CS 4700: Foundations of Artificial Intelligence

Instructor:
Prof. Selman
selman@cs.cornell.edu

Overview & Introduction
(Readings R&N: Chapter 1)
http://www.cs.cornell.edu/courses/cs4700/2017fa/
Artificial Intelligence
A Modern Approach
Third Edition
AI Methodology

Theoretical aspects
- Mathematical formalizations, properties, algorithms

Engineering aspects
- The act of building (useful) machines

Empirical science
- Experiments
What's involved in Intelligence?

A) Ability to interact with the real world
   to perceive, understand, and act
   speech recognition and understanding (*natural language*)
   image understanding (*computer vision*)

B) Reasoning and Planning
   modeling the external world
   problem solving, planning, and decision making
   ability to deal with unexpected problems, uncertainties

C) Learning and Adaptation
   Lots of data. Use to train statistical models.
   We are continuously learning and adapting.
   We want systems that adapt to us!
AI leverages from different disciplines

philosophy
  e.g., foundational issues (can a machine think?), issues of knowledge and believe, mutual knowledge

psychology and cognitive science
  e.g., problem solving skills

neuro-science
  e.g., brain architecture

computer science and engineering
  e.g., complexity theory, algorithms, logic and inference, programming languages, and system building.

mathematics, statistics, and physics
  e.g., statistical modeling, continuous mathematics, statistical physics, and complex systems.
Obtaining an understanding of the human mind is one of the final frontiers of modern science.

**Founders:**

- George Boole, Gottlob Frege, and Alfred Tarski
  - formalizing the laws of human thought

- Alan Turing, John von Neumann, and Claude Shannon
  - thinking as computation

- John McCarthy (Stanford), Marvin Minsky (MIT), Herbert Simon and Allen Newell (CMU)
  - the start of the field of AI (1956)
History of AI:  
The gestation of AI 1943-1956  
(See Russell & Norvig)

1943 McCulloch and Pitts
   - McCulloch and Pitts’ model of artificial neurons
   - Minsky’s 40-neuron network

1950 Turing’s “Computing machinery and intelligence”

1950s Early AI programs, including Samuel’s checkers program, Newell and Simon’s Logic theorist

1956 Dartmouth meeting: Birth of “Artificial Intelligence”
   - 2-month Dartmouth workshop; 10 attendees
   - Name was chosen. AI
1957 Herb Simon (CMU):

It is not my aim to surprise or shock you – but the simplest way I can summarize is to say that there are now in the world machines that think, that learn, and that create. 😊

1958 John McCarthy’s LISP (symbol processing at core)

1965 J.A. Robinson invents the resolution principle, basis for automated theorem. General reasoning procedure.

Limited intelligent reasoning in microworlds

(such as the “blocks world” --- a toy robotics domain)
The Blocks World

Requires:
--- Vision
--- Reasoning/Planning
--- Manipulation
--- Acting/Robotics

"A Microworld"
Micro-world: The Blocks World

How many different possible world states?

Core issue in AI: **Combinatorial explosion** in possible states of the world, possible futures, possible sentences, possible training examples. Need clever methods, algorithms, and representations.

How many different possible world states?

- a) Tens?
- b) Hundreds?
- c) Thousands?
- d) Millions?
- e) Billions?
- f) Trillions?
“Brainy, Yes, but Far From Handy”
New York Times 09/01/14
Making dexterous hands with human-level touch and sensing still a real challenge. [Link.]

Stacking blocks may seem like an easy task for a human, but robots have long struggled with such fine control. HDT’s Adroit manipulator uses force-sensing and vision to accomplish the delicate task.

Dynamic human touch — for example, when a finger slides across a surface — could distinguish ridges no higher than 13 nanometers, or about 0.0000005 of an inch. Individual molecules...
1) Weizenbaum’s ELIZA (“fools” users)

Capturing general knowledge is hard.

Revival: Amazon’s Chatbots

2) Difficulties in automated translation

See Babelfish

Syntax and dictionaries are not enough
Consider going from English to Russian back to English.
Early effort…

“The spirit is willing but the flesh is weak.”

