Machine Learning EN.601.475/675

DR. PHILIP GRAFF

Course Goals

01

Learn the fundamentals of machine learning

02

Learn to implement machine learning applications

03

Learn how to apply machine learning to different settings

Course Policies

- DISCUSSION ON PIAZZA
- ASSIGNMENT SUBMISSIONS AND GRADING ON GRADESCOPE
- 5 PROJECTS TOTALING 50%
- MIDTERM AND FINAL EXAMS EACH 25%
- READINGS FROM MURPHY RECOMMENDED
- CHEATING

Prerequisites

Math pre-requisites

Linear algebra (vector spaces, orthogonality, singular value decomposition)
Multivariate calculus (partial derivative, gradient, Hessian, Jacobian)
Probability and Statistics (random variables, probability distributions, expectations, mean, variance, covariance, conditional probability, law of large numbers, Bayes rule, MLE)

CS pre-requisites

Algorithms (Dynamic programming, basic data structures, complexity…)
Programming (Python 3)

General requirement

Ability to deal with “abstract mathematical concepts”

We provide some background, but the class will be fast-paced.

Course Schedule

PART 1: SUPERVISED LEARNING

1. Linear Regression
2. Logistic Regression
3. Perceptron and online learning
4. Support Vector Machines and the kernel trick
5. Decision trees, ensemble models, and boosting
6. Neural Networks

PART 2: UNSUPERVISED LEARNING AND MORE

1. Clustering
2. Expectation-Maximization and Gaussian mixture models
3. Dimensionality Reduction
4. Graphical models
5. Structured models (e.g. for sequences)
6. Practical ML and data science

General Course Advice

Read the readings

Python proficiency essential

Start projects early, don't wait

Read The Syllabus

Machine Learning Foundations

Definition 1

Machine learning allows computers to observe input and produce a desired output, either by example or by identifying latent patterns in the input.

Data
- What type of input?
- What type of output?

Patterns = Algorithm
- Intuition (empirical)
- Objective (theoretical)

Definition 2

Using **experience** to gain **expertise**

Programs that

- Learn “rules” from data
- Adapt to changes
- Improve performance with experience

Definition 3

Fitting a **function** to **data**

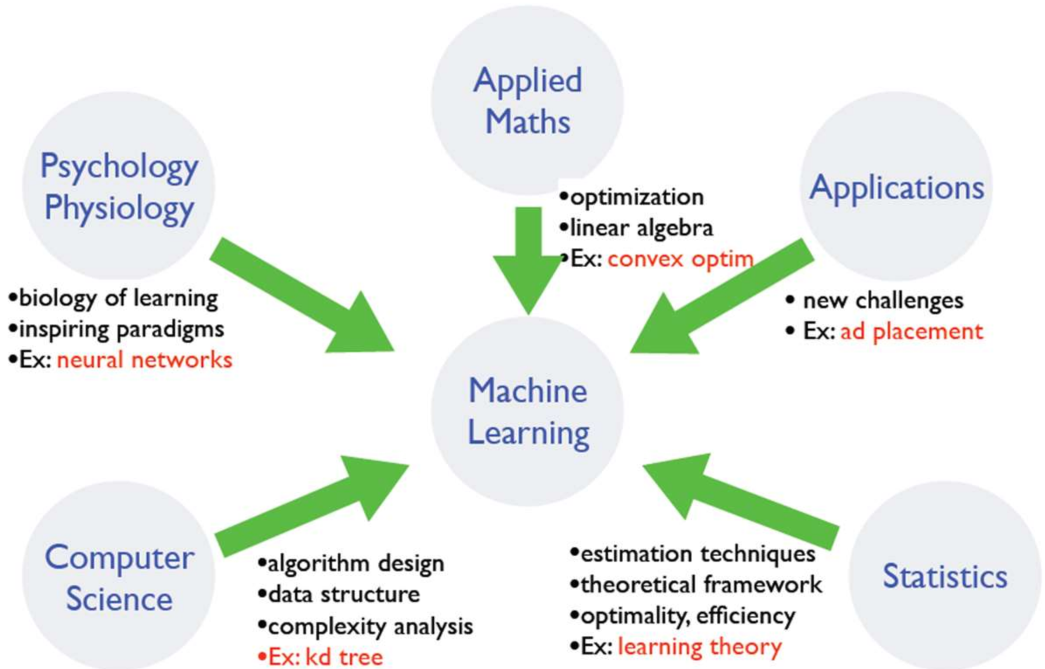
Fitting: Optimization, what parameters can we change?

