Unified Learning Model in Multiagent System

Vlad Chiriacescu, Derrick Lam, Ziyang Lin

Background

- Multiagent systems are environments in which multiple intelligent, autonomous agents may compete against one another to further their own interests or cooperate to solve problems that are too difficult for a single agent to solve alone
- The Unified Learning Model (ULM) developed by Shell et al. (2010) is a recent model that presents the learning process in a new light, combining various educational, psychological, and neurobiological studies and theories into a comprehensive view of human learning
- The goal of the project is to design and implement an adaptation of the ULM for multiagent systems

Overview

- **ULM Adaptation**
- Teacher-Learner Interaction
- Simulation Scenario
- Agent Design Strategy
- Experiments

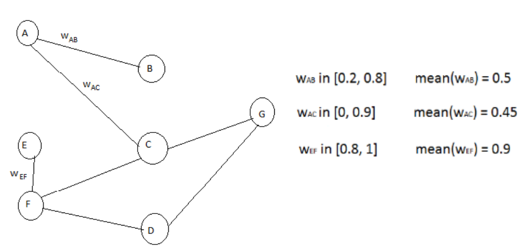
ULM Adaptation

- Three primary components
  - Knowledge, which is the "crystallized form" of information in long-term memory
  - Motivation, which reflects the agent's interest or attention
  - Working memory, which receives and processes sensory information
- Other components
  - Knowledge Decay
  - Relations to rules of learning

Knowledge

- Use a weighted graph to represent an agent's knowledge and working memory. Concepts are represented using the nodes in the graph. An edge between two concepts indicates the two concepts are considered related to each other by the agent (given its current knowledge). Edge weights indicate how related the two concepts are.
- Weights are represented by the confusion interval, the mean of this interval and the actual random weight value. Confusion interval describes how much can an agent's understanding of the current ideal weight knowledge oscillate around that perceived ideal information (the mean value). The actual random weight value represents the current understanding about the ideal knowledge weight.

Knowledge



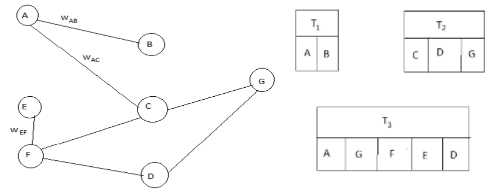
Example of knowledge topology and weights features in a given time step

Motivation

- Employs the notion of motivational score to model the concept of motivation.
- Motivational score is a function of the underlying confusion intervals of the connections related to that concept and also of all the rewards that involve that concept:
- $m\text{-}score_X = \sum_{Y \in SC_X} \frac{1}{(range(w_{XY}))} \cdot \sum_{k \in T_X} (R_k)$
- Where:
- $range(w_{XY})$ is the length of the confusion interval for weight w_{XY}
- SC_X is the set of concepts connected to concept X
- T_X is the set of tasks that require concept X
- R_k is the reward for task k, where k $\in T_X$

Motivation

$$m\text{-}score_A = \left(\frac{1}{range(w_{AC})} + \frac{1}{range(w_{AB})}\right) \cdot (R_{T_1} + R_{T_3})$$



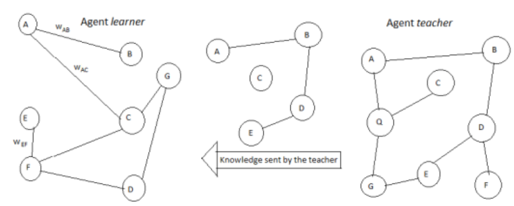
Example of computing the motivational score

Working Memory

- Working memory storage is represented in the same manner as the knowledge but it has a storage limit as opposed to the knowledge.
- Only those concepts whose motivational score is above a preset threshold can get into the working memory.
- This threshold traces a boundary between concepts that were taught in a time step but the agent is not aware of (for concepts with scores that are below the threshold) and those concepts that the agent is aware of (scores higher than threshold)
- The threshold is called awareness threshold (*AT*) and it is randomly selected from a uniform distribution
  - Different agents have different AT
  - AT for each concept is the same

Working Memory

- An agent can learn new concepts or update the weights' confusion interval among existing concepts by learning from a teacher agent.



