Michael Herrmann

University of Edinburgh, School of Informatics

17/03/2015
Overview

- MARL
- Stateless games
- Markov games
- Decentralised RL
Multi-agent learning is about cooperation or competition: The MDP model may not longer apply

- if goals are compatible, some degree of coordination may be required
- if goals are opposed, an optimal solution may no longer exist

Problems: limited information (including on the presence of other agents), complexity, non-stationarity

Applications: multi-robot systems, decentralised network routing, distributed load-balancing, traffic, finance, psychology and biology

Example: Job Scheduling

M. Brugnoli, E. Heymann, M.A. Saner: Grid scheduling based on collaborative random early detection strategies, in: 18th Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2010, pp. 35–42.
Parallel reinforcement learning: agent learn a single objective collaboratively, e.g.
  - MORL,
  - (hierarchically) divided state space
  - distributed exploration by an RL swarm

Problem: some agent may have outdated values
Solution: Use max in the learning rule assuming outdated values are smaller
This is essentially standard RL with non-standard exploration

MARL: individual goals and independent decision making capabilities
  - Nash equilibrium: no agent can improve its reward when the other agents retain a fixed policy
Benefits and/or Challenges in MARL

- Speed-up possible by parallelisation
- Experience sharing between agents by communication, mutual teaching or imitation learning
- Scalability: insertion of new agents, robustness vs. failure of some agents
- Exponential complexity in the number of agents
  - sparse interactions
  - experience sharing
- Exploration is as essential as ever, and/but may confuse other agents

Busoniu, L., Babuska, R., & De Schutter, B. (2008)
Stateless Games

- Reward depends on joint actions: \( r_k : A \rightarrow \mathbb{R} \)

- Zero-sum games or (e.g.) the prisoner’s dilemma

\[
\begin{array}{cc}
  a_1 & a_2 \\
  a_1 & (5, 5) & (0, 10) \\
  a_2 & (10, 0) & (1, 1)
\end{array}
\]

\((a_2, a_2)\) is a Nash equilibrium (which is not optimal)

- Best response for agent \( k \) if

\[
 r_k (a_{-k}, a_k) \geq r_k (a_{-k}, b_k) \quad \forall b \in A_k \quad \text{or more generally} \quad r_k (\pi_{-k}, \pi_k) \geq r_k (\pi_{-k}, \pi_k^*) \quad \forall \pi_k \in \Pi_k
\]

- In a Nash equilibrium all agents play their best response

- Stochastic policies can be optimal in MARL, e.g. “matching pennies”

\[
\begin{array}{cc}
  a_1 & a_2 \\
  a_1 & (1, -1) & (-1, 1) \\
  a_2 & (-1, 1) & (1, -1)
\end{array}
\]

best strategy is to choose any action with probability \( \frac{1}{2} \)
Example: Cooperation

Two agents, three actions each: avoid obstacle, but do not disrupt the formation

<table>
<thead>
<tr>
<th></th>
<th>L₁</th>
<th>L₂</th>
<th>S₁</th>
<th>S₂</th>
<th>R₁</th>
<th>R₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>L₁</td>
<td>10</td>
<td>-5</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S₁</td>
<td>-5</td>
<td>-10</td>
<td>-5</td>
<td>-5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R₁</td>
<td>-10</td>
<td>-5</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tie between L-L and R-R. Coordination necessary:

- “social conventions”: Agent 1 determines first, agent 2 observes and follows
- communication: agent arriving first tells the other agent (again tie may occur)

Agent 1 is heading for the goal (×) while avoiding capture by its opponent, agent 2.

Agent 2 aims at preventing agent 1 from reaching the goal, preferably by capturing it.

The agents can only move to the left or to the right.

Mini-max solution for agent one is to move left (or needs to find out what agent 2 is going to do)

Busoniu, L., Babuska, R., & De Schutter, B. (2008)
In MARL, agents do not have full access to the (stochastic) pay-off matrix, i.e. the game is unknown.

Actions of other agents are usually observable.

Goal of learning:
- Nash equilibria
- joint optimality
- evolutionary stable strategies
Adjust action probabilities according to

\[ p_i(t+1) = p_i(t) + \eta r(t)(1 - p_i(t)) \text{ if } a(t) = a_i \]
\[ p_j(t+1) = p_j(t) - \eta r(t)p_i(t) \text{ if } a(t) \neq a_i \]

\( p_i \) is the probability of playing action \( a_i \), \( \eta \) is a learning rate.

