Integrated Learning for Goal-Driven Autonomy
Combining Learning Techniques with Case Based Reasoning

presented by
ULIT JAIDEE
The learner and decision maker is called the *agent*.

Everything outside the agent is called the *environment*.

The agent takes an action, then the environment responds to the action by returning a new state and a reward.
Goal-Driven Autonomy (GDA) is a process for online planning in autonomous agents.

Key concepts:

- **Expectation**: the expected next state after executing an action.
- **Discrepancy**: the mismatch between expected state and actual state.
- **Goal**: Activity to be performed.
Planning Model vs. GDA Model

Simple conceptual model for planning (Nau, D.S., 2007)

Conceptual model for GDA (Munoz-Avila, H., Jaidee, U. et al. FLAIRS-2010, ICCBR-2010, IJCAI-2011)
Inside The Controller of GDA

- **Discrepancy detector** detect unexpected events by comparing the actual current state obtained from executing previous action in previous state with the expectation.

- **Explanation generator** reveals the cause of discrepancy.

- **Goal formulator** resolves the discrepancies may require a change in the current goal.

- **Goal manager** generates a new goal and may require its immediate focus and/or removal of some existing goals.
LGDA

Learning GDA (LGDA) is a GDA algorithm that learns two types of cases:

1. *Expectation cases* maps (state, action) pairs to a distribution over expected states.

2. *Goal formulation cases* maps (goal, discrepancy) pairs to a distribution of expected values over resolution goals.

LGDA learns these through an integration of case-based reasoning (CBR) and reinforcement learning (RL) methods.
Case Bases in LGDA

- **Policies Case Base (Π)** is a collection of goal-policy pair \((g, π)\). Policies and goals in Π are given as input.

- **Expectation Case Base (ECB)** indicates for each state-action pair \((s, a)\) which is expected state \(x\) and its probability \(p\).

- **Goal Formulation Case Base (GFCB)**. Given current goal and discrepancy, GFCB give a distribution over the expected values for formulated goals.
Inside the Controller of LGDA

- **Discrepancy Detector** compares observations (states) with expectations.
- **Goal Formulator** resolves the discrepancies may require a change in the current goal.
- **Goal Manager** generates a new goal.
LGDA Learning Algorithm

\[
\begin{align*}
\text{LGDA}(s_0, g_0, d_0, \Delta, \text{ECB}, \text{GFCB}, \alpha, \gamma, \varepsilon, G, \Pi) \\
1: & \quad s \leftarrow s_0; \quad x \leftarrow s_0; \quad a \leftarrow \phi; \quad g \leftarrow g_0; \quad g' \leftarrow g_0; \quad d \leftarrow d_0 \\
2: & \quad \textbf{while} \text{ the game-playing episode continues} \\
3: & \quad \text{wait}(\Delta); \quad s' \leftarrow \text{GETSTATE}() \\
4: & \quad \text{ECB} \leftarrow \text{UPDATE}(\text{ECB}, s, a, s') \\
5: & \quad d \leftarrow \text{CALCULATEDISCREPANCY}(s', x) \\
6: & \quad q \leftarrow \text{GET}(\text{GFCB}, g, d, g') \\
7: & \quad r \leftarrow U(s') - U(s) \\
8: & \quad q' \leftarrow q + \alpha(r + \gamma \text{ArgMax}_{g'} \text{GET}(\text{GFCB}, g, d, gl \mathcal{F}_d - q) \\
9: & \quad \text{GFCB} \leftarrow \text{UPDATE}(\text{GFCB}, g, d, g', q') \\
10: & \quad \textbf{if} \ r < 0 \\
11: & \quad \quad \textbf{if} \ \text{RANDOM}(1) \geq \varepsilon \\
12: & \quad \quad \quad g'' \leftarrow \text{ArgMax}_{g'} \text{GET}(\text{GFCB}, g, d, gl \mathcal{F}_d) \\
13: & \quad \quad \textbf{else} \ g'' \leftarrow \text{RANDOM}(|G|) \\
14: & \quad \quad g \leftarrow g'; \ g' \leftarrow g'' \\
15: & \quad \quad \pi \leftarrow \Pi(g'); \ a' \leftarrow \pi(s') \\
16: & \quad x \leftarrow \text{GET}(\text{ECB}, s', a') \\
17: & \quad \text{EXECUTEACTION}(a') \\
18: & \quad s \leftarrow s'; \ a \leftarrow a' \\
19: & \quad \textbf{return} \ \text{ECB, GFCB}
\end{align*}
\]
Experiments and Results
What is a domination game?