Working Memory

- Example of working memory allocation:
  - $m\text{-}score_A = 20 \geq AT = 10$
  - $m\text{-}score_B = 11 \geq AT = 10$
  - $m\text{-}score_C = 2 < AT = 10$
  - $m\text{-}score_D = 9 < AT = 10$
  - $m\text{-}score_E = 16 \geq AT = 10$
- In this case, working memory is allocated for concepts A, B and E
- In the process of teaching and learning, this allocation is first done by the teacher using its own knowledge and motivational scores; the teacher obtains a knowledge sub-graph that is sent to the learner; the learner applies the same allocation principle only on the concepts that are part of the communicated knowledge sub-graph.

Working Memory

- The learner then updates its knowledge based upon the contents of its working memory. This consists of updating the mean weight and confusion interval length for each of the edges involved.
- For the update of the edge weight's mean value, we alter the learner's current mean by some portion of the mean provided by the teacher:
- $w_{XY}^{learner(k+1)} = \frac{q \cdot (m\text{-}score_X + mscore_Y) \cdot w_{XY}^{teacher(k+1)} + w_{XY}^{learner(k)}}{q \cdot (m\text{-}score_X + m\text{-}score_Y) + 1}$
- Some terms may drop depending on 1) whether the connection already exists for the learner and 2) whether one concept didn't enter working memory

Working Memory

- $w_{XY}^{learner(k+1)} = \frac{q \cdot (m\text{-}score_X + mscore_Y) \cdot w_{XY}^{teacher(k+1)} + w_{XY}^{learner(k)}}{q \cdot (m\text{-}score_X + m\text{-}score_Y) + 1}$

- Where:
- $w_{XY}^{learner(k+1)}$ and $w_{XY}^{teacher(k+1)}$ are the learner's mean weight and the mean weight communicated by the teacher for the edge connecting concepts *X* and *Y* at time step $k+1$
- $w_{XY}^{learner(k)}$ is the learner's mean weight for edge *XY* at time step k
- $mscore_X$ and $mscore_Y$ are the motivational score for concepts *X* at time step $k+1$
- q is a learning factor that affects how far the learner's mean is "moved"

Working Memory

- The learner's confusion interval is updated in one of two ways depending on whether the corresponding edge in the learner's knowledge is already activated.

- If the edge already exists in the learner's knowledge, update the current confusion interval based on its motivation:
- $range\left(w_{XY}^{learner(k+1)}\right) = range\left(w_{XY}^{learner(k)}\right) - 2 \cdot \Delta range\left(w_{XY}^{learner(k+1)}\right)$

- Where:
- $range\left(w_{XY}^{learner(k+1)}\right)$ and $range\left(w_{XY}^{learner(k)}\right)$ are the confusion interval lengths for w_{XY} at time *k+1* and *k*, respectively
- $2 \cdot \Delta range\left(w_{XY}^{learner(k+1)}\right)$ is the amount that the confusion interval is to be shortened

Working Memory

- We define the $\Delta range$ function in the previous formula as:
- $\Delta range\left(w_{XY}^{learner(k+1)}\right) = k \cdot [(m\text{-}score_X - AT) + (m\text{-}score_Y - AT)]$

- Where:
- $\Delta range\left(w_{XY}^{learner(k+1)}\right)$ is the amount that (one side of) the confusion interval is updated by
- *k* is the learning rate for confusion interval shrinking
- $m\text{-}score_X$ and $m\text{-}score_Y$ are the motivational scores for concepts *X* and *Y*
- *AT* is the learner's awareness threshold

Working Memory

- If the connection between the concepts is not yet made for the learner, the initial confusion interval is based on the confusion intervals of other existing edges connected to the concepts:

- $range\left(w_{XY}^{learner(k+1)}\right) = \frac{\sum_{i \in SC_X} range(w_{Xi}^{learner(N)}) + \sum_{j \in SC_Y} range(w_{Yj}^{learner(N)})}{n(SC_X) + n(SC_Y)}$

- Where:
- SC_X and SC_Y are the sets of concepts connected to concepts *X* and *Y*, respectively
- $n(SC_X)$ and $n(SC_Y)$ are the cardinalities of the sets of concepts connected to concepts *X* and *Y*, respectively

