Option and Constraint Generation using Work Domain Analysis: Implementation for Reinforcement Learning Algorithm

Dr. Karen Feigh, Guliz Tokadli, Jesse Rosalia

The GT IML Team: Dr. Andrea Thomaz, Dr. Charles Isbel & Dr. Mark Reidl

NASA Langley’s Autonomy Incubator Seminar Series
7 November 2014
Talk Overview

❖ Project Overview: Interactive Machine Learning
❖ Background
   ✦ Reinforcement Learning
   ✦ Work Domain Analysis
   ✦ Micro-world of Pac-Man
❖ Application of WDA using Pac-Man
   ✦ Familiarization & Interview Phase
   ✦ Modeling Phase
   ✦ Option & Constraint Set Generation Phase
   ✦ Test & Evaluation Phase
❖ Conclusion
Interactive Machine Learning

❖ **Our Objective:** Robots and other machines that can learn from people unfamiliar with machine learning algorithms.
Reinforcement Learning

- A method to generate policies for an agent tasked with making decisions.
- Learning from action policy: $s \rightarrow a$
  - It maximizes the reward
Reinforcement Learning – Option & Constraint

**GOAL** is the ultimate purpose to gain.

**CONSTRAINTS** are defined as the set of all state action pairs that the agent should not do.
- Don’t do X.
- Don’t move onto unscared ghost.

**OPTIONS** are used for generalization of primitive actions to include temporally extended courses of action.

**PRIMITIVES** are the set of fundamental actions the agent can effect such as Up, Right, Left, Down.

---


Constraints

c : State x Action

❖ Corresponds to common negative advice forms:

“don’t do X”
“don’t climb on the counter”
“don’t move onto an non-scared ghost”
Options
Sutton, Precup & Singh, 1999

❖ Primitive actions are the set of fundamental actions the agent can effect: "Up", "Down", "Left" and "Right".

❖ In addition to primitive actions, make use of temporally extended actions, Options

❖ An option O: <l, π, β>
  ✤ l: *set of states* from which O can be initiated
  ✤ π(s,a): State x Action \( \rightarrow [0,1] \) (*Policy*)
  ✤ β: State \( \rightarrow [0,1] \); *termination condition*
Study Specifics

**Project Goal:** To **develop new methods** to exploit experienced humans’ knowledge to improve algorithms for machine learning

**Current practice:** Using programmer **derived or auto-derived primitives, options and constraints**, or inferring these from human coaching or demonstration

**Study Goal:** **Use work domain analysis techniques** from cognitive engineering to develop systematic method to generate options and constraints from experienced users

**How is our approach new:** Provides a **systematic way** to mine a human’s knowledge about a domain and **to translate** it to a hierarchical goal structure
Work Domain Analysis Method: Abstraction Hierarchy

- 5-level functional decomposition used for modelling complex sociotechnical systems.
- System is described at different levels of abstraction with “Why - How” relationship.

<table>
<thead>
<tr>
<th>Levels of AH</th>
<th>Relationship between levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Purposes</td>
<td>What</td>
</tr>
<tr>
<td>Abstract Functions</td>
<td>How</td>
</tr>
<tr>
<td>Generalized Functions</td>
<td>How</td>
</tr>
<tr>
<td>Physical Functions</td>
<td>How</td>
</tr>
<tr>
<td>Physical Objects</td>
<td>How</td>
</tr>
</tbody>
</table>

(4) N.A. Stanton, P.M. Salmon, G.H. Walker, C. Baber and D.P. Jenkins, Human Factors Methods, Ch. 4 Cognitive Task Analysis Methods, 2005.
## Work Domain Analysis Method: Abstraction Hierarchy

<table>
<thead>
<tr>
<th>Levels of AH</th>
<th>Meaning of Levels for Pac-Man</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Purposes (FP)</td>
<td>The goals of Pac-Man.</td>
<td>Staying Alive</td>
</tr>
<tr>
<td>Abstract Functions (AF)</td>
<td>What criteria is required to judge whether Pac-Man is achieving the purposes.</td>
<td>Avoid Ghosts</td>
</tr>
<tr>
<td>Generalized Functions (GF)</td>
<td>What functions are required to accomplish Pac-Man’s abstract functions.</td>
<td>Maneuver Pac-Man</td>
</tr>
<tr>
<td>Physical Functions (PF)</td>
<td>The limitation and capabilities of the system,</td>
<td>Distance to Nearest Ghost</td>
</tr>
<tr>
<td>Physical Objects (PO)</td>
<td>The objects and characters on the maze.</td>
<td>Ghosts, Pac-Man, Walls</td>
</tr>
</tbody>
</table>
Micro-World of Pac-Man

- Classic arcade game,
  - Invented in 1980
- Why we chose Pac-Man:
  - Helps to understand the research problem
  - Helps to build AH
    - Map linking the goals
    - Provide insight into RL policies
  - Used for many ML and RL studies
    - Allow the comparison of the results
    - Find what is needed quickly and conveniently.
Can we use work domain analysis to create options and constraints?
Pac-Man Study

