Cognitive Agents Integrating Rules and Reinforcement Learning for Context-Aware Decision Support

Teck-Hou Teng and Ah-Hwee Tan
School of Computer Engineering and Intelligent Systems Centre
Nanyang Technological University
{teng0032, asahtan}@ntu.edu.sg

Abstract

While context-awareness has been found to be effective for decision support in complex domains, most of such decision support systems are hard-coded, incurring significant development efforts. To ease the knowledge acquisition bottleneck, this paper presents a class of cognitive agents based on self-organizing neural model known as TD-FALCON that integrates rules and learning for supporting context-aware decision making. Besides the ability to incorporate a priori *knowledge in the form of symbolic propositional rules, TD-FALCON performs reinforcement learning(RL), enabling knowledge refinement and expansion through the interaction with its environment. The efficacy of the developed Context-Aware Decision Support(CaDS) system is demonstrated through a case study of command and control in a virtual environment.*

1. Introduction

Quality decision is made through astute application of choiced decision-making techniques on a selection of focused and directed information. Context-aware application achieves this end by focusing on the *who*, *where*, *when* and *what* to determine the *why* of a decision-making scenario [4]. It leverages on such contextual information to characterize the situation of a decision-maker.

There is a wide spectrum of context-aware applications such as in the area of information retrieval [1], information logistics [9], pervasive computing [10] and decision support in complex domains [6]. However, all of these works emphasized the use of context-aware information and required significant knowledge acquisition effort. The techniques for learning and autonomous expansion of knowledge for these systems were notably missing.

Context-aware Decision Support (CaDS) system exploits contextual information for focused situation assessment and goal-oriented decision support. In this paper, we present a CaDS system that is capable of learning and refining symbolic knowledge through RL. This CaDS incorporates a group of entity agents, one for each entity in the environment and a shared situation-awareness model. Specifically, we adopt a self-organizing neural model known as TD-FALCON [11] as the inference engine to each entity agent. As a generalization of ART [2], TD-FALCON learns multi-channel mapping simultaneously across multi-modal input patterns in an online and incremental manner. Due to its compatibility with rule-based knowledge representation, a class of propositional rules can be inserted conveniently into the FALCON network structure. More importantly, the refinement of such rules and the discovery of new rules is guided by temporal difference(TD) learning method.

Our proposed CaDS system is evaluated using a command and control problem domain known as *Missions-on-Mars*. The simulation is purposefully highly dynamic with hostile entity agents and unexpected occurrence such as ad-hoc goal injection and terrain changes. Our experiments show that the entity agent with the TD-FALCON inference engine achieves superior level of robustness and performance over an alternative rule-based engine known as DROOLS [8].

The rest of the paper is organized as follows. We begin with the presentation of the overall CaDS architecture and its core capabilities in Section 2. Next, the rule handling technique is detailed in Section 3. Description of the simulation scenario is provided in Section 4 and the experiment results are presented in Section 5. The final section concludes and provides a brief discussion of future work.

2. CaDS Architecture

The CaDS architecture (see Fig 1) incorporates a group of entity agents, one for each entity in the environment, and a shared situation-awareness model. The CaDS is supported by a real-time, multi-player, open source game simulation engine known as GECCO [7].

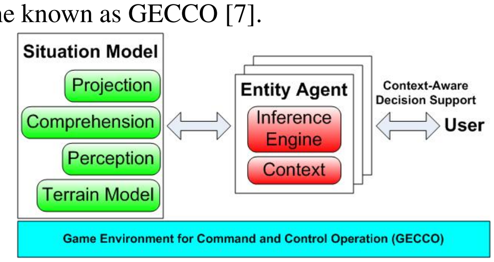

Figure 1: The CaDS Architecture.

The situation-awareness model is based on Endsley's three layer proposal [5] with an addition of a terrain model for the command and control domain. Knowledge about the environment is received at the *Perception* layer of the situation-awareness model. It is then interpreted and assessed at the *Comprehension* layer. To support anticipatory decision-making, the *Projection* layer predicts future states based on the understanding of the current situation.

The entity agent supports the decision-making processes of the user in a command and control setting for timely and effective decisions. Each entity agent has a set of goals and strategies, which form the basis of the entity's behaviors (see Fig 2). The goals and strategies are translated into executable structures such as missions, plans and actions. Together with the physical attributes such as location and health level constitute the context from which an agent is able to provide a customized set of views and services of the shared situation-awareness model to its user. The situation-awareness model is constantly updated with the user's context and the information on the environment. The important events are then highlighted to the user in accordance to their significance in the given context.

DROOLS was used as the inference engine that operates on propositional rules in a prior implementation [6]. Over here, it is substituted by novel self-organizing neural model known as TD-FALCON, that integrates rule-based knowledge and learning, as the inference engine of the entity agents. This neural model is utilised by entity agent with context-aware capabilities for *event classification*, *action recommendation*, and *mode selection*.

