Topic 0: Introduction

STAT 37710/CAAM 37710/CMSC 35400 Machine Learning

Risi Kondor, The University of Chicago

Instructors

Risi Kondor (associate prof.)
Crerar 221
`risi@cs.uchicago.edu`

TAs:
Su Yeong Lee (CAAM)
Kexiang Wang (CAAM)

Topics

1. Clustering
2. Dimensionality reduction
3. Manifold learning
4. Regression
5. Online algorithms
6. Kernel methods (Hilbert space algorithms)
7. Bayesian learning
8. Deep learning
9. Generative models

Note: this list is provisional and almost certain to change.

Prerequisites

- Competence in coding in *some* programming language.
- Mathematical maturity: ML is a mathematical subject.
- Specific areas of math needed:
    - Calculus
    - Linear algebra
    - Probability (minimal Statistics)
    - Little bit of optimization.

Support

Recitations:

- On an as needed basis, place and time TBD

Office Hours:

- Fridays 1pm Crerar 221

Online:

- `canvas.uchicago.edu` (slides, lecture notes, assignments and grades)

Resources

Books (Strictly optional! More for “further reading” than anything else.)

- Kevin Murphy: Machine Learning: A probabilistic perspective (2012)
  Warning: very Bayesian
- Zhang, Lipton, Li and Smola: Dive into deep learning (`d2l.ai`)
- Hastie, Tibshirani, Friedman: The Elements of Statistical Learning (2008)
  (available electronically on the library’s web site)

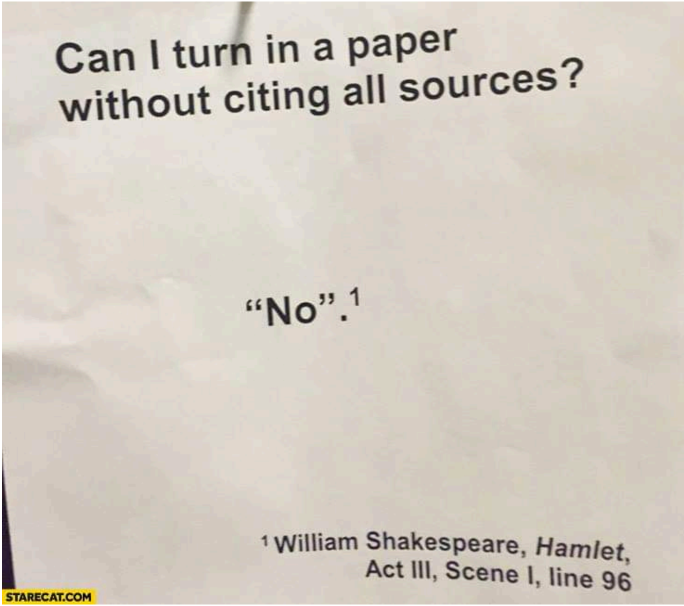
Online Courses

- Andrew White’s book “Deep learning for molecules and materials”
  `https://dmol.pub/index.html`

Links to more books, papers and videos will be posted on Canvas.

Credit

- **Assignments/projects (posted on Canvas):** $\sim 50\%$
  - Project centered course: one assignment for each topic.
  - Projects involve coding up algorithms discussed in class and running them on data.
  - Recommended language: Python.
  - Submitted work must be your own. Discussing problems is okay but must be acknowledged. Code and parts of the writeup cannot be shared.
  - Submission in .pdf via Canvas. Penalty for late submissions: 20% for 24 hours, 40% for 48 hours. No partial late homeworks.
  - For typing up assignments LaTeX is strongly preferred.
- **Midterm** $\sim 20\%$
- **Final** $\sim 30\%$


Can I turn in a paper without citing all sources?
“No”.[1]
[1] William Shakespeare, *Hamlet*, Act III, Scene I, line 96
STARECAT.COM

What is Machine Learning?

Two types of programming

1. **Explicit:** write a program that tells the computer what to do.
2. **Learning:** write a program that tells the computer how to *learn* what to do from data. $\rightarrow$ This is what Machine Learning is about.


MACHINE
LEARNING
POWER TO THE DATA

Machine Learning in the abstract

Given a **training set** $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ learn a function

$$f : x \mapsto y$$

to **predict** the y's corresonding to future x's.