“The vodka is good but the meat is rotten.”

Natural language processing (NLP) is hard.
(Ambiguity! Context! Anaphora resolution.)
3) Cars climbing up trees (at CMU)...
   Road sides look like parallel lines.
   But, unfortunately, so do trees!

   Computer vision is hard.
   (Ambiguity! Context! Noisy pixels.)

4) Limitations of perceptrons discovered
   Minsky and Papert (1969)
   Can “only” represent linearly separable functions
   Neural network research almost disappears

   Machine learning is hard.

5) Intractability of inference. NP-Completeness (Cook 72)
   Intractability of many problems attempted in AI.
   Worst-case result....

   Machine reasoning is hard.
History of AI

**Knowledge based systems (1969-79)**

Intelligence requires knowledge

Knowledge-based systems (lots of knowledge with limited but fast reasoning)

(Feigenbaum)

versus

general “weak” methods (a few basic principles with general reasoning)

(Simon and Newell)

Some success: *Expert Systems*

- Mycin: diagnose blood infections (medical domain)
- R1: configuring computer systems
- AT&T phone switch configuration

Knowledge in rules of form:

If symptom_1 & symptom_3 then disease_2 (with certainty .8)

Surprising insight: Modeling medical expert easier than modeling language / vision / reasoning of 3 year old. (not foreseen)
Expert Systems

Very expensive to code. ($1M+)

Response: Try to learn knowledge from data.

Weak with uncertain inputs / noisy data / partial information

Response: Incorporate probabilistic reasoning

Brittle! (fail drastically outside domain)

But IBM’s Watson’s knowledge modules have expert system flavor!

Leads to 1980 -- 1995:

--- General foundations reconsidered

--- Foundations of machine learning established (e.g. computational learning theory; PAC learning; statistical learning)

--- Foundations of probabilistic formalisms: Bayesian reasoning; graphical models; mixed logical and probabilistic formalisms.

From 1995 onward:

--- Data revolution combined with statistical methods

--- Building actual systems

--- Human world expert performance matched (and exceeded) in certain domains
History of AI:
1995 - present

Several success stories with high impact …
Machine Learning

In ’95, TD-Gammon.

World-champion level play by Neural Network that learned from scratch by playing millions and millions of games against itself! (about 4 months of training. Temporal-Difference learning.)

(initial games hundreds of moves)

Has changed human play.

Remaining open question: Why does this NOT work for, e.g., chess??
Some further remarks on reinforcement learning

--- reinforcement learning
  strengthen behavior with positive reward
  weaken with negative reward (punishment)

--- 2 versions of same program playing against itself.
  give more detail on output: one possible architecture
  game state and who is on play is input
Then, for each possible move:
  NN computes score/float y in range <-1,+1>.

--- After win (or loss), adjust weights in gradient descent
direction to move score y for that move up (or down).
for that move.
So, let’s say network made move “A” in play state S at because move A received a score of 0.9 by the neural net for state S.

Then, after playing out the game, the network lost. So, more likely than not, move A was quite possibly not the right move to make in state S.

Therefore, adjust weights on network a tiny bit to move down for move A, given game is in state S.

Score(state S) is a function of the state and the setting of the weights in the NN. Use basic calculus (gradient descent), to change the weights to lower (or raise) the score given the inputs representing state S.
The essence of NN is that it reduces a lot of AI/ML to gradient descent optimization, given some training set and a loss function. The loss function specifies what we want the NN to compute.

So, the NN is going to compute a complex function given the input state and the setting of its weights.

Learning / behavior etc., all comes down to modifying the weights to have the NN output something as close as possible to the training example points of the behavior/function that needs to be learned.

In a game, for each possible move, you want to know the minimax value or a good approximation thereof (“probability” first player wins between -1.0 (certain loss) and +1.0 (certain win)).

Lots of AI/ML reduced to cleverly designed optimization problems.
A mathematical conjecture (Robbins conjecture) unsolved for 60 years!