Function: Model, loss function

Data: Data/model assumptions? How do we use data?

ML Algorithms: minimize a function on some data

A Diverse Topic



Credit to Fei Sha

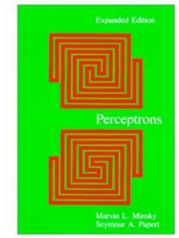
1950: Alan Turing devises the “Turing Test” for AI

1952: Arthur Samuel writes first learning program for checkers

1957: Frank Rosenblatt designs the perceptron – the seed for neural networks

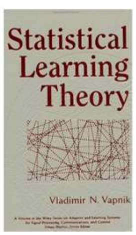
1967: Nearest neighbor algorithm created. Used for mapping routes (and for traveling salesman problem)

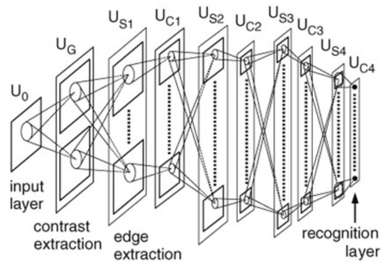
1969: Marvin Minsky, Seymour Papert (MIT)
“Perceptrons: An Introduction to Computational Geometry”
Criticism of Perceptron demonstrating limitations. Research in Perceptrons stops for almost 20 years



1970: Automatic differentiation (precursor to backpropagation) published

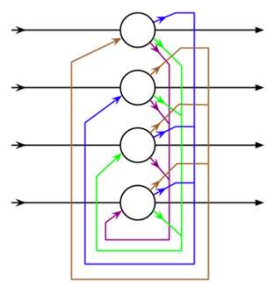
1974: Vapnik and Chervonenkis publish "Statistical learning theory" – Early idea behind SVMs. 1990- kernels, 1992- modern SVM





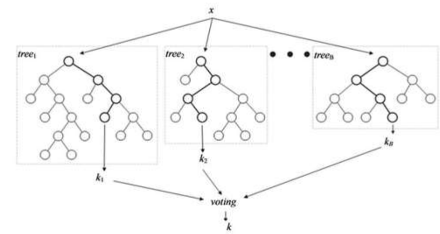
1980: Neocognitron published by Kunihiko Fukushima – type of ANN which later inspires CNNs

1982: RNNs popularized by John Hopfield



1986: Backpropagation published by David Rumelhart, Geoff Hinton and Ronald J. Williams

1989: Christopher Watkins develops Q-learning for reinforcement learning



1995: Tin Kam Ho publishes a paper on random forests

1997: IBM's Deep Blue beats Garry Kasparov

1998: MNIST handwriting dataset released by Yann LeCun – now a standard for handwriting recognition

