Predicting the Expected Behavior of Agents that Learn About Agents: The CLRI Framework

Josè M. Vidal and Edmund H. Durfee (2003).
Autonomous Agents and Multi-Agent Systems, 6(1):77-107

A seminar presented by:
Christopher Lyons
Michael Cundall

Intro

- The CLRI Framework: used to model and predict the behavior of multiagent systems with learning agents
- CLRI:
  - C - change rate
  - L - learning rate
  - R - retention rate
  - I - Impact
- Purpose: a way to model/predict the behavior of a MAS beyond observation-based experiment results

Framework for modeling MASs

- N – the set of all agents, where i is one particular agent
- W – the set of possible states of the world, where w is one particular state
- A_i – the set of all actions that agent i can take
- A decision function for agent i – a mapping which tells us which action agent i will take in each state
- A target function for agent i – a mapping which tells us what action agent i should take. Takes into account the actions other agents will take.
- An error probability function for agent i at time t – the probability that agent i will take an incorrect action (i.e., the decision function is not the same result as the target function for an agent i at time t in state w)
- Note: an action taken by an agent that is not the action indicated by the target function is "incorrect" – there is no continuum of "correctness"

Framework for modeling MASs (continued)

- Change rate – the probability that the agent will change at least one of its incorrect mappings of a decision function at a given state *w* (i.e., the likelihood the agent will change an incorrect mapping to something else, but not necessarily into the correct mapping)
- C = .5 = 50% chance that an incorrect mapping of a decision function will be modified
- Learning rate – the probability that the agent changes an incorrect mapping to the correct action at at a given state *w* (note: different than learning rate in Q-learning; we care about whether the agent correctly changes an incorrect mapping)
- L = .5 = 50% chance that an incorrect mapping of a decision function will be changed to the correct mapping (i.e., to match the target function mapping)

Framework for modeling MASs (continued)

- Retention rate – probability that a correct mapping will stay correct in the next iteration
- R = 0 = all correct mapping will be made incorrect. R = 1 = all correct mapping will persist.
- Impact – given agent *i* and agent *j,* the impact that *i*'s changes in its decision function have on *j*'s target function
- The greater an agent *i*'s Impact on agent *j,* the greater the likelihood if *i* changes its decision function that it will change *j*'s target function
- Volatility - probability the target for learning will change from world state to world state
- V = 1 = an agent's target function will always change between iterations. V = 0 = an agent's target function will always remain the same.

Volatility and Impact

- Intuitively:
  - Volatility: odds that how an agent <u>should</u> react (the correct $w \rightarrow a$ at time t, a.k.a. the target function) will change
  - Impact: how much influence one agent's decision has on how another agent <u>should</u> act: if agent i has a high impact on agent j, and agent i's decision for $w \rightarrow a$ changes, there's a good change how agent j should act $w \rightarrow a$ will also change because of it

Calculating the Agent’s Error

- 2 conditions that determine the new error - the result of volatility and whether or not the current decision function matches the target
- This results in 4 different cases to consider
  - *a* = an agent’s target function remains the same between iterations
  - *b* = an agent’s decision function is the same as its target function
  - (1) a & b (2) a & !b (3) !a & b (4) !a & !b
- (1) agent’s target function does not change, decision function is a correct mapping $w \rightarrow a$: agent has a probability of changing this mapping with probability 1 - r (i.e., the odds that it will change a correct mapping)
- (2) agent’s target function does not change, but decision function is incorrect mapping $w \rightarrow a$: agent has a probability of changing this mapping to a correct mapping of 1 - l (i.e., the odds that the agent will be incorrect in the next iteration -- odds that agent does not change the incorrect mapping)

Calculating the Agent's Error (continued)

- (3) agent's mapping of $w \rightarrow a$ is correct, but the target function changes: probability that a correct mapping does change 1 - r (if r = 1, 0% chance it will change, agent will definitely be wrong in the next iteration)
- (4) agent's mapping of $w \rightarrow a$ is incorrect, and the target function also changes
  - 1 - agent does not change its mapping (1 - c)
  - 2 - agent changes its mapping, and changes it to the correct mapping (l)
  - 3 - agent changes its mapping, does not change it to the correct mapping (c - l)

