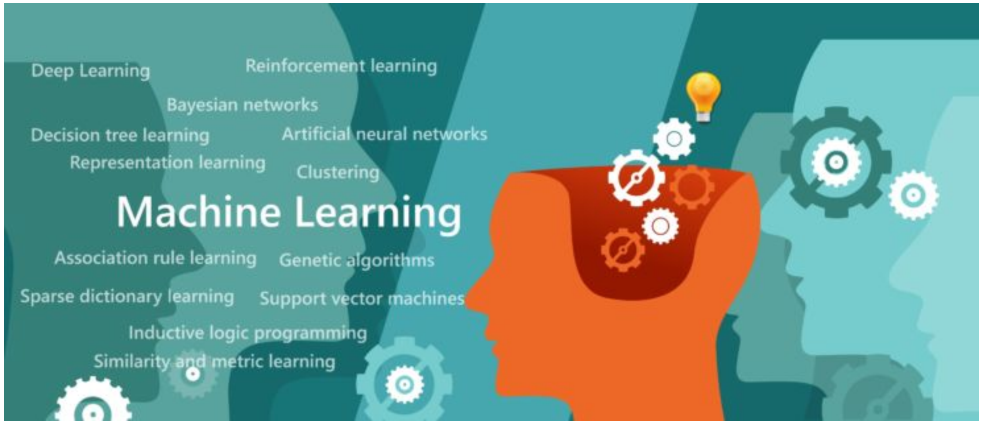
CSCE 633 - Machine Learning

Lecture 1 - Welcome!

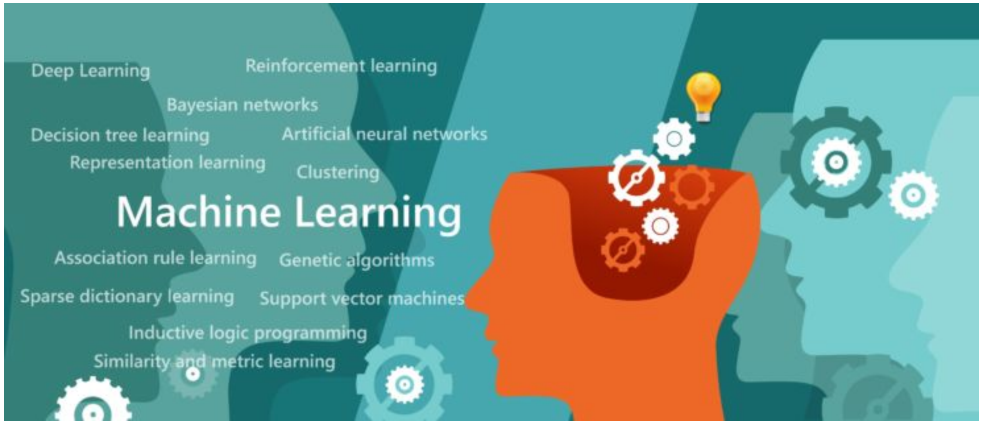
Welcome to CSCE 633!

- About this class
- Introduction to Machine Learning
  - What is Machine Learning?
  - Basic concepts



Welcome to CSCE 633!

- About this class
- Introduction to Machine Learning
  - What is Machine Learning?
  - Basic concepts



Welcome to CSCE 633!

- Instructor
  - Theodora Chaspari
  - chaspari@tamu.edu (but use Piazza for quickest reply)
  - Online office Hours: Thursday, 2.30-3.30pm (through the following link)
  - Personal Zoom Link: https://tamu.zoom.us/j/3024683671

- TA
  - Peiman Mohseni, peiman.mohseni@tamu.edu
  - Online office Hours: Thursday, 11.15am-12.15pm (https://tamu.zoom.us/j/95213141515)
  - In-person office Hours: Tuesday 11.15am-12.15pm (EAB-C; shared TA office cubicle on the left entering EAB-C)

- Grader
  - Hong-Jie Chen, raghudv@tamu.edu
  - Online office Hours: Wednesday, 1-2pm (https://tamu.zoom.us/j/5112240111)

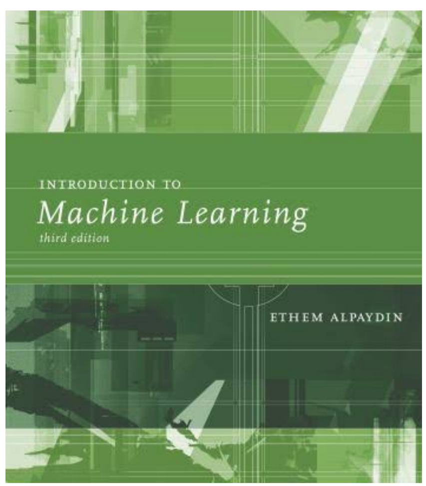
Class websites

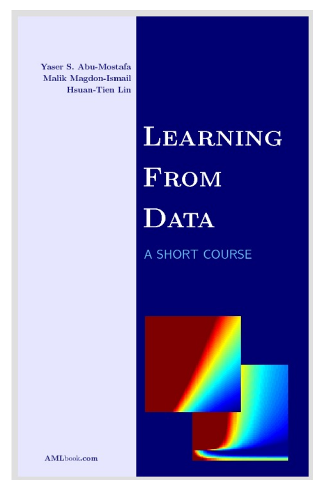
- **CANVAS**
  - https://canvas.tamu.edu/courses/98135
  - Class logistics
  - Class discussions
  - Slides
  - Homework posting, solutions, submissions
  - For sending private messages to me or the TA and the grader

- **Google drive (for supplementary material)**
  - https://drive.google.com/drive/folders/11_ZWCr_DNqDigzYnSOYT71-f6pRwEF_q?usp=sharing
  - Updated weekly roadmap, code, datasets (when needed)
  - Accessed via your **TAMU email account**

Textbook and course material

- Lecture notes (on CANVAS)
- Textbook
  - Introduction to Machine Learning, Ethem Alpaydin
  - Learning from Data, Yaser S. Abu-Mostafa
- Supplemental materials (on CANVAS)





Class structure

- 5 homework sets (4x10% + 15% = 55%)
  - 1% penalty on late submission per assignment
  - Late homework submissions can be submitted 1 week after the deadline
- 3 exams (3x15%=45%)
  - Exam 1: October 5th (during class time)
  - Exam 2: November 9th (during class time)
  - Exam 3: November 30th (during class time)

Homework Submission

- All homeworks will be submitted as a **single pdf** on CANVAS
  - The executable code (if required) needs to be included at the end of the pdf
- Programming assignments
  - Recommended languages are Python or Matlab (or Octave)
- Math assignments
  - Please submit solution produced in Latex
  - Or **very clear** handwritten solution
  - This will help our TA and grader a lot.

Active Learning

- Would you ever take a cardio class without actually participating in it?
- So why take a CS course without practicing the material in class?

