CPSC 625-600 Artificial Intelligence

- Instructor: Yoonsuck Choe (choe@tamu.edu)
- TA: Jaewook Yoo (jwookyoo@neo.tamu.edu)
- All communications out of the class will be through email registered on NEO (neo.tamu.edu), and the announcements on the web page, so regularly check out the web page.
- Class notes will be available on the web 24 hours prior to the class. It is **your responsibility** to print it out and bring it to the class. **http://courses.cs.tamu.edu/choe/13fall/625/lectures/**

Syllabus

http://courses.cs.tamu.edu/choe/13fall/625

- See handout (which is just a hardcopy of the web page on day 1).

Things You May Need

- Students with disability:
  Please contact the department office (HRBB 3rd floor) for assistance. See the syllabus for the full information.
- Computer (UNIX) accounts:
  If you don't have one, get one:
  `https://wiki.cse.tamu.edu/index.php/Getting_Started_Guide`

What is Intelligence

Textbook Definitions

- Thinking like humans
- Acting like humans
- Thinking rationally
- Acting rationally ←

However, it depends on the definition: **whatever the (intelligence) test tests.**

What is AI?

A folk (popular) view of AI

From http://www-2.cs.cmu.edu/afs/cs.cmu.edu/ user/zhuxj/www/travel/fun/images/terminator.jpg (top); Universal studio's movie "Terminator" (bottom)

But Really, What is AI?

Diverse areas: http://www.aaai.org

- Problem solving
- Reasoning
- Theorem proving
- Learning
- Planning
- Knowledge representation
- Perception and Robotics
- Agents
- **and more**

Approaches

Two basic stances

- Strong AI:
  1. Build something that actually thinks intelligently.
  2. Simulation of thought would give rise to the pheonmenology of thought (i.e., how it feels like to think).
- Weak AI:
  1. Build something that behaves intelligently.
  2. Not worried about its feelings.

Problems

- Strong AI:
  Hard to determine if something is really consciously intelligent or not (the **other minds problem** in philosophy).
- Weak AI:
  Utility of the result is limited by the stated goal. Hard to achieve a **general usefulness** as in true intelligence.

How to do AI

Why not engineer AI, in the same way people engineered airplanes?

1. Flight
   goal is simple:
   - You know when a thing is flying.
2. Intelligence
   goal is complex and hard to define clearly:
   - Intelligence is a collection of many abilities.

There are many ways to meet a single clear goal (flight), but there can be only a small number of ways to simultaneously meet a huge number of unclearly defined goals (intelligence).

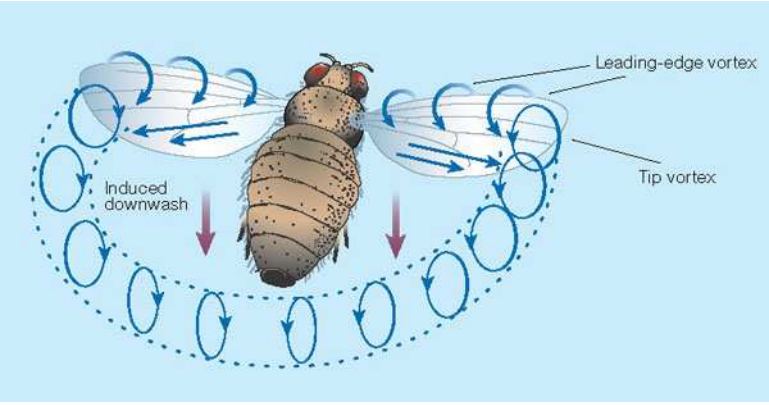
Why not Follow the Plane-Model?

Certain things may seem physically impossible (in terms of efficiency, etc.): e.g. the flight of flies.

- Flapping their wings cannot generate enough lift (for their body weight), but they do fly!
- Jet turbines cannot explain how the beetles achieve such an impossible feat.
- Recent observation:
  Flies gyrate their wings to generate a vortex to create greater lift.

Moral: if you fail to build the impossible, study an existing solution.

How Flies Fly



Source:

http://www.nature.com/nsu/010823/010823-10.html

Then, How to do AI?

