CSC 480/580: Principles of Machine Learning

Jason Pacheco

Department of Computer Science

THE UNIVERSITY OF ARIZONA

What is machine learning?

What is machine learning (ML)?

- **Tom Mitchell** established Machine Learning Department at CMU (2006).

Machine Learning, Tom Mitchell, McGraw Hill, 1997.

Machine Learning is the study of computer algorithms that improve automatically through experience. Applications range from datamining programs that discover general rules in large data sets, to information filtering systems that automatically learn users' interests.

This book provides a single source introduction to the field. It is written for advanced undergraduate and graduate students, and for developers and researchers in the field. No prior background in artificial intelligence or statistics is assumed.

- In short: algorithms adapt to data
- A subfield of **Artificial Intelligence (AI)** – computers perform "intelligent" tasks.
- Classical AI vs ML: rule-driven vs. data-driven approaches

Traditional AI vs Machine Learning (ML)

- **Traditional AI**: you encode the knowledge (e.g., logical rules), and the machine makes '**inference**', e.g. given ``a -> b and b-> c'', deduce ``a-> c''.
  - Example rule: has-feather-texture(object) and has-beak(object) -> is-bird(object).
  - Deductive reasoning (top-down)

- **ML**: given a number of input and output observations (e.g., animal picture + label), output a **function (can be a set of logical statements or a neural network)** that maps the input to the output accurately.
  - "Big data" setting => better to learn from data than to encode domain knowledge manually.
  - "statistical" / data-driven approach – inductive reasoning (bottom-up)

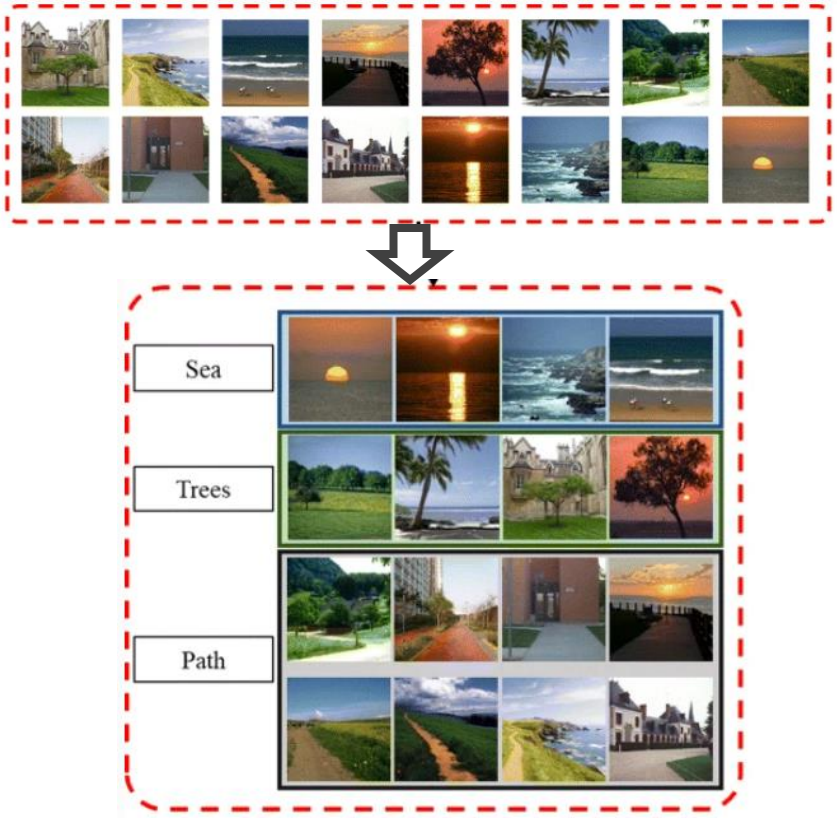
- **Note**: Traditional AI and ML can well work synergistically

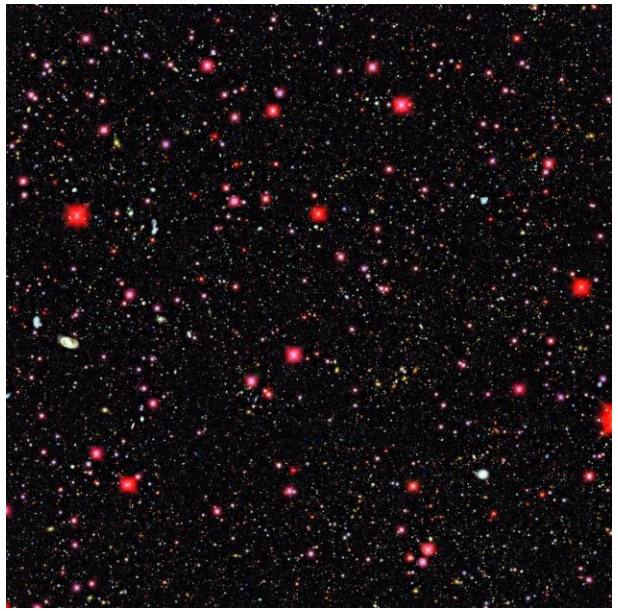
Interpretability Rules: Jointly Bootstrapping a Neural Relation Extractor with an Explanation Decoder

Zheng Tang, Mihai Surdeanu
Department of Computer Science
University of Arizona, Tucson, Arizona, USA
{zhengtang, msurdeanu}@email.arizona.edu

