Introduction to Machine Learning

CMSC 422

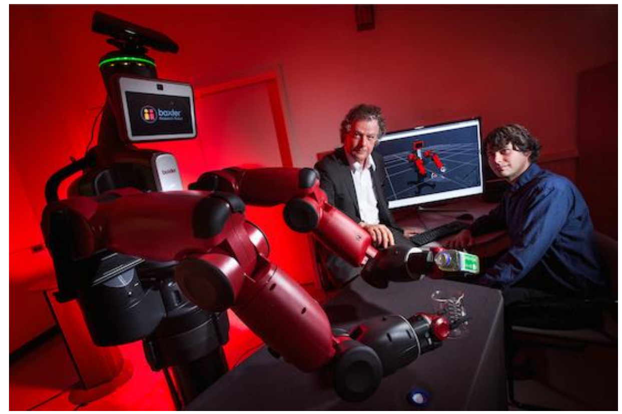
MARINE CARPUAT
marine@cs.umd.edu

What is this course about?

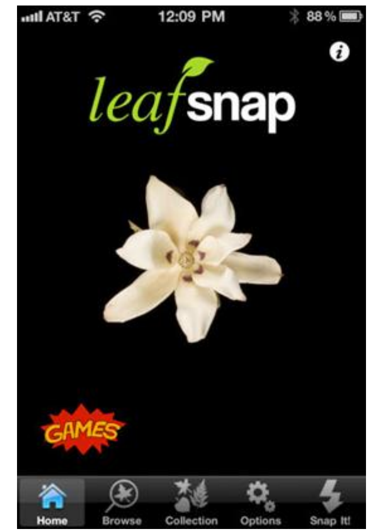
- Machine learning studies algorithms for learning to do stuff
- By finding (and exploiting) patterns in data

What can we do with machine learning?

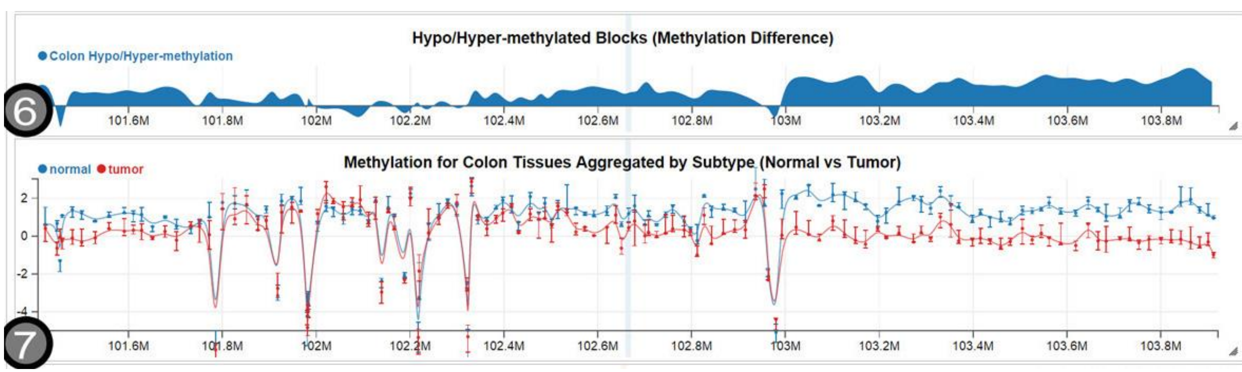
Analyze text & speech



Teach robots how to cook from youtube videos



Recognize objects in images



Analyze genomics data

Sometimes machines even perform better than humans!

nature International weekly journal of science

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

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

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?

A growing and fast moving field

- Broad applicability
  - Finance, robotics, vision, machine translation, medicine, etc.
- Close connection between theory and practice
- Open field, lots of room for new work!

Course Goals

- 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, …

Topics

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

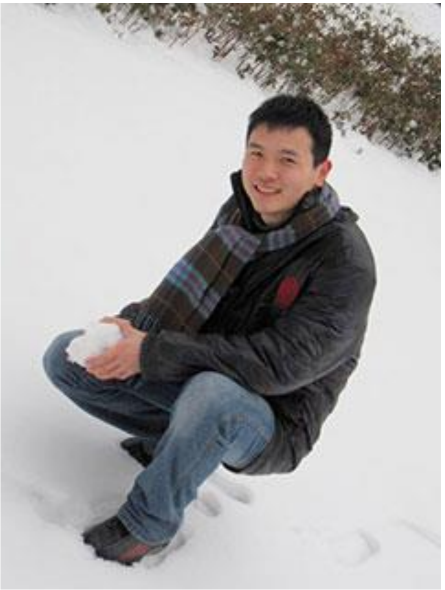
Advanced Supervised Learning
- Linear models and gradient descent
- Support Vector Machines
- Naive Bayes models and probabilistic modeling
- Neural networks
- Kernels
- Ensemble learning

Unsupervised learning
- K-means
- PCA
- Expectation maximization

What you can expect from the instructors

Teaching Assistant:

Xing Niu



We are here to help you learn by

- Introducing concepts from multiple perspectives
  - Theory and practice
  - Readings and class time
- Providing opportunities to practice, and feedback to help you stay on track
  - Homeworks
  - Programming assignments

What I expect from you

- Work hard (this is a 3-credit class!)
  - Do a lot of math (calculus, linear algebra, probability)
  - Do a fair amount of programming
- Come to class prepared
  - Do the required readings!

Highlights from course logistics

Grading

- *Homeworks (20%)*, ~10, almost weekly
- *Programming projects (30%)*, 3 of them, in teams of two or three students
- *Midterm exam (20%)*, in class
- *Final exam (30%)*, cumulative, in class.

- HW01 is due Thu 10:59am
- No late homeworks
- Read syllabus here: http://www.cs.umd.edu/class/spring2017/cmsc422//syllabus/

Where to…

- find the readings: [A Course in Machine Learning]
- view and submit assignments: Canvas
- check your grades: Canvas
- ask and answer questions, participate in discussions and surveys, contact the instructors, and everything else: Piazza
  Please use piazza instead of email

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.

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…

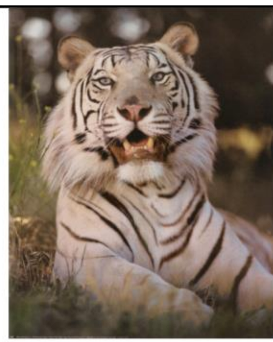
- ✓ learn a classifier from examples
- Now: predict class on new examples using what you've learned

What if I told you…


B


B

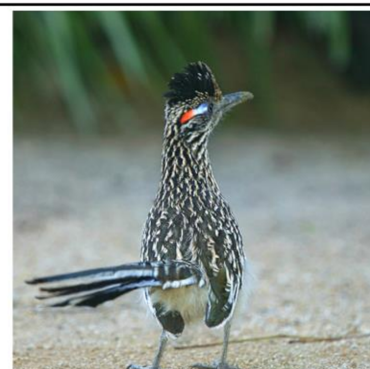

A


B


A


B


B


A


A

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 $\{(x^{(1)}, y^{(1)}), \ldots (x^{(N)}, y^{(N)})\}$

- 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 tasks before next class

- Check out course webpage, Canvas, Piazza
- Start reading
- Get started on HW01

- Let me know dates of religious holidays you observe this semester
- Let me know if you will need DSS arrangements