The Robbins problem was to determine whether one particular set of rules is powerful enough to capture all of the laws of Boolean algebra.

Mathematically:
Can the equation \( \text{not}(\text{not}(P)) = P \) be derived from the following three equations?

1. \((P \lor Q) = (Q \lor P)\)
2. \((P \lor Q) \lor R = P \lor (Q \lor R),\)
3. \(\text{not}(\text{not}(P \lor Q) \lor \text{not}(P \lor \text{not}(Q))) = P.\)

First creative mathematical proof by computer.
Contrast with brute-force based proofs such as the 4-color theorem.

[An Argonne lab program] has come up with a major mathematical proof that would have been called creative if a human had thought of it.


New: 2014 ---
Erdos Discrepancy Conjecture resolved

Note: Same order of search complexity as performed by Deep Blue per move. Quantative threshold for creativity?
Erdos Discrepancy Conjecture

A recently resolved math challenging problem using automated reasoning.

A conjecture about properties of infinite sequences of +1s and -1s.

Let’s cover some details.
Consider a sequence of 1s and -1s, e.g.:

\[ \sigma x_i \]

<table>
<thead>
<tr>
<th>-1, 1, 1, -1, 1, 1, -1, 1, -1 ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 7 8 9 ...</td>
</tr>
<tr>
<td>2 4 6 8                       ...</td>
</tr>
<tr>
<td>3 6 9 ...</td>
</tr>
</tbody>
</table>

and look at the sum of sequences and subsequences:

-1 + 1 = 0  \hspace{1cm} \text{(x_i)}
-1 + 1 + 1 = 1
-1 + 1 + 1 + -1 = 0
-1 + 1 + 1 + -1 + 1 = 1
-1 + 1 + 1 + -1 + 1 + 1 = 2
-1 + 1 + 1 + -1 + 1 + 1 + -1 = 1
-1 + 1 + 1 + -1 + 1 + 1 + -1 + 1 = 2
-1 + 1 + 1 + -1 + 1 + 1 + -1 + 1 + -1 = 1

and “skip by 1” (x_2i)

1 + -1 = 0
1 + -1 + 1 = 1
1 + -1 + 1 + 1 = 2
etc.

and “skip by 2” (x_3i)

1 + 1 = 2
1 + 1 + -1 = 1
etc.

What happens to partial sums?
How would you tackle the problem as a Computer Scientist?

What is the size of the search space? How long would your algorithm take?

Discovered in 2015: there exists a sequence of 1160 +1s and -1s such that sums of all subsequences never < -2 or > +2.

Result was obtained with a general reasoning program (a Boolean Satisfiability or SAT solver). Surprisingly, the approach far outperformed specialized search methods written for the problem, including ones based on other known types of sequences. (A PolyMath project started in January 2010.)
1160 elements
all sub-sums stay between -2 and +2

40 x 29 pattern
Aside: A Taste of Problem Size

Consider a real world Boolean Satisfiability (SAT) problem, from software & hardware verification.

The instance bmc-ibm-6.cnf, IBM LSU 1997:

```
p cnf !
-1 7 0
-1 6 0
-1 5 0
-1 -4 0
-1 3 0
-1 2 0
-1 -8 0
-9 15 0
-9 14 0
-9 13 0
-9 -12 0
-9 11 0
-9 10 0
-9 -16 0
-17 23 0
-17 22 0
```

“1” for variable \(x_1\), “2” for \(x_2\), etc.

\(x_1\), \(x_2\), \(x_3\), … our Boolean variables (set to True or False)

\((\text{not } x_1) \text{ or } x_7\)
\((\text{not } x_1) \text{ or } x_6\)

etc.

Question: Can we satisfy all statements?

Set \(x_1\) to False ??

