CMSC 422 Introduction to Machine Learning

Furong Huang / furongh@cs.umd.edu / Department of Computer Science

UNIVERSITY OF MARYLAND

What is Machine Learning?

Machine Learning studies representations and algorithms that allow machines to improve their performance on a task from experience.

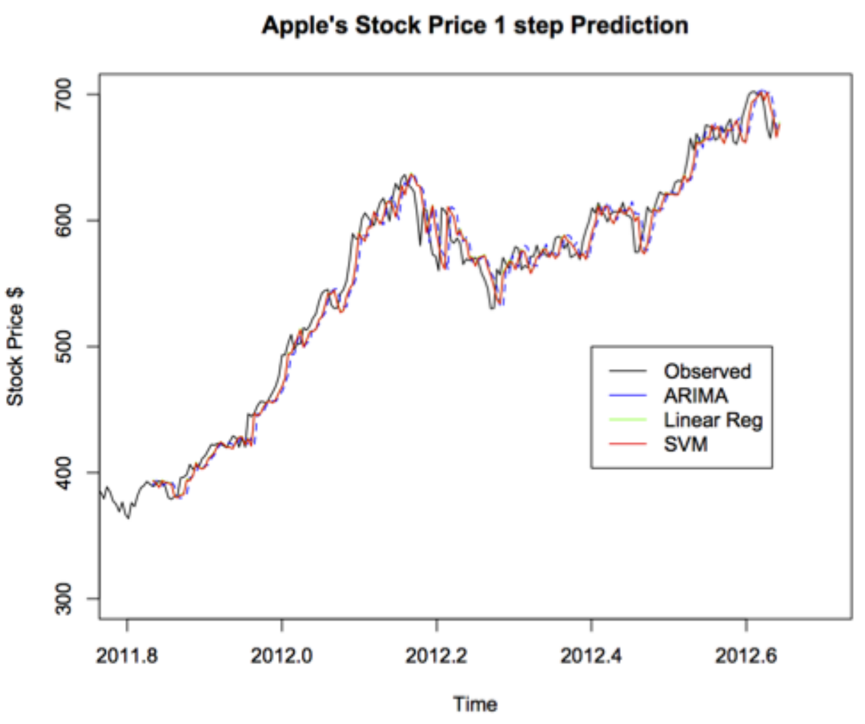
What is Machine Learning?

- Data driven science
  - Finding and exploiting patterns in data
- Algorithms
  - Data independent algorithms
- Prediction
  - Ultimate goal

What can ML do?

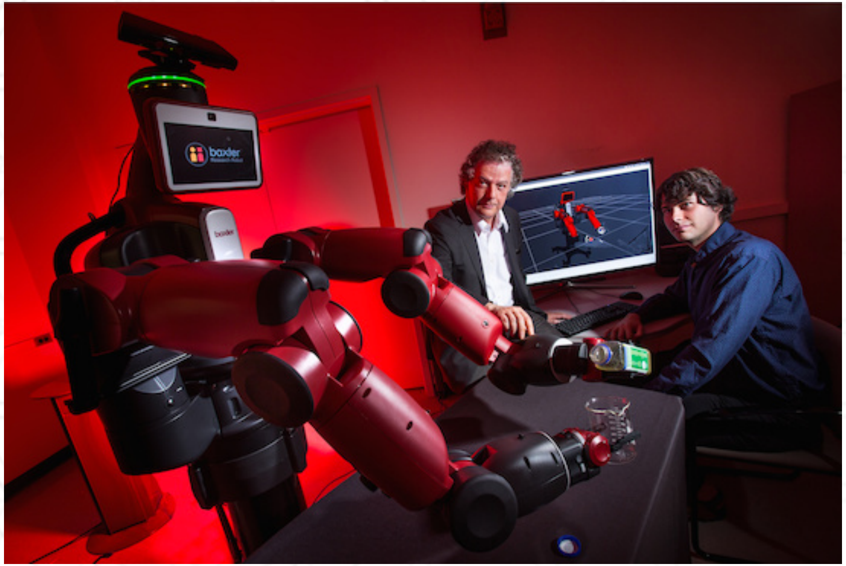
Analyze text & speech

What can ML do?