In particular, those in the **test set**
$\{(x'_1, y'_1), (x'_2, y'_2), \ldots, (x'_{m'}, y'_{m'})\}$.

Actually, this is **supervised learning.** Modern ML also encompasses many other types of learning problems.

Nomenclature

- Each (x, y) pair is called an **example** (or learning instance).
- x is called the **input** $($ $x \in \mathcal{X}$, where $\mathcal{X}$ is the **input space** $)$.
- y is called the **output** $($ $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the **output space** $)$
- The learned function

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

  is called the **hypothesis** (because the algorithm can never be sure how close it is to the “truth”).
- The space $\mathcal{F}$ from which the algorithm chooses f is called the **hypothesis class**.

Deductive vs. inductive inference

- **Deductive inference:**

$$\text{rules} \quad \longrightarrow \quad \text{data}$$

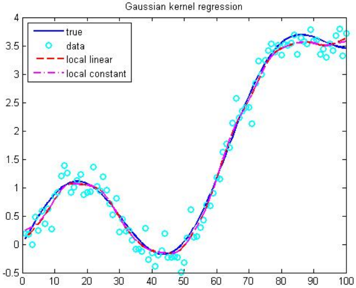
- **Inductive inference:**

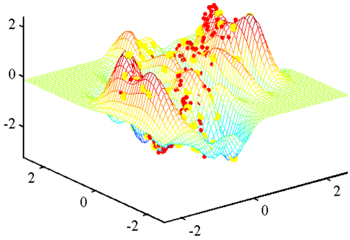
$$\text{data} \quad \longrightarrow \quad \text{rules}$$

ML is all about inductive inference $\rightarrow$ "Brave New Science of Data".
Humans are experts at induction. However, ML takes a different approach.

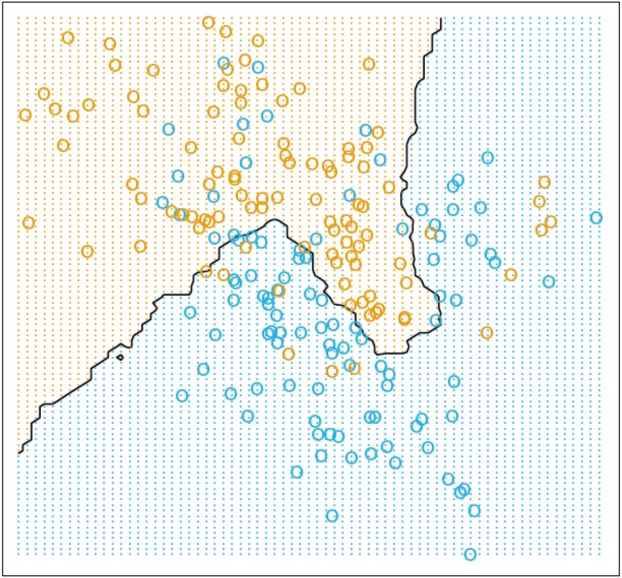
Question: Give examples of inductive vs. deductive inferential processes.
Question: What are the relative strengths of humans vs. machines in learning?