**Familiarization Phase:** 16 participants played Pac-Man for 10 minutes

**Interview Phase:** researchers interviewed the participants using a structured interview script designed to generate abstraction hierarchies

**Modeling Phase:** researchers created abstraction hierarchies for each player and composite abstraction hierarchies for high and low performing players

**Algorithm Phase:** researchers translated composite abstraction hierarchies into sets of options and constraints

**Testing Phase:** researchers evaluated the option and constraint sets on three different board sizes
Familiarization Phase
Interview Phase
Modeling Phase
Algorithm Phase
Testing Phase
Participant Performance

High Performers

Low Performers
Modeling Phase - Procedure

1. Creation of individual AH per player

2. Harmonization of statements in AHs

3. Combination of AHs of high and low performers → Performance-based AH
WDA: Performance-Based AH

Aggregated AH: Low and High Performance AHs are represented as single AH.
## WDA: Performance-Based AH

<table>
<thead>
<tr>
<th>High Performers</th>
<th>Low Performers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player perspective is as ‘<strong>Competitor</strong>’ – Maximum score and minimum time</td>
<td>Player perspective is as ‘<strong>Exploratory spirit</strong>’ – Learning Tricks -</td>
</tr>
<tr>
<td>Having both <strong>defensive and offensive</strong> actions</td>
<td>Having only <strong>defensive</strong> actions</td>
</tr>
<tr>
<td>Having dual level of protection for Pac-Man as proactive and reactive strategy</td>
<td></td>
</tr>
<tr>
<td>‘Clearing current quadrant’ as action is using as <strong>tactic</strong></td>
<td>‘Clearing current quadrant’ is using as <strong>strategy</strong></td>
</tr>
<tr>
<td><strong>Quick observation</strong> to act in minimum time interval</td>
<td>Observation on quadrant is being used for ‘Learning Tricks’</td>
</tr>
</tbody>
</table>

**Familiarization and Interview Phase**

**Modelling Phase**

**Option & Constraint Set Generation Phase**
Familiarization Phase

Interview Phase

Modeling Phase

Algorithm Phase

Testing Phase
Generation of Option and Constraint Sets

- High Performer-Defined OC Set
- Low Performer-Defined OC Set
# High Performer-Defined OC Set Creation

## Familiarization and Interview Phase
- **Staying Alive (FP)**
- **Accomplishing Level (FP)**
- **Getting Highest Score (FP)**

## Modelling Phase

### Option Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Statement (AH Level)</th>
</tr>
</thead>
<tbody>
<tr>
<td>eatClosestFood</td>
<td>Eating all dots (AF)</td>
</tr>
<tr>
<td>eatClosestPowerPellet</td>
<td>Eating big dots (GF)</td>
</tr>
<tr>
<td>eatScaredGhost</td>
<td>Eating ghosts (AF)</td>
</tr>
<tr>
<td></td>
<td>Ghosts mode (PF)</td>
</tr>
<tr>
<td>goClosestQuadrant</td>
<td>Maneuver of Pac-Man (GF)</td>
</tr>
<tr>
<td></td>
<td>Moving efficiently (GF)</td>
</tr>
<tr>
<td></td>
<td>Dividing maze into quadrants (GF)</td>
</tr>
<tr>
<td></td>
<td>Area of where Pac-Man is moving (PF)</td>
</tr>
<tr>
<td></td>
<td>Location of ghosts (PF)</td>
</tr>
<tr>
<td></td>
<td>Distance from ghosts (PF)</td>
</tr>
<tr>
<td></td>
<td>Clearing current quadrant (PF)</td>
</tr>
</tbody>
</table>

### Constraint Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Statement (AH Level)</th>
</tr>
</thead>
<tbody>
<tr>
<td>avoidUnscaredGhost</td>
<td>Avoiding ghosts (AF)</td>
</tr>
<tr>
<td></td>
<td>Providing safety of Pac-Man (AF)</td>
</tr>
<tr>
<td></td>
<td>Maneuver of Pac-Man (GF)</td>
</tr>
<tr>
<td></td>
<td>Moving efficiently (GF)</td>
</tr>
<tr>
<td></td>
<td>Escaping via tunnel (GF)</td>
</tr>
<tr>
<td></td>
<td>Distance from ghosts (PF)</td>
</tr>
<tr>
<td>eatAlQuadrantFood</td>
<td>Number and location of dots (PF)</td>
</tr>
<tr>
<td></td>
<td>Clearing current quadrant (PF)</td>
</tr>
</tbody>
</table>

## Option & Constraint Set Generation Phase

- **Staying Alive**
- **Accomplishing level**
- **Getting Highest Score**

- **Eat Dots**
- **Eat Big Dot**
- **Eat Ghost**
- **Go Fast**
- **Eat Fruit**
- **Use Tunnel**
- **Clear Quadrant**