Active Learning

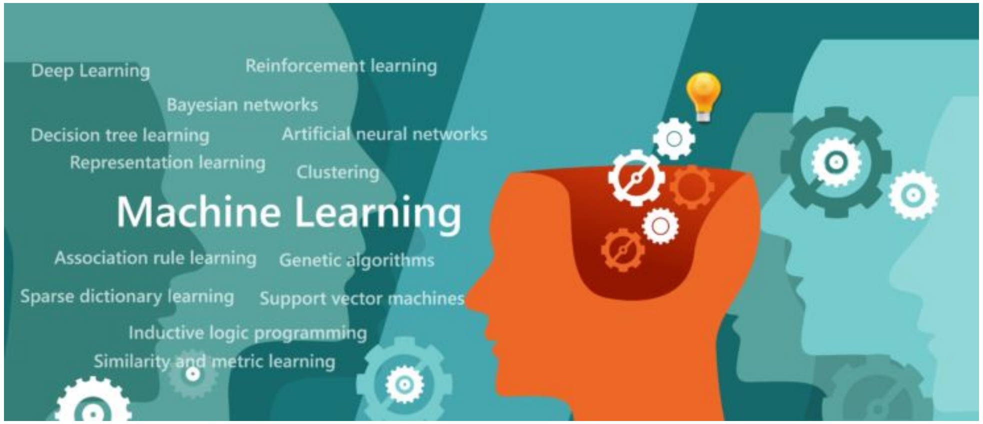
- "Anything that involves students in doing things and thinking about the things they are doing" (Bonwell & Eison, 1991)
- "Anything course-related that all students in a class session are called upon to do other than simply watching, listening and taking notes" (Felder & Brent, 2009)
- Audience attention starts to wane after 10-20 mins
- Research suggests that incorporating active learning techniques
  - encourages student engagement
  - reinforces important material, concepts, etc.
  - builds self-esteem
  - creates a sense of community

Active Learning

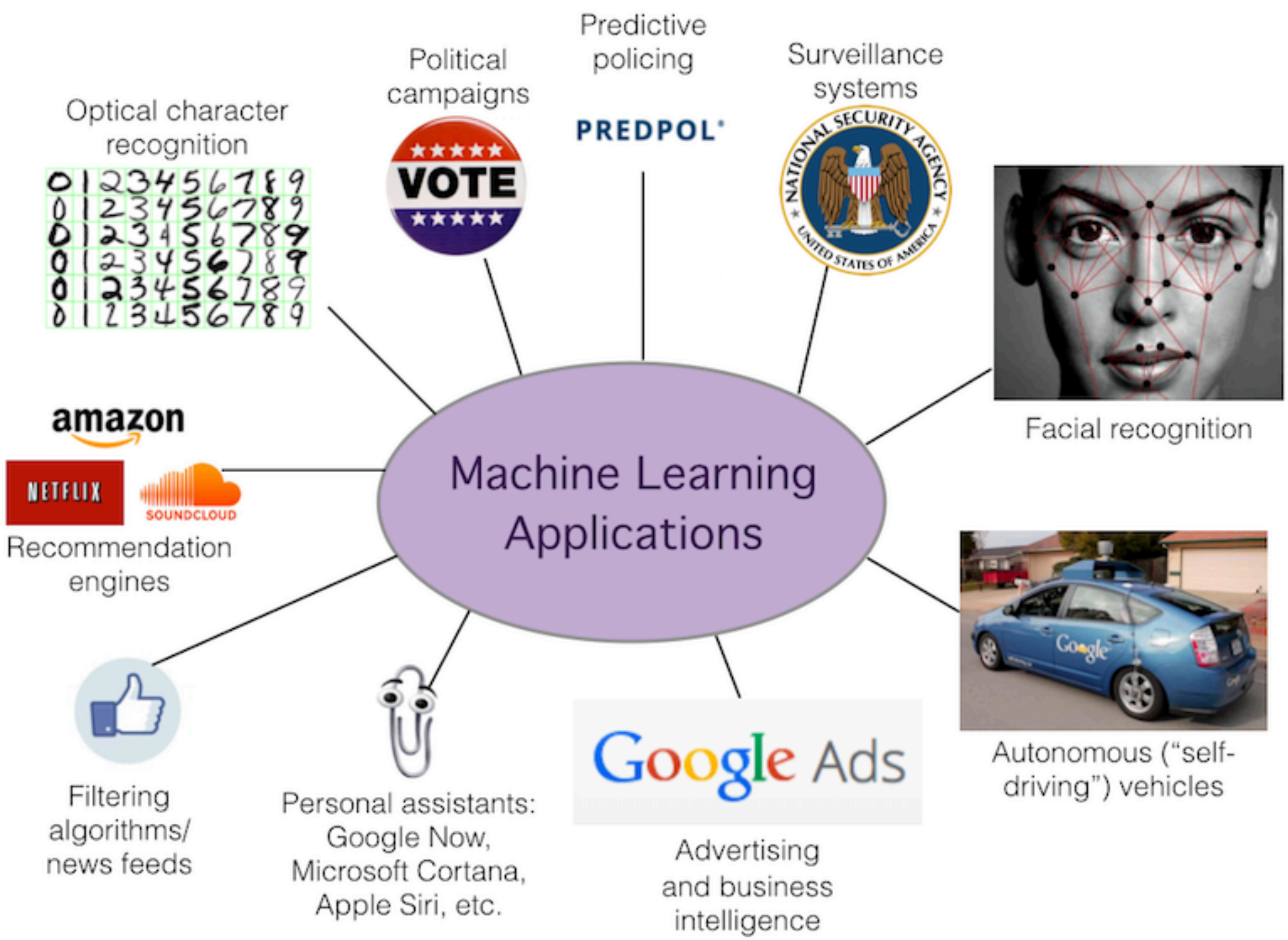
- Kahoot: free software platform that will help us answering multiple choice questions during class
- https://kahoot.it/#/
- Google Play, Apple Store
- You won’t be graded in any of these
- Just a fun way to engage and participate more in class 🙂

Outline

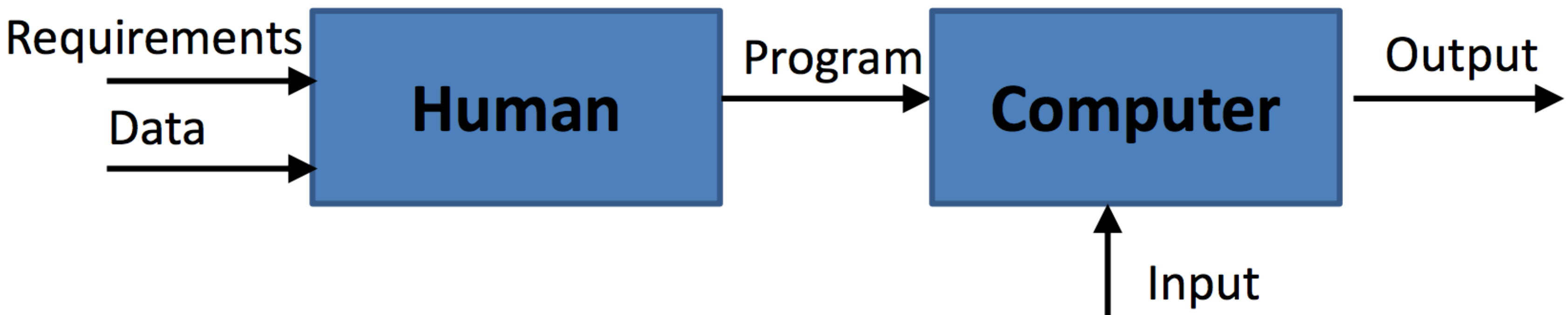
- About this class
- Introduction to Machine Learning
    - What is Machine Learning?
    - Basic challenges



