Fundementals of AI/ML

Preliminaries & the K-Nearest Neighbors

Ashley Y. Gao

William & Mary

January 21, 2026

What is learning?

- *The activity or process of gaining knowledge or skill by studying, practicing, being taught, or experiencing something.* - Merriam Webster dictionary

- *Learning can be defined as the process of acquiring knowledge, skills, attitudes, or understanding through study, experience, or teaching.* - ChatGPT

- *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.* - Tom Mitchell

What is Machine Learning?

- For many problems, it's difficult to program the correct behavior by hand
  - Recognizing people and objects
  - Understanding human speech
- Machine learning approach
  - Program an algorithm to automatically learn from data, or from experience
- Why might you want to use a learning algorithm?
  - Hard to code up a solution by hand (e.g. vision, speech)
  - System needs to adapt to a changing environment (e.g. spam detection)
  - Want the system to perform better than the human programmers

What is Machine Learning?

- It's similar to statistics...
  - Both fields try to uncover patterns in data
  - Both fields draw heavily on calculus, probability, and linear algebra, and share many of the same core algorithms
- But it's not statistics!
  - Stats is more concerned with helping scientists and policymakers draw good conclusions; ML is more concerned with building autonomous agents
  - Stats puts more emphasis on interpretability and mathematical rigor; ML puts more emphasis on predictive performance, scalability, and autonomy

Relations to Artificial Intelligence

- Nowadays, “machine learning” is often brought up with “artificial intelligence” (AI).
- AI does not always imply a learning based system.
  - Rule-based systems
  - Symbolic reasoning
  - Tree search
  - etc.
- Learning-based system $\rightarrow$ learned based on the data $\rightarrow$ more flexibility, good at solving pattern recognition problems.

Relations to Human Learning

- Human learning is:
  - Very data efficient
  - An entire multitasking system (vision, language, motor control, etc.)
  - Takes at least a few years :)
- For serving specific purposes, machine learning doesn't have to look like human learning in the end.
- It may borrow ideas from biological systems, e.g., neural networks. It may perform better or worse than humans.

What is Machine Learning

- Types of machine learning
  - Supervised learning: have labeled examples of the correct behavior
  - Reinforcement learning: learning system (agent) interacts with the world and learns to maximize a scalar reward signal
  - Unsupervised learning: no labeled examples – instead, looking
- Looking for “interesting” patterns in the data.

AI & ML in Ancient Times

- The idea of AI & ML can be traced to ancient Greece & Rome (but it is not clear where it originated)!

- Pygmalion, who created Galatea

- Hephaestus, who created Talos

History of Machine Learning

- 1957 — Perceptron algorithm (implemented as a circuit!)
- 1959 — Arthur Samuel wrote a learning-based checkers program that could defeat him
- 1969 — Minsky and Papert's book Perceptrons (limitations of linear models)
- 1980s — Some foundational ideas
  - Connectionist psychologists explored neural models of cognition
  - 1984 — Leslie Valiant formalized the problem of learning as PAC learning
  - 1988 — Backpropagation (re-)discovered by Geoffrey Hinton and colleagues
  - 1988 — Judea Pearl's book Probabilistic Reasoning in Intelligent Systems introduced Bayesian networks

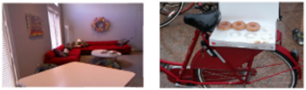
History of Machine Learning

- 1990s — the "AI Winter", a time of pessimism and low funding But looking back, the '90s were also sort of a golden age for ML research
  - Kernels and support vector machines
  - Boosting
  - Convolutional networks
  - Reinforcement learning
- 2000s — applied AI fields (vision, NLP, etc.) adopted ML
- 2010s — deep learning
  - 2010–2012 — neural nets smashed previous records in speech-to-text and object recognition
  - Increasing adoption by the tech industry
  - 2016 — AlphaGo defeated the human Go champion
  - 2018-now — generating photorealistic images and videos
  - 2020 — GPT3 language model
- now — increasing attention to ethical and societal implications

Computer Vision

- Computer vision: Object detection, semantic segmentation, pose estimation, and almost every other task is done with ML.

