

PhD proposal: Heterogeneous open agent MRS decision making by contextual action embedding - ONLY Problem Definition

Mohammad Rahmani

1 Terminology, problem definition and the objective

An open society of n heterogeneous, Intelligent Agents (AI) such as

$$I = \{i_1, \dots, i_n\} \quad (1)$$

function in an environment such that each $i_k \in I$ can perform the following $|A_k|$ actions

$$A_k = \{a_{k1}, \dots, a_{k|A_k|}\} \quad (2)$$

IA actions' utilities and costs

Performing each action such as a_{pq} for t amount of time results in a utility and a cost referred as $U_{pq}(t) \in \mathbb{R}$ and $C_{pq}(t) \in \mathbb{R}$ for all $i_p \in I$ and $a_{pq} \in A_p$.

Agent and environment statuses

Each $i_k \in I$ can adopt $|S_k|$ statuses:

$$S_k = \{s_{k1}, \dots, s_{k|S_k|}\} \quad (3)$$

As such, the available statuses in the environment will be Cartesian product of the agents in the system.

$$S = \prod_{p=1}^n S_p \quad (4)$$

1.1 Overall mission

The overall mission is to choose a **sequence of actions** from $\bigcup_{p=1}^n A_p$ such that the efficiency as formulated in Equation 8 maximizes in t amount of time under the constraints described in Section 1.2.

1.2 Constraints

1. At any time a new IA can be added to the system
2. Not only utility and cost functions for each BDI can be either negatively or positively correlated, but also, also such types of correlations may also established inter-agents.
3. No agent, can have access to implementation of U and C for any actions of another agent.
4. No agent knows about the actions that any other agents can perform.
5. No agent knows about the statuses that any other agent can adopt.
6. No agent know in which status is any other agent.

7. The means of communication is by composing a set of limited symbols such as Equation 5

$$W = \{w_1, \dots, w_{|w|}\} \quad (5)$$

to use them to compose messages as presented in Equation 6

$$m_p = (w_p)_{p \in \{1, \dots, |W|\}} \quad (6)$$

and publish them in a communication channel described in Equation 7

$$C = ((m_p)_q)_{q=1}^{|C|} \quad (7)$$

8. Each IA can have a limited storage space to store the data which builds its belief about other agents.

$$e = \sum_{q \in I} \sum_{p \in A_p} U_{i_{qp}}(t) + C_{i_{qp}}(t) \quad (8)$$

1.3 Example

For example, in an imaginary building composed of two venues, a room and a yard, with an automatic door (AD), an air conditioner (AC) and a vacuum cleaner (VC), each IA can acquire a set of statuses which are mentioned in the following paragraphs.

Statuses For example if the AC can be either on or off, that is:

$$S_{AC} = \{"on", "off"\} \quad (9)$$

, the VC can be either in the room or the yard which means:

$$S_{VC} = \{"room", "yard"\} \quad (10)$$

and the AD can be either open or close, in other words:

$$S_{AD} = \{"open", "close"\} \quad (11)$$

As such, the set of possible statuses in the environment is formed of 3-tuples defined as follows:

$$\begin{aligned} S &= S_{AC} \times S_{VC} \times S_{AD} \\ &= \{(x, y, z) | x \in S_{AC}, y \in S_{VC}, \\ &\quad z \in S_{AD}\} \end{aligned} \quad (12)$$

For example, ("off", "room", "open") $\in S$ which denotes the AC is off, the VC is in the room and the AD is open.

1.4 Actions and their resulting utilities and costs

The actions each of the three IAs can perform are as follows:

AC actions, utilities and costs: The AC can automatically turn itself on("ton") and off("tof")

$$A_{AC} = \{"ton", "tof"\} \quad (13)$$

the utility of "ton" action could be formalized as:

$$U_{AC_{ton}} = \frac{t}{|\bar{x} - n|} \quad (14)$$

where

- t is the amount of time that \bar{x} (the average temperature derived from AC's sensor) is measured.
- \bar{x} is the recorded temperatures' average during t
- n is the predefined preferred temperature the environment should have

as such the less the average temperature deviates from the preferred temperature the higher the utility will be and vise-versa.

