

COMP 755

SN 011 T/H 5:00-6:15pm

<https://wwwx.cs.unc.edu/Courses/comp755-f18/>

Jorge Silva jsilva@cs.unc.edu

Welcome to COMP 755

Textbook:

Machine Learning – A Probabilistic perspective

Kevin P. Murphy

Other helpful resources:

The Elements of Statistical Learning

<http://statweb.stanford.edu/~tibs/ElemStatLearn/>

Pattern Recognition and Machine Learning

<https://www.microsoft.com/en-us/research/people/cmbishop/#prml-book>

Prerequisites and languages

COMP 401, 410, MATH 233, and STOR 435

Basic Programming and Data Structures

Discrete Math and Probability or Stats

Languages you can use: MATLAB or Python (NumPy and SciPy)

Grading and exams

- 4 homework assignments each 10% for a max of 35%
- Midterm (25%)
- Final exam (30%)
- Participation (10%)
- Optional extra credit project (10%)

Final exam: The course final is given in compliance with UNC final exam regulations and according to the UNC Final Exam calendar.

Late policy: 0 points for late submissions unless you contact me in advance.

Week	Topic	Reading	Deadlines
1	Introduction	Chapter 1	
2	Probability Distributions and Optimization	Chapter 2	Assignment 1
3	Linear Models for Regression	Chapter 7	
4	Linear Models for Classification	Chapter 8	Assignment 2
5	Directed Graphical Models	Chapter 10	
6	Mixture Models and EM	Chapter 11	
7-8	Latent Linear Models	Chapter 12	Midterm
9-10	Sparse Linear Models	Chapter 13	Assignment 3
11	Kernel Methods	Chapter 14	
12	Markov and Hidden Markov Models	Chapter 17	
13-14	Exact inference for graphical models	Chapter 20	Assignment 4
15	Neural networks and Deep Learning	Chapter 28	
16	Neural networks and Deep Learning	Chapter 28	

TA and office hours

Your TA is Ric Poirson (poirson@cs.unc.edu)