ML Task 1: Image classification

- Predefined categories: C = {sea, trees, path, …}
- Given an image, classify it as one of the categories in C with the highest accuracy as possible.
- **Use:** sorting/searching images by category.
- Also: categorize types of stars/events in the Universe (images taken from large surveying telescopes)





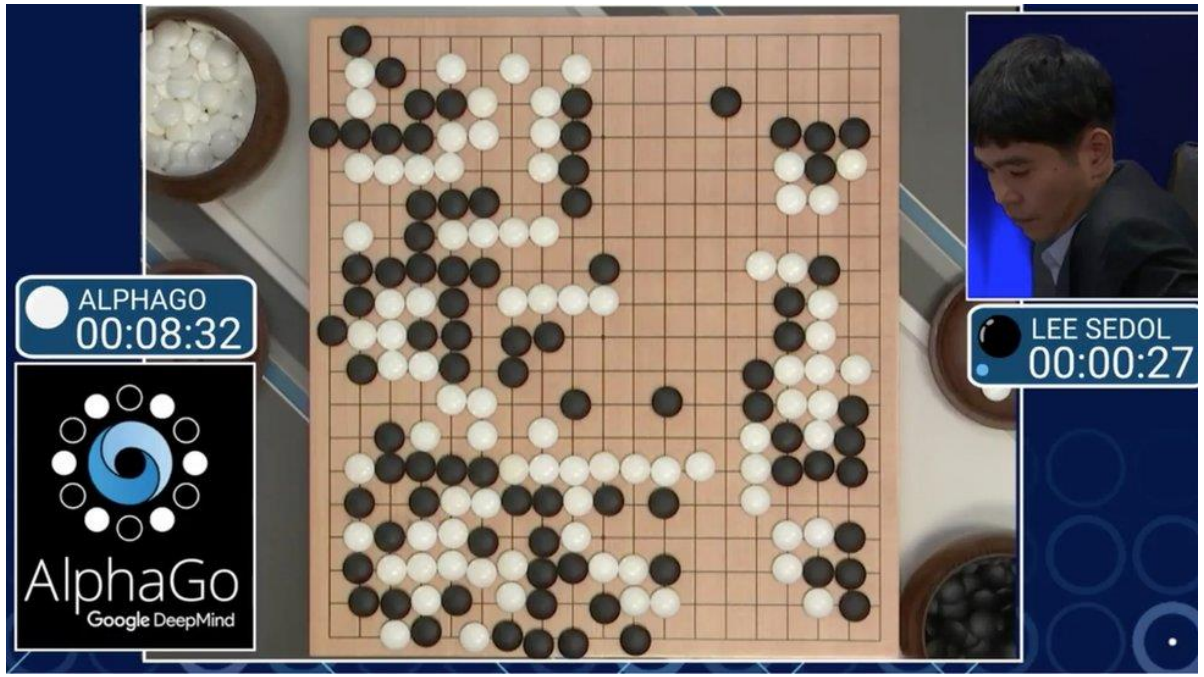
ML Task 2: Recommender systems

- Collaborative Filtering: Predict how user would rate a movie based on historical ratings
- **Use**: For each user, pick an unwatched movie with the high predicted ratings.
- **Possible approach**: compute user-user similarity or movie-movie similarity, then compute a weighted average.

	User 1	User 2	User 3
Movie 1	1	2	1
Movie 2	?	3	1
Movie 3	2	5	2
Movie 4	4	?	5
Movie 5	?	4	2

ML Task 3: games

- Predict win probability of a move in a given game state (e.g., AlphaGo)
- Traditionally considered as a “very smart” task to perform.
- **Use**: Professional go playing, leisure



Impact of Go AI on the professional Go world

ML Task 4: Generic Q&A system

- ChatGPT!
- **Use**: education, coding, appointment booking, entertainment, ..
- Good at retrieving knowledge and presenting it in natural language.
- ~~Not necessarily good at difficult tasks (e.g., reasoning).~~

JUL 21

OAI and GDM announce IMO Gold-level results with natural language reasoning, no specialized training or tools, under human time limits

Rank	First Name	Last Name	ID	Team	so...	tri...	w...	Inter...	fe...	mi...	ob...	Inter...	Global
1	Hengxi	Liu	CHN4		100	100	100	300	100	91.23	100	291.23	591.23
2	Mingyu	Woo	KOR1		100	99.33	93	292.33	100	82.45	100	282.45	574.78
3	Sizhe	Fan	CHN3		100	78.11	100	278.11	100	91.23	83	274.23	552.34
4	Rareș-Andrei	Neculau	ROU3		100	77.67	100	277.67	100	85	83	268	545.67
5	Xinyang	Chen	CHN1		100	99.02	100	299.02	66	86.4	83	235.4	534.42
–	OpenAI				100	75.29	93	268.29	100	65	100	265	533.29
6	Rain	Jiang	USA1		100	75.41	86	261.41	100	65	100	265	526.41
7	Ryan	Bai	CAN1		100	78.51	100	278.51	66	79.11	100	245.11	523.62

Work in ML

- ML skillset: complement traditional CS background – especially mathematical side, beyond discrete math.

