Week 1

20.09.2016

Tuesday 14:00 – 17:00, BM A3

Emre Ugur, BM 33
emre.ugur@boun.edu.tr
http://www.cmpe.boun.edu.tr/~emre/courses/CMPE540/
cmpe540@listeci.cmpe.boun.edu.tr
Emre Ugur
CMPE540 – Week 1
Spring 2017
October 2016
Responsibilities

- Quizzes (10): 15%
- Midterms (2): 25%
- Final: 30%
- Projects (4): 30%
## Syllabus (subject to change)

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Chapters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sep 20</td>
<td>Agents &amp; Uninformed search</td>
<td>Ch 2, 3.1-4</td>
</tr>
<tr>
<td>Sep 27</td>
<td>A* search, heuristics Local search; search-based agents</td>
<td>Ch. 3.5-6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 4</td>
</tr>
<tr>
<td>Oct 1</td>
<td>Game playing Constraint satisfaction problems</td>
<td>Ch. 5.1-5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 6.1, 6.3-5</td>
</tr>
<tr>
<td>Oct 4</td>
<td>No class</td>
<td></td>
</tr>
<tr>
<td>Oct 11</td>
<td>Midterm 1</td>
<td></td>
</tr>
<tr>
<td>Oct 18</td>
<td>Propositional logic: semantics and inference Propositional planning and logical agents</td>
<td>Ch. 7.1-4, 7.6.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 7.7</td>
</tr>
<tr>
<td>Oct 25</td>
<td>First-order logic</td>
<td>Ch. 8.1-3, 9.1</td>
</tr>
<tr>
<td>Nov 1</td>
<td>Probability Bayes nets: Syntax and semantics</td>
<td>Ch. 13.1-5 Ch. 14.1-3</td>
</tr>
<tr>
<td>Nov 8</td>
<td>Bayes nets: Exact inference Bayes nets: Approximate inference</td>
<td>Ch. 14.3 Ch. 14.4</td>
</tr>
<tr>
<td>Nov 15</td>
<td>Midterm 2</td>
<td></td>
</tr>
<tr>
<td>Nov 22</td>
<td>Markov Models, Hidden Markov Models Applications of HMMs</td>
<td>Ch. 15.1-3, 15.5 Ch. 22.1, 23.5</td>
</tr>
<tr>
<td>Nov 29</td>
<td>Decision theory Markov decision processes</td>
<td>Ch. 16.1-3, 16.5-6 Ch. 17.1</td>
</tr>
<tr>
<td>Dec 6</td>
<td>Machine learning: Classification and regression</td>
<td>Ch. 18.1-4, 18.6</td>
</tr>
<tr>
<td>Dec 13</td>
<td>Advanced Topics: Vision and robotics</td>
<td>Ch. 24, 25</td>
</tr>
</tbody>
</table>
Projects – programming + report
(and background reading)

- Programming Language: C / C++
- Programming environment: Linux
- Automatically graded (as much as possible)

- Will be announced at least 2 weeks before due date
- Late submission penalty:
  - (10 x days) up-to 3 days.

- Topics:
  - Search
  - CSP, Multi-agent, Bayesian Nets, Classification or Reinforcement Learning
Quizzes

- In every lecture
- Random time
- Mostly from the current lecture – but might be from paper readings or the previous lecture
  - Pseudo-code
  - Problem solving
  - Algorithm application
  - Explanation, definition, etc.
- Time-limit is important!

- Keep the slides with you during lectures.
Sources & credits

▶ Course notes:
  ▶ Russell and Norvig:
    ▶ http://aima.cs.berkeley.edu/
  ▶ Levent Akin, Pinar Yolum and Albert Ali Salah
    ▶ http://www.cmpe.boun.edu.tr/~akin/

▶ Projects and content:
  ▶ UC Berkeley CS188 Intro to AI
    ▶ https://inst.eecs.berkeley.edu/~cs188/fa11/assignments.html
Chapter 1:
Introduction
What is AI?

- The study of creating intelligent agents
## What is AI?