My office hours: after class until 7:15pm

Please send an e-mail to let me know you are coming

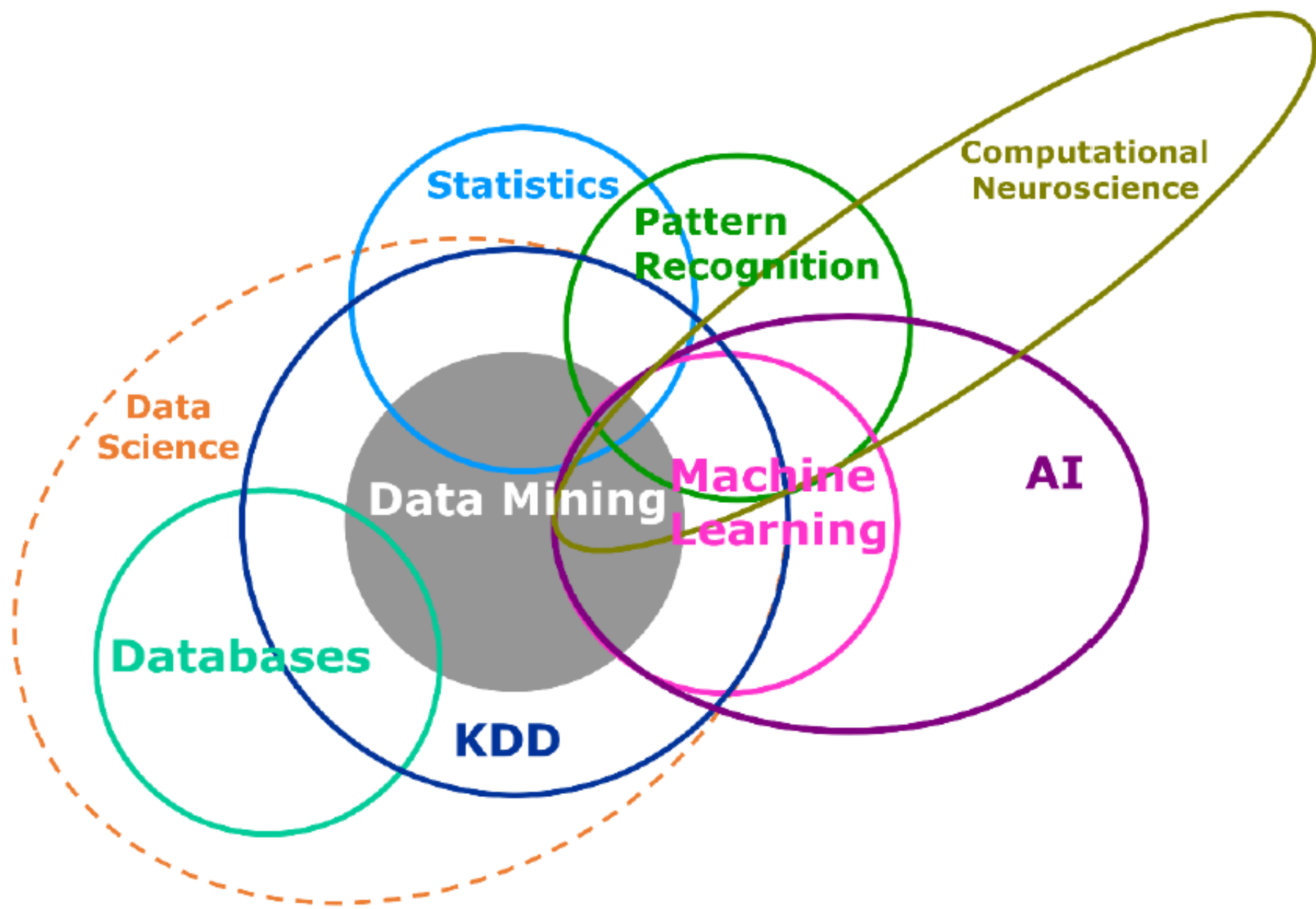
Goals

Develop an understanding of basic machine learning algorithms, their efficient implementations and their applicability to different tasks.

What is Machine Learning?

Machine learning is a branch of artificial intelligence that is concerned with building systems that require minimal human intervention in order to learn from data and make accurate predictions.

“All models are wrong, but some are useful” –George Box



Machine Learning

SUPERVISED LEARNING

- Regression
 - LASSO regression
 - Logistic regression
 - Ridge regression
- Decision tree
 - Gradient boosting
 - Random forests
- Neural networks
- SVM
- Naïve Bayes
- Neighbors
- Gaussian processes

UNSUPERVISED LEARNING

- A priori rules
- Clustering
 - *k*-means clustering
 - Mean shift clustering
 - Spectral clustering
- Kernel density estimation
- Nonnegative matrix factorization
- PCA
 - Kernel PCA
 - Sparse PCA
- Singular value decomposition
- SOM

SEMI-SUPERVISED LEARNING

- Prediction and classification*
- Clustering*
- EM
- TSVM
- Manifold regularization
- Autoencoders
 - Multilayer perceptron
 - Restricted Boltzmann machines

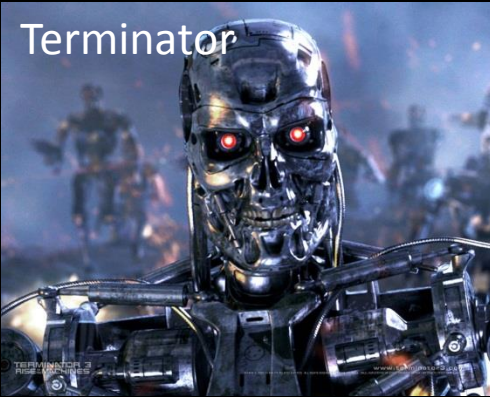
TRANSDUCTION

REINFORCEMENT LEARNING

DEVELOPMENTAL LEARNING

*In semi-supervised learning, supervised prediction and classification algorithms are often combined with clustering.

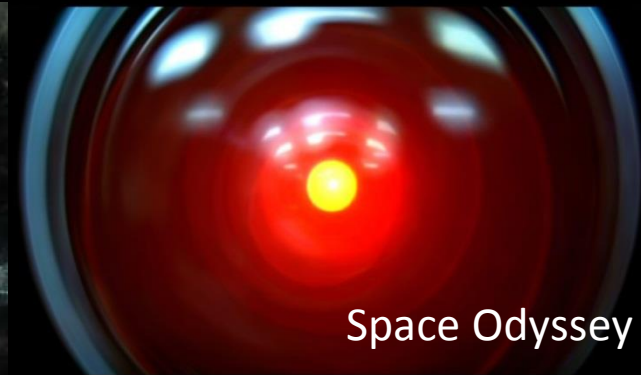
Artificial Intelligence in Art



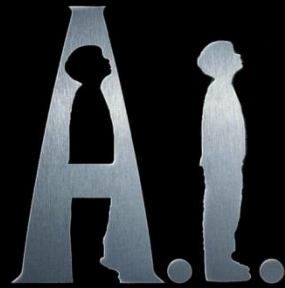
Terminator



Matrix



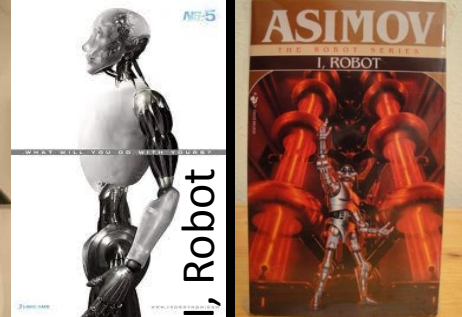
Space Odyssey



ARTIFICIAL INTELLIGENCE



Star Wars



I, Robot



Star Trek



Futurama



Wall-E



Metropolis (1927)



Mass Effect

Reality



Siri. Beta

Your wish is
its command.