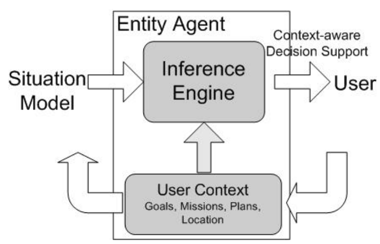

Figure 2: The entity agent model.

Event Classification - This context-aware capability prevents information overloading by presenting an event occurrence as an information, warnting or alert to the user. An event that is classified as an *information* is relevant to the user but does not affect the attainment of its goal. Event that may affect the chance of attaining the goal is issued as a *warning* while event that is certain to jeopardize the attainment of the goal is flagged as an *alert*.

Action Recommendation - An entity agent provides decision support to the user by recommending a list of possible action choices that is inline with the current situation model and the user's context. An entity agent compiles a ranked list of four action choices out of the six action choices, namely *Resume*, *Increase Speed*, *Decrease Speed*, *Reroute*, *Set to Required Speed* and *Wait*, for a given situations.

Mode Selection - The level of autonomy of the entity agent is indicated by the choice of decision modes. Three decision modes - Auto, Recommend and Don't Know - are defined in the CaDS. The entity agent automatically chooses the best option available for the current situation in the *Auto* mode. *Recommend* mode is used when the selected option is of high consequence, wherein the entity agent provides a ranked list of the options for the human operator for the final decision. *Don't Know* mode is used when the entity agent does not recognize the current situation and/or it cannot respond with an appropriate action.

3. Symbolic Rule Handling

The sensory and motor fields of FALCON correspond well to the antecedents and the consequents of propositional rules. A reward factor which corresponds to the feedback field is included as a performance metric on the rule. Therefore, *a priori* knowledge on a problem domain can be formulated as propositional rules and inserted into a FALCON network. This helps to improve learning efficiency and prediction accuracy. Prior to insertion, translation of the rules is required to bridge the gap in knowledge representation.

Rule Representation - FALCON is well-equipped to process a class of propositional rules whose consequents is implied from the antecedents with a reward factor. The reward factor is included to quantify the quality of the response as recommended by rule with respect to the overall outcome of a given task.

Formally, each propositional rule P has a set of antecedents X that gives rise to a set of consequents Y with a reward factor r, i.e., $P : X \xrightarrow{r} Y$ where X, Y and r can be defined as

$$X = \bigwedge_{n}^{N} x_n, \quad Y = \bigwedge_{m}^{M} y_m, \quad r \in [0, 1]$$

where $\bigwedge$ indicates the logical AND operator, N is the number of antecedents and M is the numer of consequents such that the propositional rule P can thus be expressed as

IF	$x_1 \wedge \ldots \wedge x_n \wedge \ldots x_N$
THEN	$y_1 \wedge \ldots \wedge y_m \wedge \ldots y_M$
REWARD	$r \in [0, 1]$

whose elements x_n and y_m can be commonly defined as $\{(a_s, v_t) | a_s \in \mathbf{X}, v_t \in \mathbf{V}_s\}$ where a_s is an attribute having a value v_t, $\mathbf{X}$ is the universe set of attributes, $\mathbf{V}_s$ is the domain of values for attribute a_s and r is the reward factor for propositional rule P. Each rule has an unique combination of antecedents and consequents. In addition, the consequents are not dependent on its previous values, i.e., $X \cap Y = \emptyset$.

Rule Translation - For insertion into the FALCON model, each feature of the symbolic rule has to be translated into an equivalent vector format. Specifically, an attribute a_s with T possible values whose attribute-value pairs (a_s, v_t) is to be translated into a vector as below.

$$[(b_{s1}, b_{s1}^c), (b_{s2}, b_{s2}^c), \ldots, (b_{st}, b_{st}^c), \ldots, (b_{sT}, b_{sT}^c)]$$

such that each of the complement-coded 2-bit vector (b_{st}, b_{st}^c) [3] is defined in accordance to

$$(b_{st}, b_{st}^c) = \begin{cases} (1,0) & \text{if } a_s = v_t \\ (0,1) & \text{if } a_s \neq v_u \text{ where } u \neq t \\ (0,0) & \text{attribute has no effect} \end{cases} \quad (1)$$

The concatenated vector of the complement-coded vectors for every attribute-value pairing of attribute a_s is known as the attribute vector and it has the dimension $(2 \times |\mathbf{V}_s|)$. The antecedent vector of the rule is a further concatenation of S attribute vectors which will cause it to have the dimension of $\sum_s^S (2 \times |V_s|)$ where $S \equiv |\mathbf{X}|$. All the attributes are represented in the antecedent vector regardless of whether they are considered in the rule P. Similarly, the consequent vector is the concatenation of the attribute vectors based on element y_m of the set of consequents. The reward vector is comprised of the reward factor r and its complement $1 - r$, i.e., $(b_r, b_r^c) = (r, 1 - r)$.

