§1. INTRODUCTION & BASIC CONCEPTS

2021.06.03

OUTLINE

- motivation.
- course logistics.
- course overview & basic concepts.

MOTIVATION

Why learn ML?
- AI is the new electricity.
- Top-desired IT skill.
- Applications in academia & industry across disciplines.

major tech disruption → opportunity to remake meaningful parts of the world, if one behaves in ethical, principled manner :)

Goal: become expert in ML in 10 weeks.
- apply state-of-the-art ML techniques to domain problems.
- Well-qualified for doing ML research.

ML applications:
- Tasks hard to be programmed by hand: handwriting recognition, autonomous driving.
- Data mining: medical records ⟶ medical knowledge.
- Daily life: fraud detection, recommender systems.

CLASS LOGISTICS

Prerequisites:
- Computer science: programming (python + numpy), big O, stacks, queues, binary trees.
- Probability: random variables, expectation, variance.
- Linear algebra: vectors, matrices, array multiplication, eigenvalue problems.

→ undergrad courses more than enough!

Class materials / grading:
- lecture notes: technical details.
- Discussion sections (optional): refreshers of prerequisites, advanced topics (eg, convex optimization, hidden markov model, time series).
- 4 problem sets: math + coding, fairly difficult.
- mid-term: timed, Take-home.
- Final project: usually applied ML. List of past projects.

Comparison of classes:
- CS229: more mathematical / theoretical, less applied / hands-on.
- CS229A: more applied, flipped classroom format (coursera + discussions).
- CS230: ~CS229A in style, but focus on DL.

→ if time permits, encouraged to take multiple classes to gain different perspectives.

ML DEFINITION

Arthur Samuel (1959):
- ML: field of study that gives computers the ability to learn w.o being explicitly programmed.
- checkers-playing program: through self-play, program learns board patterns leading to wins / losses & beats him!
  → a 1st example of computers outperforming humans in narrow tasks.

Tom Mitchell (1998):
- Well-posed learning problem: a computer is said to learn from experience E w.r.t. some task T & some performance measure P, if its performance on T, as measured by P, improves w. E.

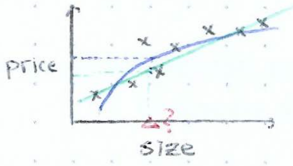
- (eg) Checkers
  E: experience of self-play.
  T: playing checkers.
  P: probability of wins.

↳ This course covers 5 major topics.

I. SUPERVISED LEARNING

↳ the most common.

⟨Ex⟩ Housing price prediction:

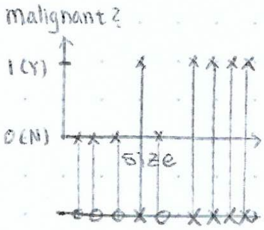


Supervised learning: given "right answers" / labeled data (x, y) pairs
input / features(s) → x, output / labels(s) → y
⇒ find functional mapping from x to y.

Regression: predict continuous y.
(eg) price.

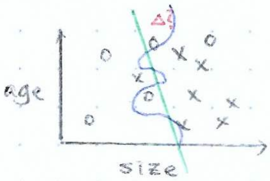
• One possible taxonomy of ML based on type of supervision / task:
supervised
unsupervised reinforcement
In reality ⇒ overlap.
(eg) semi-supervised learning
AlphaGo.

⟨Ex⟩ Breast tumor prediction.



Classification: predict discrete y
(eg) malignant, benign. → Binary classification.

↳ may have multiple categories.
(eg) benign, type A, type B. → Multiclass classification.
<> multilabel classification.



↳ may have multi-dimensional input.
(eg) tumor size, age of patient, uniformity of cell shape.
↳ How to select features?
SVM: use kernel trick to deal w. ∞-dim input!

⟨Ex⟩ Autonomous driving w. ALVINN:

- Supervised learning: get steering directions from humans. → no longer state-of-the-art.
- Regression: $y \in \mathbb{R}$.
  → discretized in actual implementation.

II. ML STRATEGY / LEARNING THEORY

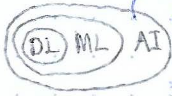
↳ general, model-agnostic principles.

- Not only give the tools, but also the know-how of applying them well.
- Most effective ML practioners: make strategic decisions on what to try next.
- This course: demystify black-box magic, codify systematic engineering principles. ※ ML Yearning book.

III. DEEP LEARNING

↳ representation learning <> manual feature engineering

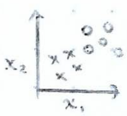
- Hottest subfield of ML.
- Can be use in supervised learning, unsupervised learning, reinforcement learning.
  ↳ focus of this course. ↳ (eg) autoencoders, RBMs ↳ (eg) deep RL.



Build programs that operate on human cognitive level.
⇒ ML: one approach (data) <> logic

IV. UNSUPERVISED LEARNING

↳ can be used as a data preprocessing step (eg. PCA → reduce dim, clustering → group pixel types for computer vision pbs)



Supervised learning: labeled data.

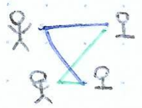
Unsupervised learning: unlabeled data.
⇒ find interesting structures in data.

Clustering: partition data into different groups.

⟨Ex⟩ Clustering:

- Google news: group articles into coherent stories.
- Genome data: divide indivisuals into different types.
- Social network analysis: identify cohesive communities.
- Market segmentation: group customers into sub-populations for target-specific marketing.
- Organizing computer clusters: find which machines' workflows are more related.

⟨Ex⟩ Cocktail party problem:



Independent-component analysis (ICA): given n recordings w. overlapping voices
⇒ figure out they can be explain by n independent sources & separate them out.

↳ an extension of PCA that finds the direction of maximum kurtosis instead of variance using SVD
⇒ optimze higher-order statistics.

⟨Ex⟩ Learning analogies from unlabeled text data:

A : B :: C : D ?

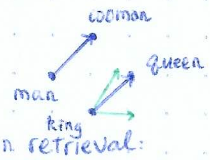
woman : man :: king : queen

tokyo : japan :: dc : usa

- Word embedding: represent words by vectors.

  word $\xrightarrow{\text{encode}}$ vector

  relation $\xrightarrow{\text{encode}}$ direction.



- Vector space model (VSM) from information retrieval:
  optimize cosine similarity between semantic relation vector for A & B and that for C & D.

V. REINFORCEMENT LEARNING

Supervised learning: learn one-shot decision making.

Reinforcement learning: learn to make a sequence of decisions.

- don't know optimal solution, but can define good / bad behavior (implicit feedback).
- agent can query feedback interactively from environment
  & learn optimal policy that maximizes / minimizes accumulative reword / penalty through trial & error.

⟨Ex⟩

- robotics
- game playing (Altari, Go)
- autonomous driving.