Instructor's perspective:

- Importance of studying brain function.
- Influence of environmental regularities on brain development and function.
- Interaction of the brain with the environment through action.

We must think about the more fundamental issues from time to time when research seems to be at a dead-end.

Back to Reality

Let's be realistic. :-)

- Study strategies employed by humans in dealing with real-world problems.
- These include all the topics listed earlier.
- The background you learn in this course will enable you to appreciate the deepness of the problems, and to pursue further interest in AI, and in human and machine intelligence in general.

Overview

- Related academic disciplines
- History of AI
- Hard Problems
- Current Trends

Foundations of AI

- Philosophy
- Mathematics
- Psychology
- Cognitive Science
- Linguistics
- Neuroscience

Philosophy of Mind

The mind-body problem:

- Dualism: Mind and body are separate entities.
- Monism: Only mind or body exist, but not both
  1. Idealism: all things are mental
  2. Materialism: all things are material
- Epiphenomenalism: mental phenomena are just side-effects of physical change in the brain (i.e. they do not have causal power over behavior).

Too many variations to mention all.

Mathematics

- Algorithm (al-Khowarazmi)
- Boole
- Hilbert
- Gödel: Incompleteness theorem
- Turing: Halting problem
- Cook and Karp: P, NP, and the like

Representation/Interpretation, Symbol/Computing: the computer/software metahpore.

Psychology

- Behaviorism: stimulus-response and conditioning
- Functionalism: internal representations and processes. Implementation independent.
- Perceptual psychology: vision, audition, etc.
- Cognitive psychology: cognition as information processing.
- Holistic vs. localist debate: emergent vs. simple summation.

Linguistics

- WW II : machine translation.
- Phonetics, syntactic theory, semantics, discourse, etc.
- Innate vs. learned? : Chomsky
- Syntax: finite automata, context free grammar, etc.
- Semantics: semantic nets
- Sub-symbolic: self-organizing maps, episodic memory, recurrent neural nets, etc.

Cognitive Science

Interdisciplinary field studying human perception and cognition, ranging over:

- Neuroscience
- Behavioral science
- Social science
- Psychology
- Computational science
- Information theory
- Cultural studies

Neuroscience

- Staining: Golgi, Nissl
- Hubel and Wiesel: orderly structure of cat visual cortex
- PET scans and CAT scans: localizing functional modules
- fMRI imaging: cognitive and perceptual tasks
- Optical imaging: orderly structure
- TMS: zap and numb your brain

Connections

Scientific discoveries came from observing unexpected connections:

- Apple and gravity
- Cloud chamber and the discovery of subatomic particles
- Looms with punch-cards and modern computers

History of AI (I)

Gestation (1943–1956)

- McCulloch and Pitts: early neural nets
- Minsky and Papert: limitations of perceptron
- Newell and Simon: physical symbol system hypothesis
  - Logic Theorist
- Dartmouth Workshop (1956): AI was born
  It is 50(+1) years old (2007)!
  http://en.wikipedia.org/wiki/AI@50

History of AI (II)

Early successes (1952–1969)

- General problem solver
- McCarthy: LISP
- Toy domains: ANALOGY, STUDENT (algebra).
- Widrow and Hoff: adalines
- Rosenblatt: perceptrons

History of AI (III)

The 60's and 70's

- ELIZA
- Genetic algorithms
- Knowledge-based systems: avoid the weak method, i.e. search
  - scientific domain
  - engineering domain
  - natural language

The 80's : 5th generation AI – Prolog.

History of AI (IV)

50th anniversary in 2006: http://en.wikipedia.org/wiki/AI@50

- Some quotes from the 50th anniversary event (Rodney Brooks):
  - the social sophisitication of 10-year-old
  - the manual dexterity of a 6-year-old
  - the language ability of 4-year-old
  - the visual object recognition of a 2-year-old

Hard Problems (I)

- Physicalism, materialism, and naturalism: brain causes mind.
- Functionlism: if it functions in the same way, a silicon brain can also be conscious.
- Dualism and homunculus: the Cartesian theatre.
- Wide vs. narrow content: real correspondence, or limited to experiential state?
- Intentionality: how can we believe in things that do not exist, such as Poseidon.

