Learning from Induced Changes in Opponent (Re)actions in Multi-Agent Games

Hoen, et al.

Presented by: James MacGlashan

Outline

- Intelligent (Multi-)Agents
- Multi-Agent games
- Improvements to be made in updating policy
- Experimental environment
- Results
- Conclusions
- Questions

Intelligent Agents

- In AI, an intelligent agent (IA) often refers to an autonomous entity that can perceive and interact with its environment (Wikipedia).
- The IA will often also have goals it wishes to achieve and will attempt to satisfy its goals through use of the available actions it can take
- IAs may make use of strategies, or policies, to define when to use any given action to arrive at their goal(s).
- Multi-Agent systems involve environments where multiple IAs exist. Each agent may have independent goals from the others, or agents may share the same goals. Various IAs may also be in competition with one another.

Multi-Agent Games

- A Multi-Agent game generally consists of:
  - defining a set of goals for agents
  - the environment the agents will exist in
  - the viable actions agents can take in each possible state
  - The reward values agents receive for being in any given state
  - The number of agents to play in the game
- An iterated game refers to a game where the same agents will repeatedly play the same game over and over
  - Agents are allowed to have any form of memory of the previous games

Playing the game

- To play the game, each agent in the game must have a strategy, or policy, for playing the game.
  - By policy we mean what action an agent chooses given the current state they are in
  - A policy is said to be pure if the chosen action is deterministic
  - A policy is said to be mixed if the chosen action is random with respect to some probability distribution

Learning Policies

- When people play games, we may start with a simple policy and then as we learn the mechanics of the game, we update our policy
  - It would be advantageous then if an IA could do the same, particularly if the programmer doesn't know the best strategy – or if there is even one
- Most IA learning frameworks are concerned with estimating what a the current policy of an opponent is, and what the expected rewards will be for actions (via Q-learning-like values)

Strategic Opponent Policy Modeling (StrOPM)

- Learning, as described earlier, ignores a critical factor – opponent policies may be changing over time and in response to an agent's own actions
- StrOPM models how an opponent IA changes its policy over time and uses that to update its policy in turn

Calculating the effectiveness of a policy

1. Linearly model the change in an opponent policy: Take current estimated opponent policy ($\pi_{-i,t}$) and their estimated policy N steps earlier and calculate $[(\pi_{-i,t} - \pi_{-i,t-N})/N] + \pi_{-i,t}$
2. Find all state loops from current state – a loop is all the state and action pairs that lead back to the current state
3. Take the probability of each loop using the probability of the agent taking that action under their policy, and estimated probability that the opponent with take the necessary actions (accounting for the estimated change in opponent policy at each step)
4. Multiply the probability of each loop by the estimated average reward for that entire loop
5. Sum above over all loops

Updating a Policy

- Consider all possible policy updates (i.e. modifying the probability of taking a certain action at a given state)
- Simulate the reward of using that updated policy via the described method
- Choose the policy update which maximizes the reward reward

