2019-12

# Modeling Large-Scale Warfighter Cognitive Reasoning and Decision- Making Using Machine Learning (ML), Artificial Intelligence (AI), and Game Theory (GT)

Zhao, Ying; Kendall, Tony A.

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/70016

# Causal Learning in Modeling Multi-segment War Game Leveraging Machine Intelligence with EVE Structures

Ying Zhao, Naval Postgraduate School, Monterey, CA, USA, yzhao@nps.edu
Bruce Nagy, NAVAIR, China Lake, CA, USA
11/8/2019

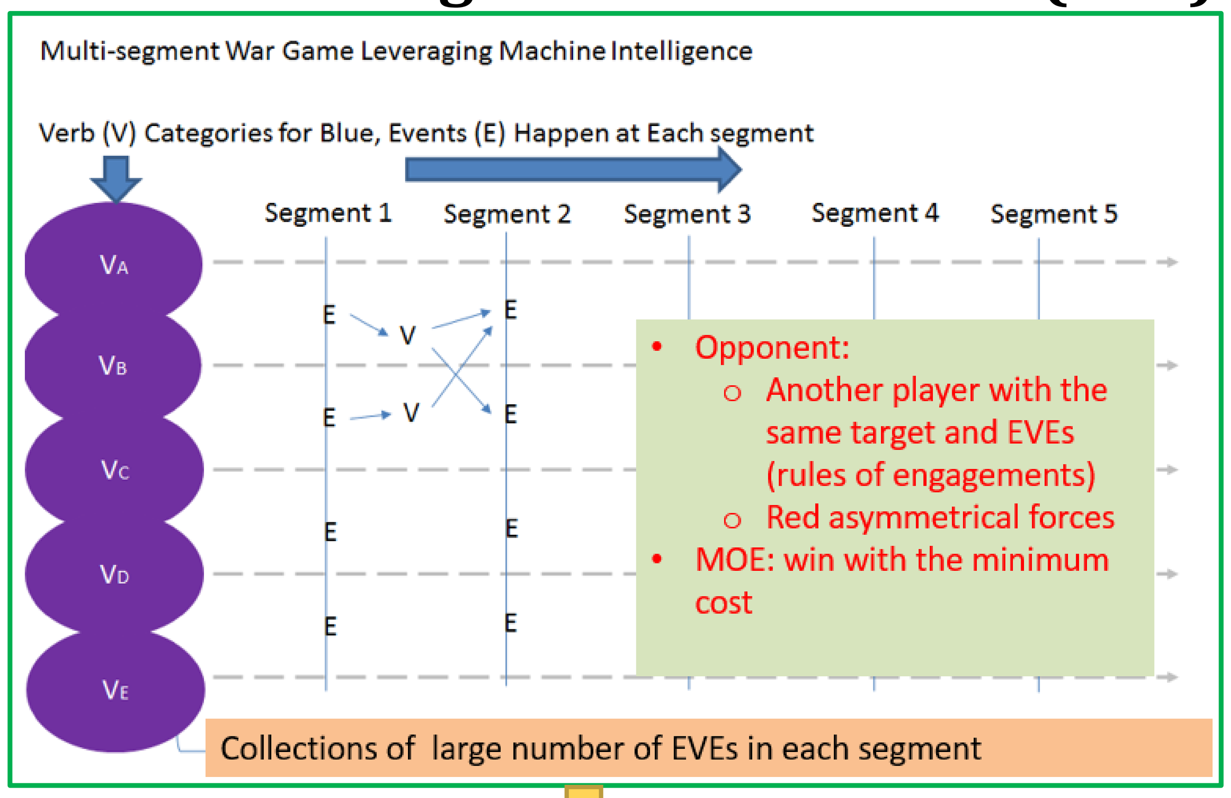
Naval Postgraduate School

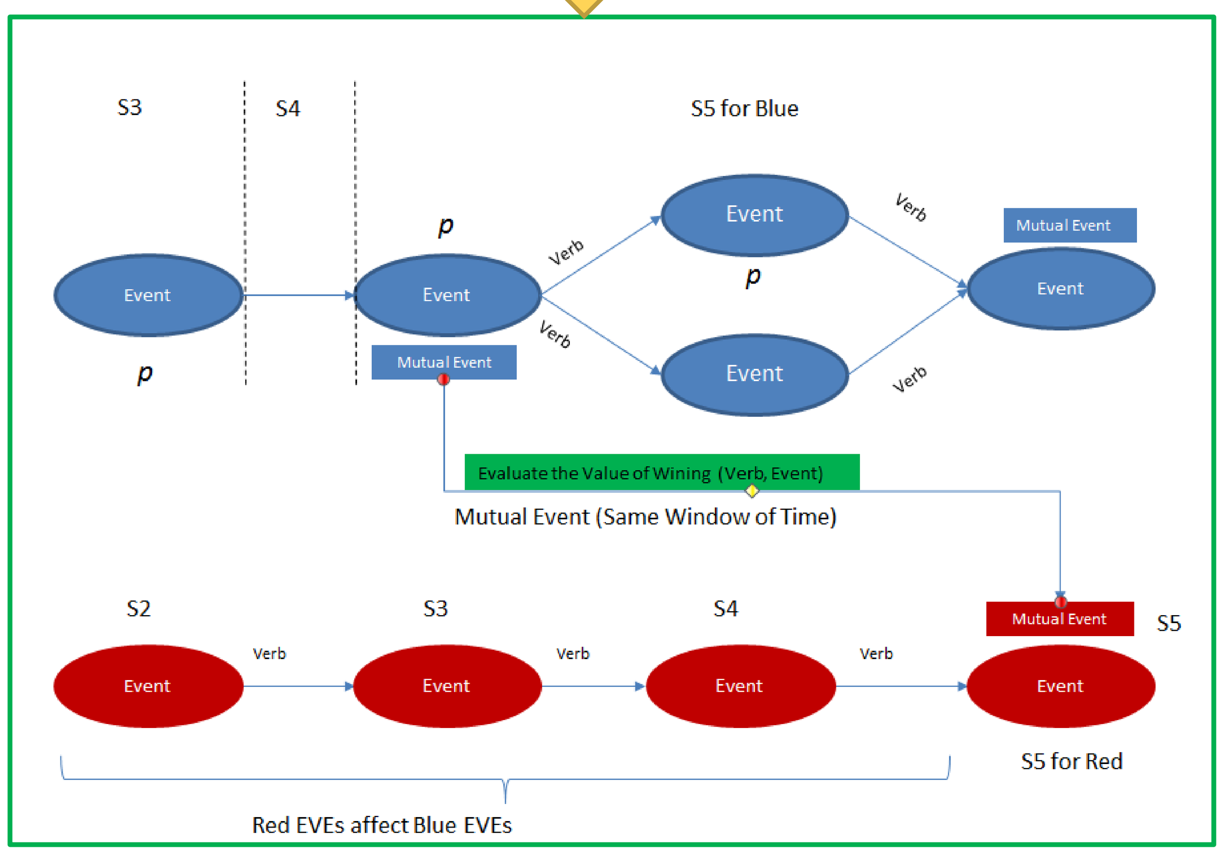
## Big Picture, Challenges, and Goals

- Cognitive architecture, algorithms, and software systems are important to model complex reasoning, cognitive functions, and decision-making in defense applications.
- Depict a generic representation of a multi-segment war game leveraging machine intelligence with two opposing players with asymmetrical rules of engagements.
- The war game uses Event-Verb-Event (EVE) rules and structures, integration of a reinforcement learning (Soar-RL) technique to achieve decision-making superiority.
- Soar is a cognitive architecture (Laird, 2012) and RL (Sutton, 2014) is a class of ML/AI algorithms capable of automating some cognitive functions of warfighters, the decision-making of a tactical action officer in a CID task (Zhao, 2016; Mooren, 2017; Zhao et al., 2017, 2018).
- Soar-RL which have a natural linkage to causal learning of three layers of a causal hierarchy - association, intervention, and counterfactuals, as well as a few other key elements of causal learning.
- Causal learning techniques has the potential for tactical decision edge for the players. Causal learning is closely related to anticipatory thinking of human cognition.
- Our method can be applied to defense applications in the vast, complex, and uncertain areas of Cybersecurity and Information Warfare, including such applications as combat identification, Battlespace Awareness, C-C4ISR, Assured C2, modeling/simulation, and mission planning and war games.

## Multi-segment War Game Using Event-Verb-Event (EVE) Rules

- Two opposing asymmetrical players with different rules of engagements
- Rules generated top down from experts or bottom up data mining from previous knowledge and databases
- EVE example: "If an opponent is found (event), then the track (verb) the opponent using tool A" and "if the opponent has been successfully tracked (event), then target (verb) the opponent using tool B."




