PF: Agent-based AI

CSC872
Pattern Analysis and Machine Intelligence

What is AI for?
### AI Prehistory

<table>
<thead>
<tr>
<th>Area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Philosophy</td>
<td>Logic, methods of reasoning, mind as physical system, foundations of learning, language, rationality</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Formal representation and proof algorithms, computation, (un)decidability, (in)tractability, probability</td>
</tr>
<tr>
<td>Economics</td>
<td>Utility, decision theory</td>
</tr>
<tr>
<td>Neuroscience</td>
<td>Physical substrate for mental activity</td>
</tr>
<tr>
<td>Psychology</td>
<td>Phenomena of perception and motor control, experimental techniques</td>
</tr>
<tr>
<td>Computer engineering</td>
<td>Building fast computers</td>
</tr>
<tr>
<td>Control theory</td>
<td>Design systems that maximize an objective function over time</td>
</tr>
<tr>
<td>Linguistics</td>
<td>Knowledge representation, grammar</td>
</tr>
</tbody>
</table>

### AI History

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1943</td>
<td>McCulloch &amp; Pitts: Boolean circuit model of brain (Began with Neural Networks!)</td>
</tr>
<tr>
<td>1950</td>
<td>Turing's &quot;Computing Machinery and Intelligence&quot;</td>
</tr>
<tr>
<td>1956</td>
<td>Dartmouth meeting: &quot;Artificial Intelligence&quot; adopted</td>
</tr>
<tr>
<td>1950s</td>
<td>Early AI programs, including Samuel's checkers program, Newell &amp; Simon's Logic Theorist, Gelernter's Geometry Engine</td>
</tr>
<tr>
<td>1960s</td>
<td>Robinson's complete algorithm for logical reasoning, AI discovers computational complexity, Neural network research almost disappears</td>
</tr>
<tr>
<td>1970s</td>
<td>First AI Boom (Inference and Search)</td>
</tr>
<tr>
<td>1980s</td>
<td>Second AI Boom (Expert Systems, Neuro networks)</td>
</tr>
<tr>
<td>1980s</td>
<td>AI becomes an industry</td>
</tr>
<tr>
<td>1986</td>
<td>Neural networks return to popularity (Back Propagation)</td>
</tr>
<tr>
<td>1990s</td>
<td>AI becomes a science</td>
</tr>
<tr>
<td>1995</td>
<td>The emergence of intelligent agents</td>
</tr>
<tr>
<td>2000s</td>
<td>Neural network research disappears again</td>
</tr>
<tr>
<td>2012--</td>
<td>Third AI Boom (Deep Neural networks)</td>
</tr>
</tbody>
</table>
**AI: What have been achieved**

- Apple’s **Siri** and Amazon’s **Alexa**
- **Google Translation**
- **IBM’s Watson**
- Proved a mathematical conjecture (**Robbins conjecture**) unsolved for decades (McCune, 1997)
- **DARPA Grand Challenge** (2005): 130 miles of mixed terrain by 7 hours by a Stanford team → **Self-Driving**
- **Deep Blue** defeated the reigning world chess champion Garry Kasparov in 1997
- **AlphaGo** by DeepMind defeated top-ranked GO players in 2015-17
- **Deep-NeuralNets** have beaten humans in image scene understanding (2015) [https://ai100.stanford.edu/](https://ai100.stanford.edu/)

**Trends**

- **6-70s’**: AI
- **80s’**: Knowledge Based, Expert Systems
- **90s’**: Neural Net, Genetic Algo, Distributed AI
- **00s’**: Agents, Evolutionary systems,
- **10s’**: Bayes, Markov, Nash, ML,
- **12~**: Deep Learning
Basic Components of AI

- Environment
- Perception
- Reasoning
- Action

Four Definitions

- The exciting new effort to make computers think ... machine with minds, in the full and literal sense” (Haugeland 1985)
- “The study of mental faculties through the use of computational models” (Charniak et al. 1985)
- “The art of creating machines that perform functions that require intelligence when performed by people” (Kurzweil, 1990)
- A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes” (Schalkol, 1990)
Four Ways to See it