Knowledge Decay

- An agent's unused knowledge on a concept diminishes over time by increasing the confusion intervals of weights related to that concept.
- If the concept that experiences weight decay enters working memory, the knowledge decay is stopped
- Each weight is updated during decay period using a widely known exponential based formula $P = P_0 \cdot e^{rt}$:
- $range\left(w_{XY}^{learner(t)}\right) = \begin{cases} range\left(w_{XY}^{learner(t-1)}\right) \cdot e^{k_{decay}}, & Z < c_{disuse} < 3Z \\ range\left(w_{XY}^{learner(t-1)}\right), & c_{disuse} < Z \text{ or } c_{disuse} > 3Z \end{cases}$
- k_{decay} is an experimental decay rate
- Z is the number of steps between last concept usage and start of decay period
- t is the current time step
- c_{disuse} is the current number of time steps that the knowledge has been unused

Relation to Rules of Learning

- **Learning Requires Attention**
- Reflected by use of motivational scores and awareness thresholds. Concepts that enter WM are those that the agent pays attention to, and therefore the connections related to those concepts are the ones altered by the learning process.
- **Learning Requires Repetition**
- Represented by confusion interval update process. An agent's knowledge is partially altered in each time step, agents need to learn repeatedly to tighten the confusion interval for the weights connecting concepts. This is how knowledge is refined.
- **Learning is about Connections**
- Reflected in our representation of an agent's knowledge. The knowledge that an agent has is represented with a weighted graph. Knowledge concepts are represented by nodes, and the connections between concepts are represented by weighted edges connecting those nodes.

Overview

- ULM Adaptation
- **Teacher-Learner Interaction**
- Simulation Scenario
- Agent Design Strategy
- Experiments

Teacher-Learner Interaction

- Teacher teaches (i.e. communicates its subgraph to) the Learner
- Learner informs the Teacher of the amount of knowledge learned during the same time step
- Teacher receives an *immediate* reward based on amount learned (i.e. change in confusion interval) and the Learner's current task

Teacher-Learner Interaction

- Change in learner's knowledge
  - $\Delta K^{learner(t)} = \sum_{e \in \boldsymbol{E}_{WM,t}} \Delta range(w_e^{learner(t)})$
  - $\boldsymbol{E}_{WM,t}$ is the set of edges connecting the concepts that entered working memory at time t
  - $\Delta range(w_e^{learner(t)})$ is the change in the confusion interval for the weight of edge e
- Teaching reward
  - Based on change in learner's knowledge and the learner's current task
  - $R_{teacher} = R_{T_{Learner}} \cdot \Delta K^{learner(t)}$

Overview

- ULM Adaptation
- Teacher-Learner Interaction
- **Simulation Scenario**
- Agent Design Strategy
- Experiments

Simulation Scenario

- A set of agents implementing ULM framework with different initial knowledge topologies
- An agent can either learn or teach in a time step
- After the agent action is determined, the agents determine the task to solve (goal task) if they do not already have one
- At the end of a step, the agents check to see whether they can solve their goal task
- If they fail to solve it for a number of steps, they abandon the task
- All tasks are available to all agents at all times, except tasks that were either solved or abandoned
- An agent that solved or abandoned a task cannot tackle attempt same task again, but agents that didn't encounter it before can choose it

Task Representation

- Task are represented with weighted graph:
  - *Nodes* represent knowledge concepts
  - *Edges* represent the connections between them
  - Edge weights specify the required knowledge to complete it
  - Does not have a "confusion interval" for its edge weights
- Reward
  - Multiplicatively increases reward with greater requirements, in terms of number of concepts and connections required
  - $R_T = R_{base} \cdot 1.1^{NumConcepts(T)-2} \cdot 1.05^{NumEdges(T)-1}$
  - R_{base} is the base reward for the smallest

Task Pool

- Fixed number of tasks
- Randomly generated at the start of the simulation
- Task generation process
  - Randomly select a set of the edges to use for the required knowledge connections
  - Randomly generate a random number in the interval [0, 1] (uniform distribution) for each edge to use as the edge weight
  - Calculate the task reward based on the number of concepts the task involves

Overview

- ULM Adaptation
- Teacher-Learner Interaction
- Simulation Scenario
- **Agent Design Strategy**
- Experiments