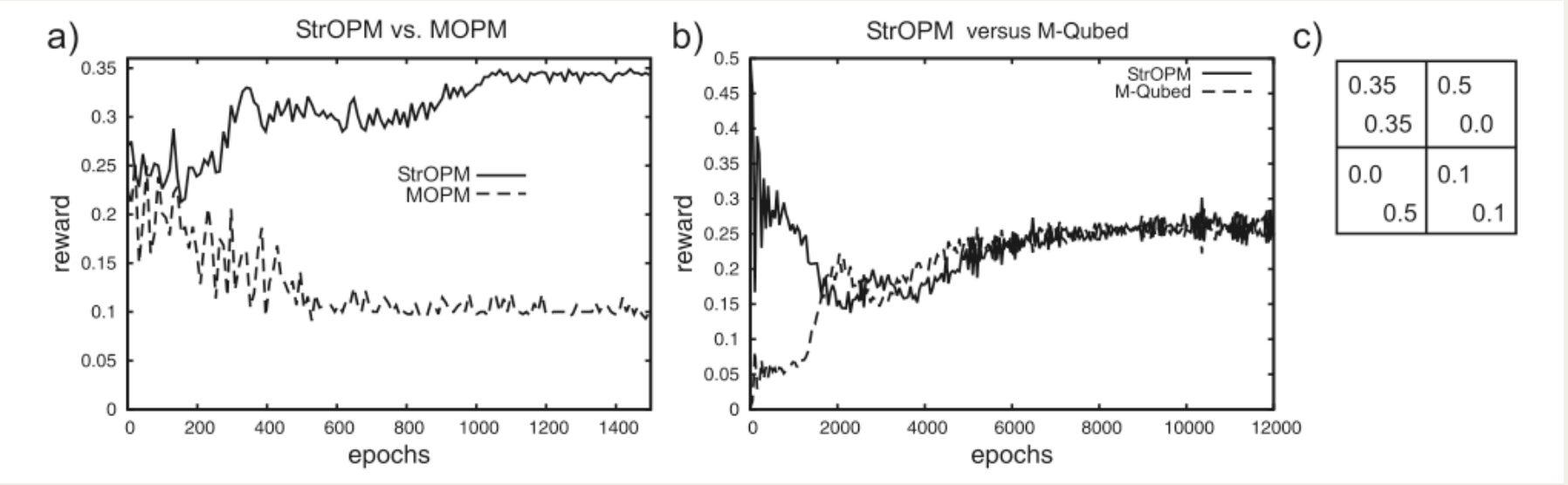
Prisoners Dilemma

- 2 player game
- Each player given the choice to Cooperate with the other player, or defect.
- Joint Actions then include {C, C}, {C, D}, {D, C}, {D,D}
- If both both cooperate, both receive a moderate reward. If both defect, both receive a lower reward. If only one defects, the defector receives a high reward and cooperator receives a very low reward
- In one shot games, the best strategy is for both to defect.
- In iterated games, it can be beneficial for both to cooperate given the ability for a player to punish the other player if they defected previously.

Results

- In StrOPM playing against another StrOPM, the joint action converged to {C,C} over iterated play
- M-Qubed is another IA framework where it mixes Q-learning-like methods with precoded policies – has been shown to be very effective in past
- StrOPM playing against M-Qubed converges slightly below the optimal outcome of {C,C} (alternates)
  - StrOPM seems to have problems predicting the policy update since M-Qubed policy changes are not always linear
- M-Qubed has other constraints since precoded strategies are not always effective
  - In a variation of the Prisoners Dilemma where the reward table is changed, StrOPM is able to effectively exploit M-Qubed

Results



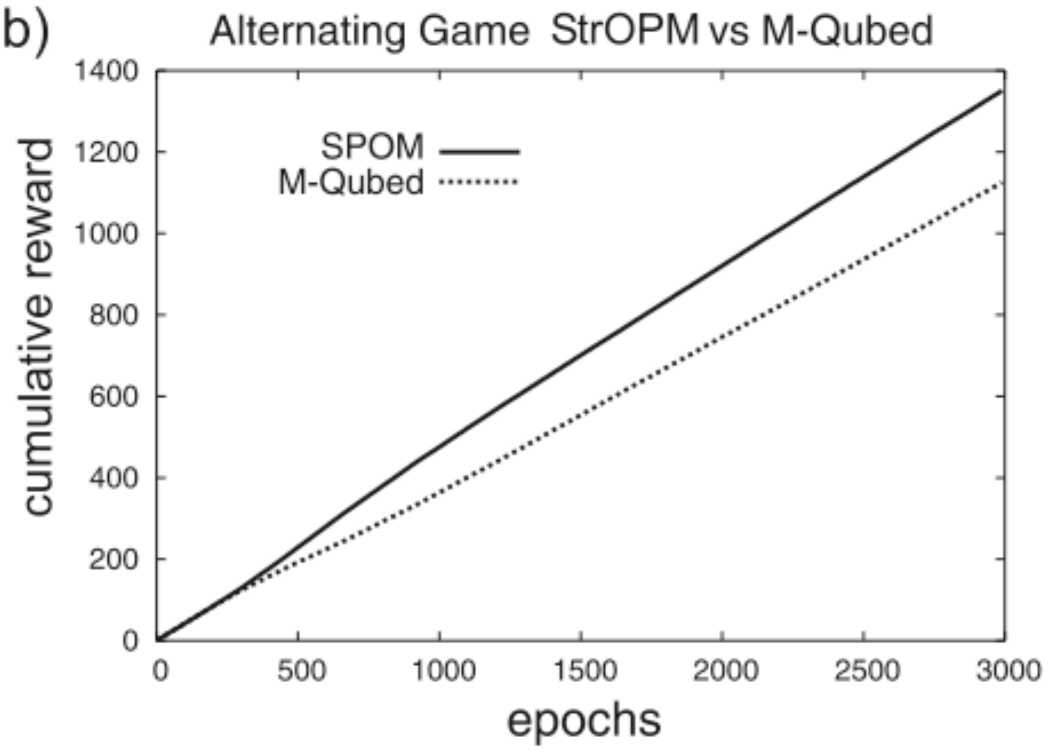
a) StrOPM playing against itself vs. StrOPM with out opponent policy modeling playing against itself
b) StrOPM playing against M-Qubed

Results cont…

a)

0.4 0.4	0.8 0.1
0.1 0.8	0.4 0.4

b)



Conclusions

- StrOPM demonstrates a better strategy for policy learning by modeling how an opponent reacts
- May have trouble with agents who do not update their policy linearly
  - Perhaps a better modeling mechanism can be used?

Questions?