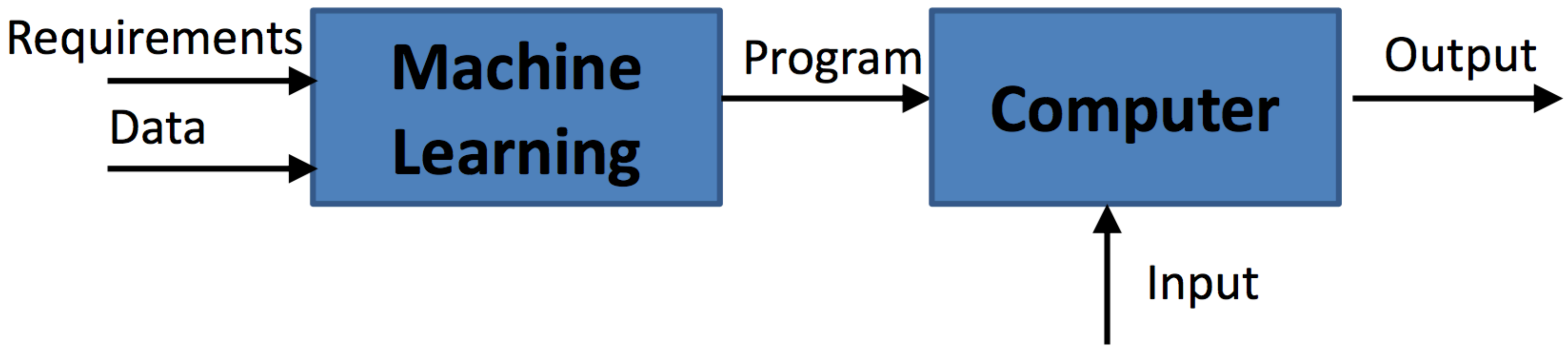
Machine learning is everywhere



What is machine learning?



Requirements and data change often



Big Data: 40 billion webpages, 100 hrs YouTube video uploaded every 1min, 1 million Walmart transactions per hour

What is machine learning?

A possible definition[1]

A set of methods that can automatically detect patterns in data, and then use those to predict future data or perform other kinds of decision making under uncertainty.

A more formal definition[2]

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T as measured by P improves with experience E

[1] From K.P. Murphy
[2] From T. Mitchell

What is machine learning?

https://kahoot.it/#/

Definition: A computer program learns from experience E with respect to some task T and some performance measure P, if its performance on T as measured by P improves with experience E

Question: Suppose your Facebook account watches the users added to your friends' list. Based on that, it learns how to suggest new friends for you. What is task T in this setting?

A. Classifying a user X according to whether or not you would possibly send them a friend request
B. Identifying the characteristics of users to which you send a friend request
C. Computing the percentage of suggested users to whom you actually sent a friend request
D. All of the above

What is machine learning?

Definition: A computer program learns from experience E with respect to some task T and some performance measure P, if its performance on T as measured by P improves with experience E

Question: Suppose your Facebook account watches the users added to your friends' list. Based on that, it learns how to suggest new friends for you. What is task T in this setting?

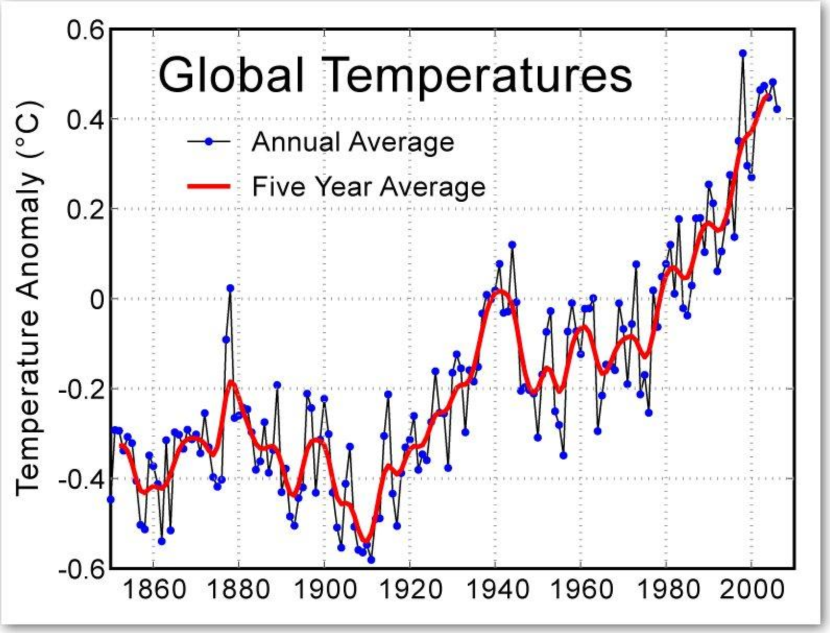
A. Classifying a user X according to whether or not you would possibly send them a friend request (task T)
B. Identifying the characteristics of users to which you send a friend request (experience E)
C. Computing the percentage of suggested users to whom you actually sent a friend request (performance measure P)
D. All of the above

Key ingredients for a machine learning task

- Data
  - collected from past observations (training data)
- Model
  - devised to capture patterns in data
  - doesn't have to be absolutely true, as long as it is close enough
  - should tolerate randomness & mistakes, i.e. uncertainty
- Prediction
  - apply the model to
    - forecast what is going to happen in the future
    - automatically make a decision for unknown data, etc.

Example: Detecting Patterns

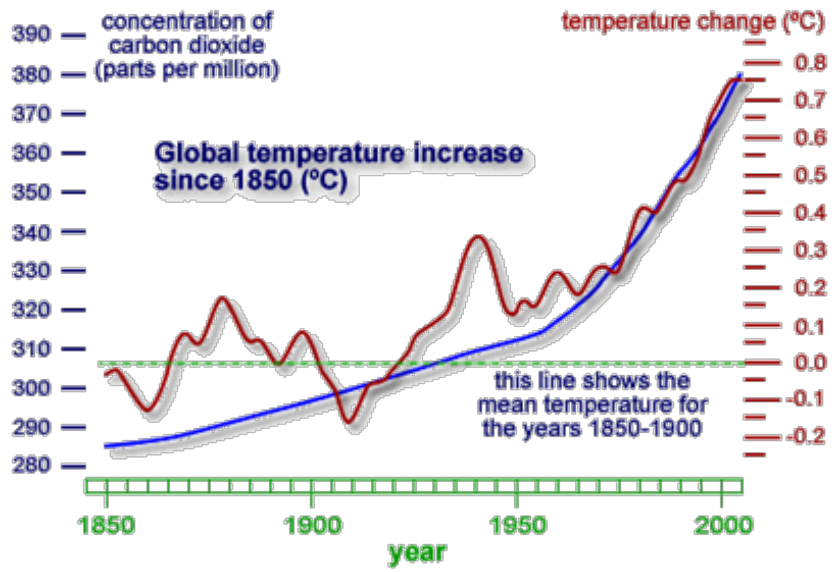
How has the temperature been changing over the last 140 years?



