CSC 480/580 Principles of Machine Learning
Spring 2025

Jason Pacheco

The University of Arizona

*some slides are from Daniel Hsu and Kwang Sung-Jun’s lecture under their permission

What is machine learning?

What is machine learning?

- **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.

- Algorithm that builds an algorithm through experience (=data)
- A subfield of **Artificial Intelligence** – algorithms to perform smart tasks. The difference from the traditional AI is “**how**” you build a computer program to do it.
- An outdated book but still has interesting discussion (and easy to read).

AI Task 1: Image classification

- Predefined categories: C = {cat, dog, lion, …}
- Given an image, classify it as one of the classes 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)

AI Task 2: Recommender systems

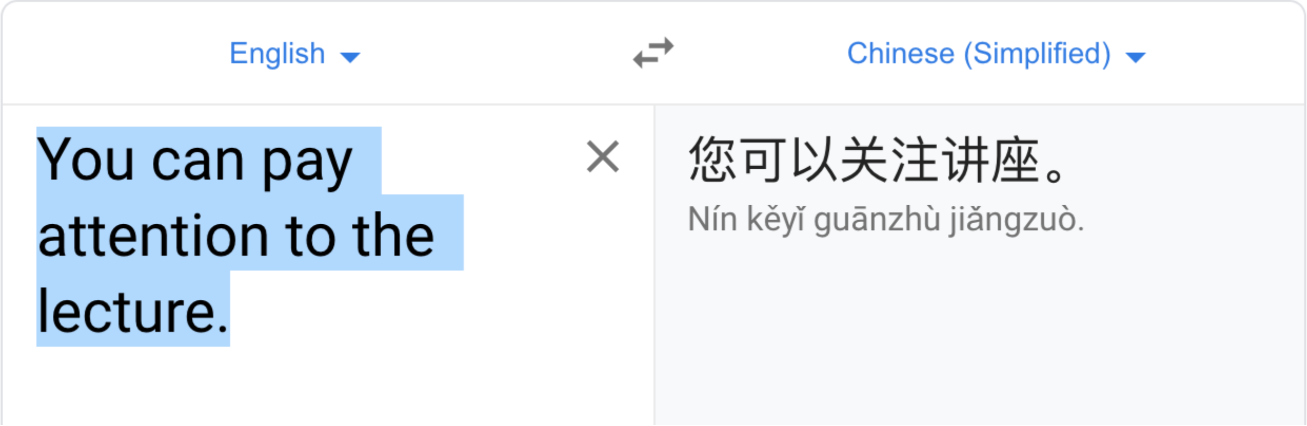
- Predict how user would rate a given movie (say 5-star rating)
- **Use**: For each user, pick an unwatched movie with the high predicted ratings.
- **Idea**: compute user-user similarity or movie-movie similarity, then compute a weighted average.

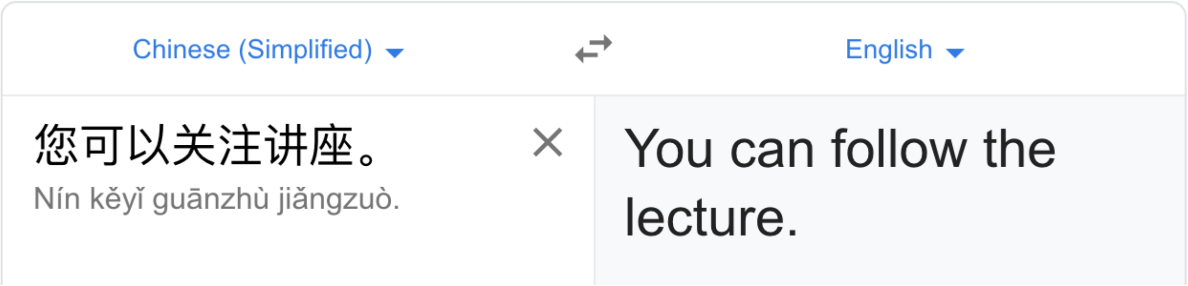
	User 1	User 2	User 3
Movie 1	1	2	1
Movie 2	?	3	1
Movie 3	2	5	2
Movie 4	4	?	5
Movie 5	?	4	2

- This particular approach is called ‘collaborative filtering’

