Introduction to Machine Learning

CMSC 422

Ramani Duraiswami

Machine Learning studies representations and algorithms that allow machines to improve their performance on a task from experience. This is a broad overview of existing methods for machine learning and an introduction to adaptive systems in general

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

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

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 and deep learning
  - Kernels
  - Ensemble learning

Topics

- Unsupervised learning
  - K-means
  - PCA
- Selected advanced topics (as time permits)
  - Expectation maximization
  - Online learning
  - Markov decision processes
  - Imitation learning

Homework

- Will try to have it at least every week
- Will not be excessive
- Essential for learning --- must do as in addition to read.
- Homework will be released on Canvas
- 20%
- No late homework

Homeworks

HW01: Warm up
Available until Feb 1 | Due Feb 1 at 2:59pm | 16 pts

HW02: Decision trees
Not available until Feb 1 | Due Feb 1 at 2:59pm | 7 pts

HW03: High Dimensional Space
Not available until Feb 2 | Due Feb 8 at 2:59pm | 3 pts

HW04: Perceptron
Not available until Feb 9 | Due Feb 15 at 2:59pm | 4 pts

HW05: Multiclass Classification
Not available until Feb 21 | Due Mar 1 at 2:59pm | 6 pts

HW06: Linear Models
Not available until Mar 3 | Due Mar 14 at 1:45pm | 6 pts

HW07: Probabilistic Modeling
Not available until Mar 24 | Due Mar 29 at 2:59pm | 9 pts

HW08: Unsupervised Learning
Not available until Apr 1 | Due Apr 5 at 2:59pm | 6 pts

HW09: Neural Networks, Kernel Methods and SVM
Not available until Apr 17 | Due Apr 26 at 2:59pm | 14 pts

Textbook, and Class Preparation

- Textbook is free and online.
- Written by a colleague, Prof. Hal Daume' III
- http://ciml.info
- Expect you to read material from the text, and other readings before the class
- Many other notes and books available and a few are listed in the syllabus

Projects

- Three projects in Python
- Project 1: Classification
- Project 2: Multiclass and Linear Models
- Project 3: PCAs and SVMs
- Remember you cannot publish or share project solutions – cheating

Exams

- Exams
  - Mid Term exam worth 20%, Date TBD
  - Final exam worth 30 %, Saturday, May 13, 8:00-10:00am
  - Closed book, Closed notes, in class
  - Allowed a “cheat sheet” in your own handwriting

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

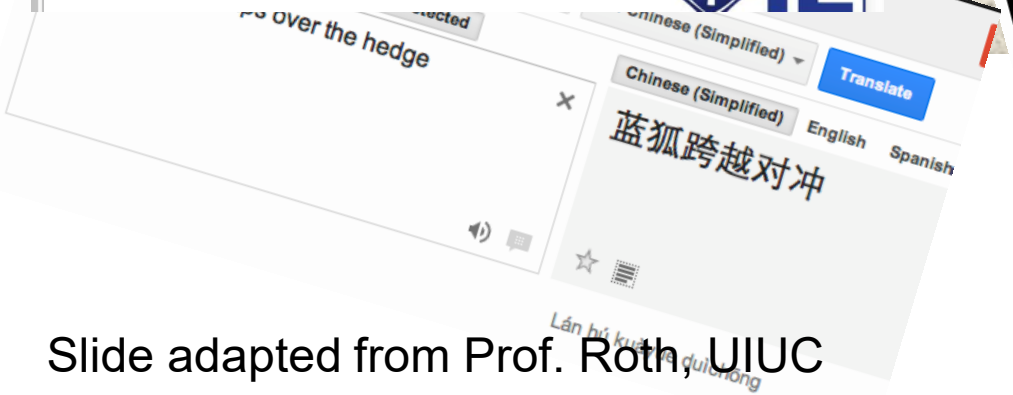
What is Learning?