Equation and Simplification of a Further nature

$$E[e(\delta_i^{t+1})] = E[\sum_{w \in W} \mathcal{D}(w)\mathbf{Pr}[\delta_i^{t+1}(w) \neq \Delta_i^{t+1}(w)]] = \sum_{w \in W} \mathcal{D}(w)($$

$$\mathbf{Pr}[\Delta_i^{t+1}(w) = \Delta_i^t(w) | \delta_i^t(w) = \Delta_i^t(w)] \cdot \mathbf{Pr}[\delta_i^t(w) = \Delta_i^t(w)] \cdot (1 - r_i)$$

$$+ \mathbf{Pr}[\Delta_i^{t+1}(w) = \Delta_i^t(w) | \delta_i^t(w) \neq \Delta_i^t(w)] \cdot \mathbf{Pr}[\delta_i^t(w) \neq \Delta_i^t(w)] \cdot (1 - l_i)$$

$$+ \mathbf{Pr}[\Delta_i^{t+1}(w) \neq \Delta_i^t(w) | \delta_i^t(w) = \Delta_i^t(w)]$$

$$\cdot \mathbf{Pr}[\delta_i^t(w) = \Delta_i^t(w)] \cdot (r_i + (1 - r_i) \cdot B)$$

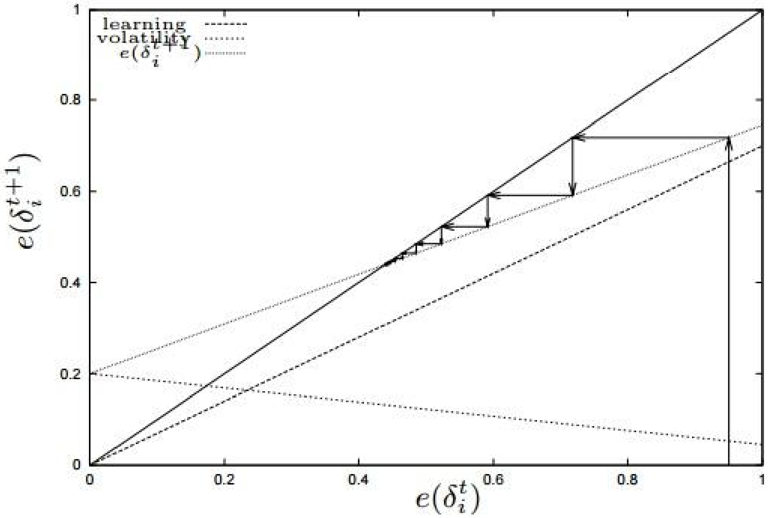
$$+ \mathbf{Pr}[\Delta_i^{t+1}(w) \neq \Delta_i^t(w) | \delta_i^t(w) \neq \Delta_i^t(w)] \cdot \mathbf{Pr}[\delta_i^t(w) \neq \Delta_i^t(w)]$$

$$\cdot (1 - c_i)D + l_i + (c_i - l_i)F.$$

Equation and Simplification of a Further nature

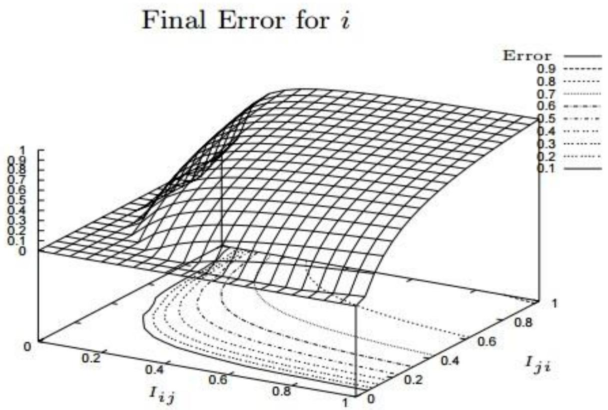
- 2 assumptions can be made to simplify the equation above
  - that actions chosen when either mapping changes are chosen from a flat probability distribution of actions
  - the probability that the target changes is irrelevant to the probability that an agent's mapping is correct (this was not the case in the matching game)

$$E[e(\delta_i^{t+1})] = 1 - r_i + v_i\left(\frac{|A_i|r_i - 1}{|A_i| - 1}\right) + e(\delta_i^t)\left(r_i - l_i + v_i\left(\frac{|A_i|(l_i - r_i) + l_i - c_i}{|A_i| - 1}\right)\right)$$