Speech Recognition

Natural Language Processing

Ontology-based task delegation

Cognitive Assistant that Learns and Organizes
(CALO) DARPA funded project carried out at SRI



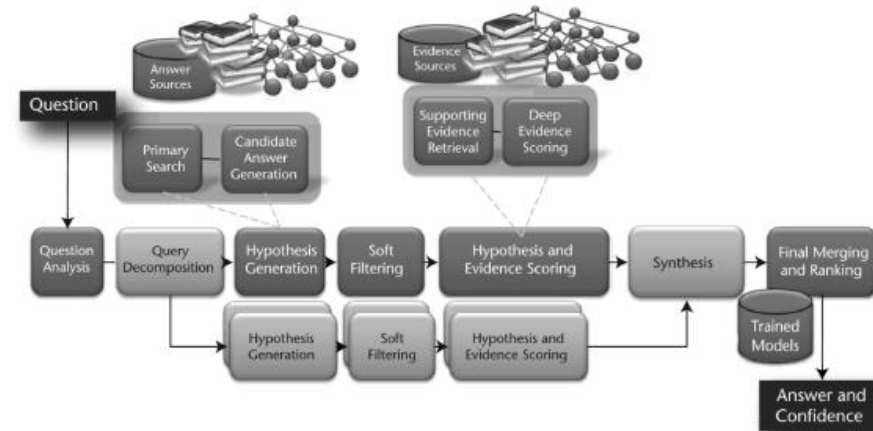
Reality



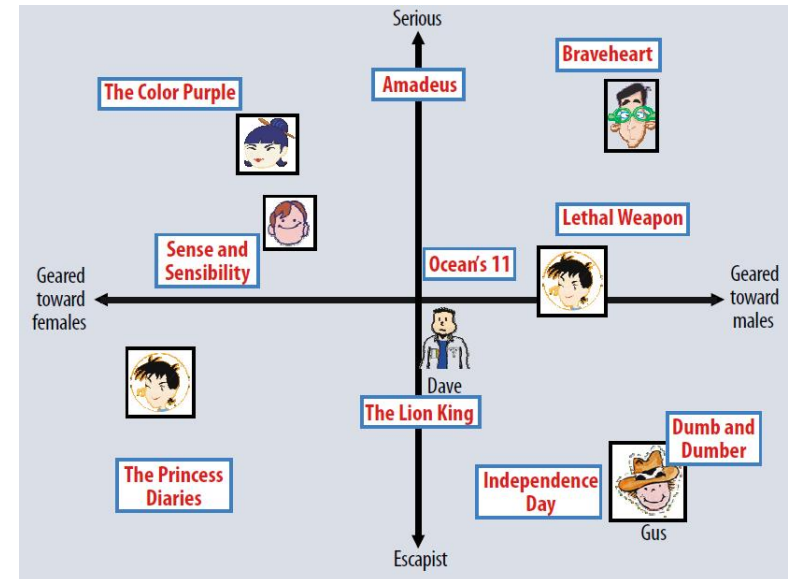
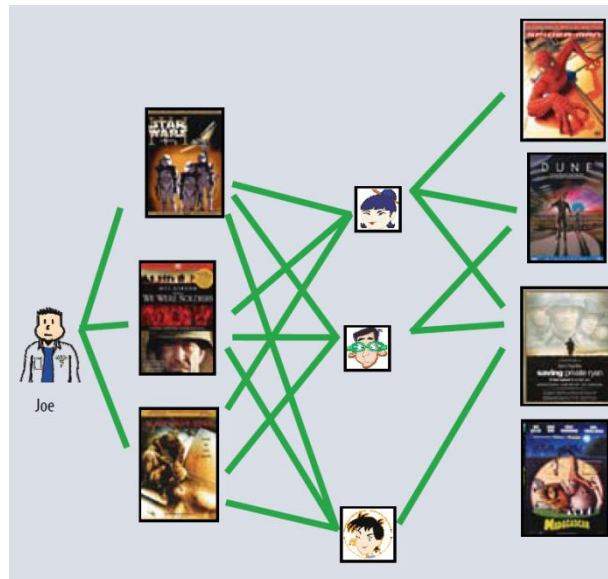
IBM Watson



Natural Language Processing
Question Answering
Decision theory



Reality



Reality

Google autonomous car

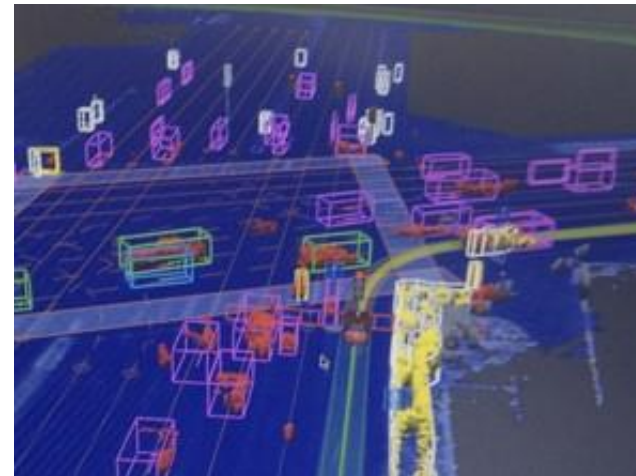


Toyota autonomous car



Computer Vision Path/motion planning

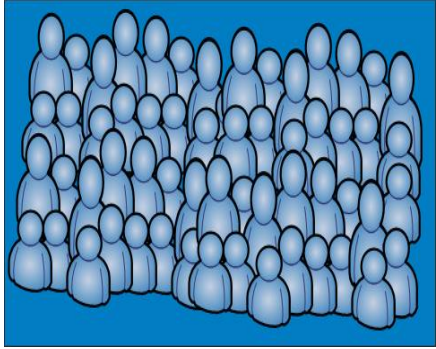
Visualization of an autonomous car's
world representation