Demo: Domination Game
DOM Game
# Adversaries in DOM

<table>
<thead>
<tr>
<th>Opponent Team</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dom1Hugger</td>
<td>Sends all bots to domination point 0.</td>
</tr>
<tr>
<td>1&lt;sup&gt;st&lt;/sup&gt;HalfOfDomPoints</td>
<td>Sends an bot to the first half +1 domination points. Extra bots patrol between the 2 points.</td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;HalfOfDomPoints</td>
<td>Sends an bot to the second half +1 domination points; extra bots patrol between the two points.</td>
</tr>
<tr>
<td>EachBotsToOneDom</td>
<td>Each bot is assigned to a different domination point And remains there for the entire game.</td>
</tr>
<tr>
<td>SmartOpportunistic</td>
<td>Sends bots to each domination point the team doesn’t own. If possible, it will send multiple bots to each un-owned point.</td>
</tr>
<tr>
<td>PriorityTeam</td>
<td>The strategy is trying to capture the highest priority DOM point. The unowned DOM points are the highest priority and then the ones belonged to enemy. The DOM points that own by our team are the lowest one.</td>
</tr>
</tbody>
</table>
Results of LGDA

Comparison between the CB-GDA, Retaliate, LGDA in different battles among the adversaries

<table>
<thead>
<tr>
<th>Adversary</th>
<th>CB-GDA</th>
<th>Retaliate</th>
<th>RGDA</th>
<th>LGDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dom1Hugger</td>
<td>77.38</td>
<td>74.26</td>
<td>71.36</td>
<td>61.38</td>
</tr>
<tr>
<td>FirstHalf</td>
<td>75.47</td>
<td>58.88</td>
<td>74.23</td>
<td>64.91</td>
</tr>
<tr>
<td>SecondHalf</td>
<td>65.36</td>
<td>65.79</td>
<td>66.28</td>
<td>63.03</td>
</tr>
<tr>
<td>SmartOp</td>
<td>54.85</td>
<td>-10.62</td>
<td>-36.59</td>
<td>45.27</td>
</tr>
<tr>
<td>EachBotToOne</td>
<td>0.46</td>
<td>-47.13</td>
<td>-68.48</td>
<td>-50.11</td>
</tr>
<tr>
<td>Priority</td>
<td>45.14</td>
<td>-6.37</td>
<td>-45.28</td>
<td>23.08</td>
</tr>
<tr>
<td>Learning method</td>
<td>None</td>
<td>RL</td>
<td>CBR</td>
<td>CBR+RL</td>
</tr>
</tbody>
</table>
Results of LGDA

- Average learning curves for comparing LGDA performance vs. non-learning and agents that use only RL or only CBR.
- The trend lines were generated using a polynomial fit to the raw curves.
Conclusions

- LGDA is the first GDA agent that automatically acquires state expectation and goal selection knowledge.
- Using a case-based reasoning method to map (state, action) pairs to a distribution over expected states, and (goal, discrepancy) pairs to a value distribution over discrepancy-resolution goals.
- Using a reinforcement learning method to learn the goals’ expected values.
- Under a complex first-person shooter gaming task, LGDA outperforms ablations that use only one of its two learning methods.
- LGDA also learns to play nearly as well as an expertly-engineered GDA agent.
Publications so far

- Applying Goal Driven Autonomy to a Team Shooter Game, *the 23rd International FLAIRS* (The Florida Artificial Intelligence Research Society) Conference (*FLAIRS*-2010)
- Goal-Driven Autonomy with Case-Based Reasoning, *the 18th International Conference on Case-based Reasoning* (*ICCBR* 2010)
- Integrated Learning for Goal-Driven Autonomy, *The International Joint Conference on Artificial Intelligence* (*IJCAI*) 2011
- Case-Based Learning in Goal-Driven Agents for Real-Time Strategy Combat Tasks. *Workshop on Case-Based Reasoning for Computer Games, 19th International Conference on Case Based Reasoning* (*ICCBR* 2011)