Views of AI fall into four categories (Humanly vs. Rationally)

- Acting rationally, Thinking humanly, Acting humanly, Thinking rationally,

```
```

- The exciting new effort to make computers think ... machines with minds, in the full and literal sense" (Haugeland, 1985)

- The automation of activities that we associate with human thinking, activities such as decision-making, problem solving, learning ..." (Bellman, 1978)

- The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990)

- The study of how to make computers do things at which, at the moment, people are better" (Rich and Knight, 1991)

- The study of mental faculties through the use of computational models" (Charniak and McDermott, 1985)

- The study of the computations that make it possible to perceive, reason, and act" (Winston, 1992)

- A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" (Schalkoff, 1990)

- AI ... is concerned with intelligent behavior in artifacts.” (Nillson, 1998)
Acting humanly: Turing Test

- Turing (1950) "Computing machinery and intelligence":
  - "Can machines think?" "Can machines behave intelligently?"
- Operational test for intelligent behavior: the Imitation Game
- Predicted that by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes
- Anticipated all major arguments against AI in following 50 years
- Suggested major components of AI: knowledge, reasoning, language understanding, learning
Thinking humanly: cognitive modeling

- 1960s "cognitive revolution": information processing psychology
  - Requires scientific theories of internal activities of the brain
- How to validate? Requires
  - 1) Predicting and testing behavior of human subjects (top-down)
  - or 2) Direct identification from neurological data (bottom-up)
- Both approaches (roughly, Cognitive Science and Cognitive Neuroscience) are now distinct from AI
Acting rationally: rational agent

- Rational behavior: doing the right thing
  - The right thing: that which is expected to maximize expected goal achievement or outcome, given the available information
- Doesn't necessarily involve thinking – e.g., blinking reflex – but thinking should be in the service of rational action
Rational agents

- An agent is an entity that perceives and acts
- This course is about designing rational agents
- Abstractly, an agent is a function from percept histories to actions:
  \[ f: P^* \rightarrow A \]
- For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance
- Caveat: computational limitations make perfect rationality unachievable
  - Design best program for given machine resources
State-of-the-art

- Play a decent game of table tennis
- Drive safely along a curving mountain road
- Drive safely along Minibüs Caddesi
- Buy a week’s worth of groceries on the web
- Play a decent game of bridge
- Discover and prove a new mathematical theorem
- Design and execute a research program in molecular biology
- Write an intentionally funny story
- Give competent legal advice in a specialized area of law
- Translate spoken English into spoken Swedish in real time
- Converse successfully with another person for an hour
- Perform a complex surgical operation
- Clean the floors without further guidance
- Unload any dishwasher and put everything away

Real-world interactions
Creativity
Quiz 1

Some sub-fields of AI are / have been receiving less attention these days. Why?

Very short answer.
Chapter 2: Intelligent Agents
Overview

- PEAS (Performance, Environment, Actuators, Sensors)
- Environment types
- Agent functions and properties
- Agent types
What is an Intelligent Agent?

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.
- Human agent: eyes, ears, and other organs for sensors; hands, legs, mouth, and other body parts for actuators.
- Robotic agent: cameras and infrared range finders for sensors; various motors for actuators.
Agents include humans, robots, softbots, thermostats, etc.
The agent function maps from $P^*$ to $A$
$f: P^* \rightarrow A$
The agent program runs on the physical architecture to produce $f$
Vacuum-cleaner world

Percepts: location and contents, e.g., [A; Dirty]
Actions: *Left, Right, Suck, NoOp*
*Simplest reflex agent?*
A vacuum-cleaner agent

<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>

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

What is the right function?
Can it be implemented in a small agent program?
Rationality

- A rational agent is one that does the **right thing**.
- More precisely, what is rational at any given time depends on four things:
  - The performance measure that defines the criterion of success.
  - The agent’s prior knowledge of the environment.
  - The actions that the agent can perform.
  - The agent’s percept sequence to date.
Performance measure

- For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

- Design measures according to what you want (as a behavior) and not according to what you think the agent should behave!

- Vacuum-cleaner?
  - Amount of dirt cleaned up.
Rationality

- Rationality maximizes expected performance while perfection maximizes actual performance.
Task Environment

- **PEAS**: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent (odometer, engine sensors, ?)
- Consider, e.g., the task of designing an automated taxi driver (“beyond the capabilities of existing technology”):
  - **Performance measure**:
    - Safe, fast, legal, comfortable trip, maximize profits and ?
  - **Environment**:
    - Roads, other traffic (?), pedestrians, customers, weather
  - **Actuators**:
    - Steering wheel, accelerator, brake, signal, horn, ?
  - **Sensors**:
    - Cameras, sonar, speedometer, GPS, odometer, engine sensors, ?
PEAS for Internet shopping agent