Typical ML task 1: Regression





Typical ML task 2: Classification



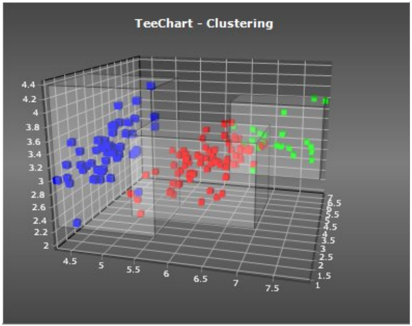
Typical ML task 3: Ranking

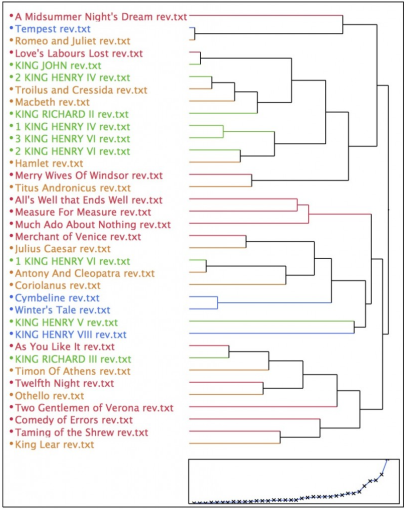
Elections

Sports

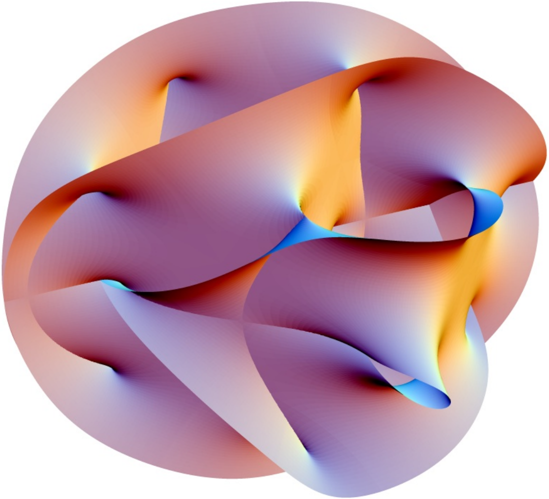
Internet search

Typical ML task 4: Clustering





ML task 5: Dimensionality Reduction



Applied vs. theoretical ML

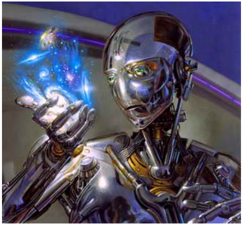
- **Practitioners** focus on solving real-world problems with ML (building autonomous cars, finding disease genes, earning lots of money, etc.).
- **Theorists** work on devising new general purpose learning algorithms and analyzing their behavior.

"Much of the art of machine learning is to reduce a range of disparate problems to a fairly narrow set of prototypes. Much of the science of machine learning is to then solve those problems and provide good guarantees." (Smola & Vishwanathan)

This course will focus on the fundamental algorithms rather than specific applications.

Origins: Classical Artificial Intelligence

AI vs. ML



$\leftrightarrow$

Attempts to replicate human intelligence in general.

Solves practical problems which humans *think* require intelligence.

Early attempts

The “Mechanical Turk” (Wolfgang von Kempelen, 1770)

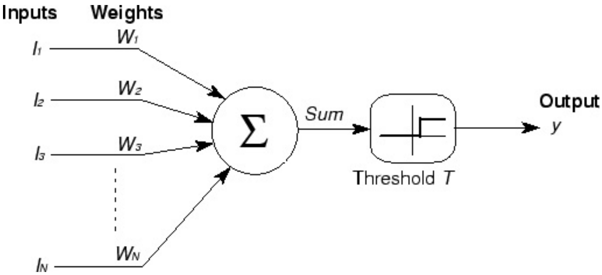
Formal reasoning = intelligence?

- Formal logic (Frege (1879) and others)
- Mathematics as a formal system (Russell & Whitehead, $\sim$1910)
- Gödel's incompleteness results (1931)
- Turing machines and universality (1936)

"Since formal systems are the pinnacle of human achievement, intelligence must be synonymous with formal reasoning."

Is the brain just a computer?

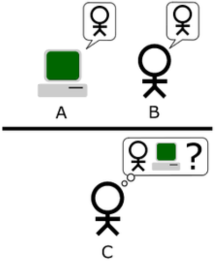
Pitts & McCullogh show that neurons appear to perform simple logical operations (1943)

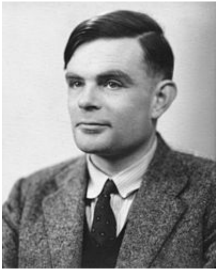


“So if all that the brain does is such mechanistic operations, then it should be easy to imitate on Turing machines (i.e., computers)”

The Turing test

In his landmark 1950 paper "Computing Machinery and Intelligence" Turing proposes a positivist approach: "If a machine can fool a human into thinking that it is a human, then it must be intelligent" $\rightarrow$ **Weak AI**





Prediction: "By the year 2000 machines with 120MB of memory would be able to fool 30% of human judges in a 5min test".

Objections to the Turing test

Even if a computer passes the Turing test it cannot be truly intelligent because...