AI Task 3: Machine translation

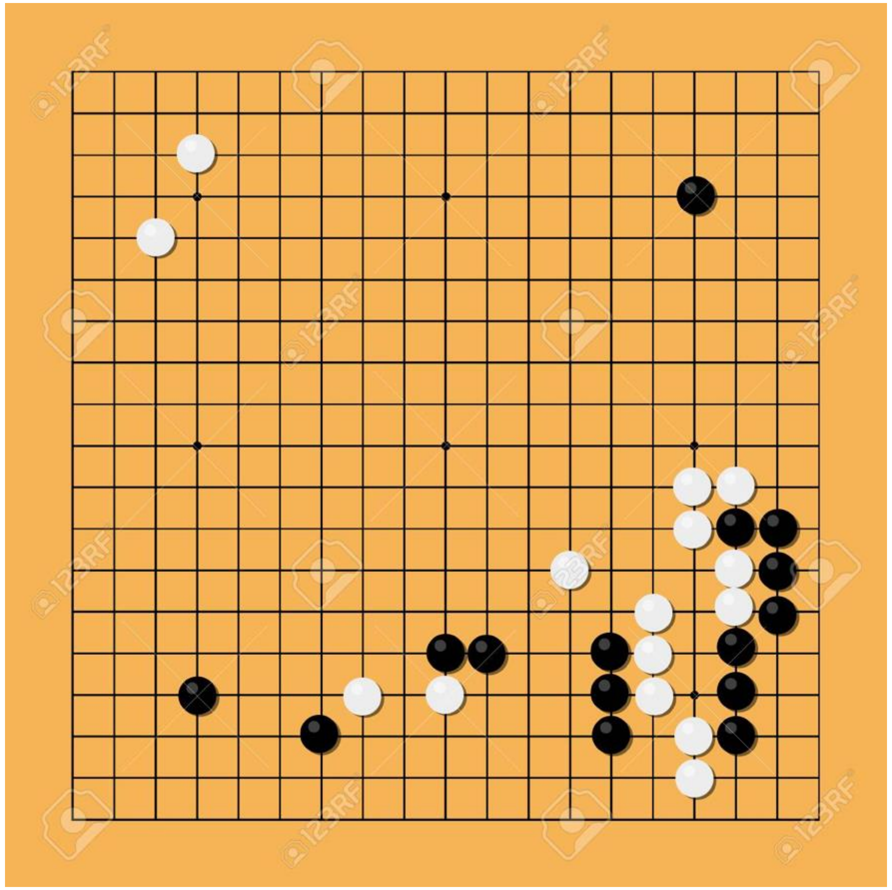
- No need to explain how useful it is.





AI Task 4: Board game

- Predict win probability of a move in a given game state (e.g., AlphaGo)
- Traditionally considered as a “very smart” task to perform.
- **Use**: From the AI Go player, you can do practice play or even learn from it.
  - These days, when people broadcast go game, they show the winning rate of each move!



AI Task 5: ChatGPT

- No need to explain.
- Good at retrieving knowledge and presenting it in natural language.
- Not necessarily good at difficult tasks (e.g., reasoning).

Traditional AI vs Machine Learning (ML)

- **<u>Traditional AI</u>**: you encode the knowledge (e.g., logic statements), and the machine executes it, with some more '**inference**' like if a -> b and b-> c, then a-> c.
  - e.g., if you see some feather texture with two eyes and a beak, classify it as a bird.

- **<u>ML</u>**: I give you a number of <u>input</u> and <u>output</u> observations (e.g., animal picture + label), and you give me a **function (can be a set of logical statements or a neural network)** that maps the input to the output accurately.
  - As the "big data" era comes, data is abundant ⇒ far better to learn from data than to encode domain knowledge manually.
  - "statistical" approach // "data-driven" approach
  - *"Every time I fire a linguist, the performance of the speech recognizer goes up." – 1988,* Frederick Jelinek, a researcher who worked on speech recognition.

- <u>Note</u>: ML approach to logic-based system: decision tree (simple rules) / inductive logic programming (complex rules)

Work in ML

- The usual CS background is often not sufficient – especially mathematical side, beyond discrete math.

- Data scientists: may not necessarily use ML (e.g., find associations between age and disease)
- Applied ML
  - Collect/prepare data, build/train models, tune hyperparameters, measure performance.
- ML research
  - Design/analyze models and algorithms
  - Theory: Provide mathematical guarantees. E.g., If I were to achieve 90% accuracy, how many data points do we need? => generalization bound.

Prereqs

- Math
  - Linear algebra, probability & statistics, multivariate calculus, reading and writing proofs.
  - Q: how many of you are familiar with eigen decomposition?