- Applied data scientist
  - Collect/prepare data, build/train models, analyze errors
- ML engineer
  - Implement/fine-tune ML algorithms and infrastructure
- ML researcher
  - Design/analyze models and algorithms
  - Theory: Provide mathematical guarantees. E.g., If I want to achieve 90% accuracy, how many data points do we need? => sample complexity (studied in depth in CSC 588)

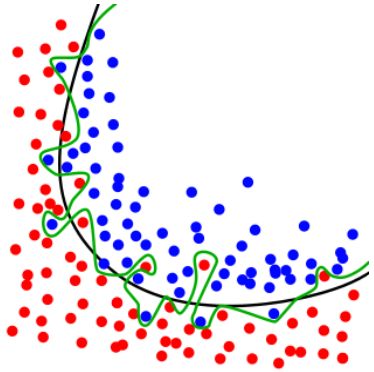
Prerequisites

- Math
  - linear algebra, probability & statistics, multivariate calculus, reading and writing proofs.
  - Quick show of hands: how many of you know the terminology “eigenvector”?

- Software/programming
  - You need be familiar with at least one programming language
  - You need to be fluent at writing classes and functions and using them efficiently.
  - Much ML work is implemented in python with libraries such as numpy and pytorch.

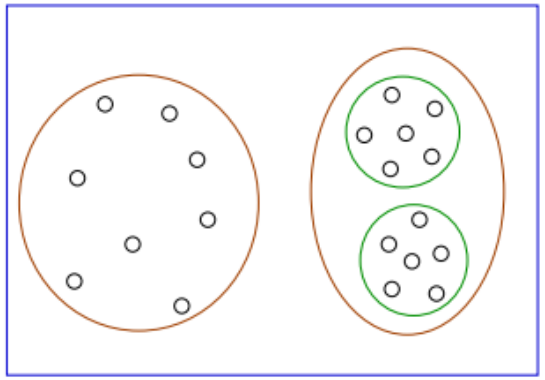
Overview of ML problems

Supervised learning



about 70% of the lecture

Unsupervised learning

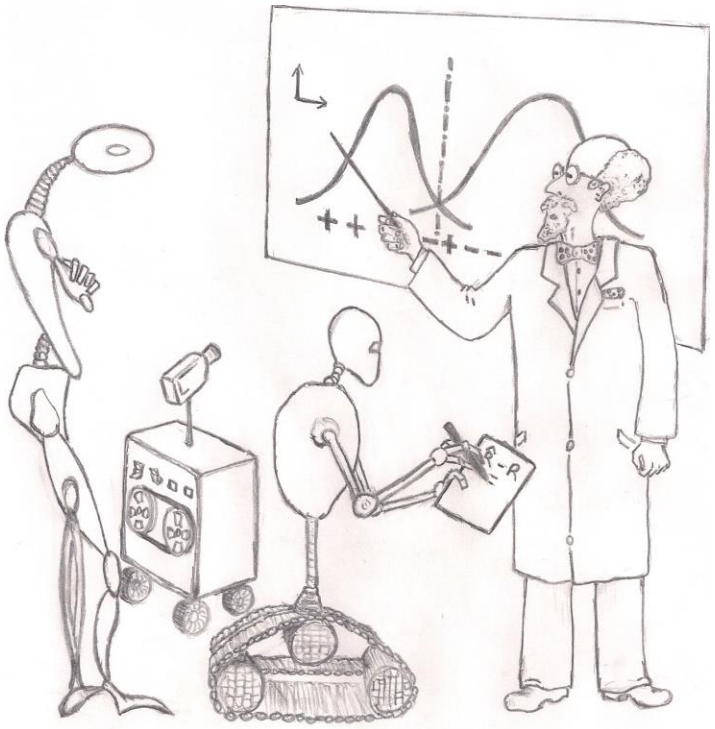


about 25% of the lecture

Interactive learning

about 5% of the lecture

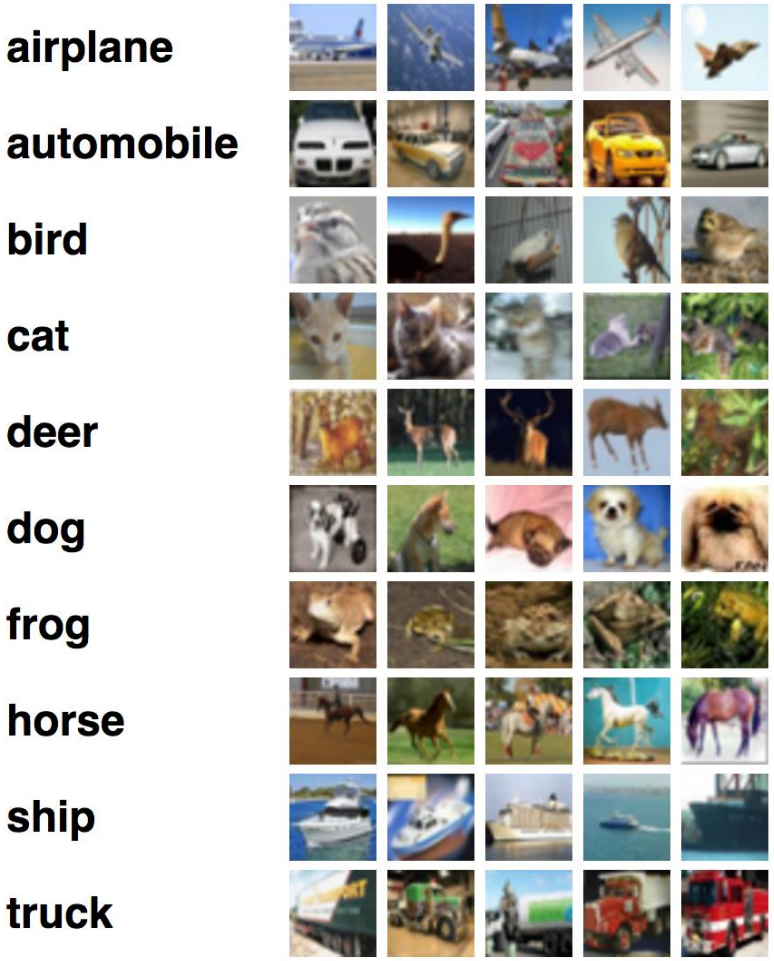
Supervised Learning