- **Avoid Ghost**
- **Maintain Perimeter**

- **Up**
- **Down**
- **Left**
- **Right**
Low Performer-Defined OC Set Creation

### Goal Name
- Staying Alive (FP)
- Accomplishing Level (FP)
- Using Past Experience (FP)

### Option Name
- **eatClosestFood**
  - Eating all dots (AF)
- **goClosestQuadrant**
  - Clearing current quadrant (AF)
  - Eating big dots (GF)
  - Maneuver of Pac-Man (GF)
  - Area of where Pac-man is moving (PF)
  - Distance from ghosts (PF)
- **runAwayFromGhost**

### Constraint Name
- **avoidUnscaredGhost**
  - Avoiding ghosts (AF)
- **avoidGhostQuadrant**
  - Avoiding ghosts (AF)
  - Ghost mode (PF)

---

**Familiarization and Interview Phase**

**Modelling Phase**

**Option & Constraint Set Generation Phase**
# High Performer-Defined Option & Constraint Set

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>eatClosestFood</td>
<td>Eat the closest food Pac-Man can reach</td>
</tr>
<tr>
<td>eatClosestPowerPellet</td>
<td>Eat the closest power pellet Pac-Man can reach</td>
</tr>
<tr>
<td>eatScaredGhost</td>
<td>Eat the closest scared ghosts Pac-Man can reach, if there exists a scared ghost in the game.</td>
</tr>
<tr>
<td>goClosestQuadrant</td>
<td>Take the shortest path to the nearest position that leaves the current quadrant</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>avoidUnscaredGhost</td>
<td>Do not allow actions that could result in walking into an unscared ghost</td>
</tr>
<tr>
<td>eatAllQuadrantFood</td>
<td>Do not allow actions that would leave quadrant while it still contains reachable food (food that can be reached by walking through the current quadrant)</td>
</tr>
</tbody>
</table>
## Low Performer-Defined Option & Constraint Set

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>eatClosestFood</td>
<td>Eat the closest food Pac-Man can reach</td>
</tr>
<tr>
<td>goClosestQuadrant</td>
<td>Take the shortest path to the nearest position that leaves the current quadrant</td>
</tr>
<tr>
<td>runAwayFromGhost</td>
<td>Move to maximize distance from all ghosts</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>avoidUnscaredGhost</td>
<td>Do not allow actions that could result in walking into an unscared ghost</td>
</tr>
<tr>
<td>avoidGhostQuadrant</td>
<td>Do not allow actions that would enter or remain in quadrants containing an unscared ghost</td>
</tr>
</tbody>
</table>
# Researcher Defined Option & Constraint Set

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>eatClosestFood</td>
<td>Eat the closest food Pac-Man can reach</td>
</tr>
<tr>
<td>eatClosestPowerPellet</td>
<td>Eat the closest power pellet Pac-Man can reach</td>
</tr>
<tr>
<td>eatClosestGhost</td>
<td>Attempt to eat the closest unscared or scared ghost</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>avoidUnscaredGhost</td>
<td>Do not allow Pac-Man to occupy positions that are 2 or fewer squares from an unscared ghost</td>
</tr>
</tbody>
</table>
Familiarization Phase

Interview Phase

Modeling Phase

Algorithm Phase

Testing Phase
Simulation Setup

- Implemented Policies for each Option & Constraint Set
- Q-learning with constrained options from Irani et al.

Experiments
- 200 independent trials
- 10,000 training episodes
- Scores were averaged over all 200 trials
- Discount of 0.9, epsilon of 0.4 → 0 over episodes
Boards

- Small Board
- Original Board
- Medium Board
Results of Option-Constraint Set Testing – Small Board

Performance (High to Low):
High performers > Researcher defined agent > Low performers
Results of Option-Constraint Set Testing – Medium Board

Performance (High to Low):
High performers > Low performers > Researcher defined agent

Cumulative Number of Learning Episodes

High Performer OC
Researcher OC
Primitive
Results of Option-Constraint Set Testing – Original Board

Cumulative Number of Learning Episodes

Reward

Researchers OC

High Performer OC

Low Performer OC

Primitive
What happened with the large board?

The way we operationalized our options and constraints overly constrained it. Needed to investigate what was happening.
Differences for High Performers – Original Board

High Performer OC - exception
High Performer OC - quadrant
High Performer OC
Results of Option-Constraint Set Testing – Original Board

Performance (High to Low):
High performers > Researcher defined agent > Low performers
Conclusion
Conclusion & Future Directions

- Derived option and constraint sets for low and high performers with the process of Work Domain Analysis
- Implemented option and constraint sets separately on RL algorithms and evaluation of the performers – able to show the differences between performers
- Will work toward automating more of the process to auto-generation of option and constraint sets
Any Questions?

Special Thanks to
The IML Team at GT
Dr. Andrea Thomaz
Dr. Charles Isbel
Dr. Mark Riedl