In the context of FALCON, for each propositional rule P, the antecedent vector is known as the state vector $\mathbf{S}$, the consequent vector is known as the action vector $\mathbf{A}$ while the reward vector $\mathbf{R}$ is taken as it is. Hence, each propositional rule P will translate to an unique pattern set F comprising of $\mathbf{S}$, $\mathbf{A}$ and $\mathbf{R}$, i.e., $P\{X, Y, r\} \rightarrow F\{\mathbf{S}, \mathbf{A}, \mathbf{R}\}$

Rule Insertion - After translation, the state $\mathbf{S}$, action $\mathbf{A}$ and reward $\mathbf{R}$ vectors are inserted into FALCON through the iterative performance of the code activation, code competition, template matching and template learning procedure as described in [11]

During rule insertion, the vigilance parameters ρ^{ck} are each set to 1 to ensure that only identical attribute vectors are grouped into the same recognition category. The *perfect mismatch* phenomenon detects contradictory rules in which the system tries to raise the sensory field vigilance ρ_s above 1 in response to a mismatch in the motor field.

All the inserted rules are assumed to be distinct. Unless there is a perfect match to any existing nodes, any other dissimilar patterns are learned by FALCON as a cognitive node. Hence, there are as many cognitive nodes as the number of inserted propositional rules.

4. Case Study: Mission-on-Mars

The *Mission-On-Mars* platform aims to be an adequate model for a complex and dynamic decision-making problem domain. The Explorer ER100 and the Alien are two entity agents while the HQ and Base Alpha are passive landmarks within this platform. The entity agents navigate through the terrain with the aid of a sensor network (see Fig 3) that is deployed over the terrain. Both agents are assumed to have complete knowledge about the terrain through this sensor network. For example, information on whether the path is blocked or unblocked is available through the sensor network.

The Explorer ER100 is tasked to navigate through the sensor network to its destination using the most efficient and safest route. The Alien is tasked to patrol along a section of the sensor network. The Alien is hostile to the Explorer ER100. It can inflict health damages to the Explorer ER100 in a close encounter. Thus, the Explorer ER100 is expected to avoid contact with the Alien as much as possible. In addition, the Explorer ER100 is able to respond to three types of goals and two types of terrain events.

The challenge is for Explorer ER100 to respond to adhoc goal injection, terrain changes as well as threats from the Alien entity agent. The Explorer ER100 can choose an action from the set of action choices mentioned in Section 2. It is expected to select one of these action choices based on the situation-awareness model and the context information (see Fig 2) with the aid of the inference engine.

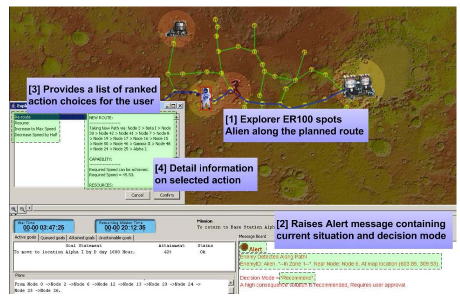

Figure 3: An illustration of context-aware decision support.

5. Experiments

The objective of the experiments is to evaluate the suitability of TD-FALCON as an inference engine for the entity agent operating in a complex decision-making domain. It is required to provide accurate response to scenarios using *a priori* knowledge as well as learned knowledge. The experiments compare the prediction accuracy of TD-FALCON trained using two learning paradigms - RL and supervised learning (SL), each with or without *a priori* knowledge and also against another rule inference engine known as DROOLS. SL and DROOLS are included for baseline comparison against the RL approach.

There are nine rules for action recommendation, another nine rules for mode selection and six rules for event classification with a total of 24 attributes as the rule antecedents and three attributes as the rule consequents. For demonstration purpose, it is sufficient to only include the results on the action recommendation context-aware capability over here.

The rule inference engines are evaluated using 281 realistic situations from the *Mission-On-Mars* platform. TD-FALCON is separately trained using SL and RL. TD-FALCON is taught the expected response when an incorrect response is provided during SL. With RL, TD-FALCON is only provided with a reward signal indicating the correctness of its recommendation.