- Generally increasing patterns
- Local oscillations

Example: Describing Patterns

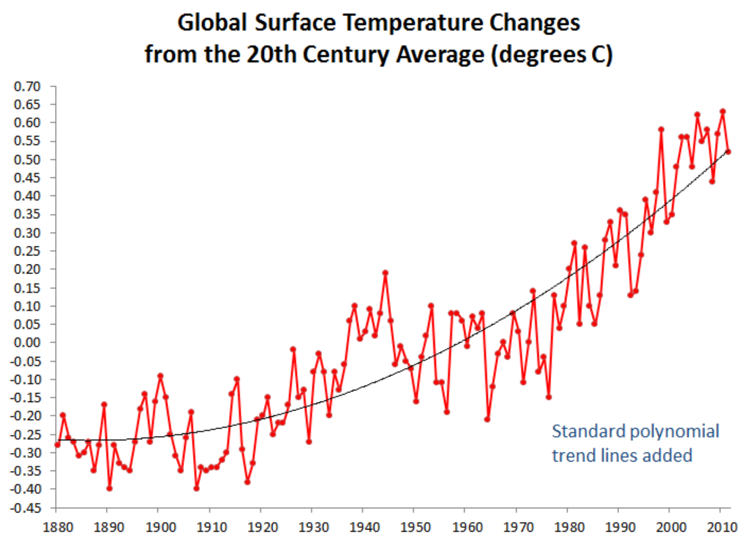
Build a model: fit the data with polynomial function



- Quadratic model is not accurate for every year
- But captures the general trend

Example: Predicting Future Value

What is the temperature of 2010?



- Again the model is not accurate for that specific year
- But it is close enough

The three components of learning

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

a learner must be represented in some formal language

an evaluation function assesses the performance of a learner

find the highest-scoring learner

Source: P. Domingos, 2014

Types of Learning

- Supervised (or predictive) learning
  - learn mapping of inputs to outputs given a set of labelled pairs
  - training data includes desired outputs
  - obvious error metrics, e.g. prediction accuracy
  - cancer prediction, stock prices, house prices, spam detection
- Unsupervised (or descriptive) learning
  - find hidden/interesting structure in data ("knowledge discovery")
  - training data does not include desired outputs
  - less well-defined problem with no obvious error metrics
  - topic modeling, market segmentation, clustering of hand-written digits, news clustering (e.g. Google news)
- Reinforcement learning
  - the learner interacts with the world via actions
  - finds the optimal policy of behavior based on "rewards" it receives
  - robot navigation, game playing, self-driving cars

Supervised Learning

- Learning a mapping from inputs $\mathbf{x}_i$ to outputs y_i given a labelled set of input-output pairs (N samples)

$$\mathcal{D} = \{(\mathbf{x_i}, y_i)\}_{i=1}^{N}$$

- Data Matrix (N samples, D features)

$$\mathbf{X} = [\mathbf{x}_\mathbf{1}^T \ldots \mathbf{x}_\mathbf{N}^T] \in \Re^{D \times N} \qquad \mathbf{x_i} \in \Re^{1 \times D}$$

- Function approximation, function f is unknown and we approximate it

$$y = f(\mathbf{x})$$

- Classification
  - y_i is categorical or nominal (C classes): $y_i \in \{1, \ldots, C\}$
- Regression
  - y_i is a real-valued scalar: $y_i \in \mathbb{R}$

Supervised Learning: Classification

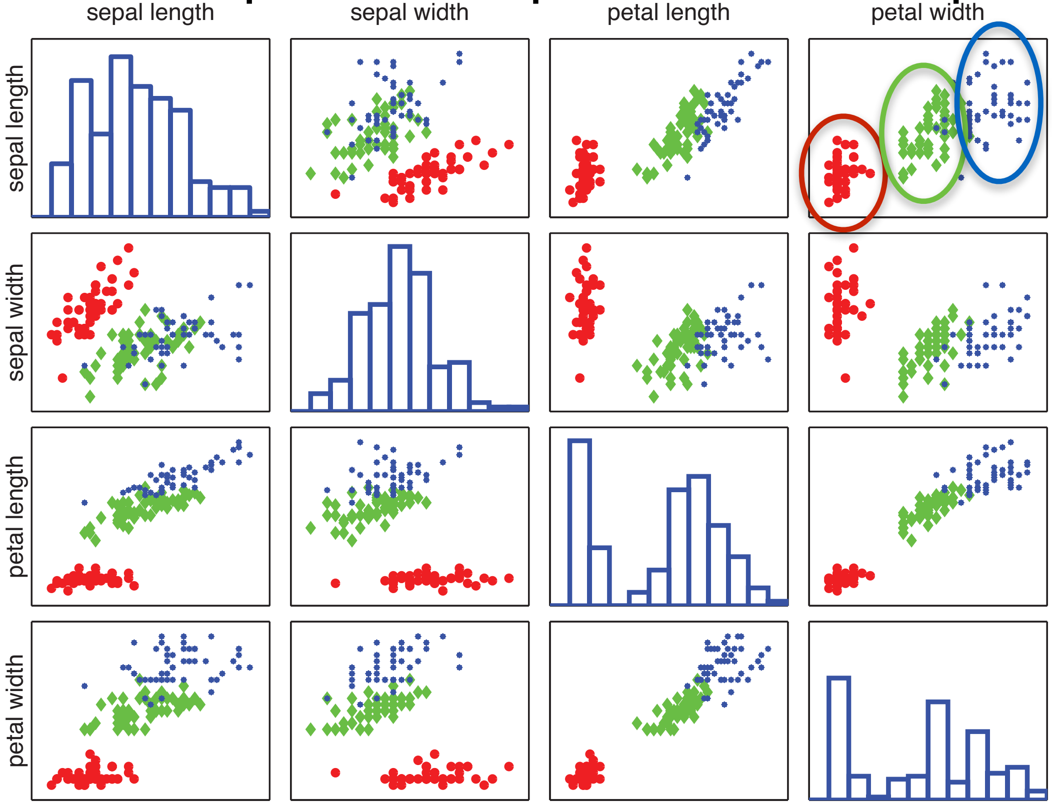
Recognizing types of Iris flowers (by R. Fisher)

setosa “●”

versicolor “◆”

virginica “✳”

Scatter plots of all possible feature pairs

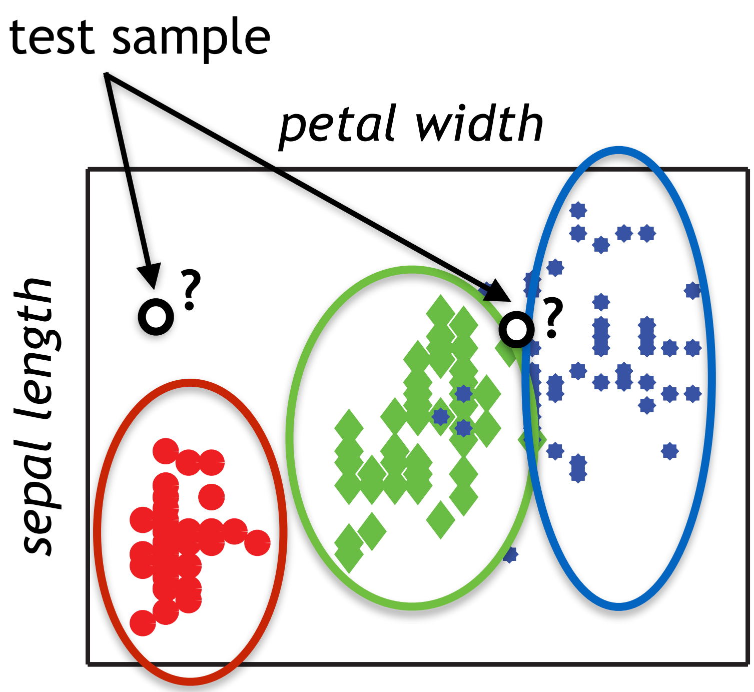


Exploratory data analysis (intuition)