SAT problem lies at the core of computer science
Prototypical NP-complete problem (from P vs. NP)
10 pages later:

\[ I.e., (x_{177} \lor x_{169} \lor x_{161} \lor x_{153} \ldots \]

\[ x_{33} \lor x_{25} \lor x_{17} \lor x_{9} \lor x_{1} \lor \neg x_{185}) \]

clauses / constraints are getting more interesting…

\textit{Note} x_{1} \ldots
4000 pages later:

```
10236 -10050 0
10236 -10051 0
10236 -10235 0
10008 10009 10010 10011 10012 10013 10014
10015 10016 10017 10018 10019 10020 10021
10022 10023 10024 10025 10026 10027 10028
10029 10030 10031 10032 10033 10034 10035
10036 10037 10038 10039 10040 10041 10042
10047 10048 10049 10050 10051 10235 -10236 0
10237 -10008 0
10237 -10009 0
10237 -10010 0
...
```
Finally, 15,000 pages later:

\[-7 \ 260 \ 0 \ 7 \ -260 \ 0 \ 1072 \ 1070 \ 0 \ -15 \ -14 \ -13 \ -12 \ -11 \ -10 \ 0 \ -15 \ -14 \ -13 \ -12 \ -11 \ 10 \ 0 \ -15 \ -14 \ -13 \ -12 \ 11 \ -10 \ 0 \ -15 \ -14 \ -13 \ -12 \ 11 \ 10 \ 0 \ -7 \ -6 \ -5 \ -4 \ -3 \ -2 \ 0 \ -7 \ -6 \ -5 \ -4 \ -3 \ 2 \ 0 \ -7 \ -6 \ -5 \ -4 \ 3 \ -2 \ 0 \ -7 \ -6 \ -5 \ -4 \ 3 \ 2 \ 0 \ 185 \ 0\]

Search space of truth assignments:

\[2^{50000} \approx 3.160699437 \cdot 10^{15051}\]

Current reasoning engines can solve this instance in a few seconds! (no satisfying assignment exists + proof)
Consider a sequence of 1s and -1s, e.g.:

\[-1, 1, 1, -1, 1, 1, -1, 1, -1 \ldots\]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

and look at the sum of sequences and subsequences:

\[-1 + 1 = 0\]
\[-1 + 1 + 1 = 1\]
\[-1 + 1 + 1 + -1 = 0\]
\[-1 + 1 + 1 + -1 + 1 = 1\]
\[-1 + 1 + 1 + -1 + 1 + 1 + -1 = 1\]

and “skip by 1”

\[1 + -1 = 0\]
\[1 + -1 + 1 = 1\]
\[1 + -1 + 1 + 1 = 2\]
\[1 + -1 + 1 + 1 + -1 = 1\]
\[-1 + 1 + 1 + -1 + 1 + -1 + 1 + 1 + -1 + 1 + 1 + -1 + 1 + -1 + 1 + -1 = 1\]

etc.

and “skip by 2”

\[1 + 1 = 2\]
\[1 + 1 + -1 = 1\]

etc.
Back to sequences of +1/-1s

Logic / SAT encoding has variables for the sequence $X_1, X_2, \ldots, X_N$
(we interpret True for +1 and False for -1)

but also e.g.

Proposition: “sum_of_first_2_terms_of_skip_by_2_subseq_=_2”
(for any given setting of $X_1 \ldots X_N$ this is either True or False)

and statements of the form:

IF ((sum_of_first_2_terms_of_skip_by_3_subseq_=_2 == True)
    AND (X_9 == False))
THEN (sum_of_first_3_terms_of_skip_by_3_subseq_=_1 == True)

Encoding: 37,418 variables and 161,460 clauses / constraints.

Sequence found in about 1 hour (MacBook Air).

Perhaps SAT solver was “lucky” in finding the sequence?
Remarkably, SAT solver also shows that each sequence of 1161 or longer leads to +3 (or -3) somewhere. (Erdos discrepancy conjecture) (Again, think of the size of the search space!)

Encoding: 37,462 variables and 161,644 clauses / constraints. Proof of non-existence of discrepancy 2 sequence found in about 10 hour (MacBook Air).