Picture Credit: Samory Kpotufe

Basic setting: Supervised learning

- Training data: dataset comprised of *labeled examples*: pairs of (feature, label)



Training data

supervised learning algorithm

training

Test example

function (”classifier”)

cat

test

Example classifier 1: Decision tree

- Task: predict the rating of a **movie** by a **user**
- If age >= 40 then
  - if genre = western then
    - return 4.3
  - else if release date > 1998 then
    - return 2.5
  - else ..

    …

    end if
- else if age < 40 then

  …
- end if

Example classifier 2: Linear

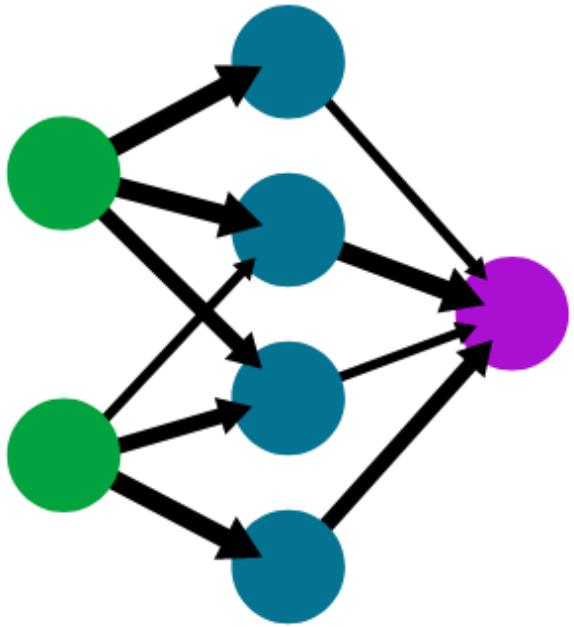
- E.g., Image classification
- Let x be a set of pixel values of a picture (30x30 resolution) => 900-dimensional vector x.
- If $0.124 \cdot x_1 - 2.5 \cdot x_2 + \cdots + 2.31 \cdot x_{900} > 2.12$ then
  - return cat
- else
  - return dog
- end

"linear combination"

- Coefficients: signed "importance weights"

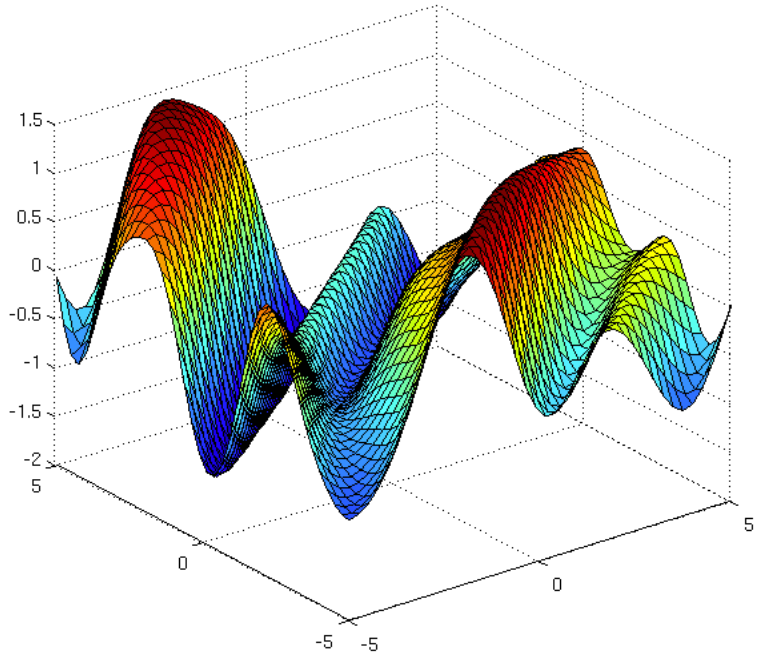
Example classifier 3: Nonlinear

Neural network



(stacked **linear** models with nonlinear **activation functions**)

Kernel classifiers

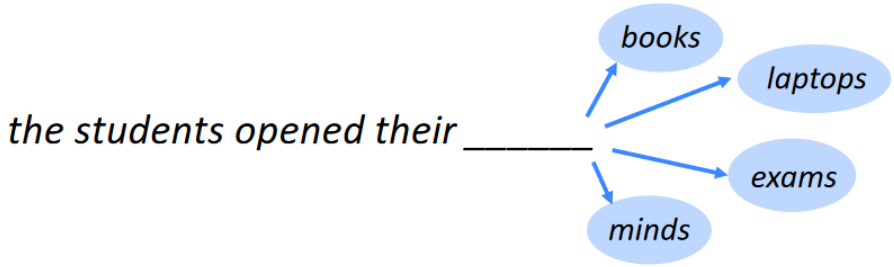


(**linear** in the induced feature space)