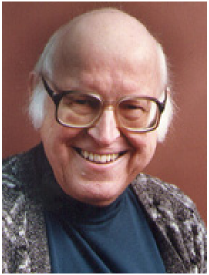
1. Theological: computers have no soul
2. "Head in the sand": it would be too scary
3. Mathematical: Godel incompleteness and such
4. Consciousness: Searle's Chinese room argument
5. Disabilities: a machine will never be able to do fall in love/invent jokes/tell right from wrong/etc.
6. Lady Lovelace's: will never do anything original
7. The brain is not digital
8. The brain is not predictable
9. Extra-sensory perception

"If I can't feel love, what was the point of making me so damn good-looking?"

The Dartmouth conference (1956)

John McCarthy
(1927–2011)

Marvin Minsky
(1927–2016)



Allen Newell
(1927–1992)

Herbert Simon
(1916–2001)

”within a generation ... the problem of creating ’artificial intelligence’ will substantially be solved” (Minsky)

True beginnings: from philosophy to building things

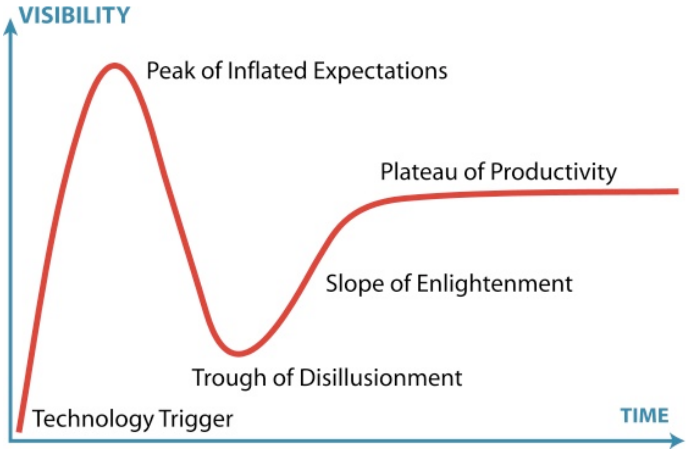
"We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer."

McCarthy et al., 1955

Early successes

- Newell and Simon's "General Problem Solver" (1959)
- ELIZA (Weizenbaum, 1966)
- SHRDLU's block world (Winograd 68–70)
- Prolog and expert systems 70's–

AI winters '74-'80, '87-'93



New beginnings: Machine Learning

The birth of Machine Learning

Starting in late ’80’s, AI was transformed by a sequence of outside influences:

- Efficiently trainable neural network models
- Input from Physics community
- Influence of Bayesian Statistics
- Black box “geometric” learning algorithms
- Huge influence of the internet
- Firm foundations in Statistics
- Strong connections to optimization, signal processing, harmonic analysis, probability, CS theory, ...
- **MASSIVE PRACTICAL DEMAND**

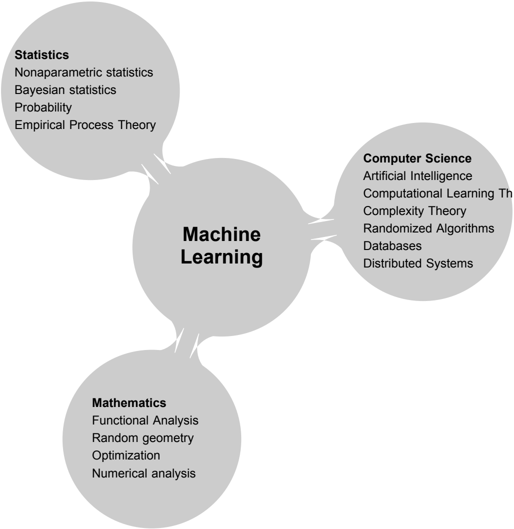
The old vs. the new AI

Early: aiming for "general intelligence", trying to imitate humans, tangled up in formal systems and philosophy