- **Performance Measure?**
  - price, quality, appropriateness, efficiency

- **Environment?**
  - current and future WWW sites, vendors, shippers

- **Actuators?**
  - display to user, follow URL, fill in form

- **Sensors?**
  - HTML pages (text, graphics, scripts)
PEAS for Part-picking robot

- Performance measure:
  - Percentage of parts in correct bins
- Environment:
  - Conveyor belt with parts, bins
- Actuators:
  - Jointed arm and hand
- Sensors:
  - Camera, joint angle sensors
Agent Characteristics

- **Situatedness**: The agent receives some form of sensory input from its environment, and it performs some action that changes its environment in some way. Examples of environments: the physical world and the Internet.
- **Embodiment**: Having a physical body
- **Autonomy**: The agent can act without direct intervention by humans or other agents and that it has control over its own actions and internal state.
Agent Characteristics

- **Adaptivity**: The agent is capable of
  - (1) reacting flexibly to changes in its environment
  - (2) taking goal directed initiative (i.e., is pro-active), when appropriate;
  - and (3) learning from its own experience, its environment, and interactions with others.

- **Sociability**: The agent is capable of interacting in a peer-to-peer manner with other agents or humans.
Environment Types – Categorize in different dimensions

- **Fully observable vs. partially observable.**
  - If an agent’s sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable. Example?

- **Deterministic vs. stochastic.**
  - Guaranteed effect.
  - If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise it is stochastic. Example? Board games? Chess, backgammon

- **Episodic vs. sequential.**
  - In an episodic task environment, the agent’s experience is divided into atomic “episodes.”
  - Episode: Single cycle of an agent perceiving and taking an action
  - Episodic: If the choice depends on the current episode and not on previous episodes
    - Easier to operate.
  - In sequential environments, the current decision may affect all future decisions. Examples? Chess?
Environment Types

- **Static vs. dynamic.**
  - If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise it is static.
  - If the environment itself does not change with the passage of time but the agent’s performance score does, then we say the environment is semidynamic. Example?

- **Discrete vs. continuous.**
  - The discrete/continuous distinction can be applied to the state of the environment, to the way time is handled, and to the percepts and actions of the agent. Example?
  - Possible to convert continuous environments into discrete environments (with loss of precision)

- **Single-agent vs. multi-agent.**
  - Taxi driver? Whether B's behavior is best described as maximizing a performance measure depending on A's performance measure
  - Competitive
  - Cooperative
<table>
<thead>
<tr>
<th>Environment types</th>
<th>Solitaire</th>
<th>Backgammon</th>
<th>Internet Shopping</th>
<th>Taxi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observable?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Episodic?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single-agent?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Environment types

<table>
<thead>
<tr>
<th></th>
<th>Solitaire</th>
<th>Backgammon</th>
<th>Internet Shopping</th>
<th>Taxi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observable?</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Deterministic?</td>
<td>Yes</td>
<td>No</td>
<td>Partly</td>
<td>No</td>
</tr>
<tr>
<td>Episodic?</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Static?</td>
<td>Yes</td>
<td>Semi</td>
<td>Semi</td>
<td>No</td>
</tr>
<tr>
<td>Discrete?</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Single-agent?</td>
<td>Yes</td>
<td>No</td>
<td>Yes (except auctions)</td>
<td>No</td>
</tr>
</tbody>
</table>

The environment type largely determines the agent design. The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent.
Structure of agent

- **Agent = architecture + program**

- **Agent Function:**
  - Mathematically speaking, we say that an agent’s behavior is described by the agent function that maps any given percept sequence to an action.

- **Agent Program:**
  - Take the current percept as input and return action.
  - The implementation of the agent function for an artificial agent is called the agent program.
  - Why only current percept?
    - Only information provided by the environment.
Agent Functions and Programs

function TABLE-DRIVEN-AGENT(percept) returns an action

static:
  percepts, a sequence, initially empty
  table, a table of actions, indexed by percept sequences, initially fully specified

append percept to the end of percepts
action ← LOOKUP(percepts, table)
return action
Problems of table lookup

- Let $P$ be the set of possible percepts, $T$ be the lifetime of agent. Lookup table size?
- Consider the taxi:
  - 640x480 camera, 30fps, 24 bit color representation
  - makes 27 Mb. per second
  - Lookup table with $10^{250,000,000,000}$ entries for 1 hour driving
  - Even lookup table for chess has $10^{150}$ entries
- Structured representation vs. raw data!
Agent types