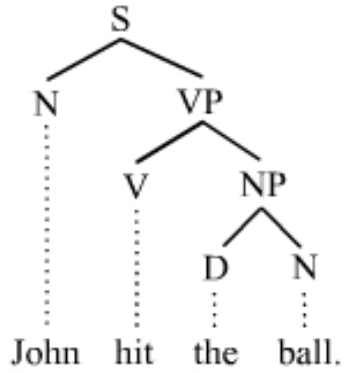
Supervised learning: Types of prediction problems

- Binary classification
  - Given an email, is it spam or not?
- Multi-class classification
  - Given a sequence of words, predict the next word

    (basis of ChatGPT)



- Regression: the label is real-valued (e.g., price)
  - Say I am going to visit Italy next month. Given the air ticket price trends in the past, what would be the price given (the # of days before the departure, day of week)?
- Structured prediction: outputting more than just a label
  - Given a sentence, what is its grammatical parse tree?

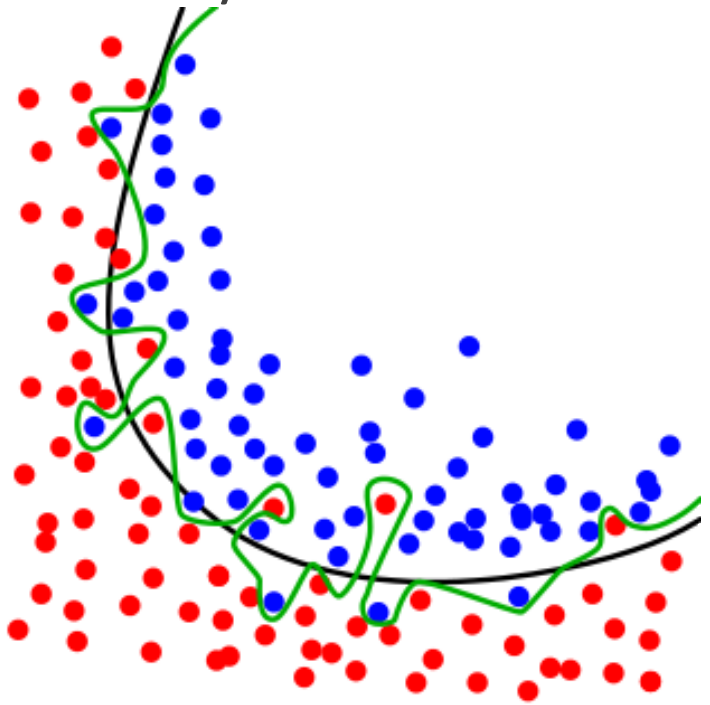


The challenge: generalization

- Why not learn the most complex function that can work flawlessly for the training data and be done with it? (i.e., classifies every data point correctly)
- Extreme: memorization.
  - For a test example, if it exactly matches some training example, output the corresponding label
  - Otherwise, output some default label, say blue
- It does not work. Why?

- Generalization: the ability to predict well on *future unseen examples*
- Need to learn from training dataset, but don't "over-do" it.
- This also applies to *our own learning*

Classify tables vs chairs

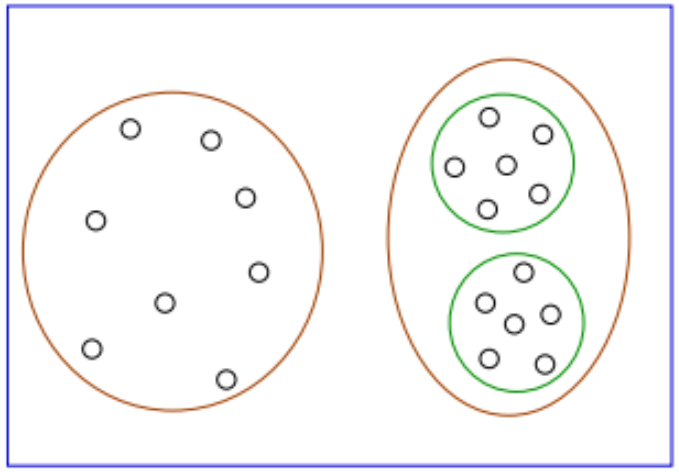


green: memorization
black: optimal decision boundary

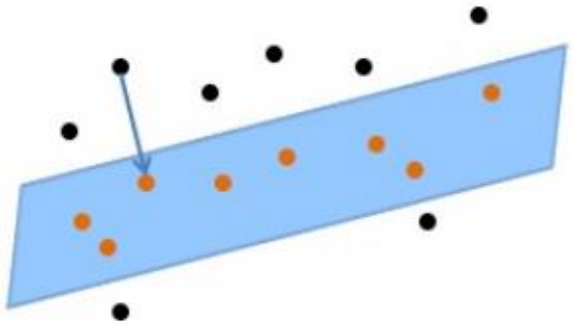
Unsupervised learning

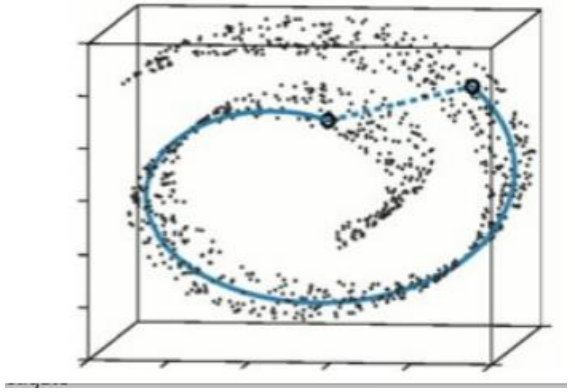
- Finding *structures* in data, e.g.

