ECE 6254
Statistical Machine Learning
Spring 2024

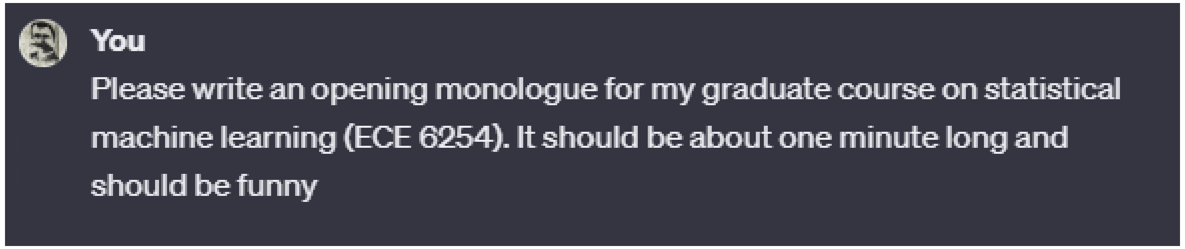
Mark A. Davenport
Electrical & Computer

Georgia Institute of Technology

Disclaimer!

None of what I just said was written by me…

Me at 11pm last night:



You

Please write an opening monologue for my graduate course on statistical machine learning (ECE 6254). It should be about one minute long and should be funny

- Lesson learned: If you want comedy (and nonsense), use Bard

Caveats

- I ***disavow*** the statement “this course is not all about math and equations”
- Natural language processing and self-driving cars are ***not*** going to be a central focus

Statistical machine learning

- How can we
  - *learn effective models* from data?
  - *apply these models* to practical inference and signal processing problems?

- Example problems: classification, regression, prediction, data modeling, clustering, and data exploration/visualization

- Our approach: ***statistical inference***

- **Main subject of this course**
  - how to reason about and work with probabilistic models to help us make inferences from data

What is machine learning?

learn: gain or acquire knowledge of or skill in (something) by
study, experience, or being taught

How do we learn that this is a tree?

My daddy told me that a tree is a perennial plant with an elongated stem, or trunk, supporting leaves or branches.

This has a trunk and branches.

Therefore it is a tree.

What is machine learning?

learn: gain or acquire knowledge of or skill in (something) by study, experience, or being taught

How do we learn that this is a tree?

EXAMPLES!

A good definition of learning for this course:

"using a set of examples to *infer* something about an underlying process"

Why learn from data?

Traditional signal processing is "top down"

Given a model for our data, derive the optimal algorithm

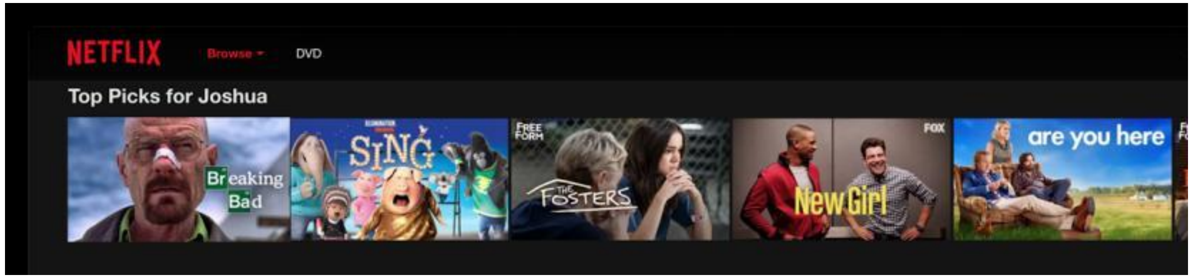
A learning approach is more "bottom up"

Given some examples, derive a good algorithm

Sometimes a good model is ***really*** hard to derive from first principles

Examples of learning

The Netflix prize (2007)



