CSE 3521 Notes

Brendan Whitaker

AU17

Lecture: Friday, August 25th

Office Hours: 3:30 - 4:30 Monday. 1-2 Wednesday.

Homework Assignment: Attend at least one of the AI Seminar Series talks. Write 1-2 paragraphs on what the speaker was trying to approach and were they successful. Remember that homework 1 is due on **8/30**.

AI is making robots that think rationally and act rationally.

Fundamental Question: **How do you turn a real-world problem into a problem that you can solve with AI?**
Much of AI is concerned with **agents** that are acting on their **environment**.

Performance - measuring desired outcomes
Environment - what populates the task's world?
Actuators - what can the agent act with?
Sensors - how can the agent perceive the world?

Arnold's PEAS - kill John Connor; people, weapons, vehicles; hands, feet, muscles; thermal imaging, HUD.

Automated Taxi PEAS:

- Performance - Safe, fast legal, comfortable trip, maximize profits.
- Environment - Roads, other traffic, pedestrians, customers
- Actuators - Steering, accelerator, brake, signals, horn, display
- Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, microphone/keyboard

Agent - an entity that perceives its environment through sensors, and acts on it with actuators.
percepts - are constrained by sensors and environment

actions - constrained by actuators and environment
A rational **agent** always acts to **maximize its expected performance measure**, given current state/percept.

Pacman

Percepts - squares around Pacman.
Actions - move U/D/L/R.
Environment - map with walls, dots, and ghosts.
Model - which squares have I eaten dots in?

Spam detector

Percepts - sender, subject line, body of current email
Actions - mark Spam/not Spam
Environment - your email inbox
Model - per-sender message history

Agent function:
if haveExchangedEMails(sender): NOTSPAM
if hasNigerianPrince(body): return SPAM
else: return NOTSPAM

Reflex agents:

- Choose action based on current percept (and maybe memory)
-

Spam detector is a good example of this. (Maybe some info about the past)

Goal-based agents

Choose action(sequence) to get from current state to some **goal** with **maximum utility along the way**.

Pacman

- Percepts - squares around Pacman.
- Actions - move U/D/L/R.
- Environment - map with walls, dots, and ghosts.
- Goal - Eat all the dots in as short a path as possible

Note that Spam Detector is not a goal based agent because it doesn't have some overarching goal that extends beyond each immediate task (email).

Types of agents

Reflex agents - act on current state (and maybe past). Simple - current precepts only. Model - current precept and rest of the
Goal - based agents - from current state to desired future

AI - agents and environments

Two different types of environments: PEAS environment (world). The **task environment** is all of PEAS.

6 common properties to distinguish tasks (not exhaustive):

1. Fully observable vs. partially observable

Fully observable - agent is able to sense everything in the environment.
Partially observable - noisy, inaccurate, or incomplete sensors.

2. Single agent vs. multiagent

Single agent - only one dude has performance measure.
Multiagent - task involves more than one agent, each with its own performance measure. (May be competitive or cooperative) (measures align or oppose)

3. Deterministic vs. stochastic

Deterministic - next state of the world is fully determined by the current state and agent action.
Stochastic - not deterministic.

4. Episodic vs. sequential

Episodic - Each step/decision is independent **Sequential** - missed it.

5. Static vs. dynamic

Static - world doesn't change while agent choosing action.
Dynamic - decision time matters!

6. Discrete vs. continuous

Self explanatory.

Bonus: level of agent knowledge

Environment is **known** - agent knows the rules of the world.
Environment is **unknown** - agent has partial knowledge of the world.

Help determine how we approach problems

Static ⇒ can focus on getting really high accuracy/utility
Dynamic ⇒ trade some utility for higher efficiency (speed!)

Episodic ⇒ reflex agent with a great model
Sequential ⇒ need a goal-oriented agent

Homework 3

The approximate Q-function takes the following form

$$Q(s,a) = \sum_{i=1}^{n} f_i(s,a) w_i,$$

where each weight w_i is associated with a particular feature $f_i(s,a)$. In your code, you should implement the weight vector as a dictionary mapping features (which the feature extractors will return) to weight values. You will update your weight vectors similarly to how you updated Q-values

$$w_i \leftarrow w_i + \alpha \cdot \text{difference} \cdot f_i(s,a),$$

$$\text{difference} = (r + \gamma \max_{a'} Q(s',a')) - Q(s,a).$$

Note that the difference term is the same as in normal Q-learning, and r is the experienced reward.

Extra Credit: AI coined in 1956 in Dartmouth.

MARCUS Deep Learning Overview
What are systematic compositional skills.