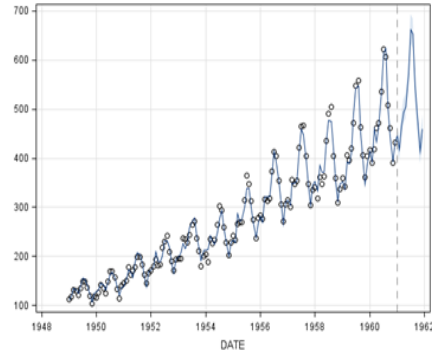
Data types

Type		Example
Numeric	Integer	-1, 0, 1, 2, 56, 100001
	"Float", "Real"	2.335, -6792.29388675465435687
Categorical	Binary	{0, 1}, {yes, no}, {alive, dead}
	Nominal	{cat, dog, mouse}, {red, blue, green, yellow}
	Ordinal	lowest, lower, middle, higher, highest

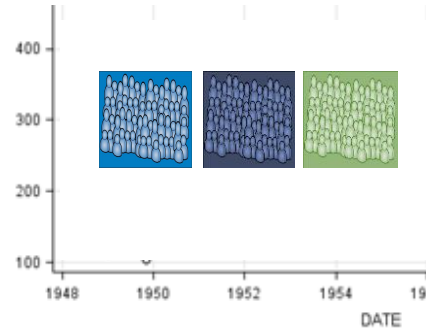
DATA Galore



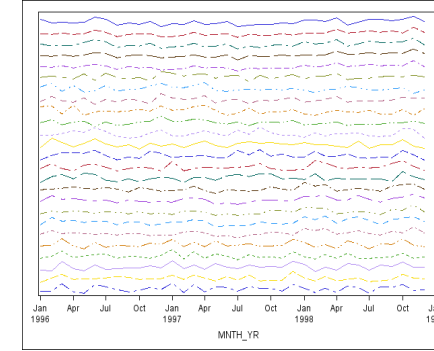
Cross-Sectional



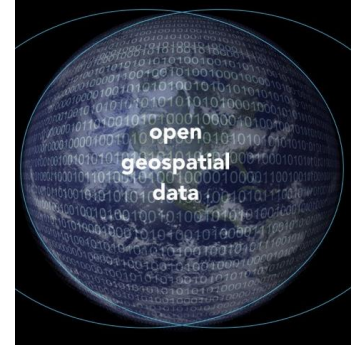
Time Series



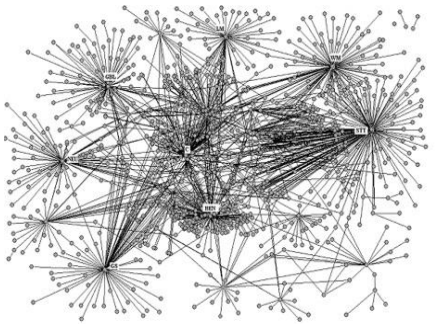
Panel



Streaming



Spatial



Network



Link



Text

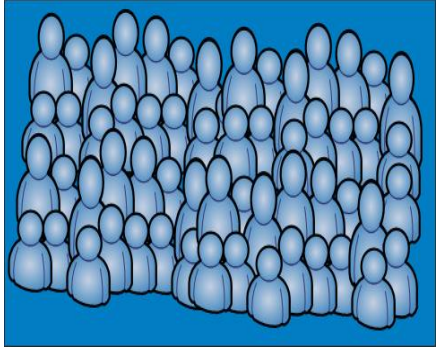


Sound

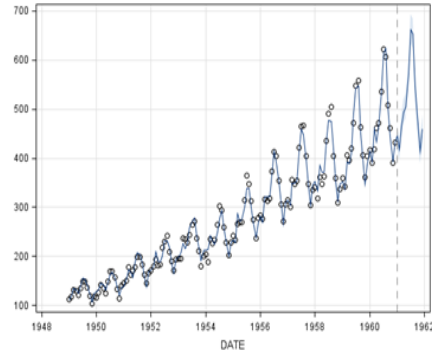


Image/Video

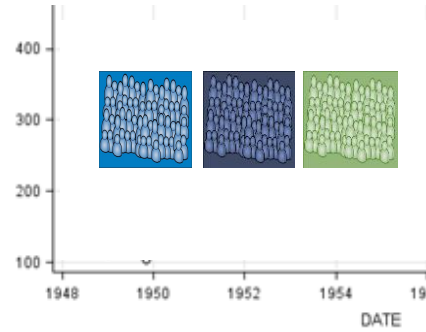
Business problems



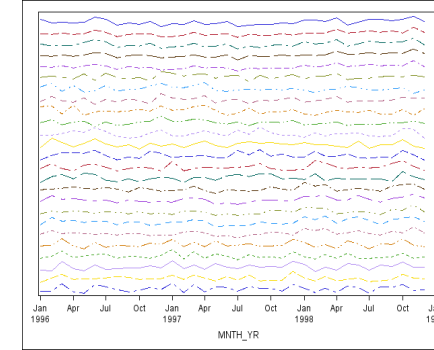
Who will respond to a campaign?