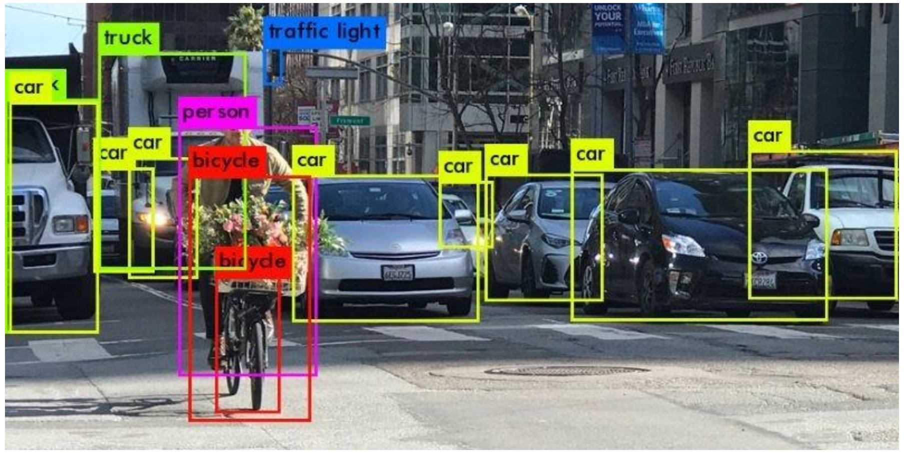
Predict how a user will rate a movie

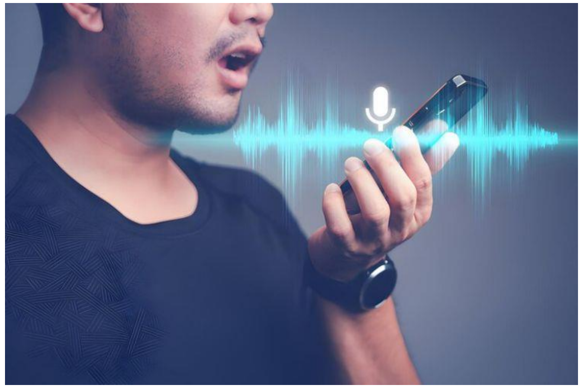
10% improvement = $1 million prize

- Some pattern exists
  - users do not assign ratings completely at random – if you like Godfather I, you'll probably like Godfather II
- It is hard to pin down the pattern mathematically
- We have lots and lots of data
  - we know how a user has rated other movies, and we know how other users have rated this (and other) movies

Examples of learning

- Recommendation systems
- Speech recognition
- Image classification
- Object detection
- Language modeling
- Spam filtering
- Machine translation
- Time series forecasting (traffic, weather, markets, etc.)
- Search
- Fraud detection
- Medical diagnosis
- …





Google Translate

Supervised learning

We are given input data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$

Each $\mathbf{x}_i$ represents a measurement or observation of some natural or man-made phenomenon

- may be called input, pattern, signal, feature vector, instance, or independent variable
- the coordinates may be called features, attributes, predictors, or covariates

In the supervised case, we are also given output data $y_1, \ldots, y_n$

- may be called output, label, response, or dependent variable

The data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ are called the ***training data***

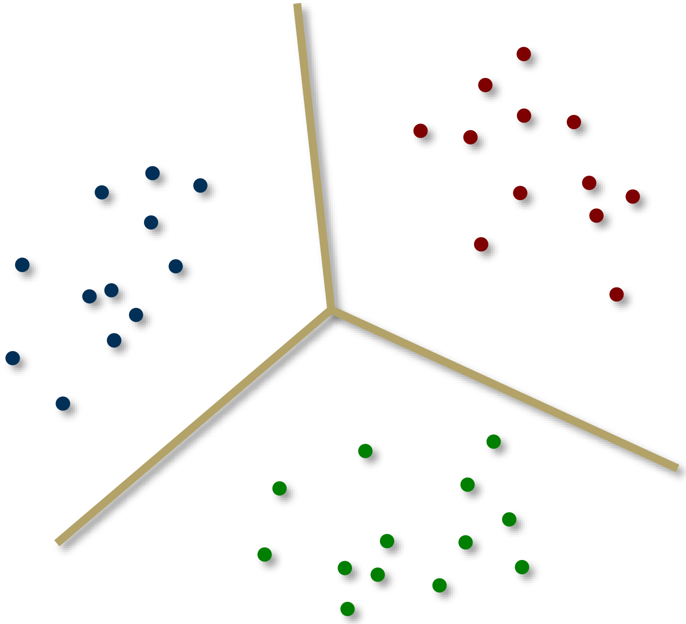
Supervised learning

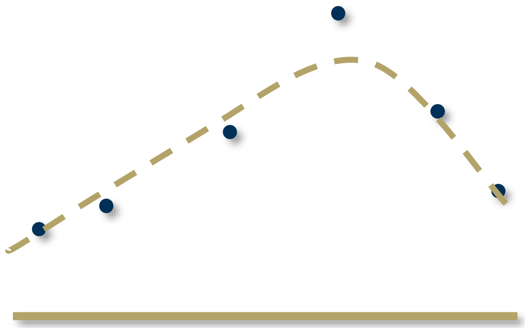
We can think of a pair $(\mathbf{x}_i, y_i)$ as obeying a (possibly noisy) input-output relationship

The goal of supervised learning is usually to ***generalize*** the input-output relationship so that we can predict the output y associated with a previously unseen input $\mathbf{x}$