Stock price prediction

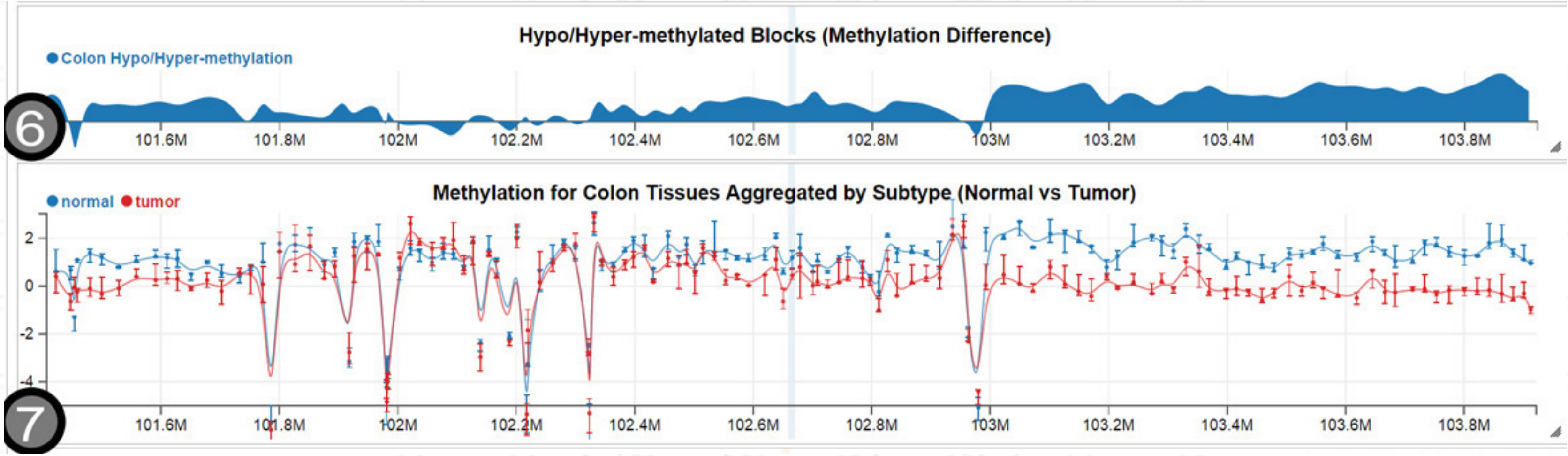
What can ML do?



Recognize objects in images

Teach robots how to cook from youtube videos

What can ML do?



Analyze genomics data

exploration and understanding of correlations between various genome features

What can ML do?

- Intelligent Machine
  - As smart as humans
    - Mechanical turk evaluation, cloud sourcing
  - Smarter than human
    - Computation Power
    - Memory

What can ML do?

nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | For A

Current Issue > Articles > Article

NATURE | ARTICLE

日本語要約

Mastering the game of Go with deep neural networks and tree search

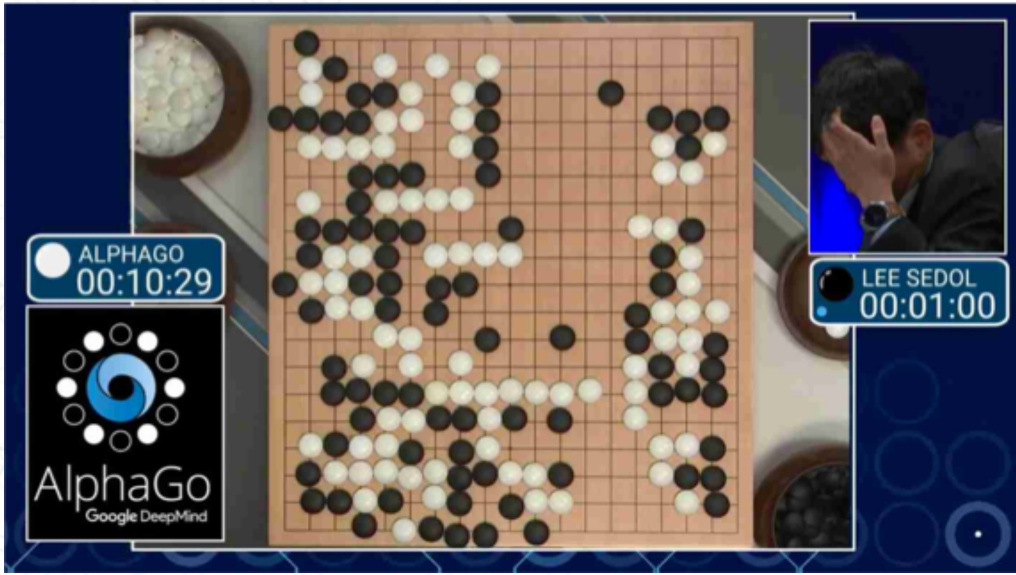
David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel & Demis Hassabis