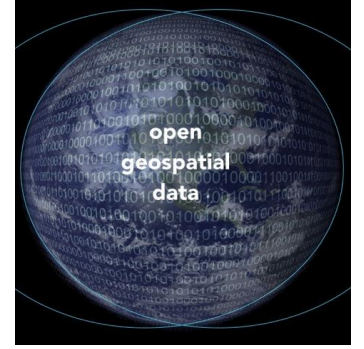
What will future demand look like?



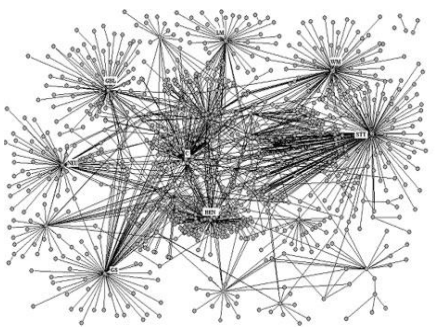
Will regulations have an impact?



Are there anomalies?



Where should we look for oil?



Who are influencers?

	2			4	5	2.94*
	5		4			1
			5		2	2.48*
		1		5		4
			4			2
	4	5		1		1.12*

Which item should I recommend?



How is our product perceived?



Emerging problem in our product?



Can we prevent insurance claims?

Classical programming ...

TASKs like sorting an array

1,4,6,5,2,3  1,2,3,4,5,6

are easy to formalize, relatively easy to implement, and straightforward to test.

Array `arr` is sorted if and only if

$$arr[i] \geq arr[i + 1] \ \forall i \in \{0, \dots, \text{length}(arr) - 2\}.$$

Write a program that takes an array as input and returns sorted array containing the same elements.

Traditional approach is not always feasible

Write a program that detects whether there is a dog in an image ...

```
bool hasDog(Image im) {  
    return random() > 0.5;  
}
```

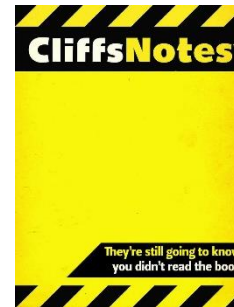
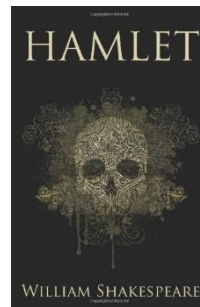
Some **TASKs** are difficult to formalize.

Why do we need machine learning?

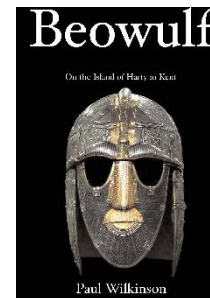
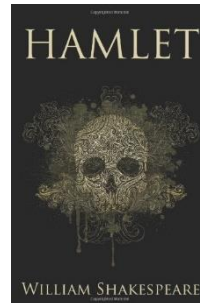
Object recognition



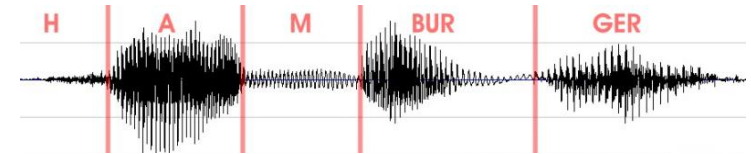
Product recommendation



OR



Speech-to-Text



Election result prediction



Traditional Programming Approach



Code

Implementation

[illegible]

Machine Learning Approach



Machine Learning Method



Data

Code

[illegible]


Implementation

Machine learning algorithms use examples, **DATA**, to learn how to accomplish a **TASK**.


Machine Learning – a toy example

Learning to detect dogs in images.

Data: {( , 1), ( , 0), ( , 0) ...}

Each example is composed of an image and a label ( , 1)

Typically we denote such pairs as (\mathbf{x}, y) where \mathbf{x} is a **feature** vector and y a **label**.

Feature vector  -> [0.25, 0.3,] vector of color intensities

Label **1**

The **TASK**: predict label y from feature vector \mathbf{x}

Machine Learning – a toy example

Data: {(, 1), (, 0), (, 0) ...}

The TASK: predict label y from feature vector x

A machine learning algorithm **produces** a **function** $f(\cdot)$ such that $f(x)$ **frequently matches** the true label y .

Machine Learning – a toy example

Machine learning algorithm produces a function $f(\mathbf{x})$ such that $f(\mathbf{x})$ frequently matches the true label y .

$$\mathbf{x}^1 = \img alt="A small, fluffy, light-colored dog sitting on grass." data-bbox="658 211 708 289"/> \quad y^1=1 \quad = f(\mathbf{x}^1) = 1$$

$$\mathbf{x}^2 = \img alt="A grey and white cat sitting inside a cardboard box." data-bbox="658 319 736 409"/> \quad y^2=0 \quad = f(\mathbf{x}^2) = 0$$

$$\mathbf{x}^3 = \img alt="A ginger and white cat sitting on a stone path." data-bbox="658 422 725 502"/> \quad y^3=0 \quad \neq f(\mathbf{x}^3) = 1$$

Q: How do we measure performance of function $f(\mathbf{x})$?