Supervised Learning: Classification

Recognizing types of Iris flowers (by R. Fisher)

setosa "•", versicolor "◆", virginica "✳"



K-Nearest Neighbor (K-NN) classifier

- Test sample **x** is assigned to the most common class among its neighbors [N]

$$y = f(\mathbf{x}) = arg \max_{c=1,\dots,C} v_c$$

most common class

number of votes from class c

Brief probability review

Probability

- P(A): probability that event A is true
  - A: “it will rain tomorrow”
  - p(A)=0.2: “there is 20% chance of rain tomorrow”

Conditional probability

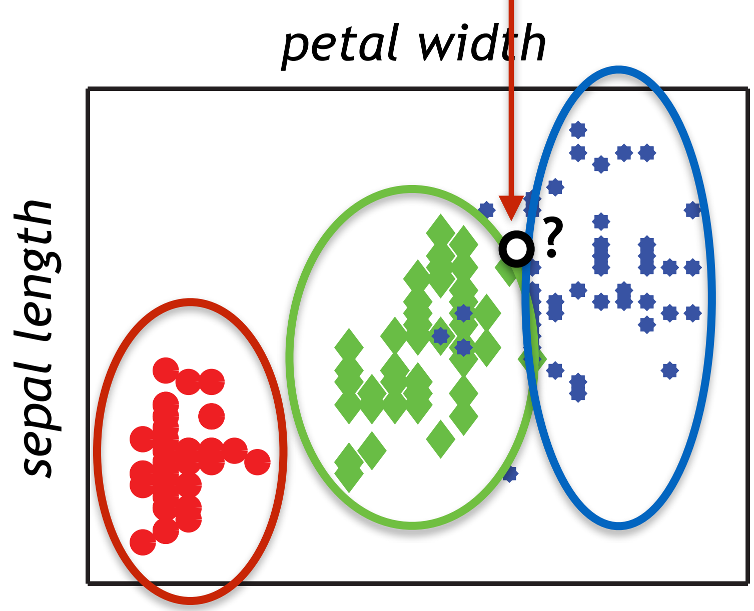
- P(A|B): probability of event A, given that event B is true
  - A: “it will rain tomorrow”
  - B: “today is humid”, C: “today is windy”
  - p(A|B): “chance of rain tomorrow, given that today is humid”, e.g. p(A|B)=0.6
  - p(A|B∧C): “chance of rain tomorrow, given that today is humid and windy”, e.g. p(A|B∧C)=0.7

Supervised Learning: Classification

Recognizing types of Iris flowers (by R. Fisher)

setosa "•", versicolor "◆", virginica "✳"

ambiguous test sample



The need of probabilistic predictions

- The right class of testing samples is unclear
- Return probabilities to handle ambiguity

$$y = f(\mathbf{x}) = arg \max_{c=1,\ldots,C} p(y = c|\mathbf{x}, \mathcal{D})$$

most **likely** class

posterior probability: probability of test sample belonging to class c given input vector **x and** training set D

MAP estimate (maximum a posteriori)

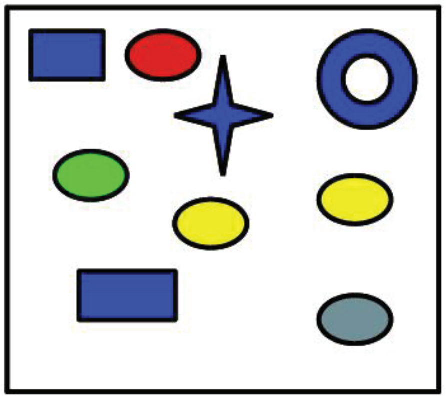
Biomedical applications (e.g. tumor classification)
DeepQA for IBM Watson, etc.

Why is it important to model uncertainty?

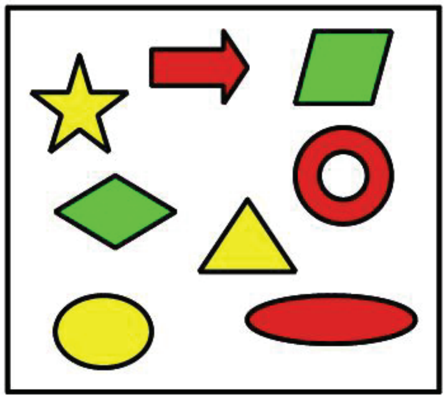
Question: Given the training data bellow, what would be a reasonable probability that a classifier would assign to the following test samples?

Training Set D

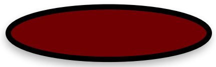
Class A



Class B



Test Set

Test sample s1: 

Test sample s2: 

A. $P(s1 \in A|D)=0.9$, $P(s2 \in A|D)=1$

B. $P(s1 \in B|D)=0.9$, $P(s2 \in B|D)=0.1$

C. $P(s1 \in B|D)=0.9$, $P(s2 \in A|D)=0.5$

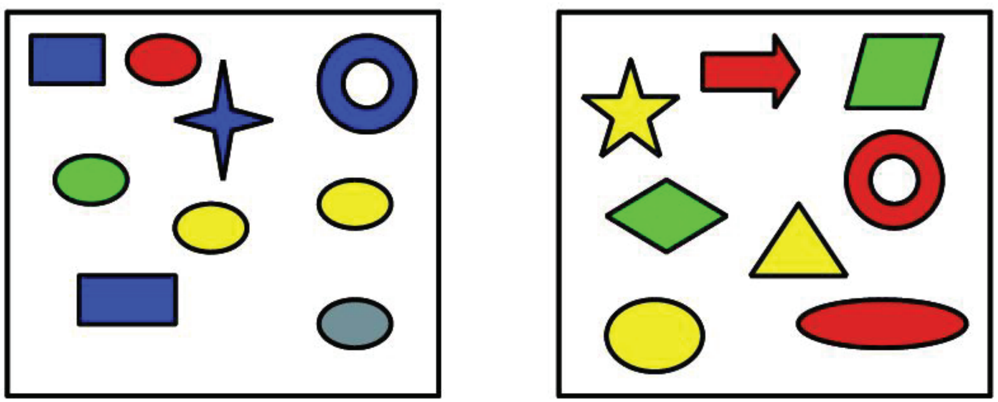
D. None of the above

Why is it important to model uncertainty?

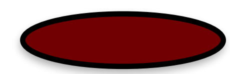
Question: Given the training data bellow, what would be a reasonable probability that a classifier would assign to the following test samples?

