CS 446/ECE 449: Machine Learning

Zhizhen Jane Zhao

University of Illinois at Urbana-Champaign

Spring 2023

L1: Overview and Nearest Neighbor

Goals of this lecture

- Organizational details of this class
- Introduction to machine learning
- k-Nearest Neighbor

Reading Material

- K. Murphy; Machine Learning: A Probabilistic Perspective; Chapter 1

Course Information

Course links

Lectures: Tuesday and Thursday 12:30pm–1:45pm

Lectures Zoom Info: Zoom link
Meeting ID: 829 7630 7784 Password: 210940

Recordings: Available on Illinois Media Space

Instructor office hours: 45 minutes after each lecture. Use the Lecture Zoom link.

TA office hours: Zoom link
Meeting ID: 856 8436 0417 Password: 194760

- Monday: 9am-10am, 3pm-4pm
- Tuesday: 8pm-9pm
- Wednesday: 6pm-7pm
- Thursday: 9am-10am
- Friday: 9am-10am

Course management

Lecture slides and Homework pdf will be posted on course website:
https://courses.grainger.illinois.edu/ece449/sp2023/

Homework submission will use gradescope:
https://www.gradescope.com

Post questions on Campuswire: https://campuswire.com/

Computing resources: We recommend Google Colab (free)
https://colab.research.google.com

Syllabus and Evaluation

Syllabus: see course website

Course deliverable:

1. Homework: 6 in total, due dates are available on course website
2. Quiz 1
3. Quiz 2 (non-accumulative)

Grading:
3 credit: 60% homework (drop 1 homework), 20% Quiz 1, 20% Quiz 2
4 credit: 60% homework (drop 0 homework), 20% Quiz 1, 20% Quiz 2

Grading policy is subject to change.

Notes

Make sure to familiarize yourself with necessary math topics:

- Basics of linear algebra
- Probability and statistics
- Basic algorithms
- Comfort with python programming

Welcome, looking forward to a fun semester!

Introduction to Machine Learning

- What is machine learning?
- What are some applications of machine learning?

T. Mitchell (Machine Learning)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

History (1930s to 1950s)

- Neurology research showed brain as electrical network of neurons
- Cybernetics (Norbert Wiener) described control and stability in electrical networks
- Information theory (Claude Shannon) described digital signals
- Theory of computation (Alan Turing) described digital nature of computation
- Electronic brain/neural net (Walter Pitts and Warren McCulloch)

Marvin Minsky (1955):

A Proposal for the

DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

June 17 - Aug. 16

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

Mentioned aspects:

- Automatic computers
- Programming a computer with language
- Neuron nets
- Theory on complexity of calculation
- Self-improvement
- Abstractions
- Randomness and Creativity

More recent applications of ML. Let's see some examples:

Example: Differentiating between 1000 image categories

Which object is illustrated?

- Car
- Truck
- Recreational Vehicle
- Ambulance truck
- Fire truck

Example: Instance level video segmentation

Example: Describing an image

Object Labels: 'person'
AG-CVAE sentences:
a man and a woman standing in a room
a man and a woman are playing a game
a man standing next to a woman in a room
a man standing next to a woman in a field
a man standing next to a woman in a suit

Object Labels: 'person', 'remote'
AG-CVAE sentences:
a man and a woman playing a video game
a man and a woman are playing a video game
a man and woman are playing a video game
a man and a woman playing a game with a remote
a woman holding a nintendo wii game controller

Object Labels: 'person','bus'
AG-CVAE sentences:
a man and a woman sitting on a bus
a man and a woman sitting on a train
a man and woman sitting on a bus
a man and a woman sitting on a bench
a man and a woman are sitting on a bus

Object Labels: 'person', 'train'
AG-CVAE sentences:
a man and a woman sitting on a train
a woman and a woman sitting on a train
a woman sitting on a train next to a train
a woman sitting on a bench in a train
a man and a woman sitting on a bench

Example: Asking questions about an image

What else?

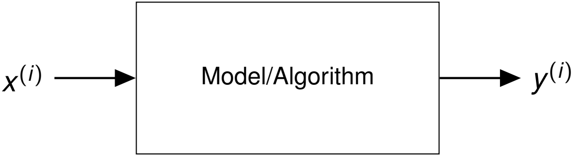
- Chess play (Deep Blue, IBM, 1997)
- Jeopardy (Watson, IBM, 2011)
- Playing Atari games (DQN, Deepmind, 2015)
- Game of Go (AlphaGo, Deepmind, 2016)
- Autonomous Driving (Audi, BMW, Mercedes, Uber, Waymo, Lyft, NVIDIA, Intel, ...)
- Dialogue applications (Open AI GPTChat, Google LaMDA, ...)
- Protein folding (AlphaFold, RoseTTAFold, ...)