Proof: 13 gigabytes and independently verified (50 line proof checking program). Proof is around a billion small inference steps.

Machine understands and can verify result easily (milliseconds); Humans: probably never. Still, we can be certain of the result because of the verifier.
Observations

1) Result different from earlier “computer math” results, such as the proof of the 4 color theorem, because here we don’t need to trust the theorem prover. Final proof (“certificate”) can be checked easily by anyone.

2) It’s not a brute force search. Earlier SAT solvers cannot find the proof. Specialized programs cannot find the proof.

Brute force proof is of order $2^{1161} = 3.13 \times 10^{349}$. Current solver finds complete proof with “only” around $1.2 \times 10^{10}$ steps. Clever learning and reasoning enables a factor $10^{339}$ reduction in proof size.

3) In part inspired by discrepancy 2 result, Terence Tao proved several months later the general Erdos conjecture (for any discrepancy). Deep and subtle math.

4) But, does not fully supersede the 1161 result for the discrepancy 2. Future math may build further on these types of computational results. (I.e. true, verifiable facts but not human accessible.)
1997: Deep Blue beats the World Chess Champion

Deep Blue had Kasparov in deep thought
(CNN)

I could feel human-level intelligence across the room
Gary Kasparov, World Chess Champion (human...
Deep Blue vs. Kasparov

Game 1: 5/3/97: Kasparov wins
Game 2: 5/4/97: Deep Blue wins
Game 3: 5/6/97: Draw
Game 4: 5/7/97: Draw
Game 5: 5/10/97: Draw
Game 6: 5/11/97: Deep Blue wins

We’ll discuss Deep Blue’s architecture, when we cover multi-agent search.

The value of IBM’s stock increased by $18 Billion!

Note: when training in self-play, be careful to randomize!
On Game 2

Game 2 - Deep Blue took an early lead. Kasparov resigned, but it turned out he could have forced a draw by perpetual check.

Interestingly, if Kasparov had been playing a human he would most likely not have resigned!

This was real chess. This was a game any human grandmaster would have been proud of.

Joel Benjamin
grandmaster, member Deep Blue team
Kasparov on Deep Blue

1996: Kasparov Beats Deep Blue

“I could feel --- I could smell --- a new kind of intelligence across the table.” (CNN)

1997: Deep Blue Beats Kasparov

“Deep Blue hasn't proven anything.” 😊

Current strongest play: Computer-Human hybrid
May, '97 --- Deep Blue vs. Kasparov. First match won against world-champion. "intelligent creative" play. 200 million board positions per second!

Kasparov: ... still understood 99.9 of Deep Blue's moves.

Deep Blue considers 60 billion boards per move! Human? Around 10 to 20 lines of play. Hmm…

Intriguing issue: How does human cognition deal with the search space explosion of chess? Or how can humans compete with computers at all?? (What does human cognition do? Truly unknown...
Concepts (briefly)
(more details with multi-agent search)

--- Minimax search on game tree to get optimal move
(large tree >= $10^{80}$ chess)

Size tree: $b^d$ (b --- average branching; d --- depth)

alpha-beta pruning: $b^{(d/2)}$  [key technique]

--- Board evaluation or utility function when
you can’t search to the bottom

--- Board eval is linear weighted some of features; can be
trained via learning. (Reinforcement learning /
AlphaGo)

--- Chess complexity?

O(1) (formally speaking…)

--- 2017: AlphaGo beats world human Go champion
From Robocup to Warehouse Automation

Kiva Systems $700M
2005 Autonomous Control: DARPA GRAND CHALLENGE

October 9, 2005
Stanley and the Stanford Racing Team were awarded 2 million dollars for being the first team to complete the 132 mile DARPA Grand Challenge course (Mojave Desert). Stanley finished in just under 6 hours 54 minutes and averaged over 19 miles per hours on the course.

