Lecture 1

STAT161/261 Introduction to Pattern Recognition and Machine Learning

Spring 2019

Prof. Allie Fletcher

UCLA

Outline

- Course Information and Details
- What and why Machine Learning?
  - Machine learning vs. expert knowledge
  - Classification, regression, unsupervised, reinforcement learning
  - Why machine learning today?
- Principles of Supervised Learning
  - Model Selection and Generalization
  - Overfitting and Underfitting
- Decision Theory (1.5 Bishop and Ch3 Alpaydin)
  - Classification, Maximum Likelihood and Log likelihood
  - Bayes Methods: MAP and Bayes Risk

Course Admin

- People:
  - Prof. Allie Fletcher.
  - TA:
- Where:
  - MW 3:30-4:45pm, Public Affairs Bldg 2238
  - Discussion: Friday, TBA
- Grading:
  - C261: Midterm 20%, Final 35%, HW and labs 25%, Quizzes&Participation 10%, Project 10%,
  - C161: Midterm 20%, Final 35%, HW and labs 35%, Quizzes&Participation 10%
  - Project is for graduate students only (see below)
  - Homework will include programming assignments
  - Midterm tentatively May 8
  - Midterm and final are closed book. Equation sheet is provided.

Pre-Requisites

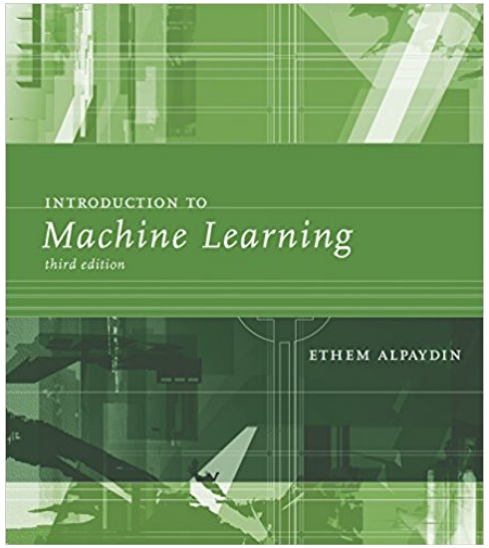
- Note: This class is primarily aimed at graduate students and a few advanced undergraduates
  - It will be move at a **fast** pace. It will assume you know the background material well.
  - There are many other stat undergraduate electives available.
- Undergraduate probability and statistics
  - UCLA STAT100B or equivalent
  - Random variables, distributions, multivariate distributions, multivariate Gaussians
  - ML estimation, hypothesis testing, regression
- Undergraduate linear algebra
  - UCLA MATH 33A or equivalent
  - Linear systems of equations, matrices, determinants
  - Eigenvectors, eigenvalues, SVDs
- Programming:
  - You can do assignments in any language of your choice (R, Python, …)
  - Later part of class will use Tensorflow but you can use any deep learning environment
  - But, the assignments will be given in python. You will need to translate if needed

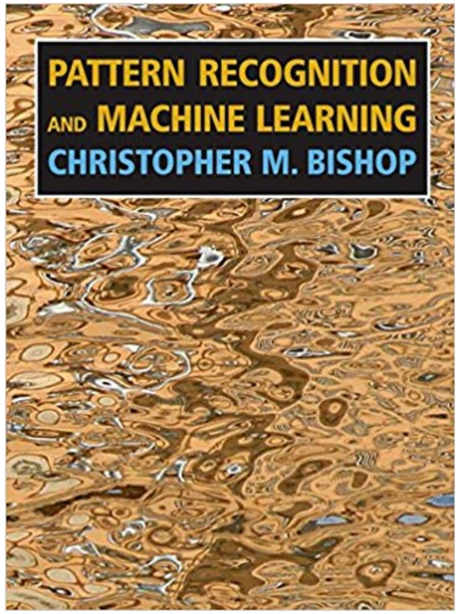
Machine Learning Project

- Graduate students only
- Perform an interesting machine learning task of your choice
- Many possible areas:
  - Machine vision, brain-computer interfaces, natural language processing, …
  - Anything that interests you
- Groups of 2 to 4
  - 2 paragraph project proposal around the midterm
  - Final project: github with your code,
  - 3-5 page write-up : any jupyter notebook, latex, or word,
  - Short presentation (5 minutes) in 2nd to last or last lecture
- Use real data
  - Many datasets available today
- Write code
- You may use other people's code as a starting point. But:
  - You MUST cite any code you use
  - You must go beyond their code. Not simply re-run their code
  - Plagiarizing code without citation will be severely punished

