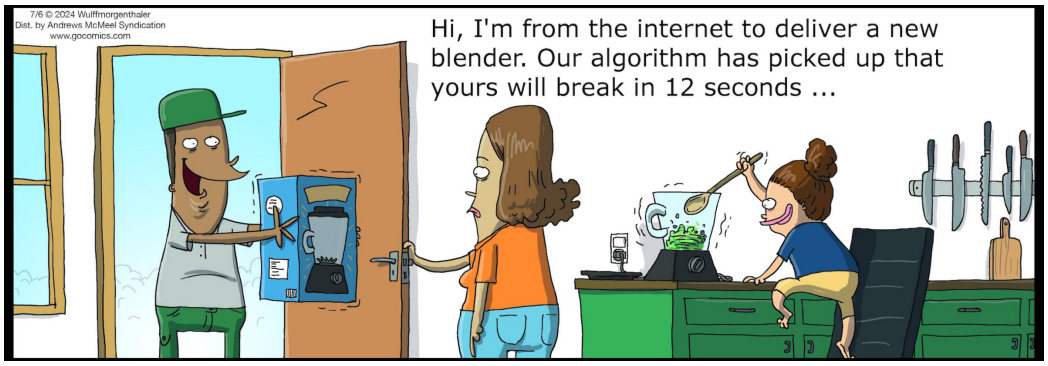
CSCI 3202

Lecture 1 Notes

September 22, 2025



Course Topics

In this course, we will study modeling and probability as a way predict complex outcomes. Our topics for this semester include:

- **Search algorithms:** Survey of modern search algorithms through a discrete data set to find an optimal solution. This includes algorithms that evaluate the entire search space, as well as algorithms that can be used to prune the search space and reduce run-time complexity.
- **Search heuristics:** Methods for designing heuristic methods that can be used to find approximate solutions to problems.
- **Bayes nets and probabilistic reasoning:** Introduction to Bayesian reasoning and calculating probability of outcomes given a set of state variables. This also includes the probability of an event given an existing set of events that has already occurred.
- **Game trees:** Introduction to search methods when there are two players with conflicting objects influencing the search space.
- **Markov decision processes:** Introduction to algorithms for navigating a search space given a set of rules and current state, and how these algorithms can be applied to accomplish a task.
- **Hidden Markov models:** Introduction to inference of a hidden state that is unobserved given observed data.
- **Reinforcement Learning and Q Learning:** Methods or learning from data that are particularly useful with Markov models

- **Machine Learning:** Using decision trees, construct a model that allows you to train your model on data and determine the probability of a particular result
- **Deep Learning:** Using neural networks and deep learning, construct a model that predicts complex behavior such as identifying the type of animal in a photograph or recognizing handwriting
- **Large Language Models:** Understand how Large Language Models use many of the techniques above to predict complex patterns after being trained on large amounts of data

Course

- The course will use Canvas, Piazza and Jupyter/JupyterLab/JupyterHub
- The class is taught in Python using Jupyter Notebooks as a way to combine code, text and images
- You may use any IDE, but assignments will be given out and turned in using a Jupyter Notebook
  - I have written and run the entire course on a Jupyter Notebook
  - For me, the biggest issue with Jupyter Notebooks is the lack of a debugger
  - You can use print() statements to debug if you'd rather not learn a new IDE
  - There is a new version of Jupyter Notebooks that includes a debugger, Version 7. If you don't have it, you may want to upgrade
  - JupyterLab is a newer variant of the Notebook, which includes a debugger
  - The Computer Science department has a server with JupyterLab running on it
  - As a student in this class, you can use the department server for free
- The Anaconda distribution includes Jupyter Notebooks, JupyterLab and most of the packages you will need in this class
  - Anaconda requires ~5 Gb of space on your SSD/HDD
  - If you want to use Anaconda, there is a *tiny* "skip registration" link that allows you to download without giving up your E-Mail address
- Many IDEs will support Python, but can't read or write Jupyter Notebooks
  - IDEs that support Jupyter Notebooks are:
    - PyCharm
    - Visual Studio Code
    - Google CoLab
- You will need to download files from a GitHub repository to do many of the homework problems in this class. You do not need set up version control, only download the necessary files
- If you are in doubt about what to do, download and install Anaconda then run

either a Jupyter Notebook or JupyterLab

Python Overview

- A Python Overview is available on Canvas
- We have posted a list of Python Resources on Canvas
- See Notebook, A Brief Intro to Python.ipynb
- We will not discuss Python programming in class