Hard Problems (II)

- Semantic content and syntactic symbols: how can syntacic constructs posess intenionality?
- Symbol grounding: sensory devices produce grounded symbols, and composite symbols can be constructed.
- Problem of qualia: why do we feel in such a way?
- Turing test and Searle's Chinese Room
  - system reply
  - robot reply

Hard Problems (III)

- However, the assumption that a collection of unconscious things are unconscious is invalid: think about organic vs. inorganic, life vsinanimate matter.
- Searle's point of view: mind is an emergent phenomena of the neural substrate (biological naturalism).

Current Trends

- Learning: instead of hand-coding or strict reasoning.
- Neural networks and statistical methods
- Genetic algorithms (Evolutionary algorithms)
- Embodied robotics; Dynamical systems approach
- Computational Neuroscience
- Distributed Agents
- Some thoughts on consciousness: Crick and Koch

Latest Achievements

- AI in computer games (1980's–current).
- Game playing (Chess, etc.): IBM's Deep Blue (1997)
- Jeopardy: IBM's Watson (2011)
- Deep learning: Schmidhuber's neural networks beating human performance in hand written digit recognition and other tasks (circa 2010).

Unix Environment (Self Study)

The following slides are FYI: they won't be covered in the class. If you have any question about these, please see the instructor or the TA.

- Remote access
- Shell
- Files and Directories
- File and Dir Permissions
- Customizing the environment
- Execution of programs
- Getting More Information
- Editor: pico

Remote Access

- Telnet (insecure) vs. SSH (secure)
- http://www.freessh.org/
- For windows, use PUTTY.EXE
  - use the SSH mode.
- On-campus:
  sun.cs.tamu.edu, interactive.cs.tamu.edu, etc.
- Off-campus:
  Only by using TAMU VPN
- Use TAMU vpn to access other sunhosts.

Shell

- Like DOSPROMPT, but much more powerful.
- Several variations:
  csh, sh, **tcsh**, bash, zsh, ksh
- tcsh, derived from csh is your default.
- Finding out your default shell:

```
$ ps
sun: /> ps
PID TTY TIME CMD
964 pts/12 0:03 tcsh
```


Files and Directories

- Same as in MSDOS except:
  - longer filenames
  - case sensitive
- Path delimiter is `/`, not `\`
  `/user/choe/filename`
- Why MSDOS differs?
  They use `/` for command options:
  `FORMAT /S`

The most important directory is your home directory:
`/user/youridhere/.`

File and Directory Cmds

- Directory Listing
  - `ls` : short (`DIR /W`)
  - `ls -a` : list dot files also
  - `ls -l` : long (`DIR`)
  - `ls -t` : newest file first
  - `ls -F` : mark with dir(`/`), sym links (i.e. shortcut; `@`), executable (`*`), etc.
- Any combination is allowed:
  `ls -alF`
- Changing direcotry: `cd dirname`.

Cont'd

- Creation and deletion:
  `mkdir dirname`
  `rmdir dirname`
- File copy:
  `cp srcfile destfile`
  `cp srcfile destdir`
- Moving and renaming:
  - `mv srcfile destfile` : rename file
  - `mv srcfile destdir` : move file to different directory

File and Dir Perms

```
unix:~/> ls -l
total 568
-rw-------   1 choe     staff      22016 Oct  1 11:26 AdobeFnt.lst
-rwx------   1 choe     staff        600 Nov 20 17:37 PUTTY.RND*
drwx------   2 choe     staff       4096 Jan 14 22:24 RCS/
-rw-------   1 choe     staff       4869 Jan  9 09:34 aaai02
drwx------   2 choe     staff       4096 Jan 13 15:45 acct/
 * snip *
```

r: read, w: write, x: execute (allow cd for dir) permission.

Changing File and Dir Perms

`chmod [u|g|o][+|-][r|w|x] [file|dir]`

- `chmod o-rwx file`
- You can also use octal:
  `rwxr-xr-x`
  == 111101101 (binary)
  == 755 (octal)
- `chmod 600 filename`
  `rwx------`
- `chmod 755 dirname`
  `rwxr-xr-x`

Directory needs to be readable to do `ls`, and executable to do `cd`.