Views of AI fall into four categories:

<table>
<thead>
<tr>
<th>Thinking humanly</th>
<th>Thinking rationally</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acting humanly</td>
<td>Acting rationally</td>
</tr>
</tbody>
</table>

The textbook advocates "acting rationally"
- algorithmic
- computational
- quantitative

Different Approaches

**theoretical vs experimental (basic or applied)**

<table>
<thead>
<tr>
<th>Thinking humanly</th>
<th>Thinking rationally</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acting humanly</td>
<td>Acting rationally</td>
</tr>
</tbody>
</table>

**Biological**: based on the idea that since humans are intelligent, AI should study humans and imitate their psychology or physiology.

**Phenomenal**: based on studying and formalizing common sense facts about the world and the problems that the world presents to the achievement of goals.
Act Humanly: The Turing Test

- Alan Turing's 1950 article: *Computing Machinery and Intelligence*
  
  - “Can machines think?” \(\iff\) “Can machines behave intelligently?”
  
  - Turing test (The Imitation Game): Operational definition of intelligence.
  
  - By 2000, 30% chance of fooling a lay person for 5 minutes by machines

• Problems:
  
  - Not reproducible, constructive, and amenable to mathematic analysis.
  
  - What about physical interaction with interrogator and environment?

AI-Complete

• Suggested AI Components by Turing
  
  - Language Understanding: to communicate with tester
  
  - Knowledge Representation: to store and retrieve information
  
  - Automated Reasoning: to answer questions and draw new conclusions
  
  - Machine Learning: to adapt to new circumstances

• Total Turing Test: requires physical interaction and needs perception and actuation.
  
  - Computer Vision: to recognize the tester’s actions and various objects represented by the tester
  
  - Other Senses: e.g., audition, smell, touch etc
  
  - Motor Control: to act upon objects as requested

• AI-Complete (a la NP-Complete)
Think Humanly: Cognitive Science

- 1960 “Cognitive Revolution”: information-processing psychology replaced behaviorism
- Cognitive science brings together theories and experimental evidence to model internal activities of the brain
  - What level of abstraction? “Knowledge” or “Circuits”?
  - How to validate models?
    - Predicting and testing behavior of human subjects (top-down)
    - Direct identification from neurological data (bottom-up)
- Both approaches (roughly, Cognitive Science and Cognitive Neuroscience) was distinct from AI for long time until the advent of Deep Learning

Think Rationally: Laws of Thoughts

- Aristotle (~ 450 B.C.) attempted to codify “right thinking”
  “What are correct arguments/thought processes?”
  - “Socrates is a man, all men are mortal; therefore Socrates is mortal”
- Several Greek schools developed various forms of logic:
  notation plus rules of derivation for right thoughts.
- Direct line through mathematics and philosophy to modern AI
- Problems:
  1) Uncertainty: Not all facts are certain (e.g., the flight might be delayed).
  2) Not all intelligent behavior is mediated by logical deliberation
  3) Meta: What is the purpose of thinking? What thoughts should I have?
  4) Resource limitations:
     - Not enough time to compute/process
     - Insufficient memory/disk/etc
Act Rationally: Agents (PF)

- Agents: can be human, robots, softbots, thermostats
- The *agent function* maps percept sequence to actions
  \[ f : P^* \rightarrow A^t \]
  - agent = architecture + program
- The *agent program* runs on the physical architecture to produce \( f \)

The Vacuum-Cleaner World

- Environment: square A and B
- Percepts: [location and content] e.g. [A, Dirty]
- Actions: left, right, suck, and no-op
The Vacuum-Cleaner World