Affiliations | Contributions | Corresponding authors

Nature **529**, 484–489 (28 January 2016) | doi:10.1038/nature16961

Received 11 November 2015 | Accepted 05 January 2016 | Published online 27 January 2016

PDF | Citation | Reprints | Rights & permissions | Article metrics



What can ML do?

Question Answering system beats Jeopardy champion Ken Jennings at Quiz bowl!

Other ML applications?

Machine Learning

- Paradigm: “Programming by example”
  - Replace “human writing code” with “human supplying data””
- Most central issue: generalization
  - How to abstract from “training”” examples to “test” examples? Memorization vs learning?
  - Analogy with human learning?

Machine Learning vs Programming

- Algorithm
  - ML: usually with uncertainty
  - Programming: usually deterministic
- Goal
  - ML: prediction, do better in the future using data in the past
  - Programming: computing
- Data dependency
  - ML: universal for all or a wide class of data
  - Programming: data specific

A fast moving field

- Broad applicability
  - Daily life: finance, natural language processing, computer vision, robotics, healthcare, medicine, biology
- Open/young field
  - Deep learning, gap between theory and practice
- Fear of AI/ML
  - Understand what's in the black box

CMSC 422 Goal

- This is a broad overview of existing methods for machine learning and an introduction to adaptive systems in general.

CMSC 422 Goal

- By the end of the semester, you should be able to
  - Look at a problem
  - Identify if ML is an appropriate solution
  - If so, identify what types of algorithms might be applicable
  - Apply those algorithms

- This course is not
  - A survey of ML algorithms
  - A tutorial on ML toolkits such as Weka, TensorFlow, …

CMSC 422 Topics

Foundations of Supervised Learning

- Decision trees and inductive bias
- Geometry and nearest neighbors
- Perceptron
- Practical concerns: feature design, evaluation, debugging
- Beyond binary classification

Advanced Supervised Learning

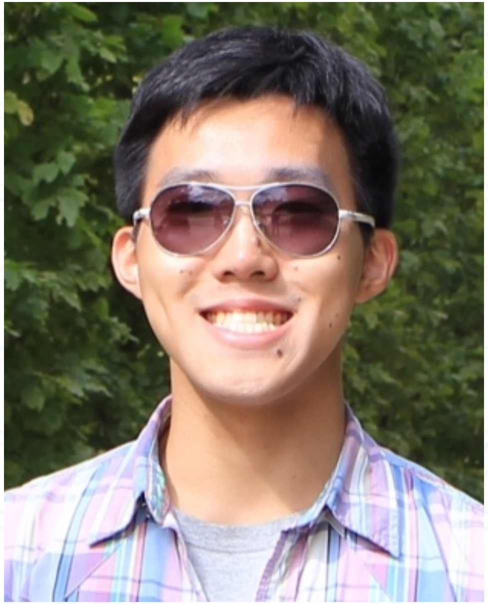
- Linear models and gradient descent
- Support Vector Machines
- Naive Bayes models and probabilistic modeling
- Neural networks
- Kernels
- Ensemble learning

CMSC 422 Topics

Unsupervised learning
- K-means
- PCA
- Expectation maximization

Selected advanced topics (as time permits)
- Expectation maximization
  - –Online learning
  - –Markov decision processes
- Imitation learning

Teaching Assistants



Xuchen You

Joseph Thomas Bergman

Justin Shen

What you can expect from the instructors

- Introducing concepts from multiple perspectives
  - Lecture, reading material, office hours, online discussion (Piazza)
- Providing opportunities to practice
  - Homework, programming projects