Q: Let's say that 75% of the images in the dataset are dogs and 25% are cats. If $f(x) = 1$ (always predict that dog is in the image) how well are we doing?

Machine learning – a toy example

One way to measure performance of your prediction function $f(x)$ is to count fraction of mistakes.

$$\text{Error} = \frac{\sum_{t=1}^T [f(x^t) \neq y^t]}{T}$$

where

$$[v] = \begin{cases} 1, & \text{if } v = \text{true} \\ 0, & \text{otherwise.} \end{cases}$$

$$x^1 = \img alt="A golden retriever puppy running on grass." data-bbox="675 435 725 515"/> \quad y^1=1 = f(x^1) = 1$$

$$x^2 = \img alt="A grey cat sitting inside a cardboard box." data-bbox="675 544 755 635"/> \quad y^2=0 = f(x^2) = 0$$

$$x^3 = \img alt="A ginger cat sitting on a stone path." data-bbox="675 645 745 729"/> \quad y^3=0 \neq f(x^3) = 1$$

$$\text{Error} = 1/3$$

Machine Learning – a toy example

Machine learning algorithm **produces** a function $f(\mathbf{x})$ such that $f(\mathbf{x})$ frequently matches the true label y .

We want to find a function that minimizes the error.

Direct optimization of the count of mistakes is very hard – there is no feedback on how to adjust the function to improve prediction.

Machine Learning – probabilistic view

Instead of making hard predictions our prediction functions can compute probabilities

$$P(y = 1|x, \theta)$$

Probability that there is a dog in an image given pixels

$$P(y = 1 | \img alt="A golden retriever dog standing on grass." data-bbox="285 505 330 585"), \theta) = 0.75, P(y = 1 | \img alt="A ginger cat sitting on a wooden surface." data-bbox="598 505 660 585"), \theta) = 0.25, \dots$$

Since our predictions are no longer discrete we can get back feedback on how to adjust them.

Probabilistic view enables use of optimization, and it allows us to quantify uncertainty.

Recap: Tasks, Data, Optimization

TASKs are defined implicitly through DATA. For example,

Data = {(, 1), (, 0), (, 0) ...}

Machine Learning Algorithms minimize ERROR to find a mapping $f(\mathbf{x})$ that performs well on the TASK.

Optimization and probabilistic view of the learning allows us to accomplish this.

Unsupervised Machine Learning

TASKs thus far have had a defined input and output ( , 0)

This is called SUPERVISED Machine Learning.

UNSUPERVISED Learning does not have a pre-specified output.

Unsupervised Machine Learning

UNSUPERVISED learning seeks to find compact data representations from which original data can be reconstructed.

100% fidelity
Image is 725kB



90%
250kB



10%
37kB



1%
20kB



Unsupervised Machine Learning

Compression is an example of **UNSUPERVISED** machine learning.

ZIP, RAR, BZIP2 find a **REPRESENTATION** for the data.

More formally find function such that:

$h = f(x)$ such that $f^{-1}(h) \approx x$ (Lossy reconstruction)

$h = f(x)$ such that $f^{-1}(h) = x$ (Lossless reconstruction)

Here, h **REPRESENTS** the **DATA** x .

Error for Unsupervised Learning

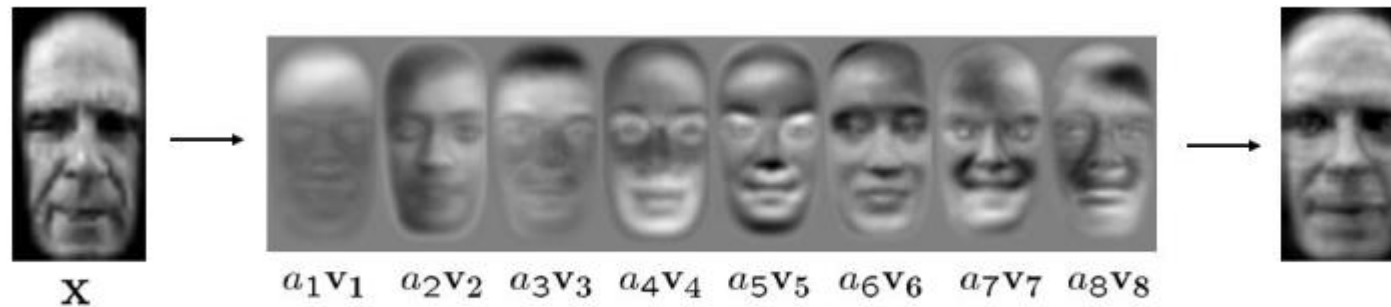
In unsupervised learning reconstruction error is optimized:

$$\text{Error} = \frac{1}{T} \sum_{\{t=1\}}^T \text{Distance}(f^{-1}(f(x^t)), x^t)$$

Typically the Distance is a smooth function enabling optimization.