Example

Let's get a computer to recognize whether there is a cat in the image.

How do we formulate the learning problem?

- input data/value/vector: $x^{(i)}$
- label/output: $y^{(i)}$



How do we call this process?

- Inference
- Prediction

What is learning here?

$$x^{(i)} \longrightarrow \boxed{\text{Model/Algorithm}} \longrightarrow y^{(i)}$$

- Model/Algorithm depends on parameters $\boldsymbol{w}$
- Learning/Fitting of parameters $\boldsymbol{w}$
- Based on dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$

How could we recognize a cat? What should the algorithm do?

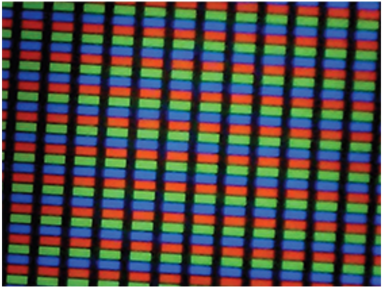
How does an image look like for a computer?

Demo

69	69	70	70	71	72	72
79	80	80	81	82	83	83
87	88	88	89	90	91	91
95	96	96	97	98	99	99
108	108	109	110	110	111	112
119	119	120	120	121	122	122
130	131	131	132	133	134	134
144	144	145	145	146	147	148
158	158	159	160	162	163	164
172	172	173	174	175	176	177
187	188	189	190	191	192	193
198	198	199	200	202	203	204
207	207	208	209	210	212	212

How does an image look like for a computer?

Cell phone screen:



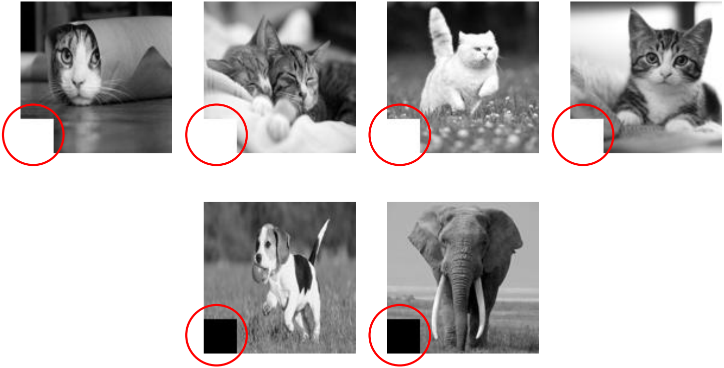
Given that we know how a computer 'sees' an image, how can it recognize cats?



Still very hard to describe what a cat looks like.

Let's look at tons of examples (Dataset).

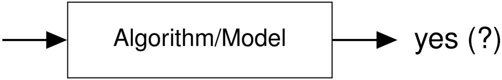
Dataset (Thousands of examples):



Instead of asking to recognize cats in general, let's recognize cats in this dataset

Algorithm: if bottom left corner is black (0) say no, otherwise say yes

Algorithm: if bottom left corner is black (0) say no, otherwise say yes



Works perfect on our dataset. :)
But does not generalize to other data. :(

Conclusion:

We designed a simple "classifier" that works on this dataset but doesn't work on real data.

To our rescue:

Machine learning found mechanisms to search for mappings which generalize.

Scope of this class:

In this class we talk about algorithms and models. A detailed treatment about generalization is left to lectures on learning theory.

Problem class

There are many different *problem classes* in machine learning. They vary according to what kind of data is provided and what kind of conclusions are to be drawn from it.

- Unsupervised learning
- Supervised learning
- Sequence learning
- Reinforcement learning
- Other settings

Categorization of machine learning algorithms according to

- Available annotated data (supervised vs. unsupervised)
- Complexity of model (linear vs. non-linear)
- Structure of output (independent vs. structured)
- Modeling of data ($x^{(i)}$) or label ($y^{(i)}$) (generative vs. discriminative)

Let's start simple. Classification tasks:

- Spam filter
- Cancer classification
- ...

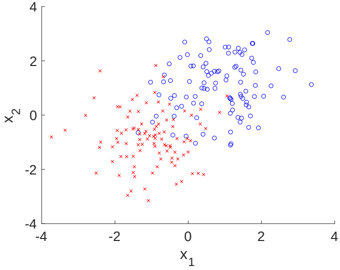
How to address those problems?