$$C_{AC_{ton}} = -xt \quad (15)$$

where

- t is the amount of time the AC was on
- x is the amount of energy AC consumes per time unit

the minus before the right side of Equation 15 represents the negative nature of turning the AC on from energy consumption perspective. Similarly

$$U_{AC_{tof}} = -C_{AC_{ton}} \quad (16)$$

$$C_{AC_{tof}} = -U_{AC_{ton}} \quad (17)$$

AD actions, utilities and costs: AD actions' list includes opening (opn) and closing (cls) the door which could be shown as:

$$A_{AD} = \{"opn", "cls"\} \quad (18)$$

Explicitly, the AD's actions are just costly without any utility (implicitly it can help other agents to increase their utility or decrease their cost). As such,

$$C_{AD_{opn/cls}} = -xrt \quad (19)$$

where

- x number of times the door opens or closes in t amount of time
- r is energy rate which is consumed in each opening or closing
- t is amount of time in which the system is being studied

VC actions, utilities and costs: The VC can change the venue ("chv") from yard to the room and vice-versa. In other words:

$$A_{VC} = \{"chv"\} \quad (20)$$

$$U_{VC_{chv}} = \frac{x}{t} \quad (21)$$

in which x stands for cleanses rate achieved by VC at time unit (For example, it can be measured by a ceiling camera) in t amount of time. "chv" cost can be presented as:

$$C_{VC_{chv}} = -xt \quad (22)$$

in which x is the average electricity consumption and t is the amount of time the consumption rate was measured.

Overall mission The overall mission is to choose a sequence of actions from the aforementioned IA actions to maximize e in Equation 23 in t amount of time.

$$e(t) = U_{AC_{tof}} + C_{AC_{ton}} + U_{AC_{tof}} + C_{AC_{tof}} + C_{AD_{opn}} + C_{AD_{cls}} + U_{VC_{chv}} + C_{VC_{chv}} \quad (23)$$

while restrictions in Section 1.2 are being observed. That is, no IA knows about implementation detail of other IAs cost and utility functions etc and the only way to gain information is referring to the messages published in public channel in t amount of time.

Implicit contributing factor Additionally, considering the fact that the only three contributing indexes in described utilities and costs are **energy consumption, keeping the temperature the closest possible to n**

and **cleanness**, then the overall mission is to **keep the ultimate energy consumption at lowest possible rate** while the **temperature is kept at most possible balance** as **cleanness is kept at highest rate** during t amount of time.

2 Sample solution

Building the training set : To build the training set each member such as $m_p' \in C'$ will be composed of $m_p \in C$ such that a contributing symbols of m_p is replaced by another. Then members in C will be corresponded with 1 and members in C' will be corresponded with 0 to build a training set for classes **1/true/meaningful** and **0/false/unmeaningful** to feed the neural network that its architecture is described in the next section.

2.1 Architecture of a very simple, exemplary neural network

A shallow neural network with an input, a hidden and an output layer will be used. The output layer will represent the two classes, 1 and 0, representing meaningful and unmeaningful messages. The input layer will have $|W|$ nodes and will be fed once with hot-encoded (Equation 24) members of C as training samples of class 1 and another time with hot-encoded members of C' as training samples of class 0. That is, if w_k has been used in a message (either from C or C') for $n_{m_p}(w_k)$ times, at k th component of the input vector $n_{m_p}(w_k)$ will be placed there and if it is not used in that message then 0 will be placed (Figure 1). As

such model v in Equation 25 is trained such that it maps every message to a 2-D plane.

$$h(m_p) = (n_{m_p}(w_1), \dots, n_{m_p}(w_{|W|})) \quad (24)$$

$$v : \mathbb{R}^{|w|} \rightarrow \mathbb{R}^2$$

$$v(h(m_p)) = (x, y) \quad (25)$$

Generating vectors for symbols

To generate semantic vectors from symbols, for each $w_i \in W$ the network will be fed with another hot-encoded vector in which the i th position is filled with 1 and the rest are filled by 0. v in Equation 25 transforms the resulting vector to a point in \mathbb{R}^2 . As such a set of points in \mathbb{R}^2 will be generated as follows:

$$\Psi = \{\vec{w}_1, \dots, \vec{w}_{|w|}\} \quad (26)$$

2.2 Triggering a decision

Random message If an agent publishes a (random) message such as m_p (such as "big car") in C then $v(h(m_p))$ will map it to a vector such as $\vec{m}_p = (x_{m_p}, y_{m_p})$.

