CS5331: Concepts in Artificial Intelligence & Machine Learning systems

Rattikorn Hewett
Department of Computer Science
Texas Tech University
Spring 2008

About the course

- **Contents:** Fundamentals of AI (Artificial Intelligence) & ML (Machine Learning) systems
- **Prerequisite:**
  - high level programming language, algorithms, database
  - formalisms such as logic, discrete mathematics
  - statistics and probabilities
- **Evaluation:** HW 30%, Midterm 30%, Final 30%, Participation 5%
- Let's see details in the course syllabus
At the end of this course

Students should

● learn concepts and techniques in AI
  ■ differentiate AI & non-AI techniques
  ■ identify problems that requires AI or software that uses AI to find solutions
  ■ recognize misconceptions, AI topics and AI applications

● learn concepts and techniques of ML
  ■ understand why ML
  ■ where does it "learn"?

● be able to continue study particular areas in depth

Breadth + some depth in AI & ML

Overview

Luger, Chapter 1 and references
Outline

- Intelligent systems & AI: What is AI?
  What is “intelligence”?
- AI Applications: Past to Present
- Why AI?
- AI problems and their characteristics
- Programming in AI

What is AI?

- Study of intelligent behavior through computational theories
- A branch in computer science that is concerned with the automation of intelligent behavior
  Intelligent software (systems)
What is intelligent behavior?

Hallmarks of intelligent behavior
- Ability to solve complex problems
- Adaptability
- Self-awareness
- Learning from experience
- Dealing with incomplete information
- Acting under pressure
- Creativity
- Common sense reasoning

Man versus Machine

Can you give an example of the following?
- Easy tasks for men that are difficult for machines
  
  *Low level perceptual tasks such as recognizing patterns, voices and understanding stories.*

- Easy tasks for machines that are difficult for men
  
  *Highly complex and well specified computation.*
Other definitions of AI

- The science of making machines do tasks that humans can do or try to do
- The study of computations that make it possible to perceive, reason and act

“Intelligence” of Agents ~ degree to which they are successful at performing tasks

The term “AI” was coined in 1958 by John McCarthy

Questions

- How does the human mind work?
- What is a theory of intelligence?
- How can we build a more capable computer?
- ......
How can we answer them?

- Study human behavior - *Psychology, Cognitive Science*
- Study human hardware - *Neurobiology*
- Think hard! - *Philosophy*
- Build computers that exhibit intelligence - *Experimental CS*
- Analyze computational complexity of tasks requiring intelligence - *Theoretical CS*

AI is multidisciplinary

<table>
<thead>
<tr>
<th>Behavior</th>
<th>Intelligence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reasoning</td>
<td>Human Performance</td>
</tr>
<tr>
<td>Action</td>
<td>Philosophy: Cognitive Model</td>
</tr>
<tr>
<td></td>
<td><em>THINK</em> like HUMAN</td>
</tr>
<tr>
<td></td>
<td>Mathematics: Logic</td>
</tr>
<tr>
<td></td>
<td>Philosophy: Turing Test</td>
</tr>
</tbody>
</table>

*THINK* RATIONALLY

*ACT* RATIONALLY
**Think like human**

Cognitive Models:

More concerned with how human reasons to solve a problem than a solution to the problem.

**Goal:**

To construct theories of the working of the human mind through introspection or psychological experiments.

**Act like human**

Turing Test:

Ability to achieve human-level performance in all cognitive tasks, sufficient to fool an interrogator.

### Alan Turing

*Man or Machine?*
Can we really make machines intelligent?

• **Weak AI position**: Turing’s argument on
  • disability: A machine can never have a sense of humor, have initiative, learn from experience, fall in love, use words properly, do something really new, ……
  • informality: Intelligent behaviors can’t be captured by formal rules

• **Strong AI position**:
  Machines that act intelligently cannot have real conscious minds.

RESULTS: Inconclusive – hard to prove or disprove
  More work on commonsense reasoning, flexible controls

---

**Think rationally**

Laws of thought:
Find laws that govern the mind.
Emphasize on correct inferences to codify and characterize “right thinking” (e.g., by using logic).

Problems:
- Not easy to formulate informal knowledge
- High computational cost
Act rationally

Rational Agents:
Ability to achieve goals given certain beliefs.
Emphasize on building an artifact with this property.

Advantages:
- More general than the “laws of thought” since correct inference is not all of rationality
- Amenable to scientific development since rationality is clearly defined

Which is the most relevant to

Intelligent Systems?