Class Details

- Text: Artificial Intelligence a Modern Approach, 4th Edition by Stuart Russell and Peter Norvig
  - You may use either the US or an International Version
  - You may use the 3rd edition if you prefer, but you are responsible for translating the readings, figures and algorithm numbers
  - The text is available digitally through the CU Bookstore Book Access program or online from a variety of sources
- The class has 9 HW assignments, a project, a midterm and a final
  - After each topic, there is a HW assignment
  - Students will code the algorithm, test it and solve a problem using real-world data, and answer questions about the efficiency or use of the algorithms
  - Midterm and Final are in-class
  - Project is partially graded via interview grading
- Readings from the text are regularly assigned. You are responsible for reading them *before* coming to class
- The readings often go beyond what we will cover it class
- The Syllabus and Class schedule are available on Canvas. Please review them and ask any questions you may have in our class on Wednesday.
- Assignments
  - Assignments in this class can be written, programming, or in-class exercises Assignments are are due at 11:59 pm on the day listed
  - Assignments received after this time are late and will be penalized as indicated below
  - All assignments submitted the due date are penalized 5% per day for 1-3 days and then will not be accepted
  - Missing assignments will be assigned a 0
- Late Policy
  - Assignments may be submitted up to 3 days late for a penalty of 5% per day

    - Example: An assignment is due Wednesday at 11:59 pm
        - If it is submitted on Thursday, it will be penalized 5%
        - If it is submitted on Friday, it will be penalized 10%
        - If it is submitted on Saturday, it will be penalized 15%
        - If it is submitted later than Saturday, it will not be graded
- Python/Jupyter
    - All programming assignments will be written in Python 3, and all assignments will be submitted as Jupyter notebooks
    - We will make extensive use of objects in Python
    - If you're not familiar with Python, you will need to learn it
        - If you are not familiar with Jupyter notebooks and Markdown, embrace this opportunity to learn tools that are ubiquitous in the fields of AI/ML/data science
- Canvas
    - We will use Canvas to distribute notes and class materials, submit homework assignments electronically and as a gradebook: https://canvas.colorado.edu/courses/122390
    - We will use Piazza for class discussion and management. The system is highly catered to getting you help efficiently from classmates, the graders, and the instructor
    - Rather than emailing me questions, I request that you post your questions on Piazza. If your question is of a private nature, Piazza allows you to send private messages to the instructors. It is your responsibility to check the web page on a regular basis.
    - There is a link to the course Piazza site on Canvas, or you can access it directly at: https://piazza.com/colorado/fall2025/csci3202/home
    - Before you can use Piazza, you will need to sign up using the following link: https://piazza.com/colorado/fall2025/csci3202
- Course lectures are recorded and will be available approximately 1-3 hours after class through the "Course Videos" the Canvas left navigation menu
- We have 2 TAs, 1 Grader and 2 Course Assistants for the class
- Office hours are posted on Canvas under Course Information > CSCI 3202 Fall 2025 Office Hours. Any changes to the schedule will be sent out in an announcement and noted on Canvas

Course work

- Course work is divided into 5 component: homework, midterm exam, project, quizzes / participation and final exam. Each part is component is weighted in computing your final average

Component	Percent of Final Grade
Homework	35%
Midterm Exam	15%
Project	20%
Quizzes / Participation	10%
Final Exam	20%

- Homework
  - Homework will be assigned roughly every two weeks throughout the course.
  - We will use Jupyter notebooks for all homework assignments.
  - You are expected to write up your solutions with full explanations and justifications, where appropriate, and your code must work.
  - Your lowest homework score will be dropped.
- Midterm and Final will be in-class consisting of multiple choice and programming problems
  - Exams are open book
- Quizzes / Participation. Beginning Friday, August 29, we will have in-class Canvas quizzes covering the readings and lectures
  - Quizzes will mainly be multiple choice, covering reading and lectures
  - You must be in class to participate in the quiz
  - Quizzes are designed to get you to attend class
- Project. You will have a approximately a month to complete a final project based on the material from this class
  - You will code an implementation and write an explanation of an algorithm you choose or one we assign
  - You may work in teams on this project