Agent Design Strategy

- Teacher-Learner Matching
- Task Selection and Performance
- Determining an Action
- Desired Emergent Behavior
- Hypotheses

Teacher-Learner Matching

- Teaching and learning in this simulation is done in a one-on-one fashion (i.e., tutoring)
- Teaching-Learner matching is made in a first come first served approach
  - Implementation as two queues (one for teachers and one for learners)
  - At each time step, each agent will decide to be either a teacher or a learner in a random order
  - Those who choose to teach are put in the teacher queue and those who choose learn will be in the learner queue
  - Agents are paired based on queue position

Teaching and Learning

- Learners
  - Adjusts mean and confusion interval size from learning as previously described
  - Gets no immediate rewards from learning but it gets the knowledge that could enable it to solve tasks in the future
  - Extra learners will miss the opportunity to learn and will idle during the time steps when they have no matched teachers
- Teachers
  - In time steps when it teaches, an agent does not shift its mean
  - However, the teacher reinforces its knowledge about taught concepts by decreasing the associated confusion intervals
  - It also receives rewards from teaching
  - Extra teachers will miss the opportunity to teach others and will not get a reward

Task Selection and Performance

- Each agent chooses a target task randomly from tasks not yet solved or abandoned
- Completing a task
  - Agent instantiates values for each of the weights associated with the requisite edges
    - Taken from within the edge's confusion interval
    - Uniform distribution
  - Compute the distance between the weights in task description and instantiated weights
  - If the distance for every edge is below a fixed upper bound k_{dist}, then the task is considered solved by that agent
- The agent will stop if there are no more tasks in the task pool

Determining an Action

- At each time step, the agent will probabilistically determine whether to teach or to learn
- In the initial setup, the probabilities of teaching or learning are determined randomly for each agent
- Probabilities are adjusted by a constant amount p_{update} based on the action at previous step, in favor of the action not taken
- Adjustment for time step *t* if the agent *learned* in time step t-1:
  - $P_t(teach) = P_{t-1}(teach) + p_{update}$
  - $P_t(learn) = P_{t-1}(learn) - p_{update}$
- Adjustment for time step *t* if the agent *taught* in time step t-1:
  - $P_t(learn) = P_{t-1}(learn) + p_{update}$
  - $P_t(teach) = P_{t-1}(teach) - p_{update}$

Desired Emergent Behavior

The desired emergent behavior is for the agents to maximize the cumulative reward of the community at the end of the simulation through teaching and learning. Confusion intervals will be smaller overall.

Hypotheses

- Reducing working memory size for all agents will result in reduced cumulative community reward. Similarly, increasing working memory size will result in increased cumulative community reward.
  - Learning about a variety of concepts, e.g. forming, shifting, and tightening confusion intervals, will be inhibited with a reduced working memory since fewer simultaneous connections can be made in one tick
  - With a reduced WM, more ticks must be spent to acquire the connections that a larger WM can create within a smaller time frame.

Overview

- ULM Adaptation
- Teacher-Learner Interaction
- Simulation Scenario
- Agent Design Strategy
- **Experiments**

Experiments

ENVIRONMENT GROUP PARAMETERS	RANGE OF VALUES
Number of Agents $\mathbf{N_A}$	5, 10, 20
Number of Time Steps	25000 ticks
Number of Concepts	5, 10, 20
Base Task Reward $\mathbf{R_{base}}$ (for task with lowest expertise requirement using fewest number of concepts)	1, 2, 5
Number of Tasks $\mathbf{N_T}$	100
Maximum Allowed Distance for Solving Tasks k_{dist}	0.05

Experiments

AGENT PARAMETERS	RANGE OF VALUES
Attention Threshold (AT)	Random integer in (0, 20], uniform distribution
Initial Confusion Interval Bounds	Random double in [0, 1], uniform distribution
Knowledge Decay Rate $\mathbf{k_{decay}}$	0.1
Teaching/Learning Probability Update Constant $\mathbf{p_{update}}$	0.05
Initial Teaching / Learning Probability	Random double in [0, 1], uniform distribution

References

- Shell D. F. et al. (2010) – The Unified Learning Model: How Motivational, Cognitive, and Neurobiological Sciences Inform Best Teaching Practices, Springer