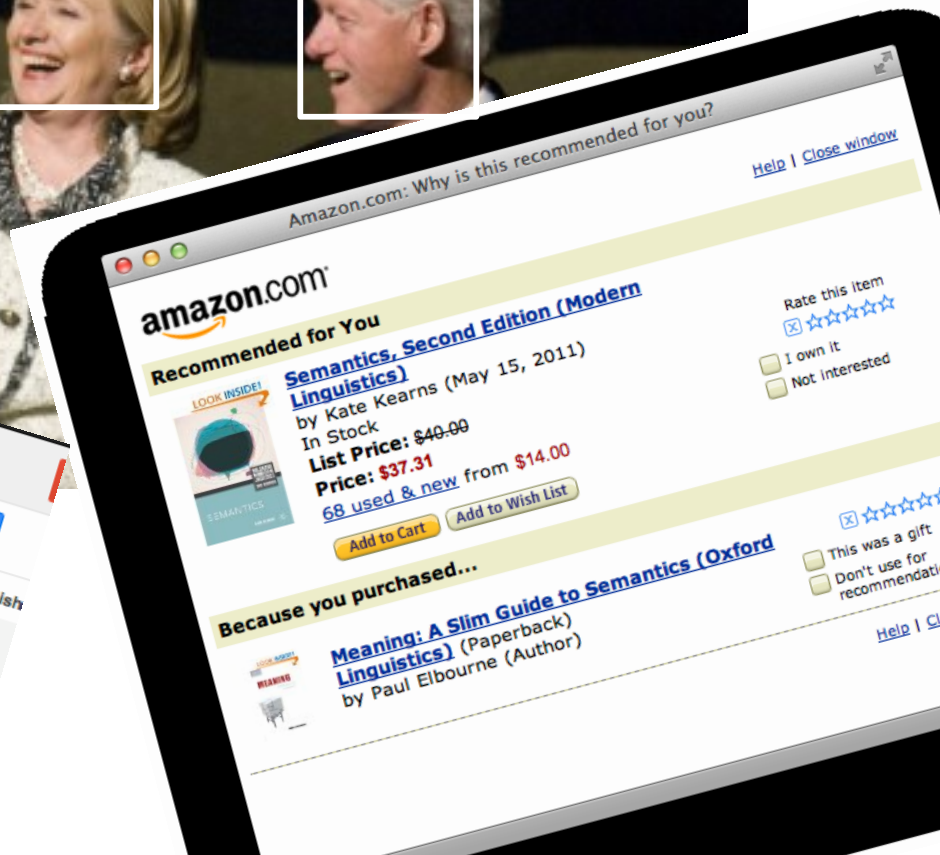
- Ability to use previous data to perform future actions
- Biological systems do it all the time

H. Simon -

"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."

Machine Learning is Everywhere





Slide adapted from Prof. Roth, UIUC

Learning is the future

- Learning techniques will be a basis for every application that involves a connection to the messy real world
- Basic learning algorithms are ready for use in applications today
- Prospects for broader future applications make for exciting fundamental research and development opportunities
- Many unresolved issues – Theory and Systems
  - While it’s hot, there are many things we don’t know how to do

Work in Machine Learning

- **Artificial Intelligence; Theory; Experimental CS**
- **Makes Use of:**
  - Probability and Statistics; Linear Algebra; Theory of Computation;
- **Related to:**
  - Philosophy, Psychology (cognitive, developmental), Neurobiology, Linguistics, Vision, Speech, Robotics,….
- **Has applications in:**
  - AI (Natural Language; Vision; Speech & Audio; Planning; HCI)
  - Engineering (Agriculture; Civil; …)
  - Computer Science (Compilers; Architecture; Systems; data bases)
  - Analytics

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…

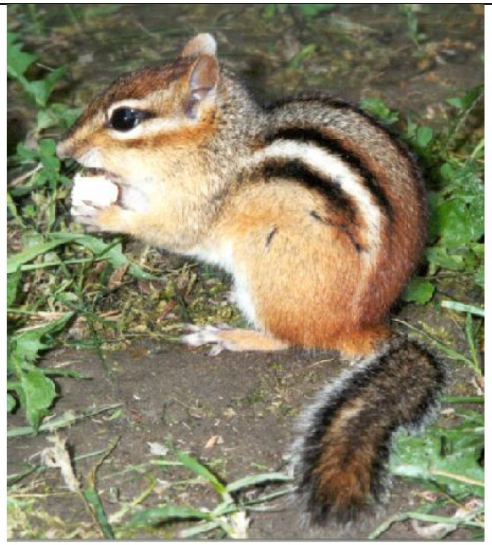
- Your task: learn a classifier to distinguish class A from class B from examples

- Examples of class A:





- Examples of class B

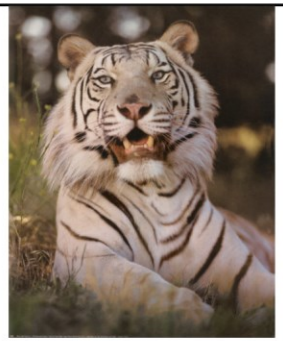
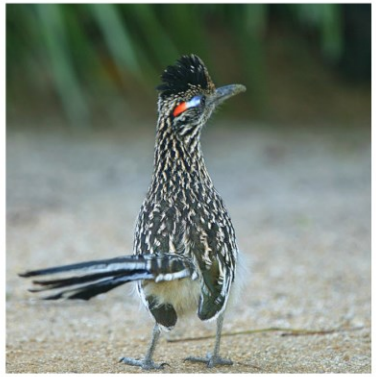




Let’s try it out…

- ✓ learn a classifier from examples

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

What if my program came up with …

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

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 it mean to “learn by example”?

- 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
- Do the readings
- Get started on HW01
  - due Thursday 10:59am