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

Overview of ML methods

supervised learning

unsupervised learning

reinforcement learning
(broadly, interactive learning)

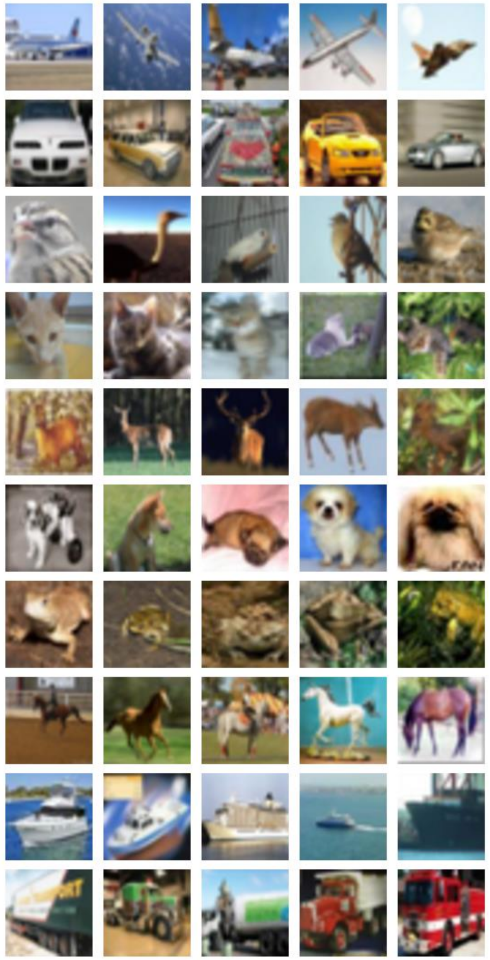
Supervised Learning

Basic setting: Supervised learning

example = data point

- Training data: dataset comprised of ***labeled examples***: a labeled example = a pair of (input, label)

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck



supervised learning algorithm → **function** ("classifier")

↓

function
("classifier")

↓

cat!

training

testing

Examples function 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

can be deeply nested!

Example function 2: Linear

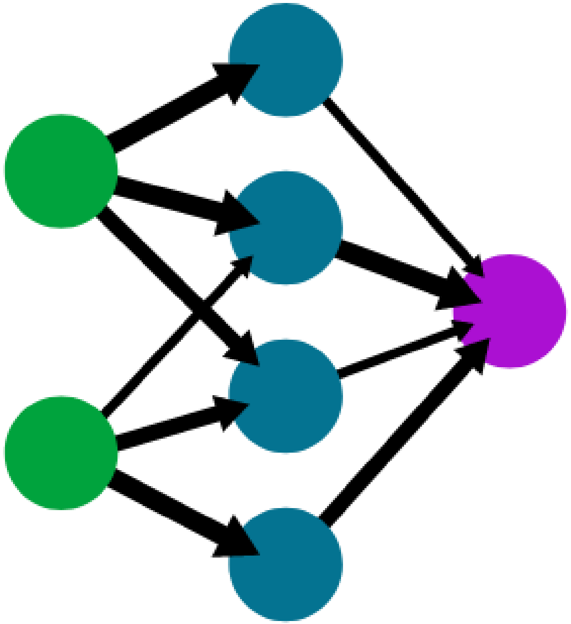
- E.g., Image classification
- Let x be a set of pixel values of a picture (30 by 30 pts) => 900 dimensional vector $x \in [0,1]^{900}$.
- If $\boxed{0.124 \cdot x_1 - 2.5 \cdot x_2 + \cdots + 2.31 \cdot x_{900}} > 2.12$ then
  - return cat
- else
  - return dog
- end

"linear combination"
"inner product"

- Coefficients: signed "importance weights"