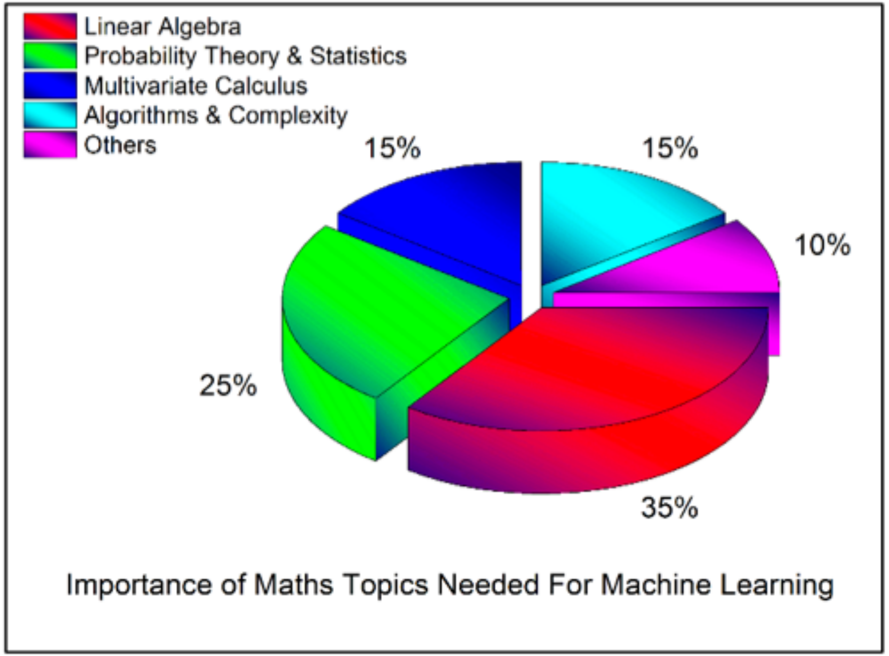
What I expect from you

- Math background
  - Calculus, linear algebra, probability
- Come to class prepared
  - Readings required
- Attend the lectures
  - Ask questions, take notes if necessary
- Complete HW independently
- Complete Projects collaboratively
- Be active on Piazza

Prerequisites

- CMSC351 (Algorithms) and CMSC330 (Programming Languages)
- Recommended: STAT400 (Applied probability and statistics) and Linear Algebra.
- These previous courses require CMSC250 (Discrete Structures), CMSC216 (Computer Systems)
- Which in turn require CMSC131 (Object oriented programming) and MATH141 (Calculus)
- Course is about data, representations, mathematical modeling, and programming

Math in ML



Linear Algebra
Probability Theory & Statistics
Multivariate Calculus
Algorithms & Complexity
Others
15%
15%
10%
25%
35%
Importance of Maths Topics Needed For Machine Learning

Sections

- Two sections
  - Prof. Marine Carpuat, 0101
  - This section, 0201
- Cover the same material, but using somewhat different slides/notes
- Same textbook
- Common online homework
- Different exams/ exam dates

Course Logistics

Grading

- Homework (20%)
  - Released on canvas weekly, due **Thursday 10:30am**
  - No late submissions (**absolute**)
- Three projects (30%)
  - Form groups of two or three
- Midterm exam(20%)
  - March 8th 11:00 am - 12:15 pm, in class, closed book/notes, one letter size cheat sheet
- Final exam(20%)
  - May 12th 8:00 – 10:00 am, in class, closed book/notes, one letter size cheat sheet

Course Logistics

Quick Links

- Class website
- Textbook: A Course in Machine Learning
- Assignments release and submission: Canvas
- Grades: Canvas
- After class discussions/surveys, ask questions and everything else: Piazza
  - **Please use Piazza instead of email**

What is Learning?

- Ability to use previous data to perform future actions
- Biological systems do it all the time

A definition due to Simon (1983) is one of the best:

"Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the task or tasks drawn from the same population more efficiently and more effectively the next time."

Simon, Herbert A. "Why should machines learn?." *Machine learning*. Springer Berlin Heidelberg, 1983. 25-37.

Today’s Topics

What does it mean to “learn by example”?

- Classification tasks
- Inductive bias
- Formalizing learning

Classification tasks

- How would you write a program to distinguish a picture of me from a picture of someone else?