The primary supervised learning problems are

- classification: $y \in \{1, \ldots, m\}$
- regression: $y \in \mathbb{R}$





Unsupervised learning

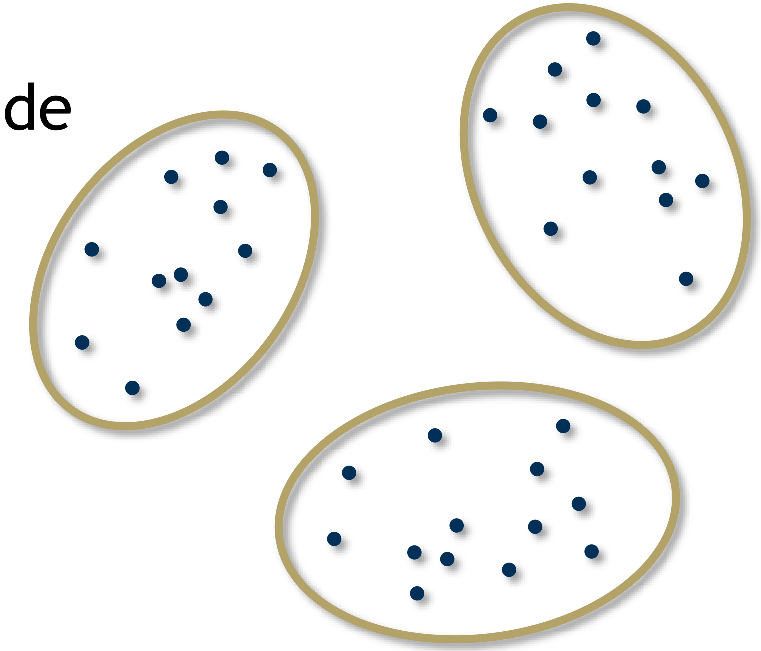
The inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are not accompanied by labels

The goal of unsupervised learning is typically not related to future observations

Instead we want to understand that structure in the data sample itself, or to infer some characteristic of the underlying probability distribution

Examples of unsupervised learning problems include

- clustering
- density estimation
- dimensionality reduction/feature selection
- visualization
- generative modeling



Other variants of learning

- semi-supervised learning
- self-supervised learning
- active learning
- online learning
- reinforcement learning
- anomaly detection
- transfer learning
- multi-task learning
- …

In general, most learning problems can be thought of as variants of traditional signal processing problems, but where we have ***no idea (a priori) how to model our signals***

Prerequisites

- Probability
  - random variables, expectation, joint distributions, independence, conditional distributions, Bayes rule, multivariate normal distribution, …

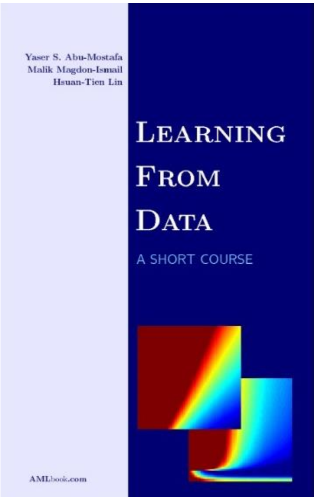
- Linear algebra
  - norms, inner products, orthogonality, linear independence, eigenvalues/vectors, eigenvalue decompositions, …

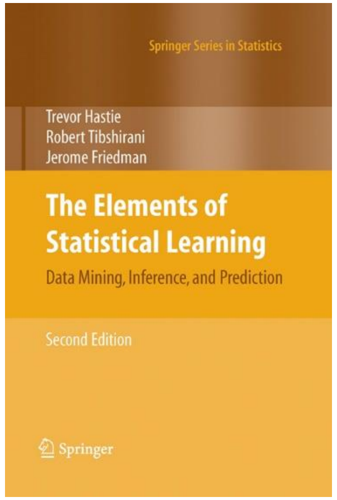
- Multivariable calculus
  - partial derivatives, gradients, the chain rule, …

- Python or similar programming experience (C or MATLAB)

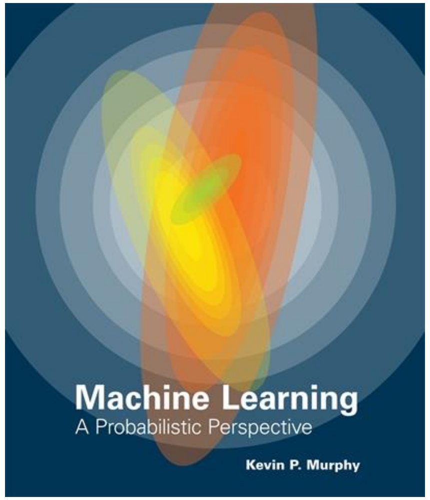
Text

There is no formally required textbook for this course, but I will draw material heavily from these sources:





Springer Series in Statistics
Trevor Hastie
Robert Tibshirani
Jerome Friedman
The Elements of Statistical Learning
Data Mining, Inference, and Prediction
Second Edition
Springer



A list of other useful books and links to relevant papers will be posted on the course webpage

Lecture notes and slides will also be posted on the course webpage

Grading

- Pre-test (5%)
- Homework (25%)
- Data challenges (10%)
- Midterm exam (20%)
- Final exam (20%)
- Final project (20%)

Distance learning

Welcome to our online students!

Recorded lectures will be available to ***all*** students (including on-campus students)

I need your help to make this a success

Online resources:

- Course website
- Canvas
- Piazza

A brief interlude

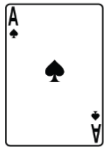
Could you learn this trick?

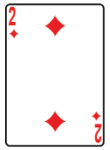
Suppose that

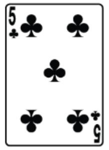
- $\mathbf{x}_i$ denotes the color of the card
  - 0 = black
  - 1 = red
- y_i denotes which card is hidden
  - E.g., Ace of Spades, Queen of Hearts, …

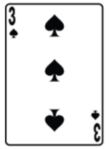
You observe me doing this trick many times and form a dataset:

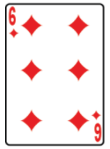
$\mathbf{x}_1 = 0$ $\mathbf{x}_2 = 1$ $\mathbf{x}_3 = 0$ $\mathbf{x}_4 = 0$ $\mathbf{x}_5 = 1$

$y_1 =$ 

$y_2 =$ 

$y_3 =$ 

$y_4 =$ 

$y_5 =$ 

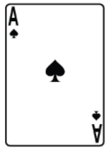
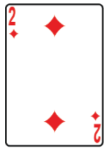
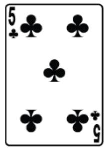
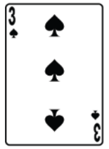
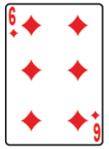
Can you learn a function such that $f(\mathbf{x})$ is a reliable predictor of y?

Another approach

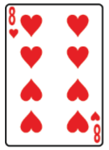
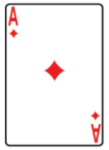
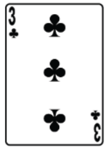
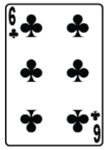
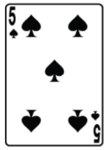
You watch me do this trick a couple times and notice I always hand out 5 cards

Suppose you instead consider

$$\mathbf{x}_1 = (0, 1, 0, 0, 1)$$

$y_1 = \Bigg($  ,  ,  ,  ,  $\Bigg)$

$$\mathbf{x}_2 = (1, 1, 0, 0, 0)$$

$y_2 = \Bigg($  ,  ,  ,  ,  $\Bigg)$

$$\vdots$$

Now, can you learn a function such that $f(\mathbf{x})$ is a reliable predictor of y?

Is learning even possible?

or: How I learned to stop worrying and love statistics

Supervised learning

Given training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, we would like to learn an (unknown) function $f : \mathcal{X} \to \mathcal{Y}$ such that $f(\mathbf{x}) = y$ for $\mathbf{x}$ other than $\mathbf{x}_1, \ldots, \mathbf{x}_n$

but…

as we have just seen, this is impossible. Without any additional assumptions, we conclude ***nothing*** about f except (maybe) for its value on $\mathbf{x}_1, \ldots, \mathbf{x}_n$

Probability to the rescue!

Any f agreeing with the training data may be ***possible***

but that does not mean that any f is equally ***probable***

A short digression

- Suppose that Javier has a biased coin, which lands on heads with some unknown probability p
  - $\mathbb{P}[\text{heads}] = p$
  - $\mathbb{P}[\text{tails}] = 1 - p$
- Javier toss the coin n times
  - $\widehat{p} = \frac{\#\text{ of heads}}{n}$

Does $\widehat{p}$ tell us anything about p ?

What can we learn from $\widehat{p}$?

Given enough tosses (large n), we expect that $\widehat{p} \approx p$

Law of large numbers

$$\widehat{p} \to p \text{ as } n \to \infty$$

Clearly, at least in a very limited sense, we can learn something about p from observations

There is always the ***possibility*** that we are totally wrong, but given enough data, the ***probability*** should be very small