Initial condition From here, if \vec{m}_p doesn't decrease the **distance correlation** of the union of $\{\vec{m}_p\}$ with Ψ (Equation 27) then the IA must modify or change \vec{m}_p as long as the condition in Equation 27 establishes.

$$dCorr(\{\vec{m}_p\} \cup \Psi) \leq dCorr(\Psi) \quad (27)$$

Lets say $\vec{w}_i \in \Psi$ has the minimum distance of r° with \vec{m}_p (Equation 28). The minimum is chosen to study the solution under the worst condition such that, the intersection of neighborhoods of elements in $\Psi \cup \vec{m}_p$ of a radius

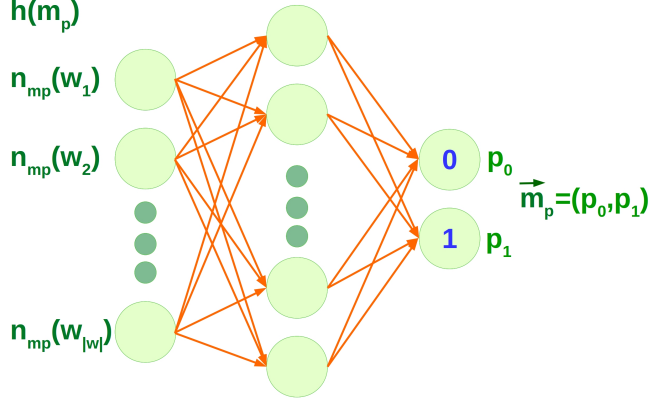


Figure 1: The architecture of a shallow neural network to map messages in C and C' to a 2-D plane

a bit greater than the half of r° also includes \vec{w}_i in the resulting set (Equation 29).

$$r^\circ = \underset{r_x \in \mathbb{R}}{\operatorname{argmin}} \{ |\vec{m}_p - \vec{w}_i| < r_x \mid \vec{w}_i \in \Psi \} \quad (28)$$

$$r = \frac{r^\circ}{2} + \epsilon^\circ \quad (29)$$

2.3 Making a decision

Extraction of greatest continuous surface : If we extract the greatest continuous set of aforementioned neighborhoods, then we have an approximate set of symbols that could be either directly or indirectly translated into a set of actions and their particularities in each agent (See Section 3).

New Cost/Utility functions If an IA publishes a message such as m_p in C which triggers $n(m_p)$ symbols in Ψ then the reward would be as of Equation 30

$$U_{\vec{m}_p} = \frac{n(m_p)}{|W|} \quad (30)$$

and the cost is as presented in Equation 31

$$C_{\vec{m}_p} = \frac{1}{|W|} \quad (31)$$

From here, the goal of each agent is to maximize Equation 32 as follows:

$$e_C = \sum_{m_p \in C} U_{\vec{m}_p} + C_{\vec{m}_p} \quad (32)$$

Implicitly, this implies that if m_p and m_q are two distinct messages, the one that includes more symbols of Ψ in Equation 26 by taking an approach in Section 2.2, is a better message, since it leads the whole system to a better equilibrium. As such these kind of messages should be rewarded more (As presented in Equation 30).

2.4 Example

If we form a vector semantic (Figure 2) around the messages formed in t amount of time using the solution discussed in Section 2, then by taking advantage of the following set of messages which have been published in C according to their sensory data we can

form a semantic vector similar to Ψ in Equation 26. Of course sensory data are slow. It takes some time that the temperature sensor understands that the temperature has varied a lot from preferred tolerance so it turns the AC on. But we look for a better or at least supplementary solution. Additionally these set of messages have been kept repeating in a logical order for a certain amount of time. Logical means the decisions were made by sensory data even though they were slow. But they were reliable decision. We want to speed up decision making.

VC published messages in C

- VC wants to clean the yard
- VC wants to clean the room
- VC is turning itself on
- VC is turning itself off
- VC wants to change location

Instructions

- VC should keep the venues clean such that the cleanliness rate doesn't fall below 70%

AD published messages in C

Messages:

- AD opens the door between the room and the yard
- AD closes the door between the room and the yard
- AD turns itself on for closing the door
- AD turns itself on for opening the door

Instructions

- Whenever someone wants to change venue, open the door.