<table>
<thead>
<tr>
<th>Percept sequence</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A, Clean]</td>
<td>Right</td>
</tr>
<tr>
<td>[A, Dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>[B, Clean]</td>
<td>Left</td>
</tr>
<tr>
<td>[B, Dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>[A, Clean], [A, Clean]</td>
<td>Right</td>
</tr>
<tr>
<td>[A, Clean], [A, Dirty]</td>
<td>Suck</td>
</tr>
</tbody>
</table>

Rational Agents

- **A rational agent** is one that does the **right thing**.
  - Every entry in the table is filled out correctly.
  - Doesn’t necessary involve thinking but it should be in the service

- What is the right thing?
  - Computational limitation makes perfect rationality unachievable
  - Approximation: the most **successful** agent.
  - **Measure of success**?

- Performance measure should be objective
  - e.g. the amount of dirt cleaned within a certain time.
  - e.g. how clean the floor is.
  - ...
Rational Agents

- **DEF:** A rational agent chooses whichever action that maximizes the expected value of the performance measure given the percept sequence to date and prior environment knowledge.

- **Rationality ≠ Omniscience**
  - An omniscient agent knows the actual outcome of its actions.

- **Rationality ≠ Perfection**
  - Rationality maximizes expected performance, while perfection maximizes actual performance.

Rational Agents

- The proposed definition requires:
  - **Information gathering/exploration:** act to modify future percept to obtain useful information
  - **Learn from percepts:** extend prior knowledge of the environment
  - **Autonomy:** act based on its own experiences
PEAS: Describing environment

- To design a rational agent we must specify its task environment.

- **PEAS** description of the task environment:
  - Performance measure
  - Environment (Physical/Virtual)
  - Actuators
  - Sensors

- PEAS = KR + PF

### PEAS: Automated Taxi Driver

- Consider, e.g., the task of designing an automated taxi driver:

  - **Performance measure:**
    - Safe, fast, legal, comfortable trip, maximize profits

  - **Environment:**
    - Roads, other traffic, pedestrians, customer

  - **Actuators:**
    - Steering wheel, accelerator, brake, signal, horn

  - **Sensors:**
    - Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard, microphone
PEAS: Medical Diagnosis System

- **Agent**: Medical diagnosis system
- **Performance measure**:  
  - Patient’s health, minimize costs, malpractice, & lawsuits
- **Environment**:  
  - Patient, hospital, staff
- **Actuators**:  
  - Screen display (questions, tests, diagnoses, treatments, referrals), speakers
- **Sensors**:  
  - Keyboard (entry of symptoms, findings, patient’s answers), microphone, video

PEAS: Combat Game AI

- **Agent**: Combat AI: autonomous players
- **Performance measure**:  
  - Reality of actions, Strength to survive, Player’s satisfaction
- **Environment**:  
  - Game’s virtual dungeon, protectors and weapons of choice
- **Actuators**:  
  - Screen Display (engaging on a combat with you), speakers
- **Sensors**:  
  - Game console, keyboard, microphone etc
### Environmental Types

- **Fully observable** (vs. **partially observable**): An agent’s sensors give it access to the complete state of the environment at each point in time.

- **Deterministic** (vs. **stochastic**): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is **strategic**)

- **Episodic** (vs. **sequential**): The agent’s experience is divided into “atomic episodes” (each episode consists of the agent perceiving and then performing a single action), and the next episode does not depend on actions taken in previous episodes.

### Environmental Types

- **Static** (vs. **dynamic**): The environment is unchanged while an agent is deliberating

- **Discrete** (vs. **continuous**): A limited number of distinct, clearly defined percepts and actions.

