CSC 580 Principles of Machine Learning

Prof. Jason Pacheco

*some slides are from Daniel Hsu's lecture under his permission

*some slides are from Prof. Kwang-Sung Jun's lecture under his 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.

- A bit outdated with recent trends, but still has interesting discussion (and easy to read).
- A subfield of **Artificial Intelligence** – you want to perform nontrivial, smart tasks. The difference from the traditional AI is "**how**" you build a computer program to do it.

AI Task 1: Image classification

- Predefined categories: C = {cat, dog, lion, …}
- Given an image, classify it as one of the set 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 movie
- **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.

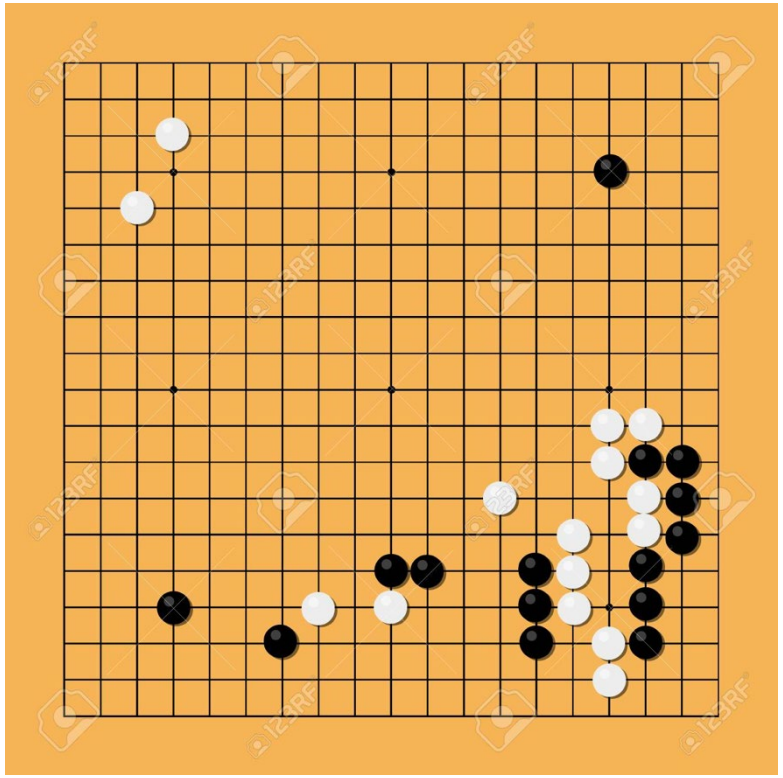
	User 1	User 2	User 3
Movie 1	1	2	1
Movie 2	?	3	1
Movie 3	2	5	2
Movie 4	4	?	5
Movie 5	?	4	2

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., AlgphaGo)
- 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.



Traditional AI vs Machine Learning (ML)

- **Traditional AI**: 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.
- **ML**: I give you a number of input and output 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 Czech-American researcher in information theory & speech recognition.
- **Note**: 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.

- Applied ML
  - Collect/prepare data, build/train models, analyze errors
- ML engineer
  - Implement/fine-tune ML algorithms and infrastructure
- 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