Examples of Unsupervised Learning

Dimensionality reduction

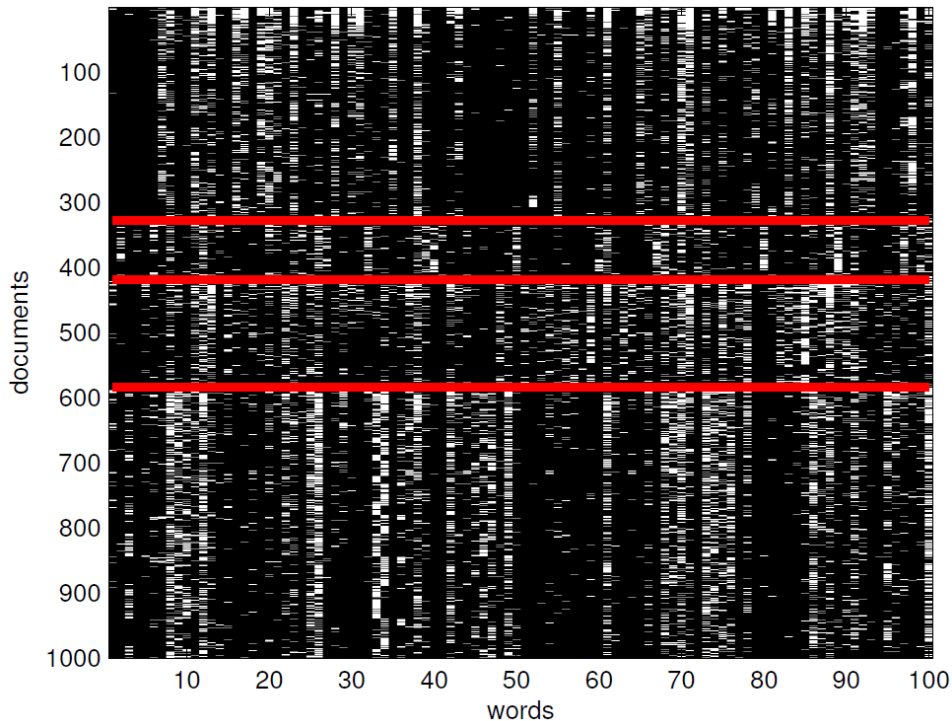


A face on the left is represented by eight real values corresponding to weight of each of the eight components.

Reconstructed face is shown on the right.

Examples of Unsupervised Learning

Clustering documents



Row is a text document; Column is a word.

White means that the document contains the word.

Red lines separate clusters of documents with similar vocabulary.

You can represent each document with the indicator of its cluster (1,2,3 or 4).

Pitfalls in machine learning

DATA is used to learn mappings $f(\cdot)$ by minimizing errors:

$$\text{Error} = \frac{\sum_{\{t=1\}}^T [f(x^t) \neq y^t]}{T}$$
$$\text{Error} = \frac{1}{T} \sum_{\{t=1\}}^T \text{Distance}(f^{-1}(f(x^t)), x^t)$$

Q: Given two machine learning methods how do we tell which one is better?

Training Data

Imagine the following scenario:

- 1) Teacher works out a quadratic equation $x^2 - 1 = 0$ in class
- 2) Teacher gives final exam with one of the problems being

$$x^2 - 1 = 0$$

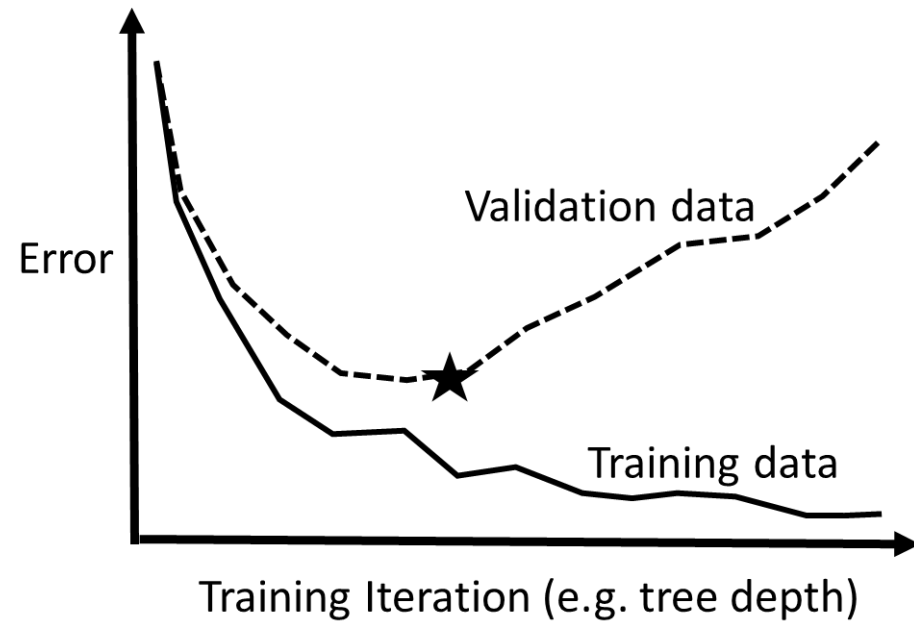
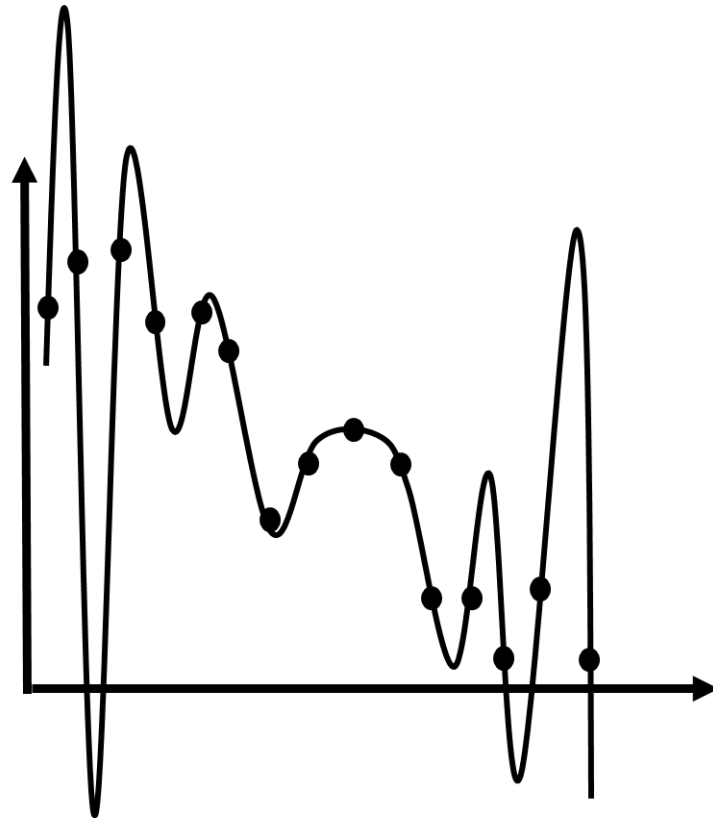
Q: Is this a good idea? Is this a bad idea? What should the teacher do?