AC published messages in C

- AC turns itself on to decrease the temperature of the room
- AC turns itself off to decrease the temperature of the room
- AC turns itself off to increase the temperature of the room
- AC turns itself off to decrease energy consumption of the room

Instructions

- AC should turn itself on whenever the temperature surpasses 27°
- AC should turn itself off whenever the temperature decreases from 27°

Two different messages for comparison:

As in Figure 3 if VC publishes a message such as $m_q = \text{"VC wants to clean the yard"}$ and another message such as $m_p = \text{"VC wants to change venue to clean the yard"}$, then m_p is a better message since it can join more symbols (5 symbols "AC", "on", "chv", "opn", "AD"). These set of symbols may result in inducing the decisions in AC to turn itself on (by observing triggered symbols "AC" and "on") ahead of the implication of its temperature sensor data and turns the AD on and open it.

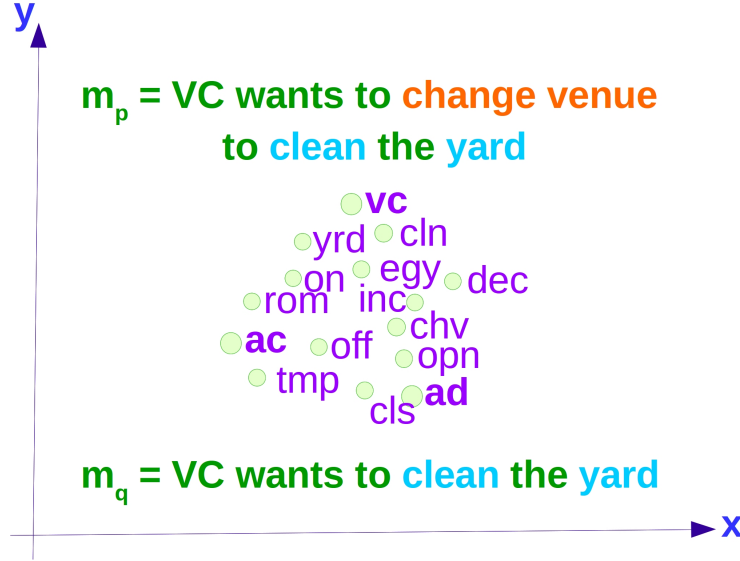


Figure 2: Vector Semantic of Symbols built around messages published in C or to build the desire or belief

Why triggering more messages cause more equilibrium

- if only AC and on were triggered, then the AC may turn itself on and consume energy, but the door never opens and temperature falls without reason.
- if the AD opens but the AC's on action is not triggered, then the temperature drops but AD haven't had understood until the sensor observes it
- and even worse, as in this example if just "cln" and "yrd" are triggered, then the VC starts cleaning without getting access to the room

As such a message that triggers more \vec{w}_i seems to be better. But this is a sub question of this proposal itself.

3 Desire

Building a semantic vector space by the symbols in Equation 5 using the instructions given to each IA such as $i_j \in I$ which is shown by Ψ_{D_j} in this document, helps with training a mapping model from triggered subsets of Ψ to actual sequence of actions(decisions). As such, if a message such as m_p is published in C (See Equation 33).

$$d : \mathbb{P}(\Psi) \rightarrow \mathbb{P}(\Psi_{D_j}) \rightarrow (a_k \in A'_j)_{A'_j \subset A_j} \quad (33)$$

where \mathbb{P} denotes power set.

4 Belief

Agent $i_k \in I$'s belief is formed upon the two models which should be trained or estimated which do the following mappings:

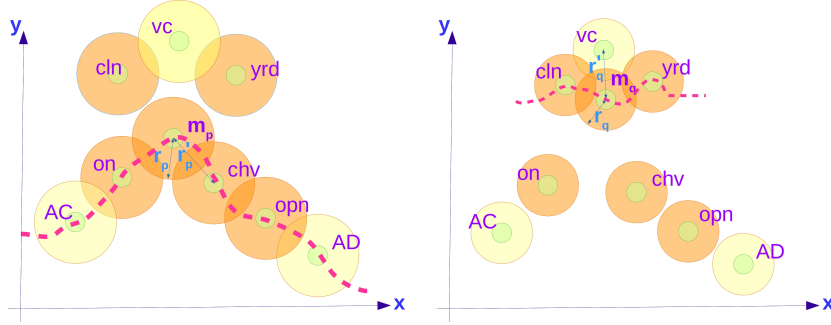


Figure 3: m_p is a better message than m_q as it introduces a longer inter-connected path using the method described in Section 2. As such, **opn** action from AD and **on** action from AC are triggered.