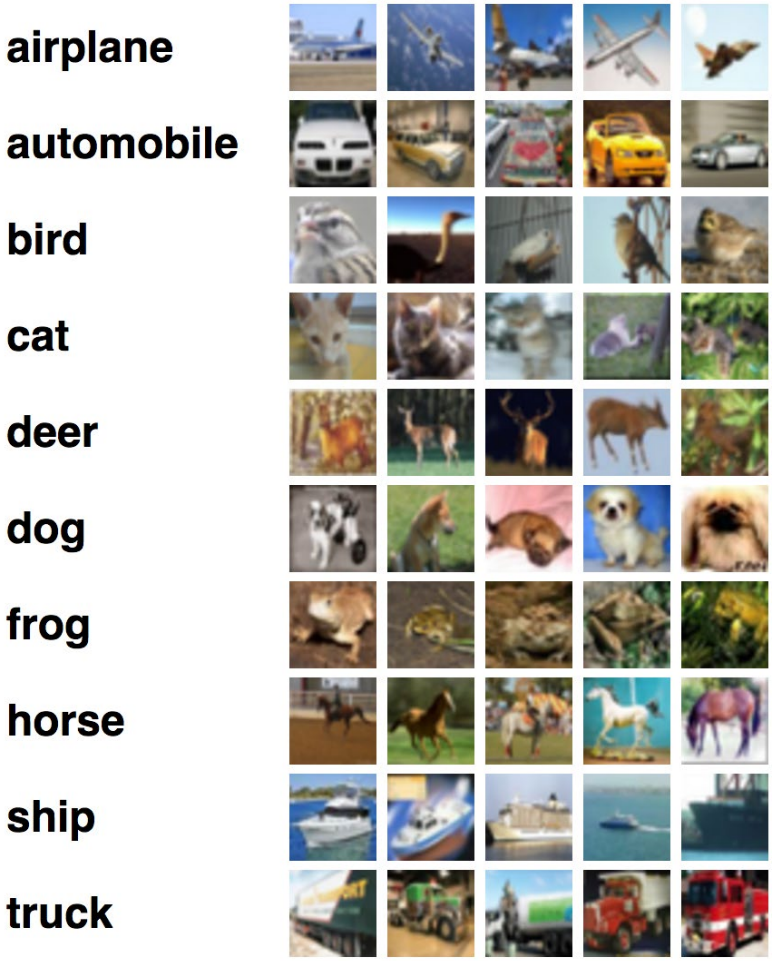
unsupervised learning

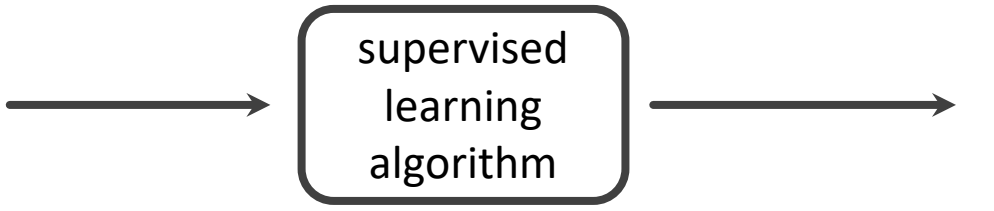
reinforcement learning

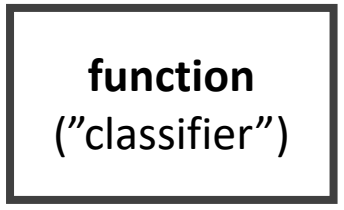
Supervised Learning

Basic setting: Supervised learning

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







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

Example function 2: Linear

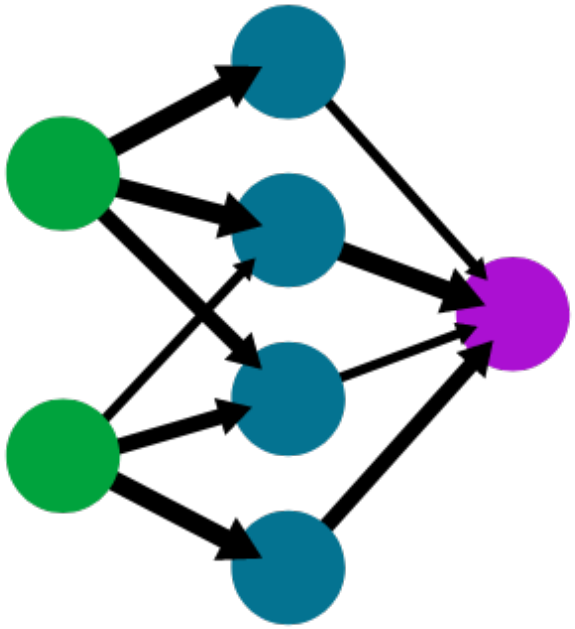
- E.g., Image classification
- Let x be a set of pixel values of a picture (30 by 30 pts) => 900 dimensional vector x.
- If $0.124 \cdot x_1 - 2.5 \cdot x_2 + \cdots + 2.31 \cdot x_{900} > 2.12$ then
  - return cat
- else
  - return dog
- end

"linear combination"

- Coefficients: signed "importance weights"

