Multi-agent Systems

Bo AN (安波)
Nanyang Technological University

August 5, 2020
Outline

- Part 1-History and current status (10min)
- Part 2-Key research areas in MAS (30min)
- Part 3-Recent advances (40min)
  - Computer poker
  - Game theory for security
  - Multi-agent RL

Points I want to make:

- Mainstream AI research
- Still at the very early age: Lots of work to be done and we can make a difference in both theory and applications
MAS Research: History

1978
CMU workshop on distributed sensor networks

1980
MIT workshop on distributed AI

1995
1st ICMAS, IFAAMAS

2002
AAMAS

40-50s
Agent concept

70s
Components for building agents (e.g., STRIPS)

80s
MAS research: coordination, cooperation, negotiation, organizations

90s
BDI, KQML, AOP, DCOP, RoboCup

00s
Game theory, social choice, TAC, DEC-MDP

10s
MAS+DRL
MAS: One of The Most Important Field in AI

➢ Over 30 years’ history: Initially called distributed AI

➢ Now one of the most important/active fields of AI

☆ IJCAI Award for Research Excellence: Victor Lesser (09), Barbara Grosz (15)

☆ IJCAI Computers and Thought Award:

△ Sarit Kraus (95), Nicholas Jennings (99), Tuomas Sandholm (03), Peter Stone (07), Vincent Conitzer (11), Ariel Procaccia (15)

☆ Large number of accepted papers at AAAI/IJCAI

☆ Researchers from top universities

☆ Quality of AAMAS, JAAMAS
MAS Research Areas

- **Theoretic foundations**
  - Coordination, DEC-(PO)MDP
  - DCOP
  - Game theory
    - mechanism design, coalition formation, social choice, negotiation
  - Multi-agent Learning
  - Organizational design

- **Applications**
  - Robotics, security, sustainability, rescue, sensor networks, games...
Cooperative Multi-Agent Systems:
Groups of Sophisticated AI systems that Work Together

- Open, dynamic, persistent systems
- Decentralized control
- Asynchronous
- Large scale (10s to 1000s)
- Partial observability
- No real-time global reward signal
- Communication delay
Partial Global Planning (PGP) [Lesser et al]

- Form a partial-global-plan (PGP) to achieve partial-global-goal
- PGP points to local models of participating plans

**Process for Each Agent**

- **Generation of local plans for an agent**
- **Recognition of relationships among other agent plans**
- **Modify plans to be coherent with other agents**

**local plans**

- $plan_{11}$
- $plan_{12}$

**node-plans**

- $nodeplan_{11}$
- $nodeplan_{12}$
- $nodeplan_{13}$
- $nodeplan_{14}$

**PGPs**

- $PGP_{11}$
- $PGP_{12}$

**TÆMS, GPGP, distributed search**

Further reading: Durfee, Lesser: Using Partial Global Plans to Coordinate Distributed Problem Solvers. IJCAI’87.
Towards Flexible Teamwork [Tambe]

Scout crashed, company waited forever

Commander returned to home base alone...

Ad-hoc: No Framework to anticipate failures

Cohen/Levesque, 91
Joint Intentions

Grosz/Kraus, 96
SharedPlans

Jennings, 95
GRATE*

Lesser/Durfee
Decker
PGP/GPGP

Sonnenberg/Rao
Tidhar/Georgeff
BDI research

STEAM

Dec-(PO)MDP Models

GO-DEC-MDP  Goldman et. al, 2004
OC-DEC-MDP  Beynier et. al, 2005

ND-POMDP  Nair et. al, 2005

Restricted Problem

Locality of Interaction

Interaction in a few
global states

Dec-(PO)MDP Models

Varakantham et. al, 2009
DPCL
IDMG  Spaan et. al, 2008

DEC-POMDP  Bernstein et. al, 2000

Independent & Structured Dependence

Factored State & Local Observability

TI-DEC-MDP
EDI-DEC-MDP

LO-DEC-MDP

Becker et. al, 2004
Becker et. al, 2004
Witwicki et al., 2010

TD-POMDP

Distributed Constraint Satisfaction & Optimization

Applications:
- Distributed sensor networks
- Resource allocation
- Meeting scheduling
- Engineering design

Organizational Design

- Human organizations
  - Businesses
  - Governments
  - Universities

- Long-term, repetitive, multi-participant

- Organizations give participants a set of responsibilities, goals, incentives and guidelines

- MAS organizations are the same
  - Assignment of goals, incentives, roles, rights, authority, expectations, partners, rules, conventions...