<table>
<thead>
<tr>
<th>Behavior</th>
<th>Psychology: Cognitive Model</th>
<th>Mathematics: Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reasoning</td>
<td>THINK like HUMAN</td>
<td>THINK RATIONALLY</td>
</tr>
<tr>
<td>Action</td>
<td>ACT like HUMAN</td>
<td>ACT RATIONALLY</td>
</tr>
</tbody>
</table>

Philosophy: Turing Test

Human Performance
Ideal Performance

Intelligence
Typical “Intelligent” Systems

Perception

Cognition

Action

Data

Sensors

Observed/Monitored Data

Interpreter

Information

Reasoner/Learner

Knowledge

Actions recommended

Actuators

Actions performed

General Goal

To develop

- effective methods for designing intelligent systems
- analysis for system evaluation, e.g., optimal criteria
- mechanisms for system adaptation to changing environments

Research is driven by applications

- Tackling new applications leads to better techniques
Outline

- What is AI? What is “intelligence”?
- AI Applications: Past to Present
- Why AI?
- AI problems and their characteristics
- Programming in AI

Tasks and Areas Studied in AI

Formal
- Game playing
- Mathematical theorem proving

Expert
- Diagnosis
- Design
- Classification
- Control

Mundane
- Speech recognition
- Vision
- Natural language understanding
- Robotics
Microworlds

Problems that appeared to require intelligence to solve

- Blockworld problem by Minsky
- Symbolic integration, typical for first year calculus students, by Slagle
- Natural language system by Winograd

Out to the real world
Game playing

*Deepthout2* ranked among top 100 players in the world

Ideal for early work:
- rules are well defined
- explicit representation and level of abstraction of knowledge content
- can focus on “search”

Game playing (cont.)

**Evolution**
- 1957 Samuels’ checkers program
- 1966 Greenblatt’s chess program beats tournament players
- 1982 Thompson’s first special purpose chess computer, *Belle*, is the first master-level program
- 1985 Berliner’s *Hitech* achieves senior master rating 1987 defeated grandmaster player
- 1988 *Deepthought* (CMU) and 1993 *Deepthought2* ranked among top 100 players in the world
- *Deep Blue* (IBM) uses parallel array of 1024 custom VLSI chips
Game playing (cont.)

Lessons
- Many new (heuristic) search methods
- Brute force search dominates knowledge intensive methods
- Specialized hardware architectures win big

Text Understanding

Evolution
- 1960s: Attempts at machine translation fail miserably
- 1970s: Success with restricted domain of discourse
- 1980s: Commercial natural language front ends for specialized applications
- Lenat’s CYC project (MCC) to build knowledge equivalent to college level engineering students
Text Understanding (cont.)

Lessons

- Understanding text is much harder than generating it
- Meaning of a word is largely in its context (e.g., bank)
- Understanding general text requires vast background knowledge of the world
- Can succeed only for narrow domains of discourse

Similarly for Natural Language Understanding

Vision

Evolution

- 1970s: Interpret line drawing of polyhedral objects
  An early vision system (U. of Edinburgh)
- 1980s: Interpret simple camera image of 2D objects without occlusions, studies of shape from shading, depth from stereo, texture from shading
- 1990s: Interpret complex 3D images with overlapping parts. Continuous vision for mobile robots
Vision (cont.)

Lessons
- Much more difficult than expected
- Vast computational demands led to parallel hardware
- Understanding the physics of vision is important
- Knowledge of world is important.
- Incorporating vision in general purpose is still open

Robotics
- Robots are designed to perform certain tasks with some degree of flexibility & responsiveness
- Assuming a robot is capable of performing atomic actions, planning attempts to find sequence of actions to accomplish the task
- Planning extends beyond the domains of robotics
Robotics (cont.)

Evolution

- 1970s Pick and place robot arms in well-controlled environment
- 1980s Mobile robots for obstacle avoidance take hours to cross room
- 1985 CMU Navlab navigates roads outdoors
- 1986 Brook’s reactive robots exhibit high-speed reactions
- 1990 Genghis learns to walk in real time

Lessons

- The task is more complex than you think. A five year old can run much faster.
- Many standard AI assumptions do not hold in mobile robotics (e.g., world state and effects of actions are not always completely known)
- Planning: Interleaving deliberating and reaction
- Simple stimulus-response systems sometimes provide effective behaviors
Expert Systems

General Philosophy:
Narrow the problem domain scope yields possibility for an expert performance.

Drawback:
Brittleness
Hard to verify correctness
Lack learning capability

Expert System (cont.)