Q: Would training a machine learning algorithm run into the similar problem? How should we fix it?

Q: Given two machine learning methods how do we tell which one is better?

Best practice for machine learning

- Do not overfit!



Pitfalls

From the reading assignments:

- (1) Statistics in the big data era: Failures of the machine:
<http://dx.doi.org/10.1016/j.spl.2018.02.028>
- (2) Adversarial Examples that Fool both Computer Vision and Time-Limited Humans: <https://arxiv.org/pdf/1802.08195.pdf>

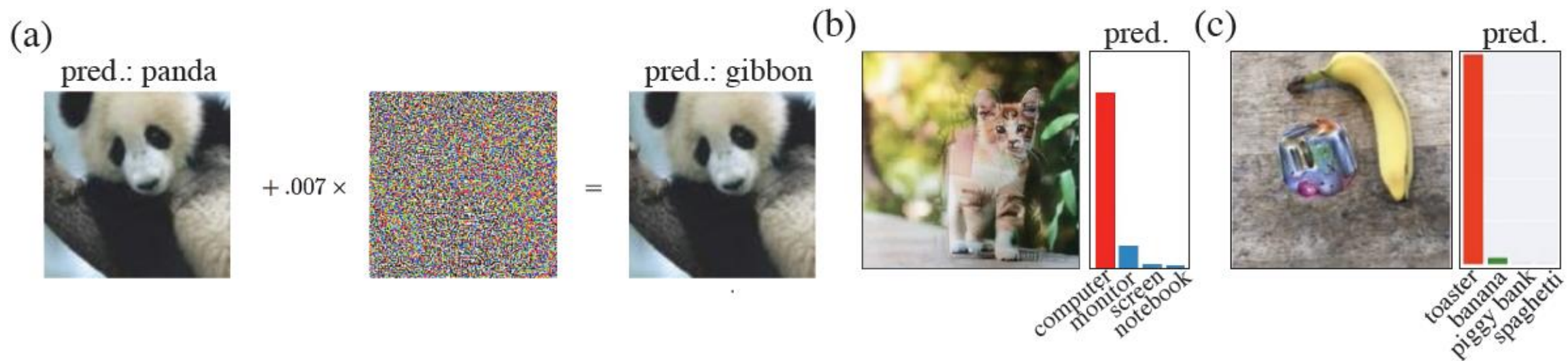
Algorithmic bias: from (1)

2.2. Fair decisions and predictive algorithms

There are numerous decisions that are made by various authorities based on their own judgment and experience; these decisions can have an enormous impact on society as a whole and on individuals. Some examples include whether and how to regulate car emissions, patrol locations and decisions of who to stop and search in policing, sentencing and bail decisions in the criminal justice system, hiring decisions, salary levels, selection of grants to fund, and tenure decisions in academics.

Of course whenever there is an individual or a small group of individuals in charge of making such decisions, there is substantial room for the decisions to not be entirely “fair” and objectively based on the data at hand but instead driven in part by implicit or explicit biases. Such biases may lead to under-regulation of pollution, policing that targets certain minority communities, more severe sentencing for individuals within those communities, and hiring/salary/tenure decisions driven in part by demographic factors [...]

Adversarial attacks: from (2)



Today

Recap:

1. In machine learning, data implicitly encodes tasks
2. Supervised learning discovers a mapping from features to labels
3. Supervised learning error is measured by comparing mapped output to the true output
4. Unsupervised learning finds compact representations of the data
5. Training and test data should be kept separate to enable accurate assessment of the method's performance