Figure 4. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

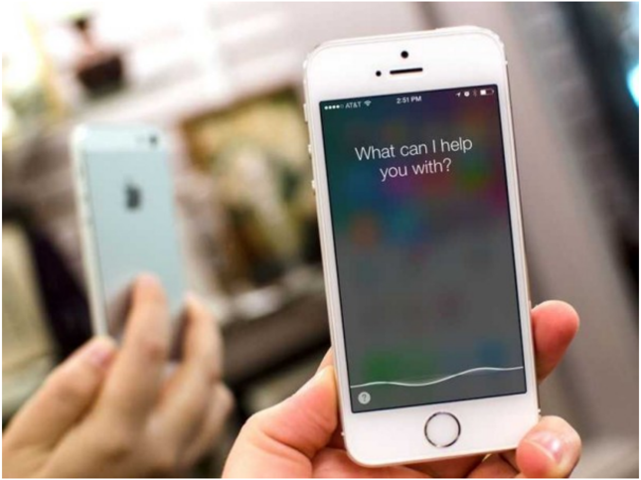


DAQUAR 1553
What is there in front of the sofa?
Ground truth: table
IMG+BOW: table (0.74)
2-VIS+BLSTM: table (0.88)
LSTM: chair (0.47)

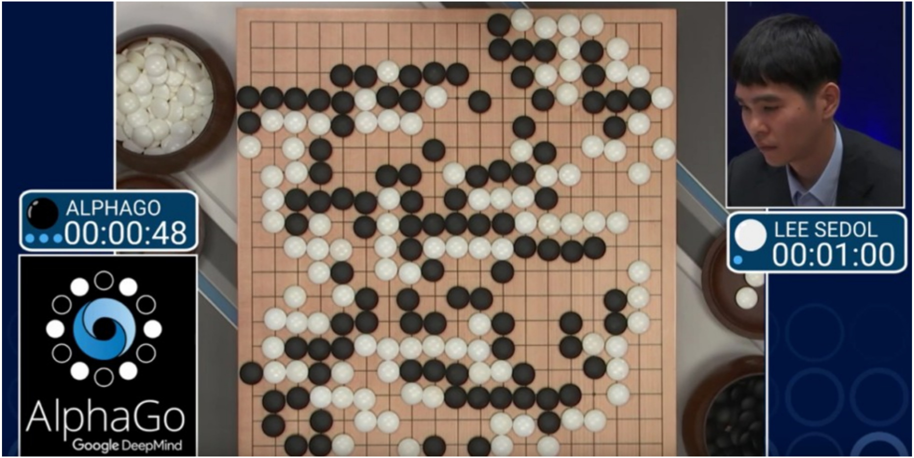
COCOQA 5078
How many leftover donuts is the red bicycle holding?
Ground truth: three
IMG+BOW: two (0.51)
2-VIS+BLSTM: three (0.27)
BOW: one (0.29)

Speech

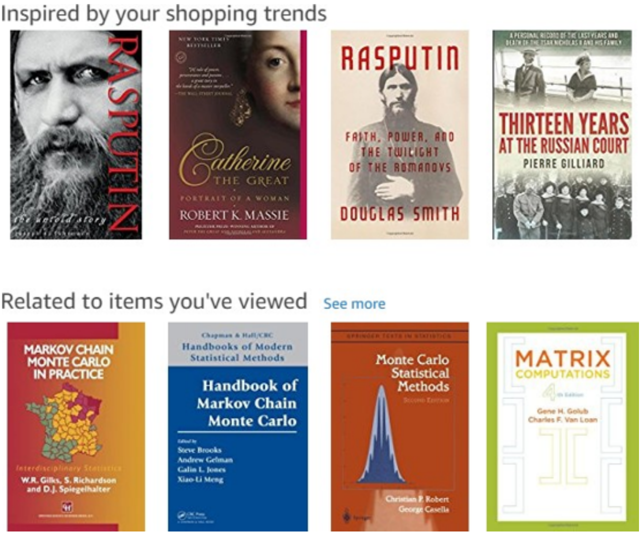
- Speech: Speech to text, personal assistants, speaker identification...