- Special case of REINFORCE (Williams, 1992)
- Properties (Sastry et al., 1994)
  - All Nash equilibria are stationary points.
  - All strict Nash equilibria are asymptotically stable.
  - All stationary points that are not Nash equilibria are unstable.
Markov Games

- Markov Games are for MAS like MPDs for single agents
- A Markov game is a tuple \((n, S, A_1, \ldots, A_n, R_1, \ldots, R_n, T)\) where
  - \(n\) is the number of agents
  - \(S = \{s_1, \ldots, s_N\}\) are the states
  - \(A_k\) are the actions of agent \(k\)
  - \(R_k : S \times A_1 \times \ldots \times A_n \times S \to \mathbb{R}\) are the rewards for agent \(k\)
  - \(T : S \times A_1 \times \ldots \times A_n \times S \to P(S)\) the action-dependent state transition function.

- Note that rewards and transitions depend on the other agents
- Task: Every agents maximises

\[
V_k^{\pi}(s) = \mathbb{E}^{\pi} \left[ \sum_{t=0}^{\infty} \gamma^t r_k(t+1) \mid s(0) = s \right]
\]

where \(r\) is a realisation of \(R\) and \(\pi = (\pi_1, \ldots, \pi_n)\) are the policies of all agents

See M. L. Littman, 1994
Markov Games: Value Iteration

- Learning with state transitions
- Combination of repeated games and MDPs
- Agent $k$ need to estimate $Q(s, a)$ for joint actions $a = (a_1, \ldots, a_n)$, not only $Q(s, a_k)$
- Action $a_k$ by agent $k$ is selected based on the observation of $a_{-k} = (a_1, \ldots, a_{k-1}, a_{k+1}, \ldots, a_n)$
- Stationary solutions may not exist: Mixed strategies, average reward schemes
MA $Q$-learning

- $t = 0$
- $Q_k(s, a) = 0 \ \forall s, a, k$
- repeat
  - for all agents $k$ do
    - select action $a_k(t)$
  - execute joint action $a = (a_1, ..., a_n)$
  - observe new state $s'$ and rewards $r_k$
  - for all agents $k$ do
    - $Q_k(s, a) = Q_k(s, a) + \eta (R_k(s, a) + \gamma V_k(s') - Q_k(s, a))$
  - until Termination Condition
How is the value $V_k(s')$ determined in MA $Q$-learning?

Various options

- **Opponent modelling** (Joint Action Learner by Claus & Boutilier, 1998)
  - count how often $(s, a_{-k})$ is played by other agents in state $s$
  - calculate frequency

\[
F(s, a_{-k}) = \frac{\#(s, a_{-k})}{\sum_{a_{-\ell}} \#(s, a_{-\ell})}
\]

- $V_k(s) = \max_{a_k} Q(s, a_k) = \max_{a_k} \sum_{a_{-k}} F(s, a_{-k}) Q(s, a)$

- Assume other agents trying to minimise your return

- Assume other agent will follow an equilibrium strategy

\[
V_k = \text{Nash}_k (s, Q_1, \ldots, Q_n), \text{ i.e. return at a Nash equilibrium}
\]

- Converges for self-play learning (convergence is often a problem in MAMG)

- Nash equilibrium is not unique in general
Interconnected Learning Automata for Markov Games (MG-ILA) by Vrancx et al. (2008)

Policy stored by learning automata: For each agent and state $LA(s,k)$

Update LAs by expected average reward

Algorithm converges to Nash equilibrium (if exists)
initialise $r_{\text{prev}}(s, k), t_{\text{prev}}(s), a_{\text{prev}}(s, k), t, r_{\text{tot}}(k), \rho_k(s, a), \tau_k(s, a) \to 0, \forall s, k, a, s \leftarrow s(0)$