- Clustering



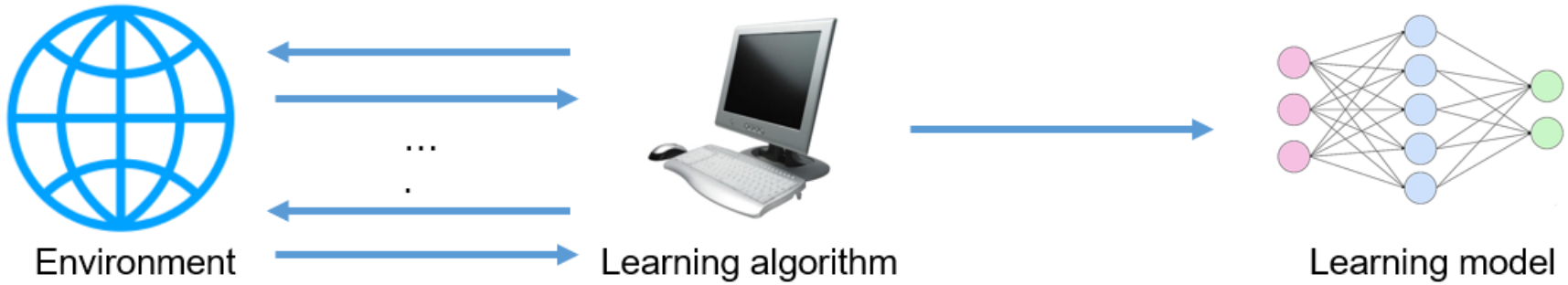
- Dimensionality reduction (e.g. autoencoder)





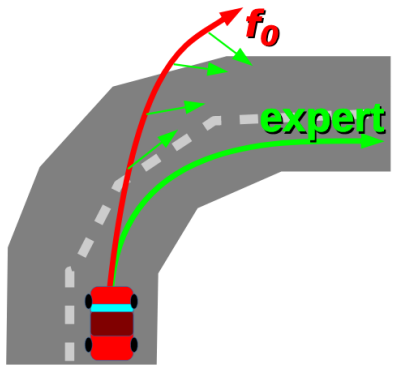
Interactive learning

- Algorithms adaptively collect and analyze data to make predictions / decisions



- E.g.:
- Reinforcement learning

- Imitation learning



What to expect in the class

- ~~How to use sklearn, pytorch, tensorflow, fine-tuning deep net algorithms.~~
  - We covered some of it in 380; you are encouraged to learn these on your own

- Algorithmic and statistical principles
  - Well-studied models and methods.
  - Those that give us some “understanding”.
  - These are and will be referred/extended/revisited in the future.

- Programming and math
  - No need to be a guru.
  - But you must be familiar enough to (1) follow popular codes and math derivations and (2) be able to adapt yourself to new programming tools and math in the future.

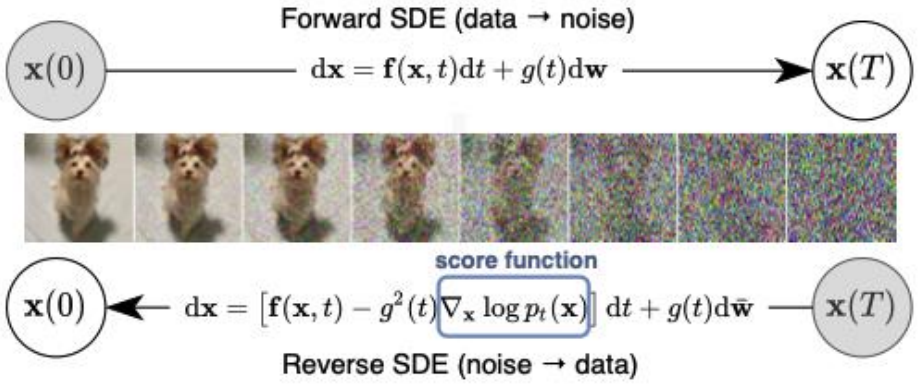
Why basic principles?