- What type of IAs contribute to C given Ψ (See T in Equation 35)
- How would IAs of type t reacts if i_k publishes a message such as m_p

As described in Section 6, Bayesian model, is a suitable model to estimate the aforementioned two models.

5 Clustering agents into different types

We can cluster/partition Ψ using methods such as **kmeans**. Such clustering may result in interesting sets of IA classes/types according to the nature of responsibilities they perform (Equation 34).

$$T = \{t_1, \dots, t_{|T|}\} \quad (34)$$

6 Modeling an equilibrium of induced decision by IAs, using Bayesian games

: Bayesian game models are useful to reach an equilibrium when there are two types **incomplete** information in a system

- Types of IAs/Players are not **completely** known to each IA.
- Given a type, then the probability by which they react (the decision) to decision/strategy of another IA is not **completely** known.

This is situation that we are exactly facing. That is a new IA in the system does not **completely** know what other types of IAs function in C /environment and they don't know completely how they react (what kind of message they publish in C or what sequence of actions they perform in real environment). As such Bayesian game seems to be a perfect choice to model the problem in order to find a

consistent equilibrium in it over the course of time. Now using the types described in Equation 34 we can introduce a model for which a Nash equilibrium could be sought as follows:

$$G = \langle I, \Omega, p, \langle A, u_k, T, \tau_k \rangle_{k \in I} \rangle \quad (35)$$

- I is the set of IAs that can publish messages in C
- Ω is possible distributions of Ψ over the course of time.
- A is the set of messages that could be composed out of W
- T is the set of the types of IAs. In the example presented in 2.4, if there have been more than one AD, VC or AC, then AD, VC and AC could have been considered as types and IAs could have been refereed with unique IDs. A function such as τ_k in Equation 36 gives the types of IAs functioning in the system, given a distribution such as Ψ° . This is the belief that each agent such as I_k should build as described in Section 4.

$$\tau_k : \Omega \rightarrow T^{|T|} \quad (36)$$

- $u_k : T \times A \rightarrow \mathbb{R}$ which denotes if agent i_k of type t_i performs an action such as publishing a message, how much would it be the summation of its utility and cost.
- p prior probability distribution over Ω (How much it is probable that upon arrival of an agent into an environment Ψ is the distribution of Ω)

Objective As such Bayesian game trains us a model to estimate u_k condition on knowing the T for agent i_k such that i_k for all $k/in 1 \dots |I|$ Equation 32 maximizes. Bayesian game model uses Nash equilibrium to determine the messages each agent can publish such that the best general equilibrium is achieved for the whole system.

Nash equilibrium Now that we have introduced the model, we should look for a Nash equilibrium as all agents are competing to publish messages in C such that they induce a longer set of actions (decision) to other IAs. Solving a Nash equilibrium can be done by the following methods.

- Dynamic Programming(DP)
- RL
- Linear programming

7 Question

If instead maximizing Equation 8, we try to maximize Equation 32 then how much would the system be successful instead? That is, if we take the condition in which all agents know existence of each other and know about the cost and utility functions of each other as the best scenario case, then if instead we use messages in C and a Nash equilibrium for decision making, then to what portion of the best scenario can we reach? This serves also as an evaluation method.

8 Notes

A great advantage : An action(a single symbol) and a decision (a se-

quence of actions / sequence of symbols) are reducible to a dimension of the same size.

Ease of Human in Loop We talked about advantage of m_p over m_q because as the symbols, we used English words.

Universal word embedding If a manufacturer uses "space" instead of "venue" in m_p , then that universal vector semantic can fill the gap.

Refining Ψ By keeping vector messages in Ψ then IAs should generate more refined messages and greater continuous surfaces will be formed which literally means better equilibrium in the final result of the system.

Why it is efficient? Why is this method efficient? because it implies that regardless of the length of m_p and variety of symbols used in it, \vec{m}_p has only one point in vector semantic space.

9 Further research

How does an IA thinks about the other? So far this document was focusing on anonymous message publication in C . But if there IA's could be identified when publishing messages then new sources of decision making are available. To present IA $i_j \in I$ belief's about $i_k \in I$, a new semantic vector space, $\Psi_{B_{jk}}$, can be formed by taking advantage of the union of messages published by i_j and i_k as presented in Equation

$$\Psi_{B_{jk}} = \bigcup_{m_j, m_k \in C} (\vec{m}_j \cup \vec{m}_k) \quad (37)$$