TD-FALCON operates in the PERFORM mode to derive a response from its knowledge base while it gets into the LEARN mode to update this knowledge base. TD-FALCON operates with baseline vigilance $\rho^{ck} = \{0.2, 0.8, 0.5\}$ for the state, action and reward fields respectively in the LEARN mode while it has $\rho^{ck} = \{0.0\}$ in the PERFORM mode. Both modes use a common set of values for the following sets of parameter: choice parameters $\alpha^{ck} = \{0.1, 0.001, 0.001\}$, learning rate $\beta^{ck} = 1.0$ for fast learning and contribution parameter $\gamma^{ck} = \frac{1}{3}$ for $k = 1, 2, 3$.

From Fig 4, DROOLS yields a consistent prediction accuracy of 97.15% while TD-FALCONs trained using SL achieve 100% prediction accuracy ealier than those trained using RL. This is expected as SL teaches the expected responses to TD-FALCON while RL requires more iterations to explore the solution space for suitable responses. The absence of learning capability accounts for the stagnant performance of DROOLS.

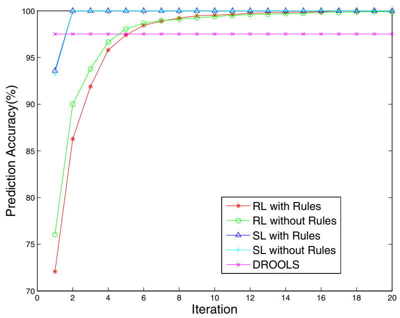

Figure 4: Comparison of action choice prediction accuracy

The profile of the prediction accuracy plots of the TD-FALCONs trained with and without rules in Fig 4 are quite closely matched. The inherent inadequacy of the inserted rules is also highlighted by the lower initial prediction accuracy of the configurations with rule insertion over those without rule insertion. This indicates the reduced role of the *a priori* knowledge after the acquisition of more sophisticated knowledge. The learning mechanism of TD-FALCON is able to supersede the less adequate rules with those that are able to provide more accurate responses.

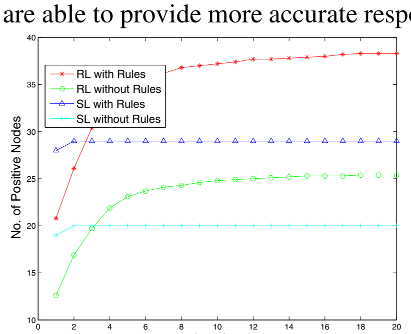

Figure 5: Comparison of the number of cognitive nodes created

Fig 5 plots the creation of the cognitive nodes from each of the experiment configurations. The production of cognitive nodes plateaus as the prediction accuracy approaches 100% accuracy. This indicates that TD-FALCON has acquired sufficient knowledge to provide the appropriate responses to all the situations. Generalization is observed as the number of positive nodes created is significantly lesser than the situations that it has to respond to.

6. Conclusions

A CaDS architecture with TD-FALCON that is able to integrate rules and perform RL for context-aware decision support has been presented. The capability of TD-FALCON as an adaptive rule inference engine has been clearly illustrated and the efficacy of our approach has been demonstrated. It is able to learn and achieve performance level which exceeds that of existing rule inference engine. Upcoming work shall include online code evaluation and pruning procedure to maintain the network at a compact size. Further enhancement to the learning algorithm can be achieved through the integration with the symbolic rule handling algorithm and the use of negative knowledge. Other opportunities for enhancements shall be explored to bring about positive and incremental contributions.

References

[1] A. Bahrami, J. Yuan, P. R. Smart, and N. R. Shadbolt. Context aware information retrieval for enhanced situation awareness. In *3rd IEEE Workshop on Situation Management*, Orlanda, Florida, Oct 2007.

[2] G. A. Carpenter and S. Grossberg. Adaptive Resonance Theory. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 87–90. MIT Press, 2003.

[3] G. A. Carpenter, S. Grossberg, and D. B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[4] A. K. Dey. Understanding and using context. *Personal and Ubiquitous Computing Journal*, 5:4–7, 2001.

[5] M. R. Endsley. Toward a theory of situational awareness in dynamic systems. *Human Factors*, 37(1):32–64, 1995.

[6] Y.-H. Feng, T.-H. Teng, and A.-H. Tan. Modeling situation awareness for context-aware decision support. *Expert Systems with Applications*, 2007.

[7] GECCO. Game Environment for Command and Control Operations. http://www.csc.kth.se/tcs/gecco/, May 2004.

[8] jboss.org. http:// www.jboss.org/ drools, Jul 2008.

[9] U. Meissen, S. Pfennigschmidt, A. Voisard, and T. Wahnfried. *Context- and Situation-Awareness in Information Logistics*, volume 3268/2004 of *LNCS*, pages 355–344. Nov 2004.

[10] B. N. Schilit and M. M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, 8:22–32, 1994.

[11] A.-H. Tan, N. Lu, and D. Xiao. Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback. *IEEE Transactions on Neural Networks*, 19(2):230–244, Feb 2008.