Competitive Gaming



E-Commerce and Recommender Systems

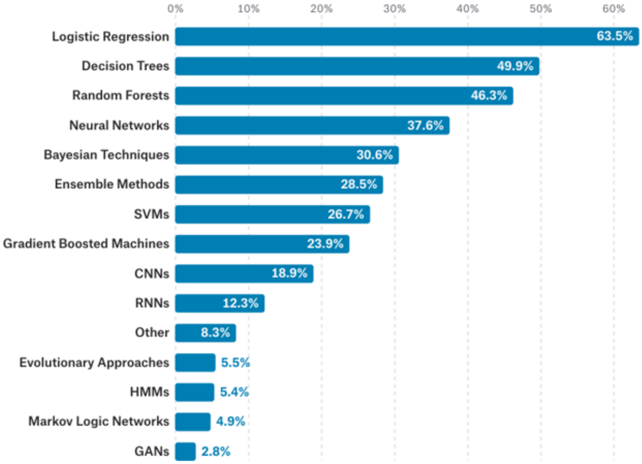


Why this Class?

- Why not jump straight to learn neural nets first?
  - The principles you learn in this course will be essential to understanding and applying neural nets.
  - The techniques in this course are still the first things to try for a new ML problem.
    - E.g., try logistic regression before building a deep neural net!

Why this Class?

- 2017 Kaggle survey of data science and ML practitioners: what data science methods do you use at work?



Why this Class?

- ML workflow sketch:
  - Should I use ML on this problem?
    - Is there a pattern to detect?
    - Can I solve it analytically?
    - Do I have data?
  - Gather and organize data.
  - Preprocessing, cleaning, and visualizing.
  - Establishing a baseline.
  - Choosing a model, loss, regularization, ...
  - Optimization (could be simple, could be a Phd...).
  - Hyperparameter search.
  - Analyze performance & mistakes, and iterate back to step 4 (or 2).

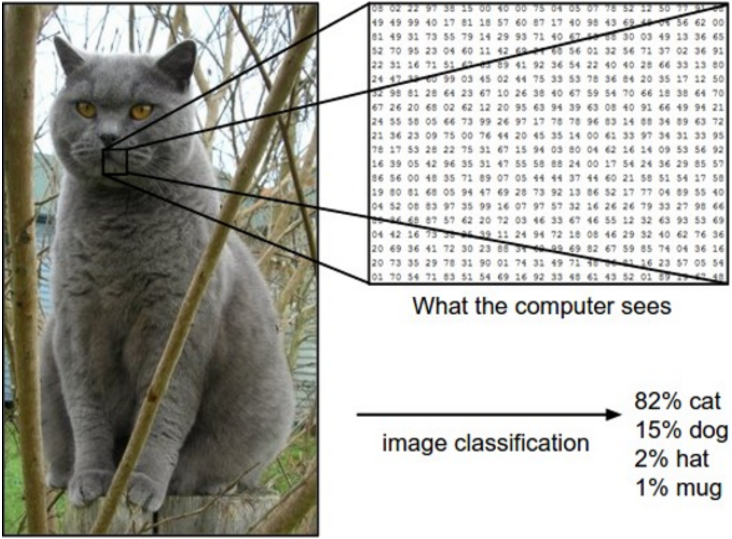
Preliminaries and Nearest Neighbor Methods

Introduction

- Today (and for much of this course) we focus on supervised learning.
- This means we are given a training set consisting of inputs and corresponding labels, e.g.