loop
for all Agents $k$ do
if $s$ was visited before then
Calculate received reward and time passed since last visit to state $s$:
\[
\Delta r_k = r_{\text{tot}}(k) - r_{\text{prev}}(s, k), \quad \Delta t = t - t_{\text{prev}}(s)
\]
Update estimates for action $a_{\text{prev}}(s, k)$ taken on last visit to $s$:
\[
\rho_k(s, a_{\text{prev}}(s, k)) = \rho_k(s, a_{\text{prev}}(s, k)) + \Delta r_k
\]
\[
\tau_k(s, a_{\text{prev}}(s, k)) = \tau_k(s, a_{\text{prev}}(s, k)) + \Delta t
\]
$LA(s, k)$ uses $L_{R-I}$ update with $a(t) = a_{\text{prev}}(s, k)$ and av. reward
\[
\beta_k(t) = \rho_k(s, a_{\text{prev}}(s, k))/\tau_k(s, a_{\text{prev}}(s, k)).
\]
$LA(s, k)$ selects action $a_k$.
For current state store: $t_{\text{prev}}(s) \leftarrow t, r_{\text{prev}}(s, k) \leftarrow r_{\text{tot}}(k), a_{\text{prev}}(s, k) \leftarrow a_k$
Execute action $a = (a_1, \ldots, a_n)$, observe rewards $r_k$ and new state $s'$
$s \leftarrow s', r_{\text{tot}}(k) \leftarrow r_{\text{tot}}(k) + r_k, t \leftarrow t + 1$
• Reduce the joint-action space
• If there is no interaction, act independently
• Learn at what states where coordination is necessary, e.g. Sparse Tabular Multiagent $Q$-learning (Guestrin et al., 2002).
• Represent dependencies that are limited to a few agents, e.g. Sparse Cooperative $Q$-learning. (Kok and Vlassis, 2004, 2006)
• Similar to SCQ with learning of coordination graphs (Utile Coordination by Kok et al. (2005))
Initialise $Q_k$ through single agent learning and $Q^j_k$;

while true do

  if state $s_k$ of Agent $k$ is unmarked then
    Select $a_k$ for Agent $k$ from $Q_k$
  else
    if the joint state information $js$ is safe then
      Select $a_k$ for Agent $k$ from $Q_k$
    else
      Select $a_k$ for Agent $k$ from $Q^j_k$ based on the joint state information $js$

  Sample $\langle s_k, a_k, r_k \rangle$

  if t-test detects difference in observed rewards vs expected rewards for $\langle s_k, a_k \rangle$ then
    mark $s_k$
    for $\forall$ other state information present in the joint state $js$ do
      if t-test detects difference between independent state $s_k$ and joint state $js$ then
        add $js$ to $Q^j_k$
        mark $js$ as dangerous
      else
        mark $js$ as safe
    if $s_k$ is unmarked for Agent $k$ or $js$ is safe then
      No need to update $Q_k(s_k)$.
    else
      Update $Q^j_k(js, a_k) \leftarrow (1 - \alpha_t)Q^j_k(js, a_k) + \alpha_t[r(js, a_k) + \gamma \max_a Q(s'_k, a)]$
MARL algorithms currently limited to simple problems
Relations to game theory, evolution theory
Main point: Uncertainty, learning
Problems: Uncertainty about actions or rewards, delayed rewards, instabilities through careless exploration
DecPOMDP (Bernstein et al., 2002): Planning takes place in an off-line phase, after which the plans are executed in an on-line phase. This on-line phase is completely decentralized.


See also: http://umichrl.pbworks.com/w/page/7597585/Myths of Reinforcement Learning
Good and evil, reward and punishment, are the only motives to a rational creature: these are the spur and reins whereby all mankind are set on work, and guided.

Locke

The human organism is inherently active, and there is perhaps no place where this is more evident than in little children. They pick things up, shake them, smell them, taste them, throw them across the room, and keep asking, “What is this?” They are unendingly curious, and they want to see the effects of their actions. Children are intrinsically motivated to learn, to undertake challenges, and to solve problems. Adults are also intrinsically motivated to do a variety of things. They spend large amounts of time painting pictures, building furniture, playing sports, whittling wood, climbing mountains, and doing countless other things for which there are not obvious or appreciable external rewards. The rewards are inherent in the activity, and even though there may be secondary gains, the primary motivators are the spontaneous, internal experiences that accompany the behavior.

Deci and Ryan, 1985 (cf. A. Barto, 2013)