Course Overview

CMPUT 261: Introduction to Artificial Intelligence

Introduction to AI

- This is a **survey course** about Artificial Intelligence
- Learn the basics of a **broad range** of core subareas

  1. Search
  2. Reasoning Under Uncertainty
  3. Supervised Learning
     - Deep Learning
  4. Reinforcement Learning
  5. Multiagent Systems

- Take specialized courses next for more **depth**

What is Artificial Intelligence?

Computing Science but not Artificial Intelligence:
Specify a good algorithm for accomplishing a **task** (with a computer)

- Send data over a possibly-congested link
- Schedule processes and threads
- Photorealistic rendering of a geometric model
- Convert source code to optimized machine code
- Meta: Properties of algorithms that make them efficient
- ...

What is Artificial Intelligence?

Artificial Intelligence:
Accomplish a **task** **without explicitly specifying** the procedure

- Find fastest route from point A to point B
- Optimal crop rotations
- Detect spam
- Detect diseased plants from images
- Play games (chess, go, poker, etc.)
- Schedule processes and threads
- Choose personalized medical treatments
- ...

Hang on...

1. Surely we explicitly specify *something?*
2. Didn't we see one of these tasks on the previous slide?

Lecture Outline

1. Course Logistics
2. Course Topics
3. AI Seminar!

After this lecture, you should be able to:

- Understand the policies and procedures of the course
- Describe the basic overview of the course material
- Decide whether you want to take the course

Course Essentials

Course information: https://jrwright.info/introai/

- This is the main source of information about the class
- Syllabus, slides, readings, deadlines

Lectures: Tuesdays and Thursdays, 9:30-10:50am in **ETLC E2-001**

- In person

Canvas:

- Discussion forum for **public** questions about assignments, lecture material, etc.
- Handing in assignments

Email: james.wright@ualberta.ca for **private** questions

- (health problems, inquiries about grades)

Office hours: By appointment, or after lecture

- TA's are available to help during lab hours
- No labs in the first week of class

Readings

We will draw from a lot of texts for this class.

BUT, they are all available online for **free**:

- [P&M] David Poole and Alan Mackworth,
  Artificial Intelligence: Foundations of Computational Agents, 2nd edition.
- [P] Simon J.D. Prince, *Understanding Deep Learning.*
- [S&B] Richard S. Sutton and Andrew G. Barto,
  Reinforcement Learning: An Introduction, 2nd edition.
- [S&LB] Yoav Shoham and Kevin Leyton-Brown,
  Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.

Optional readings for each lecture are listed on the schedule.

Evaluation

Grade breakdown

- Assignments: 30%
  - (due Sep 25, Oct 21, Nov 20, and Dec 4)
- Midterm exam: 35%
- Final exam: 35%

Late assignments

- 20% deducted for all late submissions
- 2 days late maximum
- Deadlines are **very firm**

Assignments

- There will be **four assignments**
- Types of questions:
  - **Short answer**: definitions, distinctions, etc.
    "*What is a Nash equilibrium?*"
  - **Model construction**: "*Represent XYZ as a graph search problem*"
  - **Algorithmic considerations**: "*What would be an appropriate algorithm to answer XYZ? Why?*"
  - Small **implementation** task
- Assignments are submitted electronically (via Canvas)

Collaboration Policy

Detailed version on the syllabus

You are **encouraged to discuss assignments** with other students:

1. You **may not** share or look at each other's **written work or code**.
2. You must **write up** your solutions individually

Individual work only on **exams**: No collaboration allowed

- **Exams are closed-book:** No slides, outside material, etc.

Academic Conduct

- Submitting someone else's work as your own is **plagiarism**.
- So is helping someone else to submit your work as their own.
- We report **all cases** of academic misconduct to the university.
- The university takes academic misconduct **very seriously**. Possible consequences:
    - Zero on the assignment or exam (virtually guaranteed)
    - Zero for the course
    - Permanent notation on transcript
    - Suspension or expulsion from the university

Academic Conduct: LLMs

- Submitting someone else's work as your own is **plagiarism**.
  - **Including work by an LLM**
  - Including but not limited to ChatGPT, Lex, Page, DALL-E, Google BARD/Gemini, Microsoft Bing/Copilot, DeepSeek, Qwen, Llama, ...
- If you use an LLM (including LLM-driven code-completion such as Copilot) you must **cite it** (just like any other external source like StackOverflow etc.)
- LLMs often repeat themselves
  - Two students who use an LLM will look like they copied