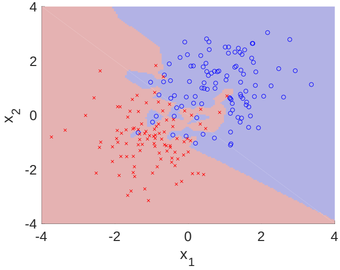
Nearest Neighbor–basics

- Dataset: $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
- New datapoint: x
- Label of new datapoint: y

How to choose y?

$$y = y^{(k)} \quad \text{where} \quad k = \arg \min_{i \in \{1,\ldots,N\}} \|x^{(i)} - x\|_2^2 = \arg \min_{i \in \{1,\ldots,N\}} d(x^{(i)}, x)$$





Nearest Neighbor–follow-up questions

- How many nearby neighbors to look at?
  k nearest neighbors. Take the majority vote or the average of nearest neighbors.

- How to measure “closeness" or “nearby"?
  Distance measurement, such as Euclidean distance, Hamming distance, cosine distance, etc.

- Optional weighting function on the neighbor points.

Discussions

Shortcomings of nearest neighbor prediction?

- Computationally expensive
- Curse of dimensionality. Nearest neighbor breaks down in high-dimensional spaces because the “neighborhood” become very large
- Memory issue
- Sensitive to outliers and easily fooled by irrelevant attributes.

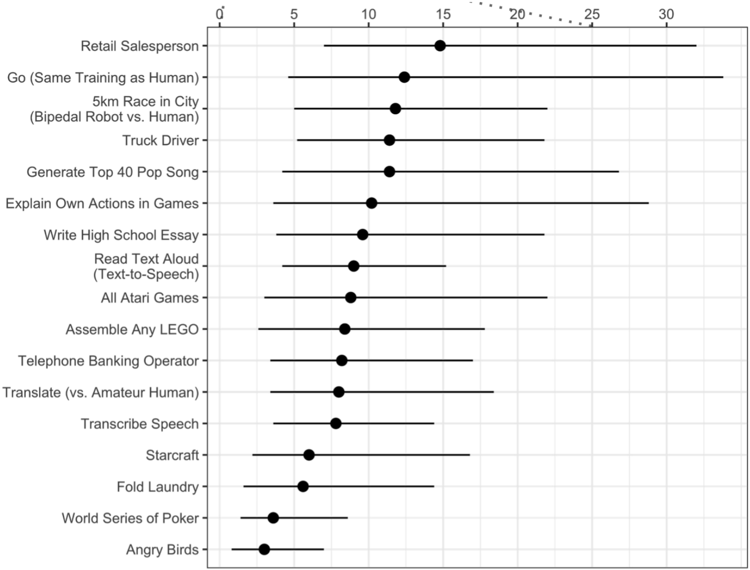
So far:

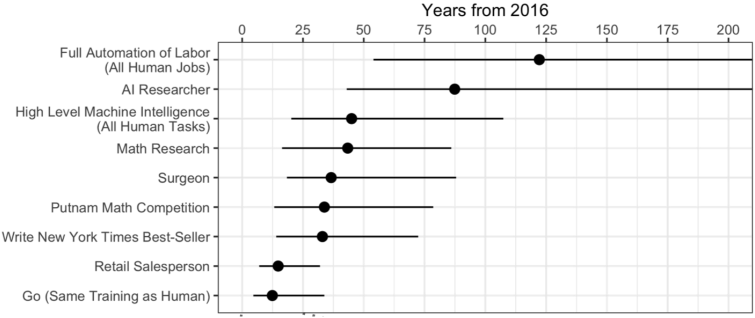
What can a computer do for you?

Now:

What will a computer be able to do for you?

Future: (years from 2016; from https://arxiv.org/pdf/1705.08807.pdf)




Years from 2016
0
25
50
75
100
125
150
175
200
Full Automation of Labor (All Human Jobs)
AI Researcher
High Level Machine Intelligence (All Human Tasks)
Math Research
Surgeon
Putnam Math Competition
Write New York Times Best-Seller
Retail Salesperson
Go (Same Training as Human)

Amazing future ahead. Let's make it happen.

Quiz:

- What is nearest neighbor?
- How do we formulate nearest neighbor?

Important topics of this lecture:

- Getting to know k-Nearest Neighbor
- Basic mechanism to address many tasks
- Think about how to solve a problem with Nearest Neighbor

Next up:

- Probability and Estimation