Evolution
- 1971 **DENDRAL**: Chemical structure elucidation
- 1976 **MYCIN**: Medical diagnosis for bacterial infections
- ~1980 **XCON**: Configure Vax orders of DEC
- 1988 IBM estimates $37.5 M of return on investment of $2.5 M on in-house expert systems
- ..... many more
**Expert System (cont.)**

**Lessons**
- AI can be practical
- Knowledge intensive expertise is easier to achieve than common sense
- Ability to explain is important
- Knowledge acquisition is a major bottleneck

**Machine Learning**

"Computer can do as they are told and therefore can’t perform original actions" [Ada Lovelace]

Machine learning aims to build programs to learn on their own, either from examples, experience, analogy, or being told

**Evolution**
- 1975 Winston: learns structural concepts, e.g., “arch”
- 1977 Lenat: AM (Automated Mathematician) learns many results of number theory
- 1986 Quinlan: ID3 learns patterns from examples
- ..... more
Neural networks

General Philosophy:
- Model the brain not the mind. Neural nets are small *learning machines*.

Advantage:
- Robust to noisy numerical data

Drawback:
- Lacks explanations, do not know what the network really learns.

Neural networks (cont.)

Example applications
- Navigation (*NavLab*, at CMU)
- Speech recognition (*NetTalk*, at Stalk Institute)
- ... many more

*Are there any other intelligence models inspired by biological systems?*
More recent areas

Intelligence inspired by biological systems:

- Evolutionary programs
e.g., Artificial life, Genetic Algorithms

  Idea:
  Spawn population of compete candidate solutions, those that better fit criteria survive and create offspring in the next generation

More recent areas (cont.)

Intelligence inspired by social systems:

- Agent-based systems

  Idea: social system provides metaphor for intelligence (e.g., insect society)

  Approach:
  A system consists of autonomous agents interacting cooperatively toward the goal
Recap: AI Timeline

17th-18th Centuries:
- Philosophy
  - mind-body problem [Descartes]
- Mathematics
  - Logic [Leibniz]
  - first-order predicate calculus [Frege]
  - introduction to theory of knowledge representation [Euler]

→ foundations for automated reasoning in AI

Recap: AI Timeline

19th Century:
- Computer
  - First programmable computer [Babbage]
  - Binary arithmatic, laws of logic [Boole]
- Early AI
  - Logical foundations for theorem-proving [Russell&Whitehead]
  - Designed intelligent machines [Turing, 1950]
Recap: AI Timeline

- 60’s: Theorem proving
  Game playing
- 70’s: Expert level performances
  Story Understanding, Perceptrons
- 80’s: Expert Systems
  Neural nets reborn
- 90’s: More theory on KR
  Commonsense reasoning
  Commercialization, Validation issues

Outline

- What is AI? What is “intelligence”??
- AI Applications: Past to Present
- Why AI?
- AI problems and their characteristics
- Programming in AI
### AI & Technologies

Some technologies influenced by AI

<table>
<thead>
<tr>
<th>AI Ideas</th>
<th>Computer Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Lisp machines</td>
<td>→ PCs</td>
</tr>
<tr>
<td>● KR techniques (frame, conceptual graph)</td>
<td>→ Object-Oriented (OO) paradigms</td>
</tr>
<tr>
<td>● AI programming language, <em>SmallTalk</em></td>
<td>→ OO program</td>
</tr>
<tr>
<td>● Expert System Development Cycles</td>
<td>→ Rapid prototyping</td>
</tr>
<tr>
<td>● Reactive Planning</td>
<td>→ Agile computing</td>
</tr>
</tbody>
</table>
Present

<table>
<thead>
<tr>
<th>AI Ideas</th>
<th>Internet Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Principles of knowledge base development</td>
<td>→ Knowledge Technology (e.g., XML, agent ontology, semantic webs)</td>
</tr>
<tr>
<td>● Machine learning techniques/data mining</td>
<td>→ Information retrieval, profiling</td>
</tr>
<tr>
<td>● Distributed AI</td>
<td>→ Software agent (e.g., e-commerce, m-computing)</td>
</tr>
</tbody>
</table>

....Future

Emerging AI related Technologies that (will) impact how we live:

- **Wearable computers**, e.g., smart shirt, non-intrusive medical treatments (with nanotechnology)
- **Digital genome**, e.g., genetic codes on a USB for healthcare
- **Neuromorphic technology**: chips to mimic human brain
- **Smart manufacturing**: distributed online factories
- **Next generation robotics**, e.g., smart drones, robots + GPS technology
- **Internet of things**, e.g., Car-to-car communication
- .......
Why AI?