- Four basic agent types in order of increasing generality:
  - simple reflex agents
  - reflex agents with state
  - goal-based agents
  - utility-based agents
- All these can be turned into learning agents
Simple Reflex Agents

- Select actions based only on the current percept, ignoring the rest of the percept history.
- Table lookup of percept-action pairs defining all possible condition-action rules necessary to interact in an environment
- • Problems
  - Possible condition-action rules too big to generate and to store (Chess has about $10^{120}$ states, for example)
  - No knowledge of non-perceptual parts of the current state
    - Get into loops - randomize
  - Not adaptive to changes in the environment; requires entire table to be updated if changes occur
Simple Reflex Agents

Diagram:
- Agent
  - Sensors
    - What the world is like now
  - Condition-action rules
    - What action I should do now
- Environment
  - Actuators
Reflex Agent with State Model-based reflex agents

- The knowledge about “how the world works” is called a **model** of the world.
- An agent that uses such a model is called a **model-based agent**.
- Encode "internal state" of the world to remember the past as contained in earlier percepts
- **Needed** because sensors do not usually give the entire state of the world at each input (what did we call this environment?), so perception of the environment is captured over time.
- Requires ability to represent change in the world; one possibility is to represent just the latest state, but then can't reason about hypothetical courses of action
  - Ability to “model” how world evolves independent of agent – other driver
  - Ability to “model” how world evolves as a result of agent actions
Reflex Agents with state

- State
- How the world evolves
- What my actions do
- Condition–action rules
- What the world is like now
- What action I should do now

Agent --> Sensors --> Environment

Agent <-> Actuators

CMPE540 – Week 1
Goal-based Agents

- A taxi in a junction, turn left or right?
- Choose actions so as to achieve a (given or computed) goal = a description of a desirable situation
- Keeping track of the current state is often not enough --- need to add goals to decide which situations are good
  - Think about a goal state where the desired goal holds
  - **Plan or search** a sequence of actions such that applying those actions will transform the current state into the goal state
  - Combine goal info and possible next states.
- Deliberative instead of reactive. Not reflex anymore.
- May have to consider long sequences of possible actions before deciding if goal is achieved --- involves consideration of the future, "what will happen if I do...?"
- Goal-based agents are more flexible – no need to rewrite lookup tables.
Goal-based Agent

- State
- How the world evolves
- What my actions do
- What the world is like now
- What it will be like if I do action A
- Goals
- What action I should do now
- Actuators
- Environment
Utility-based Agents

- When there are multiple possible alternatives, how to decide which one is best?
- A goal specifies a crude distinction between a happy and unhappy state, but often need a more general performance measure that describes "degree of happiness"
- Goals alone are not really enough.
  - Taxi: different sequences to the same goal are safer, quicker, more reliable or cheaper.
- Utility function $U: \text{State} \to \text{Reals}$
  - indicates a measure of success or happiness when at a given state
  - A state has higher utility if it is preferred over another.
  - Allows decisions comparing choice between conflicting goals, and choice between likelihood of success and importance of goal (if achievement is uncertain)
Utility-based Agents

- State
- How the world evolves
- What my actions do
- Utility

Sensors
- What the world is like now
- What it will be like if I do action A

How happy I will be in such a state

What action I should do now

Environment

Agent

Actuators
What is missing?
Learning agents - components

- **learning element**, which is responsible for making improvements,
- **performance element**, which is responsible for selecting external actions. The performance element is the entire agent: it takes in percepts and decides on actions.
- **critic** gives feedback from the on how the agent is doing with respect to a fixed performance standard.
- **problem generator** is responsible for suggesting actions that will lead to new and informative experiences.
Learning Agents
A return to the taxi example

- Performance element:
  - knowledge and procedures for selecting actions

- Learning element:
  - Formulates rules for bad actions

- Critic:
  - Observes the world for negative and positive feedback
  - Performance standard?

- Problem generator:
  - Suggests areas of exploration: new paths, braking in different environmental conditions, etc.
Summary

- **Agents** interact with **environments** through **actuators** and **sensors**
- The **agent function** describes what the agent does in all circumstances
- The **performance measure** evaluates the environment sequence
- A **perfectly rational** agent maximizes expected performance
Summary

- **Agent programs** implement (some) agent functions
- **PEAS** descriptions define task environments
- Environments are categorized along several dimensions:
- Several basic agent architectures exist:
  - Reflex, reflex with state, goal-based, utility-based
  - Learning vs. non-learning