Training Set D

Class A **Class B**



Test Set

Test sample s1: 

Test sample s2: 

A. $P(s1 \in A|D)=0.9$, $P(s2 \in A|D)=1$

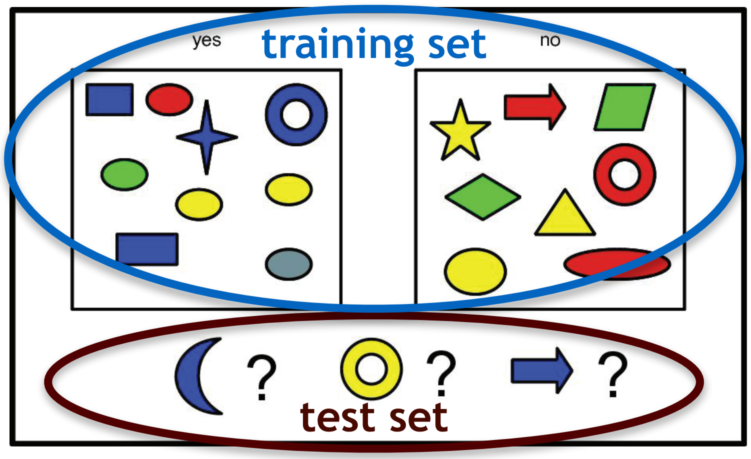
B. $P(s1 \in B|D)=0.9$, $P(s2 \in B|D)=0.1$

C. $P(s1 \in B|D)=0.9$, $P(s2 \in A|D)=0.5$

D. None of the above

Correct is C

Why is it important to model uncertainty?



D features (attributes)

N cases

Color	Shape	Size (cm)
Blue	Square	10
Red	Ellipse	2.4
Red	Ellipse	20.7

Label
1
1
0

discrete features

continuous features

training labels

- Required to generalize beyond the training set
- The right class of the testing samples is unclear
- To handle such ambiguous cases we can return a probability instead of a hard 0/1 decision

Supervised Learning: Regression

Predict the price of a used car

- Input **x**: car attributes (e.g. brand, year, mileage, etc.)
- Output y: price of car
- Model parameters **w**

- Deterministic linear model

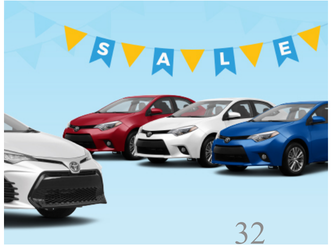
$$y = f(\mathbf{x}|\mathbf{w}) = \mathbf{w}^T\mathbf{x}$$

- Deterministic non-linear model (φ: non-linear function)

$$y = f(\mathbf{x}|\mathbf{w}) = \mathbf{w}^T\phi(\mathbf{x})$$

- Non-linear model - Probabilistic interpretation

$$y = f(\mathbf{x}|\mathbf{w}) = \mathbf{w}^T\phi(\mathbf{x}) + \epsilon, \;\; \epsilon \sim \mathcal{N}(\mu, \sigma^2)$$



Unsupervised Learning

- Discovering structure (patterns, regularities, etc.) in "unlabelled" data
- Density estimation: we want to see what generally happens and what not

$$p(\mathbf{x_i}|\boldsymbol{\theta})$$

instead of $p(y_i|\mathbf{x_i};\boldsymbol{\theta})$ (supervised learning)

- Clustering
  - identifying sub-populations in the data
- Dimensionality reduction
  - project data to a lower dimensional subspace capturing its essence
- Matrix completion
  - data imputation to infer values of non-existing entries

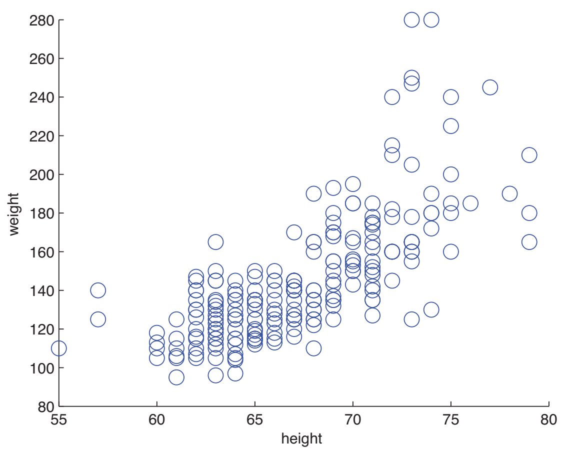
Unsupervised Learning: Clustering

- Step 1: Estimate the distribution over the number of clusters

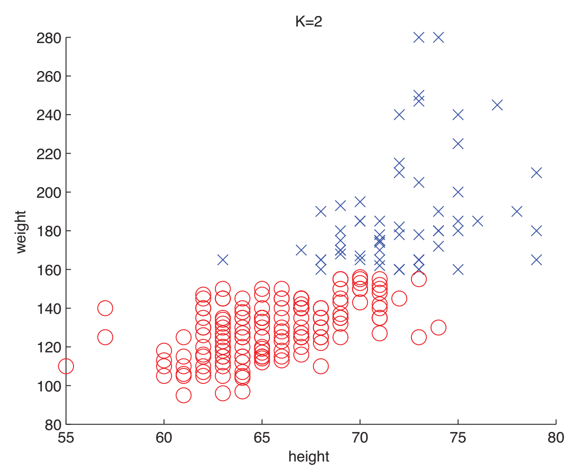
$$p(K|\mathcal{D})$$

- Step 2: Estimate which cluster each point belongs to

$$z_i^* = arg \max_{k=1,\dots,K} p(z_i = k|\mathbf{x_i}, \mathcal{D})$$



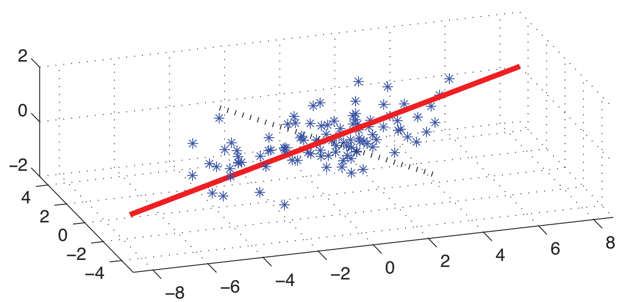
(a)



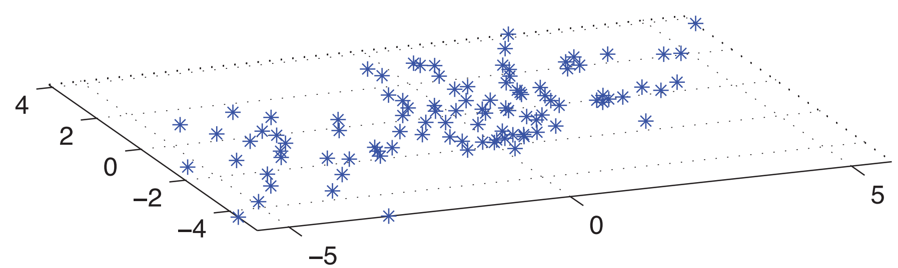
(b)

Unsupervised Learning: Dimensionality Reduction

- Lower dimensional representations can have better predictive power
  - minimized data redundancies
  - avoiding “curse of dimensionality”



(a)



(b)