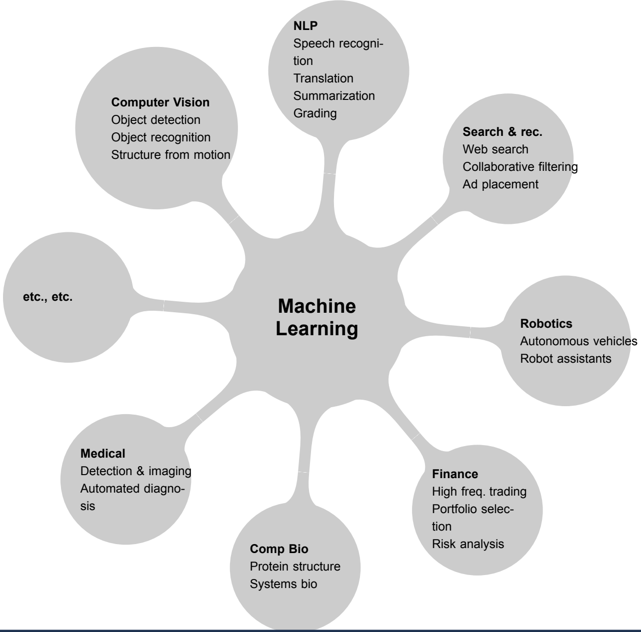
↓

New: pragmatic, focused on specific tasks, much closer ties to math and statistics than neuroscience and logic, driver behind lots of technologies

Question: Classically, the subject that deals with the art of learning from data is Statistics. So is ML just a branch of Statistics? No.


Statistics
Nonparametric statistics
Bayesian statistics
Probability
Empirical Process Theory
Machine Learning
Computer Science
Artificial Intelligence
Computational Learning Th
Complexity Theory
Randomized Algorithms
Databases
Distributed Systems
Mathematics
Functional Analysis
Random geometry
Optimization
Numerical analysis

Applications



Hallmarks of ML

ML is ambitious:

- Datasets are often very high dimensional $(\sim O(10^5))$.
- Data is often abstract (structured objects vs. just vectors).
- Datasets are massive $(\sim O(10^8)$ examples $)$.
- Really want to build actual systems that work.

ML is brutal:

- Don't need to think hard about the domain because with enough data, even black box algorithms work really well (really?).
- Butcher the statistics as much as necessary to get an algorithm which actually runs.
- Insist on algorithms that run in time $O(m^3) \rightarrow O(m^2) \rightarrow O(m) \rightarrow o(m)$.

Taxonomy of Machine Learning

Taxonomy of machine learning 1.

Based on the output space $\mathcal{Y}$:

- **Classification:** $\mathcal{Y} = \{+1, -1\}$
  Examples: spam/not spam, genuine/fraud, boy/girl,…
  (generalization: multiclass classification $\mathcal{Y} = \{1, 2, \ldots, k\}$)
- **Regression:** $\mathcal{Y} = \mathbb{R}$
  Examples: predict temperature tomorrow, price of a stock,…
  (generalization: $\mathcal{Y} = \mathbb{R}^d$)
- **Ranking:** $\mathcal{Y} = \mathbb{S}_n$ (group of permutations)
- **Structured outputs:** $\mathcal{Y} =$ anything
  Examples: translate from Chinese to English, predict folding of protein,…

Taxonomy of machine learning 2.

Based on the nature of the training data:

- **Supervised learning:** given $\{(x_i, y_i)\}_{i=1}^{m}$, learn $f\colon \mathcal{X} \to \mathcal{Y}$.
  Examples: classification, regression, …
- **Unsupervised learning:** given $\{x\}_{i=1}^{m}$, say something.
  Examples: clustering, density estimation, dimensionality reduction,…
- **Semi–supervised learning:** given a (small) amount of labeled data $\{(x_i, y_i)\}_{i=1}^{m}$ and a (large) amount of unlabeled data $\{x\}_{i=m+1}^{p}$, learn $f\colon \mathcal{X} \to \mathcal{Y}$. Examples: learning parse trees, image search

Taxonomy of machine learning 3.

Based on how the data is presented to the learner:

- **Batch learning:** see whole training set first, then predict on test examples.
- **Online learning:** examples are presented one-by-one, first try and predict y_t, then find out what y_t really is and learn from it.
- **Transductive learning:** like batch, but know test x_i''s at training time.
- **Active learning:** algorithm can ask for next data point
- **Reinforcement learning:** exploring the world incurs a cost (games, robotic control)

Taxonomy of machine learning 4.

Based on the nature of the relationship between x and y:

- **Deterministic:** x fully determines y, so there is some f_{true} out there so that
$$y = f_{\text{true}}(x).$$
- **Stochasitic:** x does not fully determine y, rather, for any given x, y is drawn from some probability distribution $p_x(y)$.

In practical problems, invariably, we cannot assume a deterministic relationship between inputs and outputs, so we use the stochastic model.