- Provide examples pictures of me and pictures of other people and let a **classifier** learn to distinguish the two.

Classification tasks

- How would you write a program to distinguish a sentence is grammatical or not?
- Provide examples of grammatical and ungrammatical sentences and let a **classifier** learn to distinguish the two.

Classification tasks

- How would you write a program to distinguish cancerous cells from normal cells?
- Provide examples of cancerous and normal cells and let a **classifier** learn to distinguish the two.

Let's try it out…

- Your task:

learn a classifier to distinguish

class A from class B

from examples

Examples of Class A





Examples of Class B





Let’s try it out…

✓ Learn a classifier from examples

- Now: predict class on new examples using what you’ve learned

What if I told you…

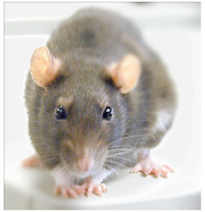
B


B

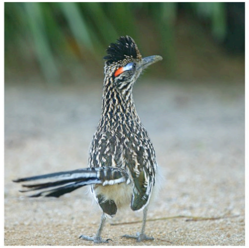

A


B


A


B


B


A


A


Plausibly other hypotheses: is the background in focus or not?

But nobody comes up with them: why? Inductive bias --- some hypotheses are more probable than others.

Key ingredients needed for learning

Training vs. test examples

- Memorizing the training examples is not enough!
- Need to generalize to make good predictions on test examples

Inductive bias

- Many classifier hypotheses are plausible
- Need assumptions about the nature of the relation between examples and classes

Machine Learning as Function Approximation

Problem setting

- Set of possible instances X
- Unknown target function $f: X \rightarrow Y$
- Set of function hypotheses $H = \{h \mid h: X \rightarrow Y\}$

Input

- Training examples $\{(x^{(1)}, y^{(1)}), \ldots (x^{(N)}, y^{(N)})\}$ of unknown target function f

Output

- Hypothesis $h \in H$ that best approximates target function f

Formalizing induction: Loss Function

$l(y, f(x))$ where y is the truth and $f(x)$ is the system's prediction

$$\text{e.g. } l(y, f(x)) = \begin{cases} 0 & if\ y = f(x) \\ 1 & otherwise \end{cases}$$

Captures our notion of what is important to learn

Formalizing induction:
Data generating distribution

Where does the data come from?

- Data generating distribution

  A probability distribution D over (x, y) pairs

- We don't know what D is!

  We only get a random sample from it: our training data

Formalizing induction: Expected loss

- f should make good predictions
  - as measured by loss l
  - on **future** examples that are also drawn from D
- Formally
  - ε , the expected loss of f over D with respect to l should be small

  $$\varepsilon \triangleq \mathbb{E}_{(x,y)\sim D}\{l(y, f(x))\} = \sum_{(x,y)} D(x,y) l(y, f(x))$$

Formalizing induction: Training error

- We can't compute expected loss because we don't know what D is

- We only have a sample of D
  - training examples $\{(x^{(1)}, y^{(1)}), \dots (x^{(N)}, y^{(N)})\}$

- All we can compute is the training error

$$\hat{\varepsilon} \triangleq \sum_{n=1}^{N} \frac{1}{N} l(y^{(n)}, f(x^{(n)}))$$

Formalizing Induction

- Given
  - a loss function l
  - a sample from some unknown data distribution D

- Our task is to compute a function f that has low expected error over D with respect to l.

$$\mathbb{E}_{(x,y)\sim D}\{l(y, f(x))\} = \sum_{(x,y)} D(x,y)\,l(y, f(x))$$

Recap: introducing machine learning

- What does “learning by example” mean?
- Classification tasks
- Learning requires examples + inductive bias
- Generalization vs. memorization
- Formalizing the learning problem
  - Function approximation
  - Learning as minimizing expected loss

Your task before next class

- Check out course webpage, Canvas, and Piazza
- Do the readings
- Get started on HW01: due Thursday 10:30am
- Let me know dates of religious holiday you observe this semester
- Let me know if you will need DSS arrangements

UNIVERSITY OF MARYLAND

Furong Huang
3251 A.V. Williams, College Park, MD 20740
301.405.8010 / furongh@cs.umd.edu