An Example

- Example: l = .2; c = 1; r = 1; 20 actions; values identical for i and j
- As impact increases, the emergent error does as well
- Abrupt change from final errors of 0 and non-zero - this aspect is a result of systems that converge or diverge
- Graph is asymmetric!
- Agent j's error makes it harder for i to come to a smaller emergent error



An Example

- If the impact of i is high, then when i's error is large, j's will increase and j will start changing its decision function
- If the impact of i is low, then regardless of j's impact, both agents will probably settle to a low emergent error
- Goal is to hit outer region, where error is 0

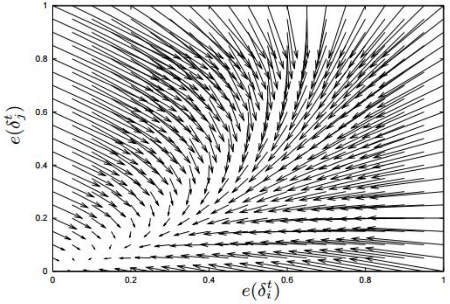
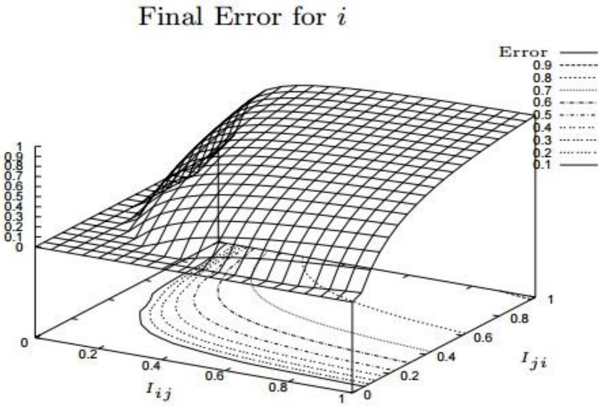
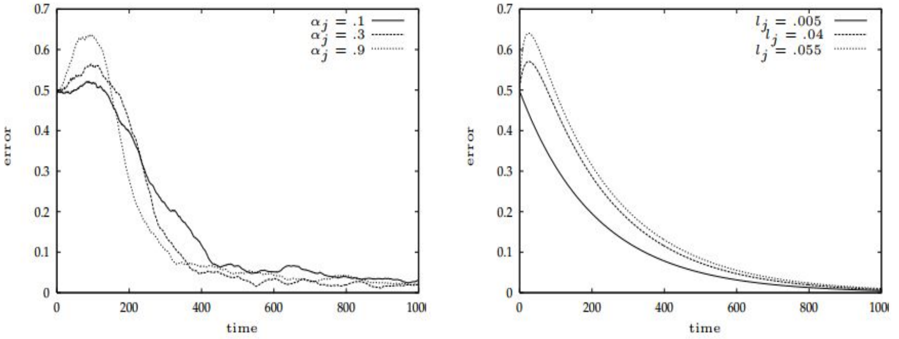


Figure 8: Vector plot for $e(\delta_i^t)$ and $e(\delta_j^t)$, where $|A_i| = |A_j| = 20$, $l_i = l_j = .2$, $r_i = r_j = 1$, $c_i = .5$, $c_j = 1$, $I_{ij} = .1$, $I_{ji} = .3$. It shows the error progression for a pair agents i and j. For each pair of errors $(e(\delta_i^t), e(\delta_j^t))$, the arrows indicate the expected $(e(\delta_i^{t+1}), e(\delta_j^{t+1}))$.