Sebastian Thrun: Google's driverless car (2011)

Cornell team stuck 😞 due to malfunctioning GPS.

http://www.youtube.com/watch?v=bp9KBrH8H04
Path Planning Overview
Path planning is the basic process by which our vehicle decides on what path to take through the world. The A.I. uses the world model created by the sensors, the GPS waypoints provided during the race by DARPA, and a road following algorithm to pick a best path.

Road Following
The road following algorithm uses color differences, shadowing, and edge-detection algorithms to detect the sides of a road (if there is a road) and then decides if the road is turning, going straight, which direction, how sharply, etc. The road following algorithm uses input from most of the vehicle sensors, and provides the A.I. with probable road characteristics.

The red line represents the ideal path picked by our A.I., and the turquoise path represents the actual path traveled by our vehicle. The differences arise because 1) we did not start our vehicle on the ideal path, and 2) our vehicle must, without exceeding its performance limits, avoid small obstacles such as boulders.

A* algorithm
Covered in search and problem solving.
Cornell: 4th!
Also, in historic
1st autonomous driverless car
collision. Rear-ended by MIT car!

2007 Darpa Urban Challenge Winner: CMU Tartan Racing's Boss

2007 Darpa Urban Challenge
The Urban Challenge will pit driverless vehicles against one another on city streets. Robots will have to handle traffic, intersections, rules of the road and other robots. The challenge is a high-stakes competition that plays out on a world stage. The prize is $2M, but the payoff for driver safety is much greater. This competition will be held November 3, 2007.

The Urban Challenge is third in a series of autonomous vehicle competitions designed to catalyze robotic technology development. On October 8, 2005, Carnegie Mellon's “Sandstorm” and “Highlander” crossed the finish line of DARPA Grand Challenge after successfully completing a 132-mile course through the Nevada desert, coming in second and third place respectively.

http://www.tartanracing.org/blog/index.html#26
Watson: Question-Answering system, 2011

Watson defeats the two greatest Jeopardy! champions

http://www.youtube.com/watch?v=dr7IxQeXr7g
Watson
Multi-layer neural networks, a resurgence!

a) Winner one of the most recent learning competitions

b) Automatic (unsupervised) learning of “cat” and “human face” from 10 million of Google images; 16,000 cores 3 days; multi-layer neural network (Stanford & Google).

c) Speech recognition and real-time translation (Microsoft Research, China).

Aside: see web site for great survey article

ML as Optimization (i.e. minimize a loss function)
Start at min. 3:00. Deep Neural Nets in speech recognition.
Other promising ongoing efforts

1) Intelligent autonomous assistants, e.g., iPhone’s Siri (still a long way to go 😊) Integrated, autonomous agents. Google Glass will be the next step. Location / context aware; rich sensing, vision and speech understanding and generation.

2) Fully self-driving car (Google; assisted driving Mercedes and BMW --- the cost of a car is becoming software and sensors Incredibly more lines of code in a Mercedes than in a Boeing 747.)

2) Google translate. Reaches around 70% of human translator performance. Almost fully a purely statistical approach.

Not clear yet how far one can go without a real understanding of the semantics (meaning). But with Big Data, statistical methods already went much further than many researchers had considered possible only 10 years ago.
Course Administration

What is Artificial Intelligence?

Course Themes, Goals, and Syllabus
Setting expectations for this course

Are you going to build real systems and robots?

NO...

Goal:
Introduce the conceptual framework and computational techniques that serve as a foundation for the field of artificial intelligence (AI).
• Structure of intelligent agents and environments.
• Problem solving by search: principles of search, uninformed (“blind”) search, informed (“heuristic”) search, and local search.
• Constraint satisfaction problems: definition, search and inference, and study of structure.
• Adversarial search: games, optimal strategies, imperfect, real-time decisions.
• Logical agents: propositional and first order logic, knowledge bases and inference.
• Uncertainty and probabilistic reasoning: probability concepts, Bayesian networks, probabilistic reasoning over time, and decision making.
So far, we discussed

Artificial Intelligence and characteristics of *intelligent* systems.

Brief history of AI

Major recent AI achievements

Reading: Chapter 1 Russell & Norvig