Game Theory for AI

- GT analyzes *multi-agent* interaction (1940s-)

- AI: study and construction of *rational* agents [Russell & Norvig, 2003]

  **Building a single agent** (1950s-70s)  **Multi-agent systems (cooperative)** (1980s-)  **Multi-agent systems (competitive)** (1995-)

- GT for AI: success in computer poker, security, auction, GAN, …
Players, strategies, payoffs

<table>
<thead>
<tr>
<th></th>
<th>Player A</th>
<th>Player B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 0</td>
<td>1, -1</td>
<td>-1, 1</td>
</tr>
<tr>
<td>-1, 1</td>
<td>0, 0</td>
<td>1, -1</td>
</tr>
<tr>
<td>1, -1</td>
<td>-1, 1</td>
<td>0, 0</td>
</tr>
</tbody>
</table>

Nash Equilibrium: no agent has incentive to unilaterally deviate

- In any (finite) game, at least one Nash equilibrium (possibly mixed) exists \[Nash, 50\]
- In 2-player zero-sum games, a profile is an NE iff both players play minimax strategies
- Computing one (any) Nash equilibrium is PPAD-complete (even in 2-player games) \[Daskalakis, Goldberg, Papadimitriou 2006; Chen, Deng 2006\]
- All known algorithms require exponential time (in the worst case)
  - Lemke-Howson, support enumeration

Mechanism design

Recent advances: Libratus and Pluribus

**Abstraction**

(offline)
- action abstraction
- card abstraction
- took the game size from $10^{161}$ to $10^{12}$

**Equilibrium Finding**

(offline)
- CFR
- CFR+
- Monte Carlo CFR

**Decomposition and Subgame Refinement**

(online)
- endgame solving
- subgame re-solving
- max-margin subgame refinement

**Deep learning:** Alberta’s DeepStack, DeepMind
Regret and Regret Matching

- For each player \( i \), action \( a_i \) and time \( t \), define the regret \( r_i(a_i, t) \) as
  \[
  \sum_{1 \leq t' \leq t-1} (u_i(a_i, a_{-i,t'}) - u_i(a_{i,t'}, a_{-i,t'}))/(t-1)
  \]

- Regret matching: at time \( t \), play an action that has positive regret \( r_i(a_i, t) \) with probability proportional to \( r_i(a_i, t) \)
  - If none of the actions have positive regret, play uniformly at random

- **Theorem** [Hart & Mas-Colell 2000]: Informally, if we pick actions according to RM in a zero-sum game, the regret for any \( a_i \) is approaching 0 as \( t \) goes to infinity

- **Folk Theorem**: In a zero-sum game at time \( T \), if both players’ average overall regret is less than \( \varepsilon \), then the average strategy is a \( 2\varepsilon \) equilibrium
**CFR: Regret Computing**

- **Key idea:** minimize regret independently in each information set of an extensive-form game by maintaining the average immediate regret:

\[
R_i^T(I, a) = \frac{1}{T} \sum_{t=1}^{T} \pi_{-i}(I) \cdot (u_i(\sigma_t^{I \rightarrow a}, I) - u_i(\sigma^t, I))
\]

- **\( \pi \)** is the probability of reaching information set \( I \) given the strategies at time \( t \) and the player \( i \) is trying to reach that information;

- **\( \mu \)** is the utility of a strategy, including that we switch our action only at that information set