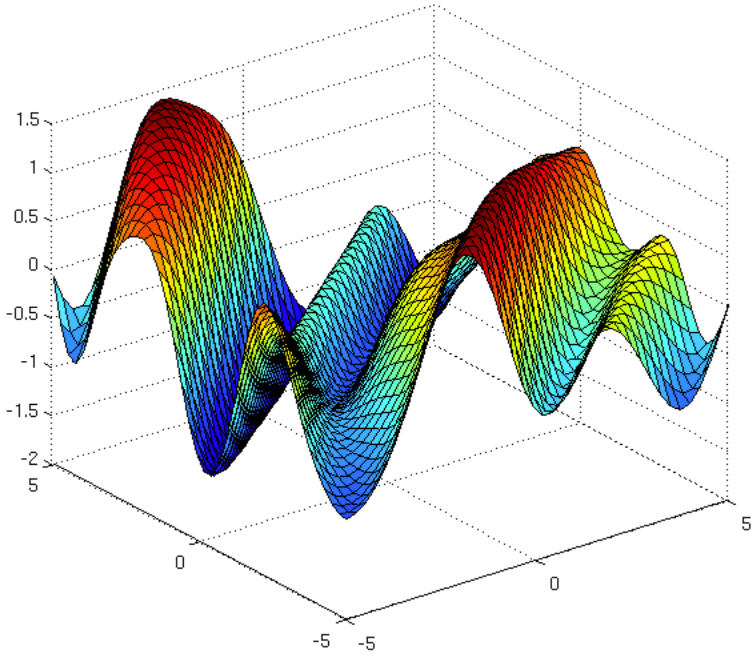
Example function 3: Nonlinear

Neural network



(stacked **linear** models with <u>nonlinear</u> **activation functions**)

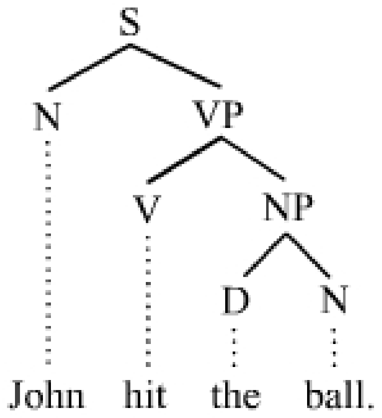
Gaussian process / Kernels



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

Supervised learning: Types of prediction problems

- Binary classification
  - Given an email, is it spam or not? (or, the probability of it being spam)
- Multi-class classification
  - Image classification with 1000 categories.
- Regression: the label is real-valued (e.g., price)
  - Say I am going to visit Italy next month. Given the price trends in the past, what would be the price given (flight destination, the # of days before the departure, day of week)?
  - Pricing: predict the price that will maximize the profit.
- Structured output prediction: more than just a number
  - Given a sentence, what is its grammatical parse tree?

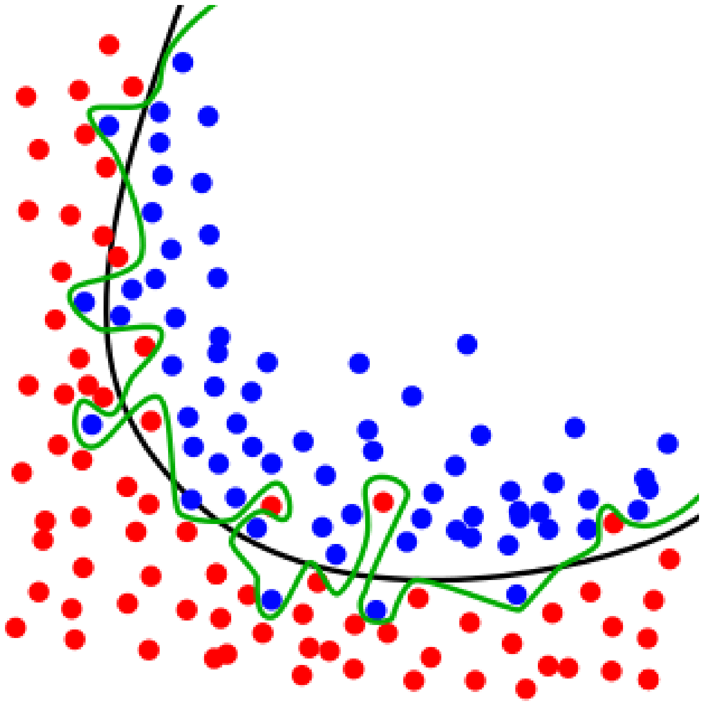


Beyond supervised learning

- Online learning (opp. “batch learning”)
  - Immediate updates are needed (e.g., personalized product/content recommendation)
  - Sequential update for fast learning / adapt to changing environment
- Unsupervised learning
  - Finds patterns/representation in the data without the help of labels.
- Reinforcement learning
  - The environment interacts with your action, transferring you to different states.
  - It learns to take ‘actions’ as opposed to making ‘predictions’.
  - When there are no states: ”**bandit**” feedback.
    - E.g., Amazon recommends you a pair of shoes. You did not click it. Amazon don’t know if you would’ve clicked had it recommended speakers or cookware.
    - The dataset is now dependent on the recommendation algorithm ⇒ biased data.
    - “bandit-logged” data.

The challenge: How to learn a function

- Okay, we have a training data. 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: let's memorize the data. To predict an unseen data, just follow the label of the closest memorized data.
- It does not work.
- You need to learn training dataset but don't "over-do" it.
- This is called "regularization" – an important notion.



green: memorization
black: true decision boundary