RL agents.
  - *Visible actions baseline* - influence agents without the influence reward (i.e. they can still see others actions, but are not incentivized to influence).
  - *Influence* - causal influence agents.
- We see the influence agents and visible actions baseline agents both out preform the A3C baseline, in terms of total collective reward.
  - This shows that knowledge of other agent's actions is valuable by itself.
- In both games, the influence agents produce the best total collective reward, by far.
  - From this, we can conclude they cooperate the best.

> KEY IDEA - Top performing *Cleanup* agents were observed to be using their movement as a form of binary code to signal to other agent's the presence of apples. If the influencing agent moved, then it signaled to the other agents that there is an apple in that direction. Example of emergent communication!

3.5 - Influential Communication

- Studies show a connection between influence and communication in human learning.
  - Children's communication skills increase when trying to influence other's behavior during cooperative tasks.
  - "Capacity to form shared goals"

- Explicit communication channel:
  - At each timestep:
    - Each agent, k, selects a discrete communication symbol, m_t^k.
    - These symbols form a message vector, $\boldsymbol{m}_t = [m_t^0,\, m_t^1,\, \ldots,\, m_t^N]$.
    - The message vector is then given to all agents at the next timestep.
- This form of communication is not effective with self-interested agents.
- Modification to the network design:
  - The message vector, m_{t-1}, is inserted into the LSTM input as alongside the image output from the feedforward layer. (See Fig. 3.5 for a picture.)
  - LSTM layer outputs an additional A3C output: a communication strategy π_m and value function V_m to decide which symbols to choose.
  - Normal policy and value function (V_e and π_e) are trained with only the environment reward e.
- We now consider an agent's message influence rather than their action. The change in causal reward is as follows:

$$c_t^k = \sum_{j=0,\, j \neq k}^{N} \Big[D_{\mathrm{KL}} \left[p(\, a_t^j \mid m_{t-1}^k,\, s_t^j \,) \,||\, p(\, a_t^j \mid s_t^j \,) \right] \Big]$$

- Rewarding influence from communication channels fixes the problems with self-interested agents learning.
  - This means its possible to influence an agent in a non-cooperative way.
- Note:
  1. Agents are free to ignore the communication channel.
  2. Agents actions are dictated by the environmental reward; therefore, they will act in their own perceived best interest.
     - Therefore - Agents will only be influenced by the communication channel if it's helpful to them.

3.5.1 - Experiment II: Influential Communication

- Same set of agents as Experiment I - A3C baseline, Comm. baseline, Influence comm.
  - The comm. baseline's communication policy is trained without the influence reward.
- Interestingly, the top performing agents in the *Cleanup* game trained their communication policies with no extrinsic reward. ($\alpha = 0$).
- Three metrics for analyzing communication behavior:
  - *Speaker consistency* - measure of how consistently an agent issues a symbol when performing a given action.

    Formally, this is defined as the entropy between $p(\, a^k \mid m^k \,)$ and $p(\, m^k \mid a^k \,)$, normalized $\in [0, 1]$.

    For instance, we expect this value to be high when agents are communicating that they're cleaning the river.
  - *Instantaneous coordination* - two measures:
    1. Symbol / Action IC ($I(\, m_t^k \,;\, a_{t+1}^j \,)$) - measure of mutual information between influencer's symbol and listener's action.
    2. Action / Action IC ($I(\, a_t^k \,;\, a_{t+1}^j)$) - measure of mutual information between influencer's action and listener's action.
       - Note: these are measures of coordination in a given timestep and does not capture communication that could transpire over multiple time steps.
- Communication metrics results:
  - Speaker consistency results show that influence agent's clearly communicate their own actions.
  - Baseline agents show no instantaneous coordination.
    Speaker says A, listener does B.
  - Influence agents demonstrate high instantaneous coordination, limited to *influential moments*.

    Influential moment - when influence was greater than or equal to the mean.

    Agent's influence is sparsely distributed across timesteps, only 10% of timesteps meet the criteria.

> KEY IDEA - Agent's only listen when it's beneficial to them, meaning most communications are ignored while a few are acted on.
>
> KEY IDEA - Agents that were most influential also obtained the highest environmental rewards.

3.6 - Modeling Other Agents

- How can we eliminate the central training assumption in the first two experiments?
  Add a *Model of Other Agents* (MOA) to each agent which allows them to estimate their own influence by predicting other agent's reactions.
- MOA architecture:
  - Takes output from convolution layer.
  - Feeds this into fully connected layers.
  - Then into an LSTM, along with the other agent's previous action, a_t .
  - LSTM layer outputs $p(\ a_{t+1} \mid a_t,\ s_t^k\ )$.
- MOA is trained by observing action trajectories and cross entropy loss.
- Imitates how humans reason about their effects on others.
- Algorithm:

Algorithm 3: Computing the influence reward for agent k using a trained model of other agents (MOA).

Require: τ, trajectory containing actions for all agents, but only the observations and rewards of agent k. Let T be the trajectory length.
Require: A trained MOA parameterized by θ_M^k
influence $= [0] * T$
for *timestep* t *in* $[0, T)$ **do**
  for *other agent* j *in* N **do**
    prob_aj $= 0$
    for *action* $\tilde{a}_t^k$ *in* $\mathcal{A}$ **do**
      Compute $p(\tilde{a}_{t+1}^k | \boldsymbol{a}_t, s_t^k; \theta_\pi^k)$ using agent k's policy network
      Compute $p(a_{t+1}^j | \tilde{a}_t^k, s_t^k; \theta_M^k)$ using MOA
      prob_aj $=$ prob_aj $+ p(a_{t+1}^j | \tilde{a}_t^k, s_t^k) p(\tilde{a}_t^k | \boldsymbol{a}_t, s_t^k)$
    end
    $p(a_{t+1}^j | s_t^j) \leftarrow$ prob_aj
    influence$[t]$ = influence$[t] + D_{KL}[p(a_t^j | a_t^k, s_t^k; \theta_M^k) || p(a_t^j | s_t^j)]$
  end
end

3.6.1 - Experiment III: Modeling Other Agents

- Intrinsic influence consistently improves learning.
- A3C and MOA baseline perform similarly.
- SOTA results!!!