Task	Inputs	Labels
object recognition	image	object category
image captioning	image	caption
document classification	text	document category
speech-to-text	audio waveform	text
⋮	⋮	⋮

Input Vectors

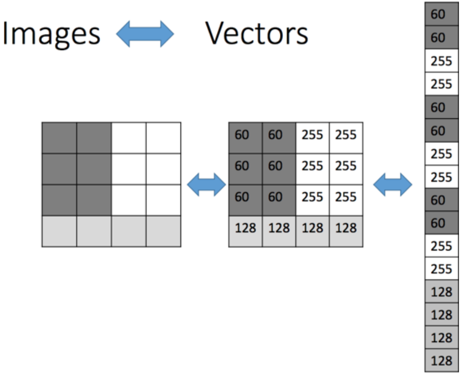


Input Vectors

- Machine learning algorithms need to handle lots of types of data: images, text, audio waveforms, credit card transactions, etc.
- Common strategy: represent the input as an input vector
  - Representation = mapping to another space that's easy to manipulate
  - Vectors are a great representation since we can do linear algebra!

Input Vectors

- Can use raw pixels.
- Can do much better if you compute a vector of meaningful features.



Input Vectors

- Mathematically, our training set consists of a collection of pairs of input vector $x \in \mathbb{R}^d$ and its corresponding target or label, t or y.
  - Regression: t/y is a real number (e.g. stock prices)
  - Classification: t/y is an element of a discrete set $\{1, ..., C\}$ (e.g. object classification)
  - t/y can also be a highly structured object, such as an image (e.g. image-to-image style transfer)
- The training set is denoted as $\{(x^1, t^1) ... (x^N, t^N)\}$ or $\{(x^1, y^1) ... (x^N, y^N)\}$
  - These subscriptions are not related to exponentiation.

The Nearest Neighbor

- Suppose we are given a novel input vector x that we want to classify.
- Idea: find in the training set the vector that is “nearest" to x in the feature space, and copy the label of that vector.
- can formalize “nearest" in terms of the Euclidean distance.

$$||x^{(a)} - x^{(b)}||_2 = \sqrt{\sum_{j=1}^{d}(x_j^{(a)} - x_j^{(b)})^2} \tag{1}$$

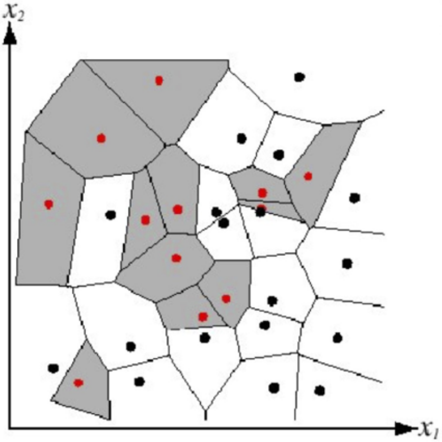
Algorithm:

Find example (x^*, t^*) from the stored training set closest to x. That is, looking through all $x^{(i)} \in$ the training set, we want to find:

$$x^* = \text{argmin distance}(x^{(i)}, x) \tag{2}$$

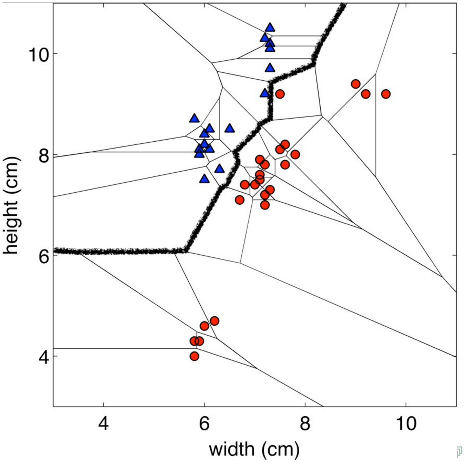
The Nearest Neighbor

- We can visualize the behavior in the classification setting using a Voronoi diagram.
- Samples are training samples.