- Define the strategy at information set \( I \) and time \( t+1 \) as

\[
\sigma_{i}^{t+1}(I, a) = \begin{cases} 
\frac{R_{i}^{t+1}(I, a)}{\sum_{a \in A(I)} R_{i}^{t+1}(I, a)} & \text{if } \sum_{a \in A(I)} R_{i}^{t+1}(I, a) > 0 \\
\frac{1}{|A(I)|} & \text{otherwise.}
\end{cases}
\]

Further reading: Zinkevich, Johanson, Bowling, Piccione: Regret Minimization in Games with Incomplete Information. NIPS’07.
Game Theory for Security [50+@ AAAI, AAMAS, IJCAI, ICML, NeurIPS]

- Global challenges for security
  - Boston Marathon bombings
  - French oil tanker hit by a boat
  - Cyber physical attacks

- Security resource allocation
  - Limited security resources
  - Adversary monitors defenses, exploits patterns

- We pioneered the first set of applications of game theory for security resource scheduling (2007-)
  - 40+ papers at premier conferences/journals, 2 best paper awards
  - INFORMS Daniel H. Wagner Prize for Excellence in Operations Research Practice (2012), etc
  - Operational Excellence Award from US Coast Guard (2012), etc
  - United States congressional hearing (4 times)
Security Games

- Security allocation: (i) Target weights; (ii) Opponent reaction
- Stackelberg game: Security forces commit first
- Strong Stackelberg Equilibrium
- Our contributions:
  - Algorithms for solving large scale games
  - Learning adversary behavior
  - Applications in the real world

<table>
<thead>
<tr>
<th>Defender Action</th>
<th>Attacker Action #1</th>
<th>Attacker Action #2</th>
<th>......</th>
<th>......</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defender Action</td>
<td>7, -3</td>
<td>-6, 2</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>Defender Action</td>
<td>-3, 4</td>
<td>4, -6</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
</tbody>
</table>
Combining techniques from AI, Game Theory, Operations Research …

• Marry theory with practice

• Approaches can be applied to other domains With proper tuning & extension
  - Incremental strategy generation
  - Construct (multiple) equivalent games
  - Exploit compact representation
  - Abstraction
  - Tradeoff between optimality and efficiency
  - Approximation

Further reading: An: Game theoretic analysis of security and sustainability. IJCAI’17.
Equilibria in Multiplayer Games
- Hard to compute: PPAD-Complete
- Hard to select: NEs are not unique
- Few results:
  - Special structure: congestion games
  - No theoretical guarantee: Pluribus (Brown and Sandholm 2019)

Team-Maxmin Equilibria (von Stengel and Koller 1997)
- A team of players independently plays against an adversary
- Unique in general
- FNP-hard to compute a team-maxmin equilibrium
  - Formulated as a non-convex program
  - Solved by a global optimization solver

Converging to Team-Maxmin Equilibria
- Existing ISG for multiplayer games
  - Converge to an NE but many not to a TME
  - Difficult to extend the current ISG to converge to a TME
- ISGT: the first ISG guaranteeing to converging to a TME
  - Conditions in ISGT cannot be further relaxed
- CISGT: further improve the scalability
  - Initialize the strategy space by computing an equilibrium that is easier to be computed

<table>
<thead>
<tr>
<th>L×W</th>
<th>5×5</th>
<th>5×5</th>
<th>5×5</th>
<th>5×5</th>
<th>5×5</th>
<th>4×4</th>
<th>6×6</th>
<th>8×8</th>
<th>10×10</th>
</tr>
</thead>
<tbody>
<tr>
<td>(p,q)</td>
<td>(0.8,0.6)</td>
<td>(0.7,0.5)</td>
<td>(0.6,0.4)</td>
<td>(0.5,0.3)</td>
<td>(0.4,0.2)</td>
<td>(0.4,0.2)</td>
<td>(0.4,0.2)</td>
<td>(0.4,0.2)</td>
<td>(0.4,0.2)</td>
</tr>
<tr>
<td>FullTME</td>
<td>∞</td>
<td>448s</td>
<td>50.4s</td>
<td>17.8s</td>
<td>0.3s</td>
<td>∞</td>
<td>&gt;1000s</td>
<td>4s</td>
<td>&gt;1000s</td>
</tr>
<tr>
<td>ISGT</td>
<td>9.8s</td>
<td>5.9s</td>
<td>4.7s</td>
<td>3.7s</td>
<td>2.3s</td>
<td>2.2s</td>
<td>8.3s</td>
<td>24s</td>
<td>57s</td>
</tr>
<tr>
<td>CISGT</td>
<td>9.8s</td>
<td>5.9s</td>
<td>4.7s</td>
<td>3.7s</td>
<td>2.3s</td>
<td>2.2s</td>
<td>8.3s</td>
<td>24s</td>
<td>57s</td>
</tr>
</tbody>
</table>