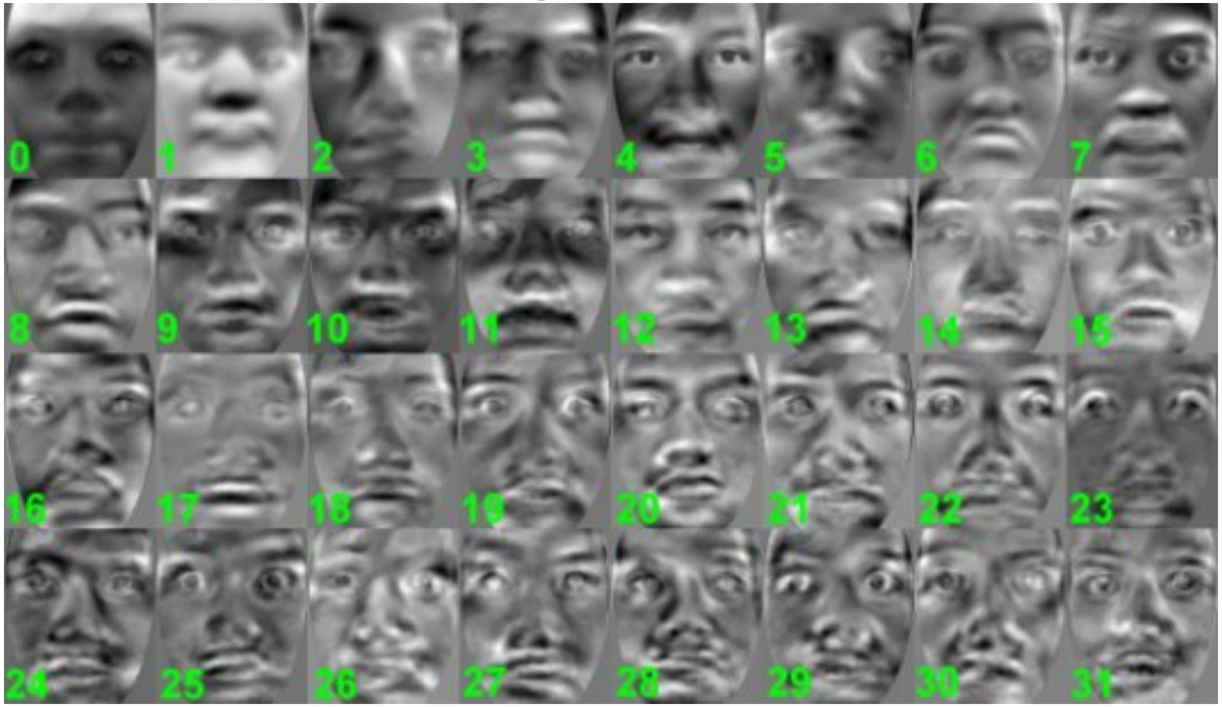
Principal component analysis (PCA)

identifies a set of uncorrelated axes that maximize the variance of the data

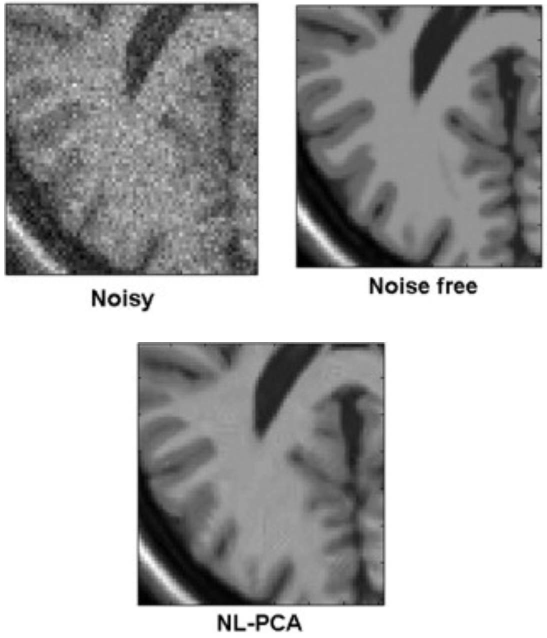
Unsupervised Learning: Dimensionality Reduction

Example applications of PCA

Eigenfaces



MRI denoising



Unsupervised Learning: Matrix completion

Recommender systems

users

movies

1		?	3	5	?
?	1				2
	4		4	5	?

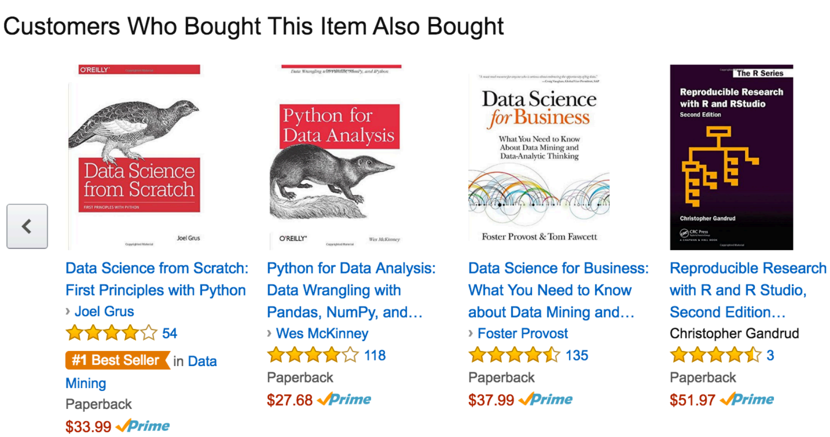
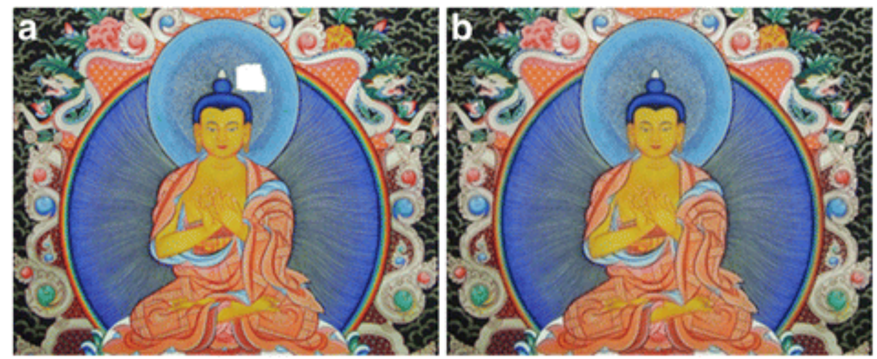


Image restoration



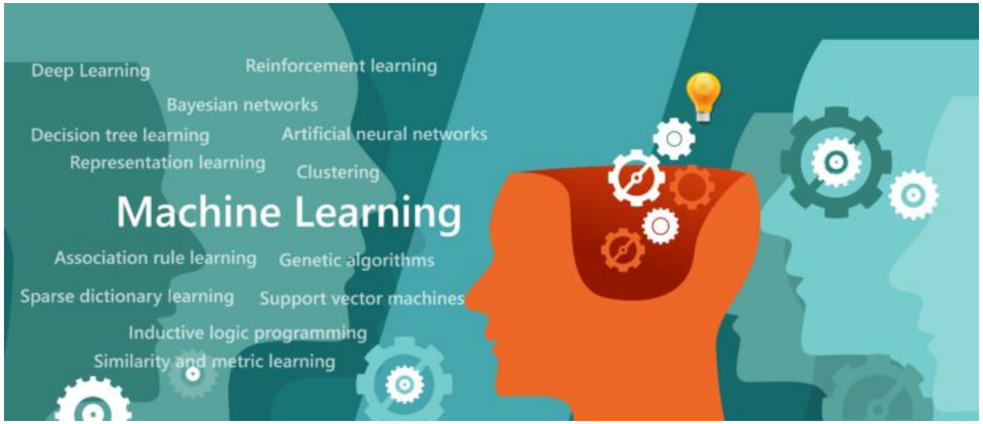
Sources: Wang & Jia, 2017; Papandreou, Maragos, & Kokaram, 2008