- Old fundamental principles are often the cornerstones of modern “fancy” algorithmic developments, e.g.:
- GRPO (Group Relative Policy Optimization, 2025): finetuning LLM using reinforcement learning – basically a variant of REINFORCE (Williams, 1992)

PPO VS GRPO

- Convolutional neural networks (LeCun, 1989): idea dates back to (Fukushima, 1969)
- Diffusion models (Song et al, 2021):

  key idea: score matching (Hyvarinen, 1995)



Logistics

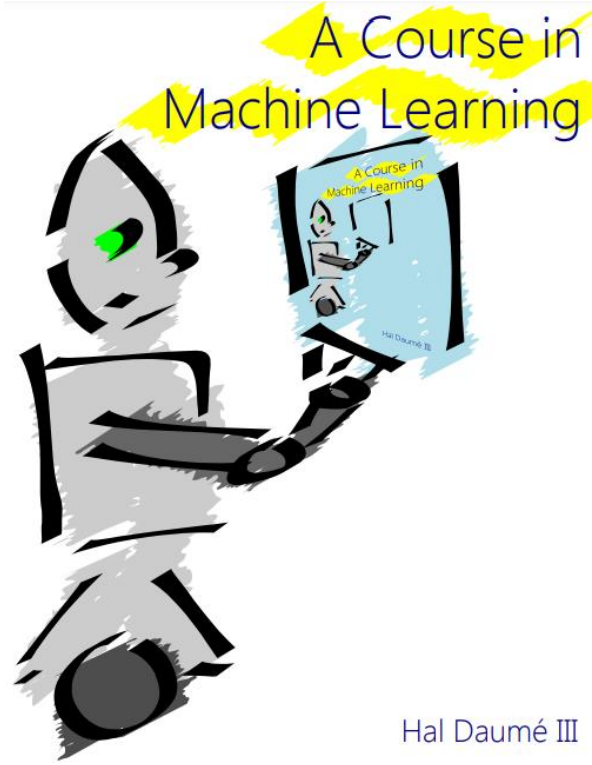
Instructor info

- Jason Pacheco
- Associate Professor at CS @ UA
- Research interests: probabilistic machine learning, uncertainty quantification
- Email: pachecoj@arizona.edu

- **TA:** Emiliano Islas, eislasq@arizona.edu
- **TA:** Mahshad Habibpourparizi, mahshadh@arizona.edu

Course structure

- D2L: for announcements & important email communications & grades
- Piazza: Q&A/discussion.
- Gradescope: for submitting homeworks



- Main book: “A Course in Machine Learning” (CIML) by Hal Daume III
- http://ciml.info/
- Unofficial errata (from Fall 2022): here
- We might also use Moritz Hardt and Benjamin Recht, “Patterns, Predictions, and Actions: A story about machine learning” for supplementary readings
- https://mlstory.org

Electronical Resources

Course Webpage: https://pachecoj.com/courses/csc480-580_fall26/

D2L 580: https://d2l.arizona.edu/d2l/home/1815948

D2L 480: https://d2l.arizona.edu/d2l/home/1815945

Piazza: https://piazza.com/arizona/fall2026/csc480580

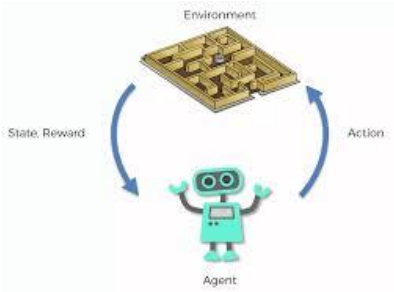
Gradescope 580: https://www.gradescope.com/courses/1364022 **Entry Code: ZPDJDB**

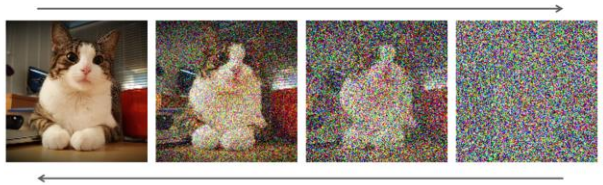
Gradescope 480: https://www.gradescope.com/courses/1364018 **Entry Code: 3ZGDGX**

Instructor Homepage: http://www.pachecoj.com

Syllabus summary

- Warm up
  - Basic supervised learning: decision tree, k-NN, perceptron
  - Practical issues in supervised learning: evaluation, feature selection, etc.
  - Bias-variance decomposition
- Learning methods
  - linear models, kernels
  - naïve Bayes, graphical models
  - neural networks
- Modern paradigms:
  - Large Language Models (LLMs)
  - Unsupervised learning: Autoencoders, Generative Adversarial Networks. Diffusion Models and Flow Matching
  - Reinforcement learning