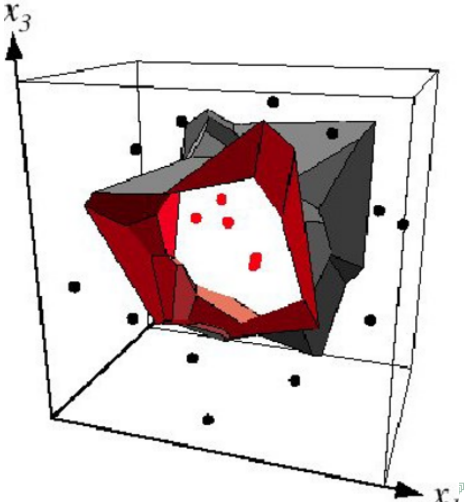
The Nearest Neighbor

- Decision boundary: the boundary between regions of input space assigned to different categories.
- Samples are training samples.

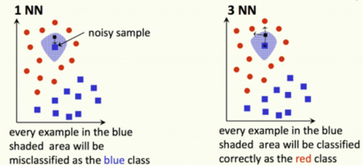


The Nearest Neighbor

- Example: 2D decision boundary
- Samples are **testing** samples (different from previous slides)!



The K Nearest Neighbors



- The nearest neighbor method is sensitive to noise or mislabelled samples (class noise).
- Smooth by having k nearest neighbors vote.

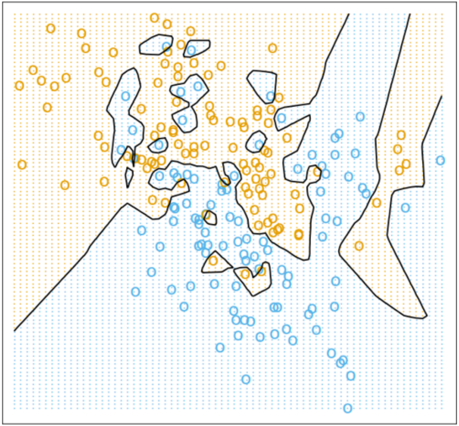
Algorithm (kNN):

- Find k examples $(x^{(i)}, t^{(i)})$ closest to the test instance x. Classification output is majority class.

$$y = \underset{t^{(z)}}{\operatorname{argmax}} \sum_{i=1}^{k} \mathbb{I}(t^{(z)} = t^{(i)}) \tag{3}$$

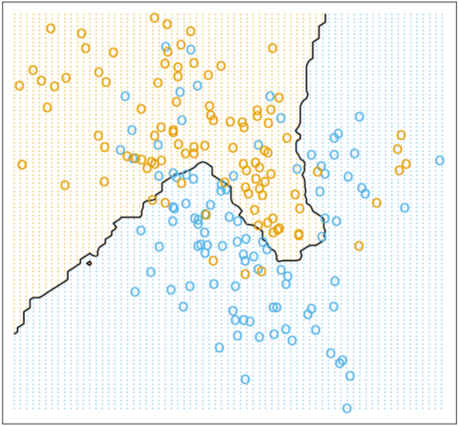
The K Nearest Neighbor

- K = 1.



The K Nearest Neighbor

- K = 15.

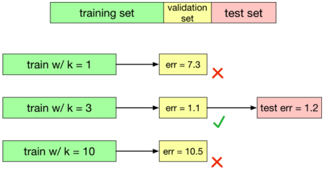


The K Nearest Neighbor

- Trade-off choosing k?
  - Small k
    - Good at capturing fine-grained patterns
    - May overfit, i.e. sensitive to the randomness in the training data
  - Large k
    - Makes stable predictions by averaging over a lot of samples
    - May underfit, i.e. fails to capture important regularities
  - Balancing k
    - Optimal value of k depending on the number of available samples, n
    - We can choose k using validation set
    - Rule of thumb: choose $k < \sqrt{n}$

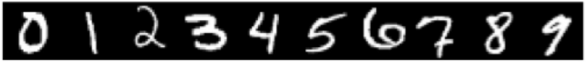
Validation and Testing Sets

- k is an example of a hyperparameter, something we can't fit as part of the learning algorithm itself
- We can tune hyperparameters using a validation set:
  - The test set is used only at the very end, to measure the generalization performance of the final configuration.