Environment Variables

- Environment variables:
  - used to configure your env.
  - `setenv` gives you the list with their values
  - `setenv VARNAME=value` to define or reset
  - Append `$` to get the value.
  - `echo $VARNAME` to view value
- Shell variables: use `set`
- Home directory (`echo $HOME`):
  - This is where all the action begins.
  - Everything below here belongs to you (almost).

Going to home: `cd` or `cd ~` or `cd $HOME` or ...

Customizing Your Shell

In tcsh:

- `.login` : only executed when you first login.
- `.cshrc` : run everytime you create a sub-shell, i.e. running a shell within a shell.
- Basically a list of shell commands, like the `BAT` file.

Just add these lines in `/.login`:

```
setenv PATH "$HOME/bin:$PATH"
set prompt="%m:%~/> "
alias ls "ls -F"
```


Execution of Programs

- `ls` is actually `/usr/bin/ls`.
- Because $PATH contains `/usr/bin/`, the above file gets executed.
- For local executables (compiled ones, etc.) in the current dir, add `./` to specify the current dir: `./execfilename`
- This is not a concern because we'll be using GCL interpreter: `/usr/local/bin/gcl`

Getting More Info

- Use the `man` command:
  ```
  man ls
  man chmod
  man mkdir
  ```
- Navigating in man pages:
  - `space (or CTRL-F)`: pgdown
  - `CTRL-B`: pgup
  - `/<string>ENTER`: search fwd for <string>
  - `?<string>ENTER`: search bwd for <string>
  - `/ or ? alone`: repeat search fwd or bwd

Editor: PICO

`/opt/csw/bin/pico`

An easy to use text editor: navigate with arrow keys, and

`CTRL-Y`(pgup) and `CTRL-V`(pgdn).

`CTRL-X`: Exit : answer carefully!.

`CTRL-O`: Save As

`CTRL-R`: Read another file into current position

`CTRL-K`: Cut (you can cut multiple lines)

`CTRL-U`: Paste (one or more lines cut with CTRL-K)

`CTRL-W`: Search (**W**here is ...)

No undo(!): I recommend `vi` or `emacs`.

Of course you may ftp. :-)

Tcsh tips

- Command and filename completion: just type a partial string and press `[tab]`.
- Use arrow keys to go backward (like in `doskey`).

Pipes

My output is your input: output of `ls` goes to `grep`.

```
sun:/user/choe> ls -al | grep no
drwx------ 3 choe staff 4096 Oct 10 10:53
.ssh2-no
-rw------- 1 choe staff 123 Jan 14 13:32 noms
drwx------ 2 choe staff 4096 Nov 12 14:41 notes
```


Useful Commands

- `grep` : pattern matching (also `awk` and `sed`)
  `grep <pattern> <filename>`
  `<some-other-command> | grep <pattern>`
- `find` : find file with a certain pattern
  `find <start-dir> <pattern> -print`
- `wc` : line, word, and char count `ls -al | wc`
- `df` : disk partitions and current usage
- `du` : disk usage under a directory

Other: `cal`:calendar, `date`:date, `uname -a`:OS-version, etc.

Little Bit of LISP

http://www.cs.tamu.edu/faculty/choe/courses/05fall/lisp-quickref.html

- CMUCL: Carnegie Mellon University Common LISP
- `/opt/apps/cmucl/bin/lisp`
- At the `*` prompt, just type the expressions.

```
unix:~/> lisp
CMU Common Lisp CVS Head 2003-07-01 16:23:01, running on unix
With core: /usr/local/lib/cmucl/lib/lisp.core
Dumped on: Tue, 2003-07-01 16:01:00-05:00 on empic5
See <http://www.cons.org/cmucl/> for support information.
Loaded subsystems:
    Python 1.1, target UltraSparc/Solaris 7
    CLOS based on Gerd's PCL 2003/06/18 09:23:09
* (+ 10 20)

30
* (quit)
unix:~/>
```

Next Time

- Lisp