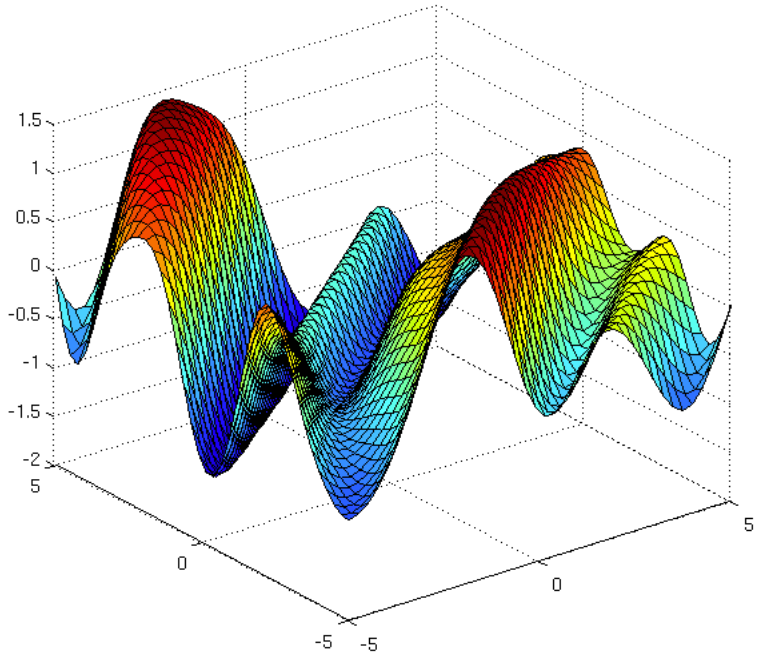
Example function 3: Nonlinear

Neural network



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

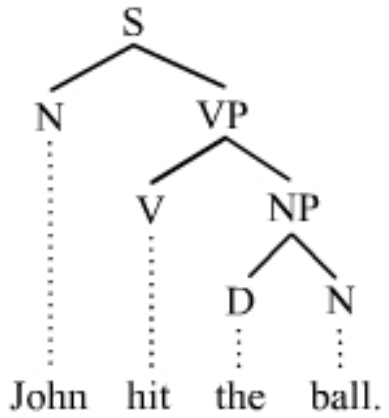
Gaussian process / Kernels



(**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 (the # of days before the departure, day of week)?
  - Pricing: predict the lowest price
- Structured output prediction: more than just a number
  - Given a sentence, what is its grammatical parse tree?

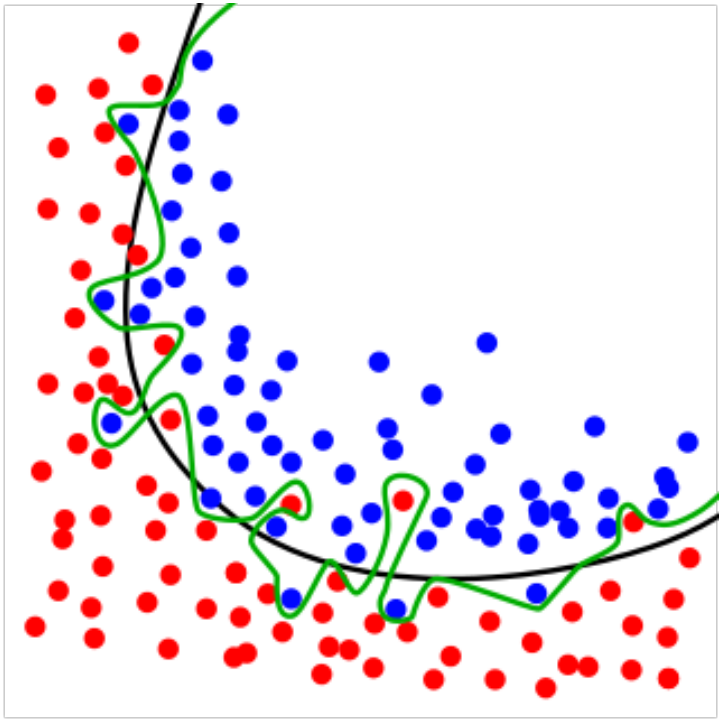


Beyond supervised learning

- Online learning (opp. “batch learning”)
  - Immediate updates are needed (e.g., personalized product recommendation)
  - Sequential update for fast learning / adapt to changing environment
- Unsupervised learning
  - Finds patterns in the data without the help of labels.
- Reinforcement learning
  - The environment interacts with your action, transferring you to different states.
  - 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

What to expect in the class

- ~~How to use sklearn, pytorch, tensorflow, fine-tuning deep net algorithms.~~
  - You can learn these on your own; their values likely decay soon

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

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

Logistics

Office Hours

All office hours online via Zoom – See Piazza for links

Office Hours are for:

- Clarification on lecture material
- Homework questions
- Other questions related to course logistics / material / ML

I prefer "Jason" or "Professor"
***But NOT** "Professor Pacheco"*

Me

Fridays @ 3:00-5:00pm
(1500 – 1700)

TA: Yinan Li

Fridays @ 10:30am – 12:30pm
(1030 - 1230)
Undergraduates Only

Electronical Resources

Course Webpage The main 'hub' with all lecture slides, schedule, etc.
http://www.pachecoj.com/courses/csc480-580_fall23/

D2L *Probably* won't use this for much beyond final grades...