2002: Torch machine learning library released

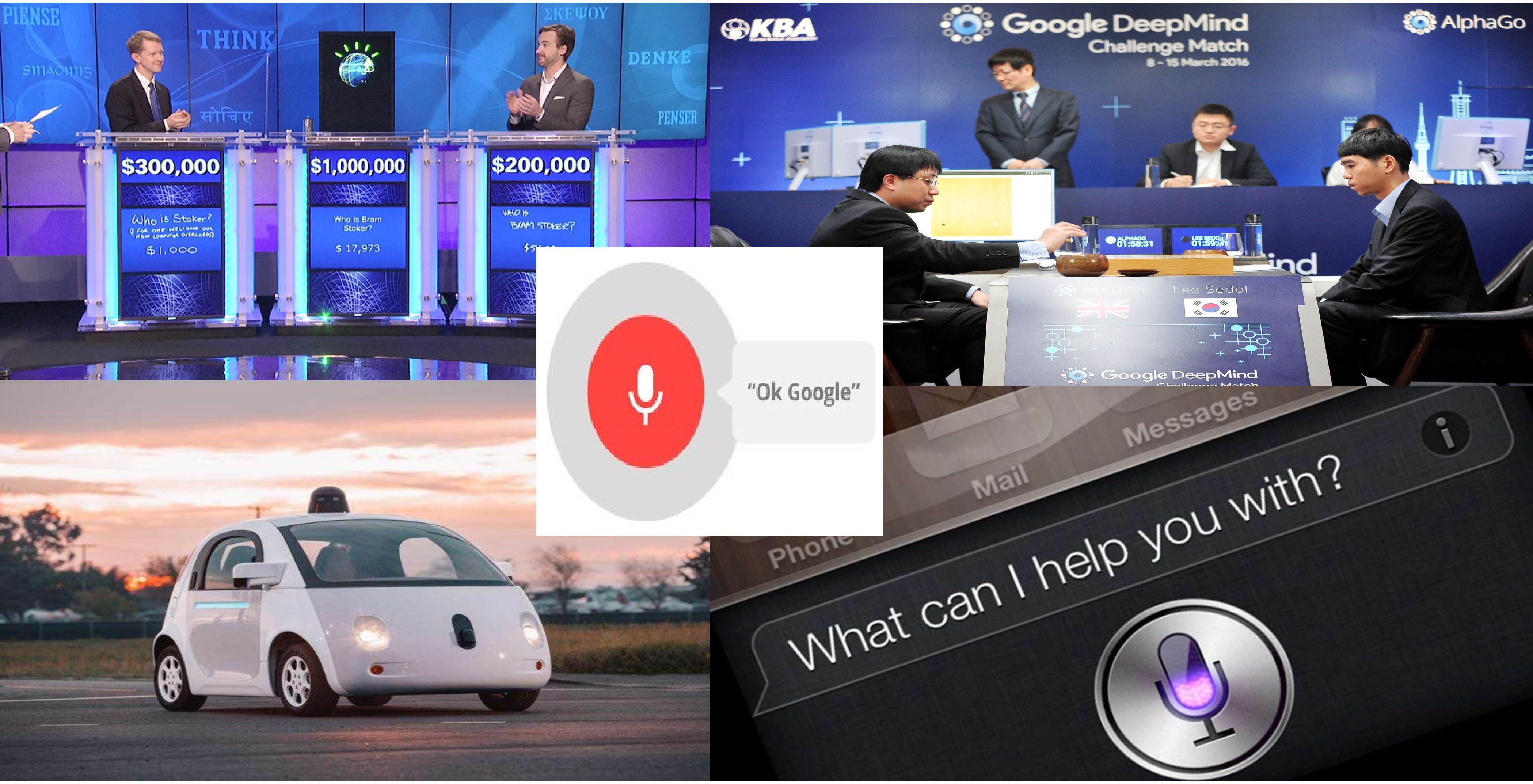
2006: Netflix prize for movie recommendations announced (no winner until 2009)

2010: Kaggle, a website hosting machine learning competitions, is launched

2011: IBM's Watson defeats human champions on *Jeopardy!*

AlphaGo

2016: Google's AlphaGo becomes first program to beat an unhandicapped human player in Go


PIENSE
THINK
ΣΚΕΨΟΥ
DENKE
PENSER
सोचिए
$300,000
Who is Stoker?
$ 1,000
$1,000,000
Who is Bram Stoker?
$ 17,973
$200,000
KBA
Google DeepMind
Challenge Match
8 - 15 March 2016
AlphaGo
Lee Sedol
Google DeepMind
"Ok Google"
Phone
Mail
Messages
What can I help you with?

Why Such Success?