Example: Digit Classification

- Decent performance when lots of data.



- Yann LeCunn – MNIST Digit Recognition
  - Handwritten digits
  - 28x28 pixel images: d = 784
  - 60,000 training samples
  - 10,000 test samples
- Nearest neighbour is competitive

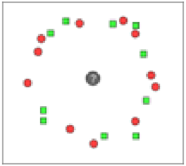
	Test Error Rate (%)
Linear classifier (1-layer NN)	12.0
K-nearest-neighbors, Euclidean	5.0
K-nearest-neighbors, Euclidean, deskewed	2.4
K-NN, Tangent Distance, 16x16	1.1
K-NN, shape context matching	0.67
1000 RBF + linear classifier	3.6
SVM deg 4 polynomial	1.1
2-layer NN, 300 hidden units	4.7
2-layer NN, 300 HU, [deskewing]	1.6
LeNet-5, [distortions]	0.8
Boosted LeNet-4, [distortions]	0.7

Pitfall: The Curse of Dimensionality

- As the dimensionality increases, the number of data points required for good performance of any machine learning algorithm increases exponentially.
- The curse of dimensionality becomes a problem in KNN when the number of dimensions in the feature space is large. As the dimensionality increases, the volume of the feature space grows exponentially, causing the data points to become more dispersed.
- This means that the distances between data points increase, and the notion of proximity becomes less meaningful.

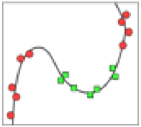
Pitfall: The Curse of Dimensionality

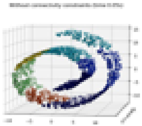
- In high dimensions, “most” points are approximately the same distance.
- Picture to keep in mind:



Pitfall: The Curse of Dimensionality

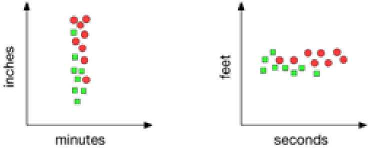
- Saving grace: some datasets (e.g. images) may have low intrinsic dimension, i.e. lie on or near a low-dimensional manifold.
- Intrinsic dimension, in the context of machine learning, refers to the minimum number of parameters or features required to represent and capture the essential characteristics of a dataset or a particular pattern within the data.
- The neighborhood structure (and hence the Curse of Dimensionality) depends on the intrinsic dimension.





Pitfall: Normalization

- Nearest neighbors can be sensitive to the ranges of different features.
- Often, the units are arbitrary:



- Simple fix: normalize each dimension to be zero mean and unit variance.
  - To achieve zero mean, the mean value of the data is subtracted from each data point. This process centers the data around zero, ensuring that the average value of the dataset becomes zero.
  - To achieve unit variance, each data point (after the previous step) is divided by the standard deviation of the dataset. This rescaling ensures that the data has a standard deviation of one, meaning that the data points are spread out evenly around the mean.

Pitfall: Normalization

- Caution: depending on the problem, the scale might be important!
- Simple fix: normalize each dimension to be zero mean and unit variance. i.e. compute the mean μ_j and standard deviation σ_j and take

$$\tilde{x}_j = \frac{x_j - \mu_j}{\sigma_j} \tag{4}$$

Pitfall: Computational Cost

- Number of computations at training time: 0
- Number of computations at test time, per query (naive algorithm)
  - Calculate D-dimensional Euclidean distances with N data points: $O(ND)$
  - Sort the distances: $O(NlogN)$
- This must be done for each query, which is very expensive by the standards of a learning algorithm!
- Need to store the entire dataset in memory!
- Tons of work has gone into algorithms and data structures for efficient nearest neighbors with high dimensions and large datasets.

Conclusion

- Simple algorithm that does all its work at test time — in a sense, no learning!
- Can control the complexity by varying k
- Suffers from the Curse of Dimensionality
- Next time: parametric models, which learn a compact summary of the data rather than referring back to it at test time.