- CSC 480 - https://d2l.arizona.edu/d2l/home/1355962
- CSC 580 - https://d2l.arizona.edu/d2l/home/1355965

Piazza mainly for Q&A/discussion (https://piazza.com/arizona/fall2023/csc480580/home)

Gradescope submitting the homework (https://www.gradescope.com/courses/578061)

- **Important** Login with **School Credentials | University of Arizona (NetID)**

Book *A Course in Machine Learning* by Hal Daumé III

- Web Version: http://ciml.info/
- PDF Version: http://ciml.info/dl/v0_99/ciml-v0_99-all.pdf

Lecture videos will be made available after each class for review

Syllabus summary

Section Warm up

- Basic supervised learning: decision tree, k-NN, perceptron
- Practical issues in supervised learning: evaluation, feature selection, etc.
- Bias-variance decomposition

Section Learning methods

- Linear & Nonlinear (Kernel) Models
- Probabilistic Modeling, Naïve Bayes, Graphical Models
- Neural Networks & Backpropagation
- Ensemble methods

Section Unsupervised learning

Section Learning theory

See course webpage for assigned readings related to each lecture

Syllabus summary

- 08/22: HW0 (calibration) Due 8/29 @ 12pm Noon
- 09/07: HW1 assigned
- 09/26: HW2 assigned
- 10/12: Midterm exam
- 10/31: Project proposal due
- 11/02: HW3 assigned
- 11/16: HW4 assigned
- 12/05: Final project due
- 12/13: Final exam at 6:00pm – 8:00pm (online)
- **Due**: HW0 is due in 7 days. HW1-4 and is due in 10 days.
- NO LATE DAYS

Grades will be on a 0-100 scale with weights:

- Homework assignments: 35%
- Project Proposal: 5%
- Project: 20%
- Midterm exam: 15%
- Final exam: 15%
- Participation: 10%

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.

Participation

- Stop me at any point to ask questions! **There are no bad questions**
- Any ideas to encourage participation?
- I **strongly** encourage off-class discussion in Piazza.
  - Students should also attempt to answer questions
  - Sometimes answering questions helps us learn better (especially if we're wrong)
- Lecture videos are for review-you should attend lecture in-person

400- vs. 500-Level Credit

- This course will be co-convened CSC 480 / 580

- The same assignments will be issued to all students

- Assignments / Exams will have questions designated **only** for CSC 580 students
  - Undergraduates should not answer these questions
  - There won't be extra credit for answering them (I will occasionally have extra credit 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 should make some improvement
  - Undergrads may implement an algorithm as-is or apply it to a dataset of their choosing

Plagiarism

- Case study 1
  - Two students turned in a final exam with nearly identical code blocks
  - The same number of lines, only variables were renamed and some lines reordered

- Case study 2
  - One student turned in a midterm exam with another student's name on it
  - On checking much of the material was nearly identical in both exams
  - When confronted both students admitted that they shared exams

- So, what happened?
- **No tolerance**. You will receive zero credit

HW0

- Calibration purpose; due on **8/29 12pm**. NO LATE DAYS. Will not accept late submissions.
- Will not be part of the homework score *unless you don't make an effort*
- I require 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. ..."

- The participation score will be deducted (-2 out of 10pts) if …
  - Empty answers
  - No nontrivial efforts to solve it.

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.

Useful Background Material

Probability

- http://cs229.stanford.edu/section/cs229-prob.pdf
- Lecture notes: http://www.cs.cmu.edu/~aarti/Class/10701/recitation/prob_review.pdf

Linear Algebra:

- http://cs229.stanford.edu/section/cs229-linalg.pdf
- Short video lectures by Prof. Zico Kolter: http://www.cs.cmu.edu/~zkolter/course/linalg/outline.html
- Handout associated with above video: http://www.cs.cmu.edu/~zkolter/course/linalg/linalg_notes.pdf

Big-O notation:

- http://www.stat.cmu.edu/~cshalizi/uADA/13/lectures/app-b.pdf
- http://www.cs.cmu.edu/~avrim/451f13/recitation/rec0828.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

Questions?