(Somewhat) Better algorithms

- Deep learning

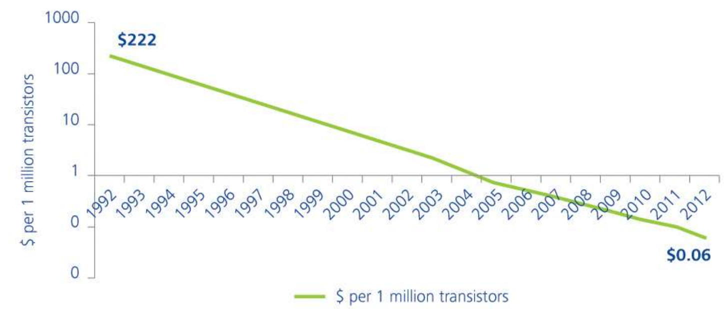
Better computers

More data

The rise of data science

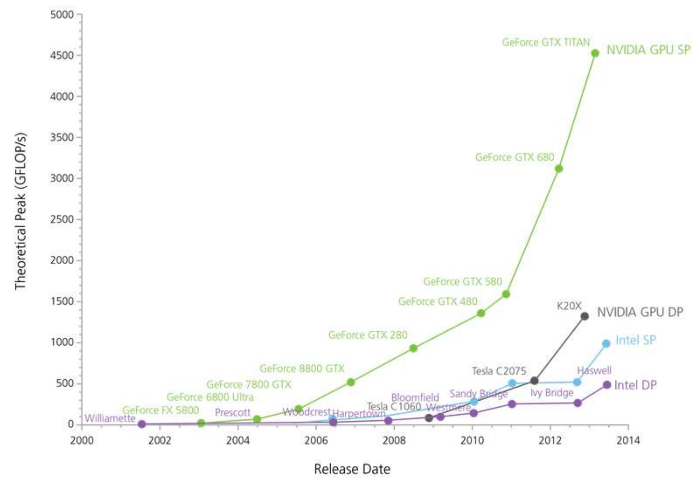
- More companies hire ML, more success (cycles)

Figure 1. Computing cost-performance (1992–2012)



Source: Leading technology research vendor

Graphic: Deloitte University Press | DUPress.com



ML vs. other Data Science

Data mining

- Emphasis on analysis and producing output for human understanding

Statistics

- Focused on explanation rather than prediction

Data science

- Many definitions. Broadly, combination of data mining, ML and statistics for “data wrangling”

Genres of ML

Recall our Definition

Fitting a function to data

Fitting: Optimization, what parameters can we change?

Function: Model, loss function

Data: Data/model assumptions? How do we use data?

ML Algorithms: minimize a function on some data

Data

It all comes down to data

- And the questions we want to answer!

What information is available for learning?

- What does the data look like?
- How is it annotated?

What output is desired?

- What should the algorithm produce?
- How will it be used?

Supervised Learning

Learning with a teacher
- Explicit feedback in the form of labeled examples
- Goal: make prediction
- Pros: Good performance
- Cons: Labeled data is difficult to find

Examples
- Regression
- Classification
  - Sort documents by topic
- Ranking
  - Sort web pages

Unsupervised Learning

Learning by oneself
- Only observed unlabeled examples
- Goal: uncover structure in data
- Pros: Easy to find lots of data
- Cons: Finding patterns of interest

Examples
- Clustering
  - Group emails by topic
- Manifold learning
  - Find a low-dimensional data representation

Semi-Supervised Learning

Labeled examples + unlabeled examples

Lots of ways to do this

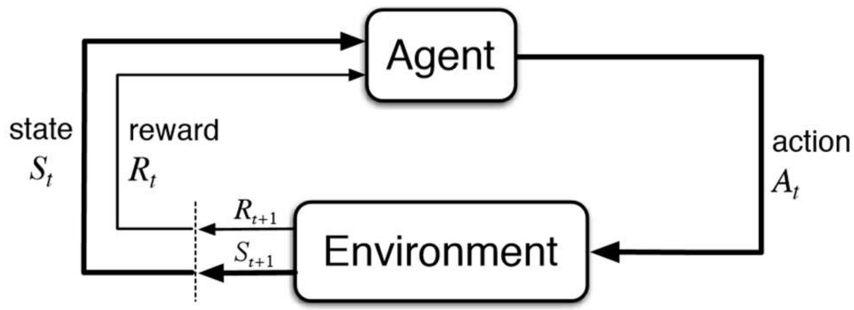
- Use unlabeled to guide learning in classification
  - Some documents labeled by topic, lots of unsorted docs
- Graph based models for labeling new data
  - Label propagation
- Other weak forms of supervision
  - A list of names, learn to extract more