## Data Set and Simulation Design

Soar-RL large-scale test data: 1.3M training combinations/400K test combinations and 50 attributes, ~25 attributes are state variables, and ~25 are action variables

Table 1: Sequential or Parallel Asymmetric Action Combinations

| Self-player | Opponent (e.g., environment or adversaries) |
|---|---|
| Action/state combination $d_1$ | $o_1$ |
| Action/state combination $d_i$ | $o_i$ |
| … | |
| Action combination $d_N$ | $o_M$ |

Table 2: Action Combination Components

| Action/state combination component | $c_1$ | $c_i$ | … | $c_k$ | End reward |
|---|---|---|---|---|---|
| | 1 | 0 | | 0 | win |
| | 1 | 1 | 0 | 1 | not win |

states or actions

Data representation: Boolean lattice including counterfactuals

Define preferences (e.g., contribution from an action combination component $c\_k$) $c\_k\_v\_1\_r\_1$, $c\_k\_v\_0\_r\_1$, $c\_k\_v\_1\_r\_0$, and $c\_k\_v\_0\_r\_0$. For example, $c\_k\_v\_1\_r\_1$ means "if an action combination component $c\_k$ is included ($v = 1$), there is a preference (probability) $c\_k\_v\_1\_r\_1$ for the self-player to win the game in the end ($r = 1$)."

We show how the preferences can be computed for the rules. Let $m$ be the number of rules and $N$ the number of data for Soar-RL to perform on-policy learning [18][19].

$$Q(s_{t+1}, a_{t+1}) = Q(s_t, a_t) + \alpha[r + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (1)$$

Since we only consider an on-policy setting or SARSA, $Q(s_{t+1}, a) = 0$ and let

$$\delta_t = \alpha(r_{t+1} - Q(s_t, a_t)) \quad (2)$$

$\alpha, r_{t+1} = 1$ for a positive reward or $-1$ for a negative reward. In order to converge, $r_* = Q(s_*, a_*)$ in Equation (2),

### Soar-RL and Counterfactuals

In our representation as in Table 2, each action combination consist of multiple components which can be represented as binary 1 (action or condition included) or 0 (action or condition not included) as a Boolean lattice[21].

When a Soar-RL learns/updates the rules, it learns/updates the preferences of the action components as well as their counterfactuals, therefore,

- $P(good\ result|action\ component\ k\ included)$,
- $P(good\ result|action\ component\ k\ not\ included)$,
- $P(bad\ result|action\ component\ k\ not\ included)$, and
- $P(bad\ result|action\ component\ k\ not\ included)$

are estimated independently. We are able to compare $P(good\ result|action\ component\ k\ included)$ with its counterfactual $P(bad\ result|action\ component\ k\ not\ included)$ for all the components and select the components that have the difference greater than a predefined threshold.