Schedule

Week	Date	Day	Topic	Readings	Quiz	HW	HW Due	Project
1	**8/22**	**F**	Introduction	Syllabus, Schedule	**Survey**			
2	**8/25**	**M**	Agents	Ch 1		HW 1		
		W	State	Ch 2				
		F	Trees, Graphs, BFS	3.1-3.3	**Quiz 1**			
3	**9/1**	**M**	**Labor Day Holiday**					
		W	DFS, UCS	3.3-3.4		HW 2	HW 1	
		F	A*, Greedy	3.5	**Quiz 2**			
3	**9/8**	**M**	Heuristics	3.5				
		W	Admissibility	3.6			HW 2	
		F	Search Review	3.1-3.6	**Quiz 3**	HW 3		
4	**9/15**	**M**	Probability Review	Class Notes, 12.1, 12.2				
		W	Optimization: Hill Climbing	4.1				
		F	Beam Finding, Simulated Annealing	4.1	**Quiz 4**		HW3	
5	**9/22**	**M**	Bayes Networks	13.1		HW 4		
		W	Bayes Networks	13.2-13.3				
		F	Conditional Independence	12.4	**Quiz 5**			
6	**9/29**	**M**	Solving Networks	13.3		HW 5	HW 4	
		W	Bayes Examples	Class Notes				
		F	Bayes Summary	Class Notes	**Quiz 6**			
7	**10/6**	**M**	Review				HW 5	
		W	**Midterm**					**In class Midterm**
		F	Games	5.1				
8	**10/13**	**M**	Minimax	5.2				
		W	Alpha-Beta Pruning	5.2		HW 6		
		F	Markov Processes	17.1	**Quiz 7**			Assigned
9	**10/20**	**M**	Markov Decision Models	17.2				
		W	Value and Policy Iteration	172			HW 6	Topic, Team
		F	Q-Learning	17.3	**Quiz 8**			
10	**10/27**	**M**	Reinforcement Learning	22.1		HW 7		
		W	Reinforcement Learning	22.2				
		F	Hidden Markov Models	14.2	**Quiz 9**			
11	**11/3**	**M**	Hidden Markov Models / Viterbi	14.3			HW 7	
		W	Learning in Probabilistic Models	14.1		HW 8		Intermediate Report
		F	Machine Learning	20.1	**Quiz 10**			
12	**11/10**	**M**	Decision Trees	19.1, 19.2				
		W	Goodness of Fit, Cross Validation	19.3			HW 8	
		F	Model Fitting Strategies	19.4	**Quiz 11**	HW 9		
13	**11/17**	**M**	Deep Learning	21.1, Class Notes				**Project Due**
		W	Video Classification	25.1-25.4				Interview Grading
		F	Natural Language Processing	23.1, 23.3, 23.5	**Quiz 12**			Interview Grading
14	**11/24**	**M**	**Fall Break**					
		W	**Fall Break**					
		F	**Fall Break**					
15	**12/1**	**M**	Large Language Models	24.1-24.4			HW 9	Interview Grading
		W	Summary	Class Notes				Interview Grading
		F	**Review**					Interview Grading
	12/6	**Sa**	**Reading Day**					
	12/7	Su	Reading Day					
Final	**12/11**	**Th**	**Final Exam**	**1:30-4:00 pm**				**In class Final**

- Schedule may change
- All changes will be posted as Announcements in Canvas

Grading

Item	%	Drop	Notes
Homework	30	Lowest 1	9 Total
Project	20		
Quizzes	10	Lowest 2	Weekly on Friday at End of Class
Midterm	20		In Class Only
Final	20		In Class Only
Total	100		

Homework due at 11:59 pm	Penalty
On Time	0%
1-3 days late	5% per day
> 3 days late	Not Graded

Example:	
Due on Wednesday	Penalty
Turn in on Wednesday	0%
Turn in on Thursday	5%
Turn in on Friday	10%
Turn in on Saturday	15%
Turn in on Sunday	Not Graded