Table 1. Computing TMEs: ∞ represents out of memory.
Manipulating a Learning Defender and Ways to Counteract

- Learn to play optimally in a Stackelberg Security Game (SSG)

[Letchford et al., 2009; Blum et al., 2014; Haghtalab et al., 2016; Roth et al., 2016; Peng et al., 2019]

- Unrealistic assumption about truthful attacker responses → What if untruthful?

- In our work:
  - Learning algorithms can be easily manipulated by untruthful attacker
  - Often optimal for attacker to deceive defender into playing a zero-sum game
  - A policy-based framework to play against attacker deception
  - A poly-time algorithm to compute optimal policy and a heuristic approach for infinite attacker types
When do We Need RL?

- RL might be more appropriate when
  - *Problem cannot be well modelled*
  - *Large scale*
  - *Non-convex and cannot be approximated*
  - *No domain structures can be exploited*

- Multi-agent RL is receiving increasing attention
  - *Agents collaborate/compete with each other without the coordinator*
  - *Often centralized training and decentralized execution (CTDE)*
  - *MADDPG, COMA, VDN, QMIX ...*
  - *More resources:*

- Does not mean multi-agent RL can always work!

- Rest of the lecture: quick overview of some of our recent works on RL
  - *Game, security, e-commerce, urban planning: competitiveness - data rich/poor*
Competitive Bridge Bidding with Deep Neural Networks [AAMAS’19]

- Imperfect information
  - Not necessary to infer opponents’ cards
  - Estimation neural network (ENN) to infer partner’s cards

- Cooperation & competition
  - Train the policy neural network (PNN) & ENN simultaneously
  - Maintain an opponent pool for competitors
  - Self-play with approximate reward from DDA

- Large state space
  - Efficient bidding sequence representation

<table>
<thead>
<tr>
<th></th>
<th>SL-PNN</th>
<th>RL-PNN</th>
<th>SL-PNN+ENN</th>
<th>RL-PNN+ENN</th>
<th>Wbridge5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL-PNN</td>
<td>N/A</td>
<td>-8.7793</td>
<td>-5.653</td>
<td>-9.2957</td>
<td>-</td>
</tr>
<tr>
<td>RL-PNN</td>
<td>8.7793</td>
<td>N/A</td>
<td>2.1006</td>
<td>-1.0856</td>
<td>-</td>
</tr>
<tr>
<td>SL-PNN+ENN</td>
<td>5.653</td>
<td>-2.1006</td>
<td>N/A</td>
<td>-2.2854</td>
<td>-</td>
</tr>
<tr>
<td>RL-PNN+ENN</td>
<td>9.2957</td>
<td>1.0856</td>
<td>2.2854</td>
<td>N/A</td>
<td>-</td>
</tr>
</tbody>
</table>

Test on 10 thousand boards | Test on 64 boards manually
Fraud transactions in e-commerce:

- Sellers buy their own products to fake popularity
- Deliberate, large scale fraud: Grey industry
- Existing approach: Machine learning for fraud detection

Reducing fraud by optimal impression allocation

- Intuition: reward honest sellers and penalize cheating sellers

MDP model:

- Solving the MDP: Deep Deterministic Policy Gradient (DDPG) + reward shaping
  - Shaped reward: \( R(M^t, w^t) = \frac{1}{n} \sum_{i=1}^{n} (r_{i+1} - \lambda f_{i+1}) \times \text{price}_{i+1} - \delta ||w^t||_2 \)
  - Avoid too large action (\( w^t \))
  - Make the reward signal continuous
  - Outperform all existing approaches using Alibaba’s real data
Improving Robustness of Fraud Detection Using Adversarial Examples [WWW’19]

- **Features** → **Classifier**
  - Fraud(0.95)
  - Normal(0.05)

- **New Features** → **Classifier**
  - Fraud(0.3)
  - Normal(0.7)

**Crafting Adversarial Examples**

- **DNN Model**
  - **Sampling**
    - **Mini Batch**
      - **Adversarial Attack**
        - **Adversarial Examples**
      - **Aggregate**
        - **New Mini Batch**
      - **One Gradient Descent Step**
Abstracted road network

An MDP formulation (state transition)

PG-β: scalable RL based on function approximation

Algorithm 1: PG-β

1. Initialize $\theta^0 \leftarrow \theta_0, \theta^t \leftarrow \theta_0, \forall t = 0, 1, \ldots, H - 1$;
2. repeat
3. Generate an episode $s^0, a^0, R^1, \ldots, s^{H-1}, a^{H-1}, R^H$;
4. for $t = 0, \ldots, H - 1$ do
5. $Q^t \leftarrow \sum_{i=1}^{H} Q^i$;
6. $\delta \leftarrow Q^t - \pi(s^t, a^t, \theta^t)$;
7. $\theta^t \leftarrow \theta^t + \beta \delta \nabla \pi(s^t, a^t, \theta^t)$;
8. $\theta^t \leftarrow \theta^t + \beta \gamma \nabla_{\theta^t} \log \pi(a^t | s^t, \theta^t)$;
9. until converge
10. return $\theta^t, \forall t = 0, 1, \ldots, H - 1$;
Scaling up Dynamic Electronic Road Pricing [IJCAI’19]

- Multi-agent reinforcement learning (MARL) for scaling up
- Edge-based graph convolutional network (GCN) for domain feature learning

Scaling up dynamic ERP

MARL Solution framework

**Algorithm 1: MARL-eGCN**

```
input: \( \theta_j^0 \in \mathbb{R}, j \in [1, ..., N] \) and \( \phi \)
for \( e \leftarrow 0 \) to MAX-EPOISODE do
  while \( t = 1 < \text{MAX-EPISODE-LENGTH} \) do
    for agent \( i \) in \( N \) do
      \( \rho_i^e = \mathbb{V}_i \times f(s^e, V); \)
      \( a_i^{e+1} = \pi(\rho_i^e, \theta_i^e); \)
      Concatenate \( a_i^{e+1}, i \in [1, ..., N] \) into \( a_e; \)
      Take \( s^e \) into traffic road graph and get \( s^{e+1}; \)
      foreach \( i \leftarrow 0 \) to \( N \) do Get \( r_i^e; \)
      for \( i \leftarrow 0 \) to \( N \) do
        \( y_i^e = \sum_{j=1}^{N} r_i^e; \)
        Update critic by minimizing the loss:
        \( \mathcal{L}(\phi) = y_i^e - Q(\rho_i^e, a_i^e); \)
        Update actor using the policy gradient:
        \( \nabla_{\theta_i^e} J = \nabla_{\theta_i^e} \pi(s_i^{e+1} | \rho_i^e) \cdot Q(\rho_i^e, a_i^{e+1}); \)
      endforeach
    endfor
    return \( \theta_i^e, i \in [1, ..., N]; \)
endwhile
endfor
```

An example of zone partitions

Edge-base GCN

Output for policy & value functions

```
X
0 1 0 1 0 1
1 0 1 0 1 1
```

adjacency matrix

feature matrix

MARL-eGCN: scaling up dynamic ERP
Learning Expensive Coordination: An Event-Based Deep RL Approach