- AI deals with computations that make it possible for machines to perceive, reason and act [Winston, MIT]
  - Engineering goal - to solve real-world problems
  - Scientific goal - to explain sources of intelligence
- AI can help us
  - solve complex problems
  - create new opportunities in various applications
  - shed new light on the science of “intelligence” → help us to become more intelligent

Outline

- What is AI? What is “intelligence”?
- AI Applications: Past to Present
- Why AI?
- AI problems and their characteristics
- Programming in AI
Goals of AI

- To understand and attempt to build intelligent agents
  - Build software to do things which at the moment people do better

Building Software Systems

- Process in software development:
  - define problems (specify requirements, I/O)
  - develop (design and analyze) algorithms
  - implement, test and evaluate software
Core of Software: Algorithms

- Algorithm ~ a sequence of computer instructions to solve a given problem

- A problem may have none or more than one algorithmic solution \(\rightarrow\) Analysis of Algorithms

- Some algorithmic solution may take far too long to compute \(\rightarrow\) Theory of Computation

Example

- Let \(n\) = size of a program (e.g., length of list to be sorted)

- Algorithm \(A\): requires \(n^2\) operations (quadratic)
  Algorithm \(B\): requires \(10^n\) operations (exponential)
  but easier to program, understand and think of

  Should we use \(B\) if we have a very fast computer?

Say, \(n = 100\) and each operation takes 1 sec
  \(\rightarrow\) \(B\) needs \(10^{100}\) sec

  How long does this take?
Cosmic Time Scale

- 0 sec: Big bang
  - GUT freezing: strong forces separate out
  - Electro magnetic force separate from weak force
  - Quark → Protons → Nuclei → Atoms → Stars, Planets

- $10^{18}$ Now
  - $10^{24}$ Stars lose planets
  - $10^{26}$ All stars burned out
  - $10^{39}$ Protons decay; Solid matter vanishes
  - $10^{71}$ Black hole starts to vanish
  - $10^{107}$ End of everything – last black holes vanish

- $10^{100}$ sec is a very long time!
- Even if we could get each operation done faster, say $10^{19}$ operations/sec
  (each operation takes less time than light travels in one angstrom unit ~ $10^{-10}$ meters - which you can’t!)
- $B$ still takes $10^{100}/10^{19} = 10^{81}$ sec .... by then your computer will disintegrate

AI Problems

- Computational Problems
  - Tractable problems
    - Realistically computable
      - Polynomial time
        - Example: Sorting problem
      - How to build agents that behave optimally given resource limitations?
  - Intractable problems
    - No algorithmic solution
    - Theoretically computable but not realistically computable
    - Exponential time or no algorithm
      - Example: Traveling Salesman problem
      - AI problems
AI Problems Characteristics

- Knowledge intensive
- Incomplete data
- Uncertain situations
- Combinatorial explosive choices that may lead to solutions

Outline

- What is AI? What is “intelligence”? 
- AI Applications: Past to Present
- Why AI?
- AI problems and their characteristics
- Approaches and Programming in AI
AI Approaches

- Use *knowledge* to reason and make/suggest decisions/actions that satisfy goals
- Basic Elements:
  - Knowledge Representation
  - Reasoning $\rightarrow$ Search
- AI Solutions:
  - when no algorithmic solution exists – find possible solutions
  - when algorithms are not computationally feasible – find efficient (but may not be optimal) solutions

Building Intelligent Software

- Define a problem
  - scope/domain
  - evaluation criteria
- Acquire, abstract and represent relevant “knowledge”
- Apply appropriate reasoning and problem-solving strategies
Programming languages for AI

Desired features:
- Support symbolic computation
  - Well-defined semantics
- Support exploratory programming methodologies
  - dynamic binding
  - flexible controls

Examples:
- LISP, Prolog (logic-based)
- Smalltalk, CLOS, Object C, C++, Java

AI Program ~ Experiment to better understanding of the problem it is trying to solve

Summary

- AI techniques are applied to intractable problems by finding
  - possible solutions for problems with no algorithms to solve
  - better solutions for problems whose algorithmic solutions are not realistically computable (e.g., quadratic time alg.)
- Can tractable problems be solved by AI techniques?
- Can non-AI techniques be used to solve intractable problems?
- Is a program that exhibits intelligent behaviors considered to be “intelligent” software?
Quiz: Which is intelligent software?

1. Computer animation
2. Character recognition
3. Video games
4. Two-player board games
5. Update financial transactions
6. Approve financial loans
7. Internet search such as Google
8. Virtual laboratory/class