- Grades will be available in Canvas
- We allow assignments to be up to 3 days late with a penalty of 5% per day
- All assignments are due in Canvas by 11:59 on the due date
- We will drop your lowest HW and lowest 2 quizzes before calculating your final grade

Academic Dishonesty

- We expect that students in this class will follow the Honor Code and Student Code of Conduct, https://www.colorado.edu/sccr/students/honor-code-and-student-code-conduct
- We will report all violations to the office of Student Conduct and Conflict Resolution for their resolution
- In addition to any actions taken by the Student Conduct and Conflict Resolution office, we may do one or more of the following depending on the type and extent of dishonesty detected:
  - Give you a 0 on the question(s) where we detect academic dishonesty
  - Give you a 0 on the assignment(s) or exams where we detect academic dishonesty

    - Give you an F for the course
- We consider work submitted under your name that was performed by another person or Artificial Intelligence agent or obtained from a website as academic dishonesty, even a code snippet
- You may discuss problems or homework with your classmates, but all code or solutions that you turn in must be your own
- Turning in work from ChatGPT or CoPilot as your own is clearly academic dishonesty
- If you use work from another person, AI agent or website, you must give proper attribution
- We will grade the work you have added to that work and grade you on your contribution. If you did nothing other than enter a prompt, you will receive a 0 for your work
- If you have any questions, please see your instructor or TA

Disability

- If you qualify for accommodations because of a disability, please submit your accommodation letter from Disability Services to your instructor in a timely manner so that your needs can be addressed.

Religious Holidays

- If you have conflicts with scheduled exams, assignments or required attendance due to religious holidays, please E-Mail the instructor with requests for accommodation.

Introduction

- See Lecture 1 Slides

Artificial Intelligence

There are two major branches of Artificial Intelligence. One branch uses probability and mathematics to model complex processes and make predictions about their outcome. Examples of this include computers playing chess or movie recommendations from Netflix. The other branch uses mathematics and other disciplines to build models of complex processes, then builds an algorithm to make predictions by examining thousands (or more) of examples using those models in a process called training. This branch of artificial intelligence is called machine learning. Examples are facial recognition or fraud detection on your credit cards.

- Artificial Intelligence is generally where programmers generate a series of rules to perform a particular task then write code to follow these rules
  - This is sometimes called "software engineering"
  - Example: Computer Chess
    - Teach computer the basic rules
    - Evaluation function
    - Evaluation tree
    - Database of beginning positions (opening book)
    - Look at moves and response to as many levels as possible
    - Database of end positions

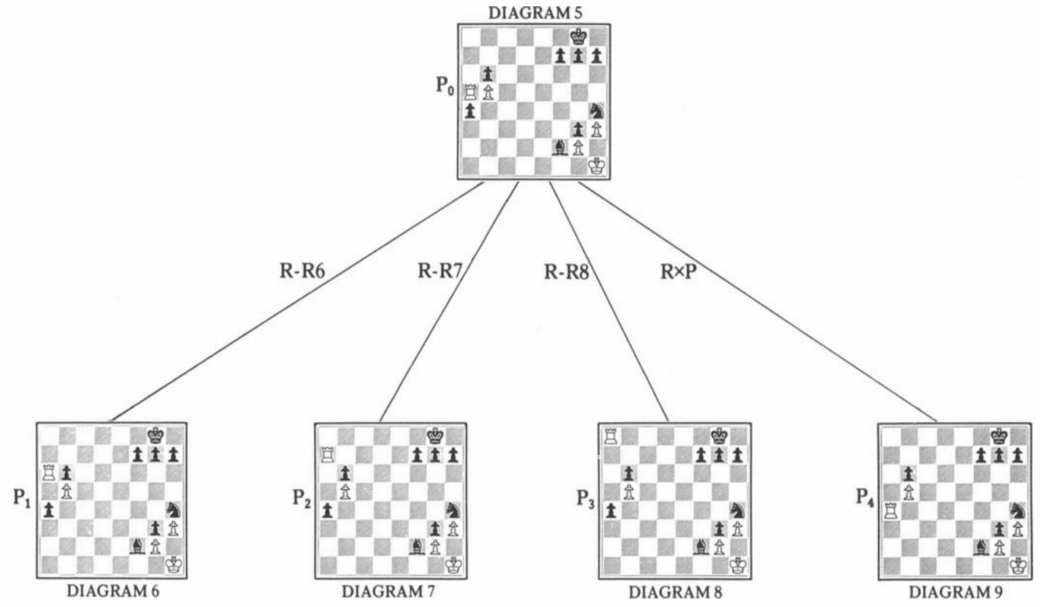


Fig. 19a
White has a choice of four moves.