Prerequisites

- Prerequisites: one of STAT 141, 151, 235, or 265, or SCI 151.
  Corequisites: CMPUT 204 or 275.
- Comfort with or interest in formal, **mathematical/algorithmic reasoning**
- Basic **probability**: random variables, expectations, conditional probability.
  (There will be a refresher lecture)
- Basic **calculus**: gradients, partial derivatives & **linear algebra** (matrix product, inner product)
  (refresher lecture for this too)
- Basic **graph theory**: Nodes, edges
- Ability to program in **Python**
  - Most assignments will have a programming component
  - TAs will run a refresher session

Course Topics

1. Search
2. Reasoning Under Uncertainty
3. Supervised Learning
    - Deep Learning
4. Reinforcement Learning
5. Multiagent Systems

1. Search

Example: Farmer's raft

A farmer needs to move a hen, fox, and bushel of grain from the left side of the river to the right using a raft.

- The farmer can take one item at a time (hen, fox, or bushel of grain) using the raft.
- The hen cannot be left alone with the grain, or it will eat the grain.
- The fox cannot be left alone with the hen, or it will eat the hen.

- We want to compute a sequence of actions:
  - from a **starting state** (all of the animals on the left bank)
  - to a **goal state** (all of the animals on the right bank)
  - while satisfying **constraints** (nothing gets eaten)
- Every action has a **known** and **deterministic** result and cost
- **Search:** efficiently compute a cost-optimal solution based on known rules

2. Reasoning Under Uncertainty

Example: When to harvest?

A farmer has planted a crop which is vulnerable to a certain type of pest

- The pest typically arrives somewhat after a run of rain (how much varies)
- The crop benefits from watering, especially right before harvest
- Rain typically appears in runs (length varies)
- Harvesting too soon means less yield; harvesting after pests means total failure

- We want to choose an **action**, but:
  - Consequences are **uncertain** but have **known probabilities**
  - Observations of one event (rain) can predict others (pests) in complicated ways
  - Different outcomes have different **costs**
- **Reasoning under uncertainty:** Use known models of probabilities to compute a solution that has an optimal **on-average** performance

3. Supervised Learning

Example: Is this food contaminated?

A spectrophotometer can measure a great many features of a grain sample

- These features can be predictive of grain contamination
- We have lots of samples of contaminated and uncontaminated grain

- We want to **learn** an unknown **predictive model**
- We don't know how the different observables related to each other, but
  - We have a lot of **examples**
  - We might have **qualitative** information about how they are related
- **Supervised learning:** Derive a predictive mapping from input **features** to output **predictions** based on examples
  - **Deep learning:** Special case where the learned model is a deep neural network

4. Reinforcement Learning

Example: Inventory management

A warehouse needs to keep a certain amount of stock on hand in order to send to retailers

- There is a delay between when it orders from suppliers and when it receives shipments
- Order volume from retailers is unpredictable and depends on demand
- Keeping unsold inventory is costly

- We need to choose a sequence of actions, but
  - We don't know the outcome of the action until we take it
  - **Rewards** (e.g., selling some stock to a retailer) may arrive a long time after actions that caused them (e.g., ordering from supplier)
  - The environment changes after each action
  - We don't know what a good policy is, and we might not have any examples
- **Reinforcement learning:** A way to learn a good **policy** for acting sequentially

5. Multiagent Systems

Example: What crop to plant?

Two farmers need to choose which crop to plant: a staple crop or a cash crop

- Cash crop is very lucrative, but only if a single farmer plants it (the market for it is small)
- If both plant cash crop, they will both make much less money

- Need to choose an action (or series of actions), but:
  - Outcomes depend on the actions of other agents
  - Other agents' outcomes depend on our actions
- **Multiagent systems:** Framework for understanding how to achieve goals and preferences when interacting with **other agents** who have **own goals and preferences**

Summary

- **Artificial Intelligence:** Accomplishing tasks without specifying how
- **Course information:** https://jrwright.info/introai/
- **Lectures:** Tuesdays and Thursdays, 9:30-10:50am in **ETLC E2-001**
- **Survey class** of core areas of artificial intelligence:
  1. Search
  2. Uncertainty
  3. Supervised Learning (including deep learning)
  4. Reinforcement Learning
  5. Multiagent Systems

AI Seminar

What: Great talks on cutting-edge AI research
External (e.g., other universities, DeepMind, local startups) and internal speakers

When: Fridays at noon

Website: sites.google.com/ualberta.ca/ai-seminar/

Announcements: Sign up for mailing list (bottom of webpage)