An Application

- Market based MAS - 3 agents: 1 buyer, 2 sellers
- Reinforcement: profit earned each round; 20 possible bid values
- Vary j’s alpha value (the weight the algorithm assigns to the latest result)
- i’s error rate shown below
- slight delay before decrease - due to high exploration rate

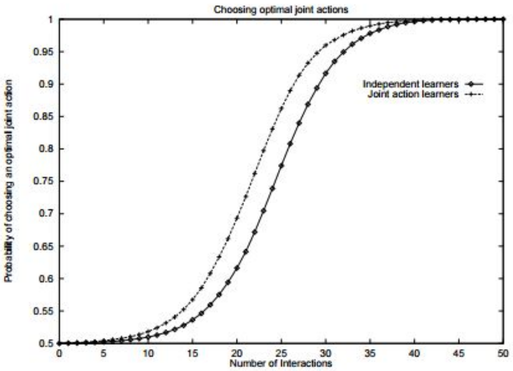


(a) Experiment

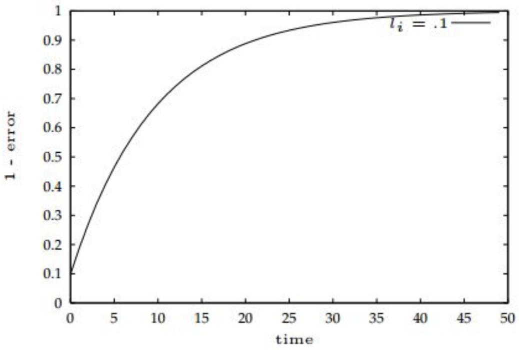
(b) Theory

Applications of the Theory

- A comparison of CLRI framework application to experimental results from the matching game explored by Claus and Boutilier
- CLRI cannot account for initial exploration yet again



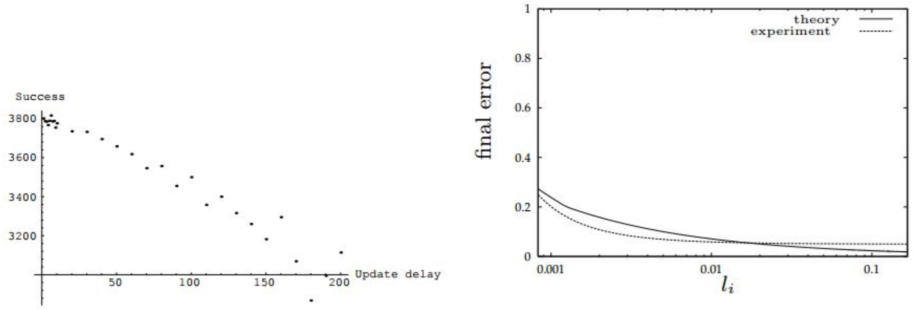
(a) Experiment



(b) Theory

Applications of the Theory

- Shoham and Tennenholtz investigate an experiment similar to the matching game known as the coordination game; all agents use their custom algorithm: HCR (highest cumulative reward)
- at every t, each agent takes an action, the agents are paired up randomly with the goal of taking the same action
- the agents take their action after a delay, this delay is varied in the experiment



(a) Original Experiment (b) Theory and Experiment

Bounding the Learning rate with Sample Complexity

- Previous examples assume *l, c, r* can be found based on an agent's learning algorithm
- What if the algorithm is highly complex or unknown?
- Probably Approximately Correct (PAC) theory: loose upper bound on number of examples an agent must observe before arriving at a PAC hypothesis
- Determine *l* and then find *c* and *r*
- With assumption that $r = 1$ and $v = 0$
- Simply put: can find the lower bound of an agent's learning rate (*l*), and can be used to determine its *c* and *r*

Conclusions

- The CLRI framework can get fairly limited as the constraints imposed on the agents are critical for its function; additionally, there is no room for non-binary degrees of success, an agent is correct or incorrect
- The world states the framework must act upon must be chosen from a uniform probability distribution and must be episodic. There isn’t much room for chaotic, shifting environments
- The description given by the CLRI is very high-level behavior, meaning it doesn’t work well when describing specific systems, but works well in predicting agents emergent behaviors