From Levy, et al. (1982)

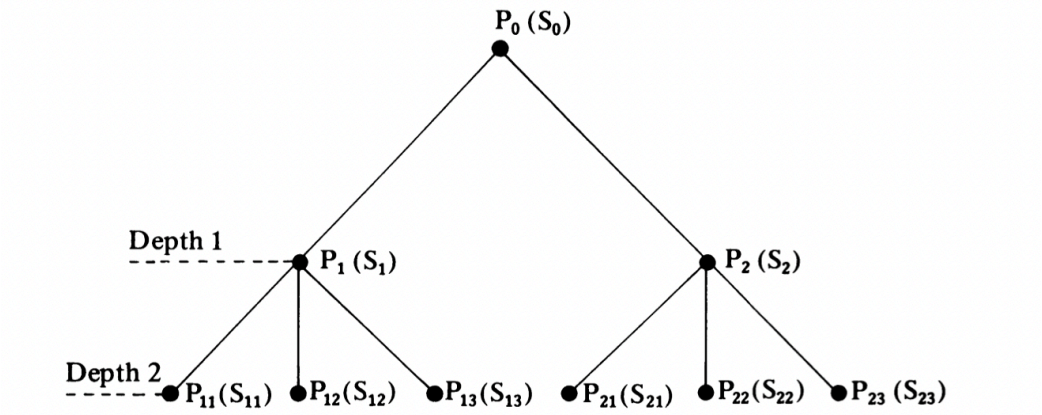


Fig. 20
A tree of depth two "half-moves".

From Levy, et al.(1982)

- Machine Learning
  - Also known as Deep learning
  - Give the computer massive amounts of data and let it generalize from the data
  - Example: Facial Recognition

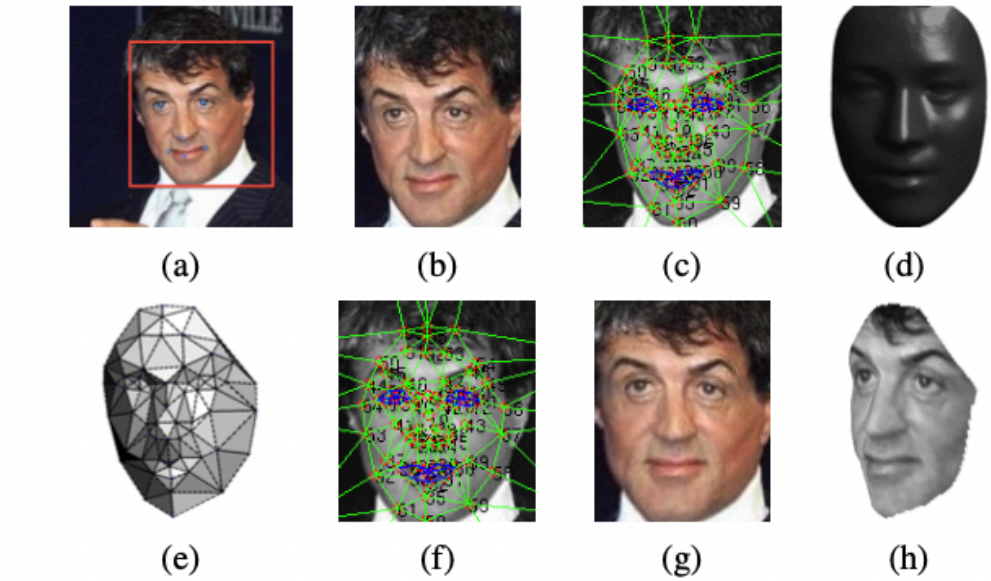


Figure 1. **Alignment pipeline.** (a) The detected face, with 6 initial fiducial points. (b) The induced 2D-aligned crop. (c) 67 fiducial points on the 2D-aligned crop with their corresponding Delaunay triangulation, we added triangles on the contour to avoid discontinuities. (d) The reference 3D shape transformed to the 2D-aligned crop image-plane. (e) Triangle visibility w.r.t. to the fitted 3D-2D camera; darker triangles are less visible. (f) The 67 fiducial points induced by the 3D model that are used to direct the piece-wise affine warpping. (g) The final frontalized crop. (h) A new view generated by the 3D model (not used in this paper).