To sum up

- Machine learning definition
- Key components of learning: representation, evaluation, optimization
- Types of learning systems: supervised & unsupervised
- Challenges in machine learning

Outline

- About this class
- Introduction to Machine Learning
  - What is Machine Learning?
  - Basic challenges

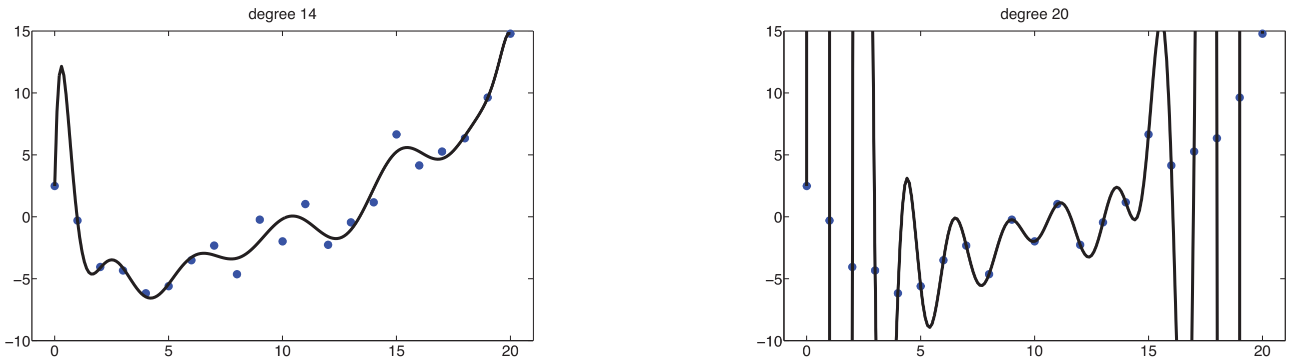


Key Machine Learning Challenges
Generalization

- Biggest ML challenge is to generalize beyond the training set
- **Never evaluate your ML system on the train data only**
  - Use test data instead
- Contamination of the ML system from the test data can occur when:
  - use test through excessive parameter tuning
    - Avoid this with **(cross-)validation**
- On the positive side 🙂
  - We may not need to fully optimize it, since the objective function is only a proxy of the true one

Key Machine Learning Challenges
Overfitting

- The risk of using highly flexible (complicated) models without having enough data
- Ways to avoid overfitting
  - (cross-)validation
  - regularization

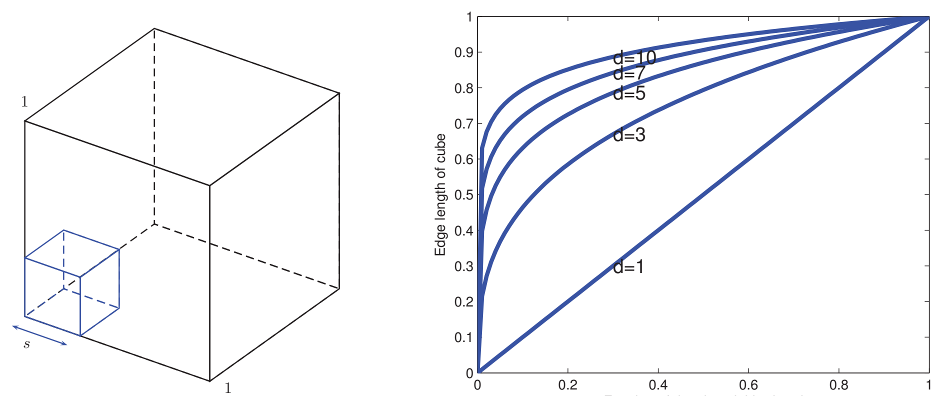


(a) (b)

Example of polynomial fit

Key Machine Learning Challenges
Curse of dimensionality

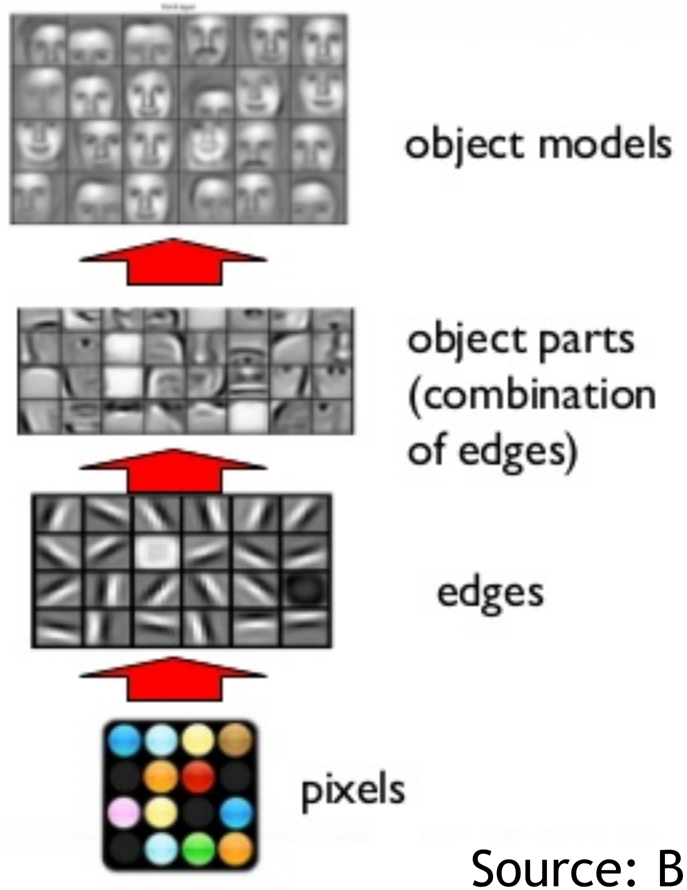
- All intuition fails in higher dimensions
- For a fixed training set, generalization gets harder in larger dimensions
  - harder to systematically search a high-dimensional grid-space
  - harder to accurately approximate a high-dimensional function
- On the positive side 🙂
  - "blessing of non-uniformity": examples aren't usually spread uniformly



(a) (b)

Key Machine Learning Challenges
Feature Engineering

- Learning is easy if you have informative features for the problem
- Automating the feature engineering process
  - Deep learning systems producing output from raw input



Source: Baidu

Key Machine Learning Challenges
"No-free-lunch" theorem

- "All models are wrong but some models are useful", G. Box, 1987
- There is no single best ML system that works optimally for all kinds of problems
- On the positive side 🙂
  - General assumptions can actually work pretty well, e.g.
    - Similar examples belong to similar classes
    - Independence and smoothness assumptions
- We might need to try lots of different ML systems and learning algorithms to cover the wide variety of real-world data.
- Machine learning is not magic: it can't get something out of nothing, but it can get more from less!

To sum up

- Machine learning definition
- Key components of learning: representation, evaluation, optimization
- Types of learning systems: supervised & unsupervised
- Challenges in machine learning

Readings:

- Alpaydin Ch1, Abu-Mostafa Ch 1
- P. Domingos, "A few things to know about machine learning"

Fun videos to watch:

- https://www.youtube.com/watch?v=R9OHn5ZF4Uo
- www.youtube.com/watch?v=ujxriwApPP4
- https://www.youtube.com/watch?v=Q-Qq8ipUHEI