Syllabus summary

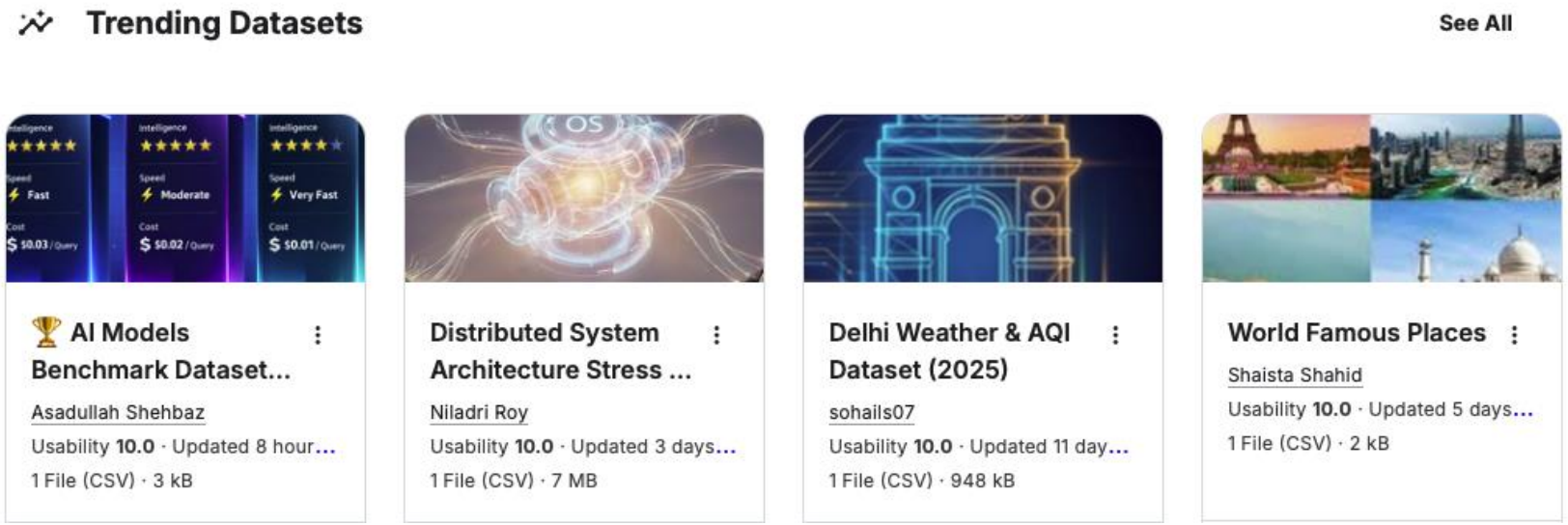
- 08/24: HW0 (calibration) assigned
- 09/02: HW1 assigned
- 09/21: HW2 assigned
- 09/30: Project Proposals assigned
- 10/14: Midterm exam (in class)
- 10/28: HW3 assigned
- 11/16: HW4 assigned
- 12/09: Final project due
- 12/17: Final exam at 1:00pm – 3:00pm

- **Late Policy:** HWs submitted 1 day late will receive 90%, 2 days late 80%, +2 days will not be accepted

- Grading scheme
  - Assignments: 40%
  - Project Proposal: 10%
  - Project Report: 15%
  - Midterm Exam: 15%
  - Final Exam: 20%
- Project
  - Pick a paper in recent ML venue and implement it
  - Pioneering new applications of ML (e.g., connect to your research)
  - Talk to me for other ideas.
  - Will explain 480 vs 580 later.

Project

- Different types of projects:
  - Evaluation of a new method: pick a paper in recent ML venue and implement the algorithm from scratch
  - Applying an existing method to a new dataset: see e.g. Kaggle
    - Or you are welcome to connect to your research or personal projects



  - Talk to me if you have other ideas!

Academic Integrity

- If you discuss high-level ideas with a friend outside group, mention their name in your solution.
- Verbatim copying of solutions from any external source (web search, ChatGPT, etc.) is not acceptable. If you make use of any external source, mention the source in your solution, and make sure that your solution is your own work.
  - We have pretty effective tools that can detect AI-generated texts.. (see this interesting article)
- If we suspect that there is academic integrity violation, we will meet and you may plead your case (worst case: may get an E)
- **Bottom line: as long as your solutions are all your own work, you will be fine ☺**

Generative AI use

- Generative AI (ChatGPT, Bard, ..) may make up or hallucinate information. These tools may reflect misconceptions and biases of the data on which they were trained and the human-written prompts used to steer them.

- You can use it for additional explanations of course material, **but you are responsible for checking facts, finding reliable sources, etc.**

- Usage for course assignments / projects: **only limited to text polishing and refining**

- Welcome to discuss with me if you are in doubt whether usage conforms to class policy

400- vs. 500-Level Credit

- Assignments / Exams may have questions designated **only** for CSC 580 students
  - 480 students do not need to answer these questions

- Expectations for the semester project will be higher for CSC 580 students
  - More emphasis on novelty
  - I.e. if you implement a paper you are expected to make some improvement

- If you modify preexisting code repositories, we ask that you provide some novel modifications to it, with each modification more than 100 lines of code
  - We expect 3 modifications from 580 and 2 modifications from 480

HW0

- Calibration purpose; due on **8/31 @ 11:59pm:**.
- We ask that you spend some time to figure out an answer to the homework.
- If you failed to figure out, please explain **what you have done to find an answer** and **where you get stuck**.
  - DON'T: "I googled it and nothing came up"
  - DO: "I read material A, and there is this statement B that seems to help, but when I tried to apply, C became an issue due to independence. ..."

- We will grade it generously though, appreciating that different students come with different backgrounds!

HW0 Submission: Gradescope

- Watch the video and follow the instruction: https://youtu.be/KMPoby5g_nE
- Please upload one PDF file.
- If you do it handwritten, then make sure you picture it well. I recommend using TurboScan (smartphone app) or similar ones to avoid looking like slanted or showing the background.

Background refreshers

Probability

- http://cs229.stanford.edu/section/cs229-prob.pdf

Linear Algebra:

- http://cs229.stanford.edu/section/cs229-linalg.pdf

Big-O notation:

- http://www.stat.cmu.edu/~cshalizi/uADA/13/lectures/app-b.pdf

Other resources:

- The matrix cookbook: https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf
- The probability and statistics cookbook: http://statistics.zone/
- Calculus cheatsheet: https://tutorial.math.lamar.edu/pdf/calculus_cheat_sheet_all.pdf

Next lecture (8/26)

- The supervised learning paradigm
- Decision-tree learning
- Assigned reading: CIML Chap. 1 (Decision Trees)

Thank you!
Questions?