- **Single agent** (vs. **multiagent**): An agent operating by itself in an environment.
**Environmental Types**

<table>
<thead>
<tr>
<th></th>
<th>Solitaire</th>
<th>Chess</th>
<th>Taxi driving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observable</td>
<td>Full</td>
<td>Full</td>
<td>Partial</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Partially</td>
<td>Strategic</td>
<td>Stochastic</td>
</tr>
<tr>
<td>Episodic</td>
<td>Sequential</td>
<td>Sequential</td>
<td>Sequential</td>
</tr>
<tr>
<td>Static</td>
<td>Static</td>
<td>Static</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Discrete</td>
<td>Discrete</td>
<td>Discrete</td>
<td>Continuous</td>
</tr>
<tr>
<td>Single agent</td>
<td>Multi</td>
<td>Multi</td>
<td>Multi</td>
</tr>
</tbody>
</table>

- The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

**Agent Types**

- All agents have the same skeleton:
  - Input = current percepts
  - Output = action
  - Program= manipulates input to produce output

- Four basic kind of agent programs will be discussed:
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents

- All these can be turned into learning agents.
The Vacuum-Cleaner World

![Function table diagram]

<table>
<thead>
<tr>
<th>Percept sequence</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A, Clean]</td>
<td>Right</td>
</tr>
<tr>
<td>[A, Dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>[B, Clean]</td>
<td>Left</td>
</tr>
<tr>
<td>[B, Dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>[A, Clean], [A, Clean]</td>
<td>Right</td>
</tr>
<tr>
<td>[A, Clean], [A, Dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table-Lookup Agent

1) Create and maintain a table of actions indexed by percept sequences
2) Append a new percept to the table
3) Find action by table lookup

• Drawbacks:
  – Huge table
  – Take a long time to build the table
  – No autonomy
  – Even with learning, need a long time to learn the table entries
The Vacuum-Cleaner World

function REFLEX-VACUUM-AGENT ([location, status])
    returns an action

    if status == Dirty then return Suck
    else if location == A then return Right
    else if location == B then return Left

Much smaller than storing the entire table!!!

Simple Reflex Agents

- Select action on the basis of only the current percept.
- Large reduction in possible percept/action situations.
- Implemented through condition-action rules
Simple Reflex Agents

function SIMPLE-REFLEX-AGENT(percept) returns an action

static: rules, a set of condition-action rules

rule ← RULE-MATCH(percept, rules)
action ← RULE-ACTION[rule]
return action

Model-based Reflex Agents

• To tackle partially observable environments.
  – Maintain internal state

• Update state using world knowledge over time
  – How does the world change?
  – How do actions affect world?
  ⇒ Model of World
Model-based Reflex Agents

function REFLEX-AGENT-WITH-STATE(percept) returns an action

static: rules, a set of condition-action rules
state: a description of the current world state
action: the most recent action.

state ← UPDATE-STATE(state, action, percept)
rule ← RULE-MATCH(state, rules)
action ← RULE-ACTION[rule]
return action

Goal-based Agents

- The agent needs a goal to know which situations are desirable.
  - Things become difficult when long sequences of actions are required to find the goal.
- Typically investigated in search and planning research.
- Major difference: future is taken into account
- Is more flexible since decision knowledge is represented explicitly and can be manipulated.
Utility-based Agents

- Certain goals can be reached in different ways.
  - Some are better, have a higher utility.
- Utility function maps states onto a real-valued number of “happiness”.
- Improves on goals:
  - Selecting between conflicting solutions finding a tradeoff
  - Select appropriately between several goals based on likelihood of success.

Learning Agents

- Learning element: introduce improvements in performance element.
  - Critic provides feedback on agents performance based on fixed performance standard.
- Performance element: selecting external actions based on percepts.
  - Corresponds to the previous agent programs
- Problem generator: suggests actions that will lead to new and informative experiences.
  - Exploration like scientists do

- Learn the programs to produce desirable actions
- Teach them instead of instructing them
- Robust toward initially unknown environments
### Summary

- **Overview**
  - AI History
  - Four Approaches
  - Rational Agents
  - PEAS
  - Environmental Types
  - Agent Types → Next few lectures will see specific implementations.

- **MATLAB exercise #1 after a break**

- **Next Lecture**
  - Search Agents: Goal-based and Utility-based Agents!
  - MATLAB exercise #2