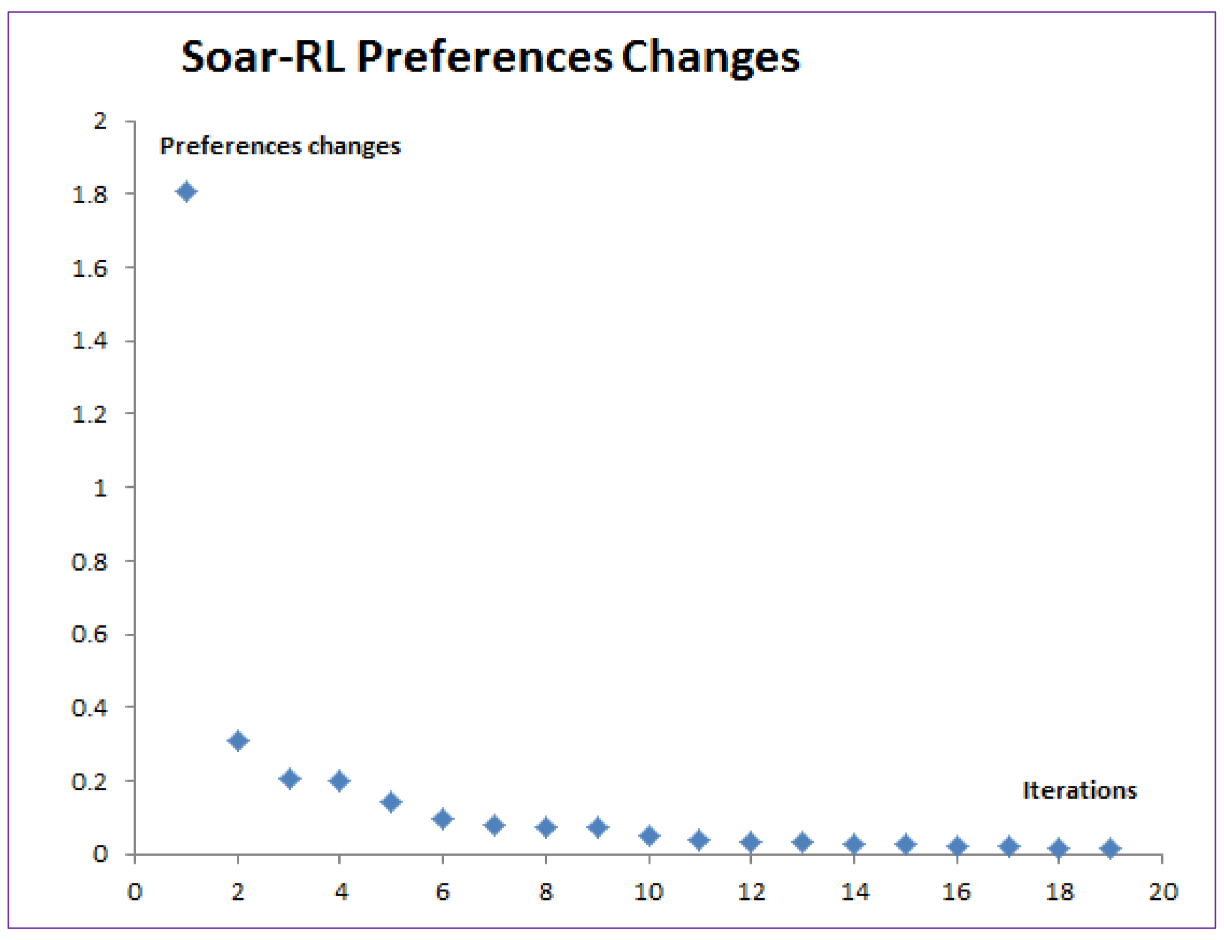
## Results: Soar-RL XAI Learning Process

```
Adaptation Mode ('b' for binary or 's' for spreadsheet):b
Input data (1 or 0 separated by comma):1,0,0,1,0,0,0,0,1,0,0,1,0,0,1
Type ground truth 1(good) or 0(bad):1
```

```
===========Soar Predicted:0
===========Ground Truth:1
===========Soar-RL's  Decision Wrong!

Rules updated:

  O5 (preference) =0
    Self_player_action_component_53 not included good

  O5 (preference) =0
    Self_player_action_component_52 not included good

  O5 (preference) =0
    Self_player_action_component_51 included good

  O5 (preference) =0
    Self_player_action_component_50 not included good

  O5 (preference) =0
    Self_player_action_component_49 not included good

  O5 (preference) =0
    Self_player_action_component_48 not included good
```

```
===========Soar Predicted:1
===========Ground Truth:1
===========Soar-RL's  Decision Correct!

Rules updated:

  O10 (preference) =7.407407407407408e-006
    Self_player_action_component_53 not included good

  O10 (preference) =7.407407407407408e-006
    Self_player_action_component_52 not included good

  O10 (preference) =7.407407407407408e-006
    Self_player_action_component_51 included good

  O10 (preference) =7.407407407407408e-006
    Self_player_action_component_50 not included good

  O10 (preference) =7.407407407407408e-006
    Self_player_action_component_49 not included good

  O10 (preference) =7.407407407407408e-006
    Self_player_action_component_48 not included good
```

## Results: Soar-RL Convergence




## Conclusions

- We showed the EVE structures and integration of reinforcement machine and causal learning techniques in a multi-segment war game.
- We showed how the EVE structures and related machine intelligence techniques reflect the critical elements of causal learning.
- The integration of machine and causal learning techniques has the potential for tactical decision edge for defense applications.

Poster to the Association for the Advancement of Artificial Intelligence (AAAI) 2019 Fall Symposium: Cognitive Systems for Anticipatory Thinking Washington, DC, November 7–9, 2019