- Large state space based expensive coordination
  - Considering the followers are self-interested
  - The leader should coordinate them by assigning incentives

- Event-based policy gradient
  - Modeling the leader’s strategy as events
  - A novel event-based policy is induced based on the events

- Action abstraction for followers
  - Accelerating the training process through action abstraction approach

![Diagram showing the state transitions and actions for followers](image)

(a) Resource Collections.  (b) Multi-Bonus Resource
(c) Navigation.  (d) Predator-Prey.
Challenge: high-level policy suffers the non-stationarity problem

- the constantly changing low-level policy leads to different state transitions and rewards of high-level policy

Key idea: the high-level policy makes decisions conditioned on the low-level policy representation

Solutions:
- Low-level policy modeling
- High-level policy stabilization via information-theoretic regularization
- Influence-based adaptive exploration with auxiliary rewards
Limited bandwidth in multi-agent communication:

Theorem: limited bandwidth constrains the message’s entropy

Key idea: compressing the communication messages

Information bottleneck principle

\[
\max_T I(T, Y) \text{ s.t. } I(X, T) \leq I_c \quad \rightarrow \quad \max_{M_i} J(\theta_i) := E_{\pi_i}[Q_i] \text{ s.t. } I(G_i, M_i) \leq I_c
\]

- lower bound for control entropy

\[
\tilde{J}(\theta_i) = E_{\pi_i}[Q_i] - \beta E \left[ D_{KL}[\pi_i^{pro}(m_i|g_i)||z(m_i)] \right]
\]

Architecture:
Learning Behaviors with Uncertain Human Feedback [UAI’20]

**Uncertain Human Feedback**
- Human may not give any feedback
- Positive feedback $\rightarrow$ sub-optimal action
- Negative feedback $\rightarrow$ optimal action

$$p(a, \lambda(s); \sigma, \mu, f) = \begin{cases} 
  p^+(a, \lambda(s); \sigma, \mu^+) & f = f^+ \\
  p^-(a, \lambda(s); \sigma, \mu^-) & f = f^- \\
  1 - p^+(a, \lambda(s); \sigma, \mu^+) - p^-(a, \lambda(s); \sigma, \mu^-) & f = f^0
\end{cases}$$

$p^+$ and $p^-$ are functions like Gauss function where $\lambda$ is the optimal action, $\mu$ and $\sigma$ are unknown parameters controlling mean and variance.

**Inferring Optimal Behavior**
- Maximizing likelihood estimation of receiving different kinds of feedback:
  $$\arg \max_{\lambda} P(h \mid \lambda; \mu, \sigma)$$
- EM + GD:
  - EM updates $\lambda$ with $\sigma$ fixed
  - $\mu$ is latent variable in integral
  - GD updates $\sigma$ with $\lambda$ fixed
  - $\mu$ is eliminated by a trick

![Rat catching](image1)

![Light controlling](image2)

![Step analysis](image3)

![Distance/state analysis](image4)
We Won Microsoft Collaborative AI Challenge
[AAAI’18]

- **Collaborative AI**
  - *How can AI agents learn to recognize someone’s intent (that is, what they are trying to achieve)?*
  - *How can AI agents learn what behaviors are helpful when working toward a common goal?*
  - *How can they coordinate or communicate with another agent to agree on a shared strategy for problem-solving?*

- **Microsoft Malmo Collaborative AI Challenge**
  - *Collaborative mini-game, based on an extension “stag hunt”*
  - *Uncertainty of pig movement*
  - *Unknown type of the other agent*
  - *Detection noise (frequency 25%)*
  - *Efficient learning*

- **Our team HogRider won the challenge** (out of more than 80 teams from 26 countries)
  - *learning + game theoretic reasoning + sequential decision making + optimization*
Recent AI breakthrough

AI for Complex Interaction

What’s next: AI for complex interaction
- Stochastic, open environment
- Multiple players
- Sequential decision, online
- Strategic (selfish) behavior
- Distributed optimization

MARL will play a very important role…