Reinforcement Learning

Learn a behavior policy by interacting with the world
- How to navigate in a world
- Success measured by rewards received by actions
- Maximize rewards - costs

No examples
- You don't know how you did till its over
- E.g. Chess: Was that a good move? Did you eventually win?

Examples
- Checkers, Chess, Go, video games
- Robot control
- Piloting an airplane
- Google data center energy study



Parametric vs. Non-Parametric Models

PARAMETRIC

- Fixed number of parameters
- Faster to use
- Make stronger assumptions about the data distribution and relationships

➢ Nearly all algorithms in this class will be of this type

NON-PARAMETRIC

- Number of parameters grows with the size of the data
- More flexible
- Can be computationally intractable for large datasets

Which Model Should I Use?

"All models are wrong, but some models are useful."

George Box (Box and Draper 1987)

Models are just an approximation of reality but can produce useful results.

"No free lunch theorem"

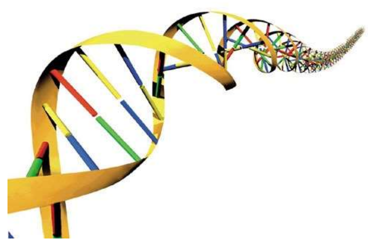
Wolpert 1996

Models that work well in one domain may work poorly in another.

Data Representation

Data is complex.

How does a computer algorithm see data?



High-Dimensional Vectors

One example is a vector of length *M*: $\mathbf{x}_i \in \Re^M$

Examples are drawn from some underlying distribution

Each dimension represents a feature

- Feature functions: $\mathbf{x}[j] = f_j(\text{item})$

A collection of *N* examples

$$\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$$

What Do Data Look Like?

Word counts from documents in a corpus

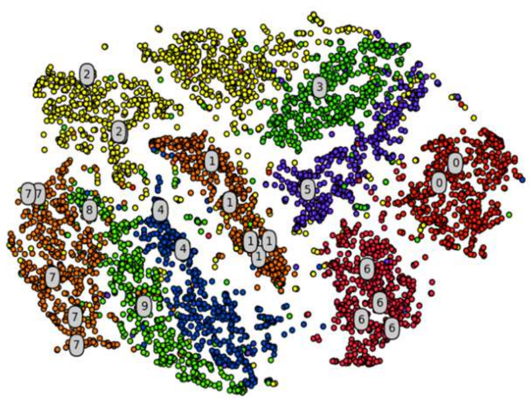
Nucleotides in a sequence of DNA

Customer satisfaction survey results for online purchases

Scatter plot of population characteristics

RGB pixel data from images or videos

Game logs from expert chess players



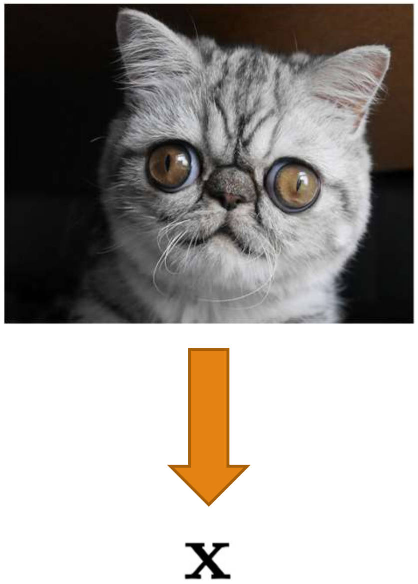
Data Features

Designing feature functions is critical
- Well designed representations greatly affect performance

How should we design features?
- Features are application specific
- You need to know about biology/vision/speech/etc.

Since this is domain specific, we won't talk much about it

More on this in last lecture



Next time: Linear Regression!