From Taigman, et al. (2014)

Limits on Artificial Intelligence and Machine Learning

- Training Bias
  https://www.theguardian.com/technology/2018/oct/10/amazon-hiring-ai-gender-bias-recruiting-engine
  - Amazon built a software system to help select the best candidates to interview from the many resumes they receive
  - They trained the system using the previous 10 years of resumes they had received
  - Most of the resumes were from men
  - The words and phrases used by men were different
  - The system favored me in its selection

- Facial Recognition and Race
  https://www.bbc.com/news/technology-50865437
  - Many facial recognition systems are less accurate for people with dark skin color
  - Some of this is due to the way the software works
  - Some of it is due to the fact that the software is trained predominantly on while and Asian faces
- X's Grok Bot Creates a False Story about NBA Player
  https://www.sfgate.com/warriors/article/klay-thompson-accused-brick-vandalism-spree-ai-19407952.php
  - Golden State Warriors player Klay Thompson shot 0 for 10 in an NBA game against the Golden State Warriors in April 2024
  - Sports commentators wrote about him "shooting bricks" against the Warriors
  - Based on these reports, X's Grok Bot created a story stating that several people had their homes damaged by Thompson with windows broken out with bricks
- Microsoft Tay Chatbot Became Offensive
  https://www.huffpost.com/entry/microsoft-tay-racist-tweets_n_56f3e678e4b04c4c37615502
  - Learned from users
  - Some users deliberately fed it offensive material
  - AI didn't know how to distinguish offensive from inoffensive
- Uber Self Driving Car Fatality
  https://www.nbcnews.com/tech/tech-news/self-driving-uber-car-hit-killed-woman-did-not-recognize-n1079281
  - Person killed was crossing the street in the middle of a block, carrying a bicycle during the daytime
  - Software did not recognize them as a person because of the bicycle
  - Software was trained that people cross streets at the intersections
  - Car had an operator in the driver's seat who was ultimately held responsible
- ICT's Intelligent Camera Operator Tracks Incorrectly
  https://devcount.com/ai-fails/
  - Camera mistook player's bald head for the soccer ball at certain angles
  - Camera tracked the player's head, not the ball
- Most AI and Machine Learning failures are related to data quality issues
  - Biased sample (mostly men, mostly white)
  - Not enough data for some cases

    - Incorrect data (pedestrians cross at intersections)
- Need to think carefully how the system will be used then train or test based that usage

Administrative

- Read the Syllabus and review the course schedule before coming to class on Wednesday
- Sign into Canvas
- If you don't have it already, download and install Anaconda or create an account on the CS Department JupyterHub
    - Anaconda has everything you need for this class already installed, but takes ~5 GB of space
    - The Anaconda installation is s-l-o-w
    - CS Department JupyterHub will retain your files for 1 year after the class ends
    - JupyterHub has all needed software for this class installed
- You can use a different IDE than Jupyter, but all assignments must be turned in as a Jupyter notebook
- Create a Piazza account for this course
    - There is a link to the course Piazza site on Canvas, or you can access it directly at: https://piazza.com/colorado/spring2025/csci3202/home
    - Before you can use Piazza, you will need to sign up using the following link: https://piazza.com/colorado/spring2025/csci3202
- We recommend that you use Piazza to ask questions. Others will likely have the same question

Next Time

- Read chapters 1 and 2 in the text
- We will discuss Agents
- HW 1 on Python will be available on Mon before class. It is due on Wed, 9/3

REFERENCES

- Levy, D., & Newborn, M. (1982). How computers play chess. In All About Chess and Computers (pp. 24-39). Springer, Berlin, Heidelberg.
- Russell, S., & Norvig, P. (2021). Artificial intelligence: a modern approach, 4th. Foundations, 19, 23.
- Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1701-1708).