Main texts

- Alpaydin, “Introduction to Machine Learning”, 3rd ed
  - Develops background in decision theory and parameter estimation
  - Will follow sequence of topics
- Bishop, “Pattern Recognition and Machine Intelligence”
  - Widely-used
  - Statistical perspective





Supplementary Material

- Hastie, Tibshirani, Friedman. Elements of Statistical Learning.
- Murphy. Machine Learning. “A Probabilistic Perspective.
- Raschka, “Python Machine Learning”, 2015.
  - http://file.allitebooks.com/20151017/Python%20Machine%20Learning.pdf
- Siraj Raval YouTube channel https://www.youtube.com/channel/UCWN3xxRkmTPmbKwht9FuE5A
- Many Coursera courses.

Topics Covered

- Supervised Learning
  - Bayes classification, Bayes risk, ROC
  - Multi-variate models and multivariate linear regression
- Model selection, bias, variance and
- Linear discrimination, logistic regression, SVM
- Dimensionality reduction, PCA
- Nonparametric methods
- Clustering, k-means, EM
- Non parametric estimation
- Multi-layer perceptrons
- Convolutional neural networks
- Deep learning

Outline

- Course Information and Details
- What and why Machine Learning?
  - Machine learning vs. expert knowledge
  - Classification, regression, unsupervised, reinforcement learning
  - Why machine learning today?
- Principles of Supervised Learning
  - Model Selection and Generalization
  - Overfitting and Underfitting
- Decision Theory (1.5 Bishop and Ch3 Alpaydin)
  - Classification, Maximum Likelihood and Log likelihood
  - Bayes Methods: MAP and Bayes Risk

What is Machine Learning?

- Learn to improve algorithms from data
- Optimize a performance criterion using example data or past experience

- Role of Statistics: Makes inference from a sample
  - Make inferences on things we haven’t seen from things we have seen

- Role of Computer science: Efficient algorithms to
  - Solve the optimization problem
  - Representing and evaluating the model for inference

Why Learn?

- Machine learning programming computers to:
  Optimize a performance criterion using example data or past experience
- There is no need to "learn" to calculate payroll: fixed algorithm
- No need to learn if you can derive the algorithm from first principles

- Learning is used when:
  - Human expertise does not exist (navigating on Mars)
  - *Humans are unable to explain their expertise (language translation, image recognition)
  - Solution changes in time or needs to be adaptive (routing on a computer network)
- Best example of this is language translation
  - Used to have professional translators come in and work with computer scientists together
- Google came at this a completely different way and combed the web for translated documents
  - So they had a large body of documents
  - Humans generated the data
  - But the humans didn't codify the rules that could be placed into a computer program

When We Talk about "Learning"?

- Learning general models from a data of particular examples

- Data is cheap and abundant
- Knowledge is expensive and scarce

- Example in retail: Customer transactions to consumer behavior:

  People who bought "Hello kitty pencil case" also bought "pink backpack" (www.amazon.com)

- Build a model that is *a good and useful approximation* to the data
  - Need not be exact just good enough for our purpose

Data Mining in Almost All Fields

- Data mining is often used *by* machine learning
  - Explores data to find connections or relationships or pattern
- Retail: Market basket analysis, Customer relationship management (CRM)
- Finance: Credit scoring, fraud detection
- Manufacturing: Control, robotics, troubleshooting
- Medicine: Medical diagnosis
- Telecommunications: Spam filters, intrusion detection
- Bioinformatics: Motifs, alignment
- Web mining: Search engines
- …

Types of Learning

- Association
- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
- Reinforcement Learning

Learning Associations

- Basket analysis:

  $P\ (Y \mid X\ )$ probability that somebody who buys X also buys Y
- X and Y are products/services.

  Example: P (chips | beer) = 0.7

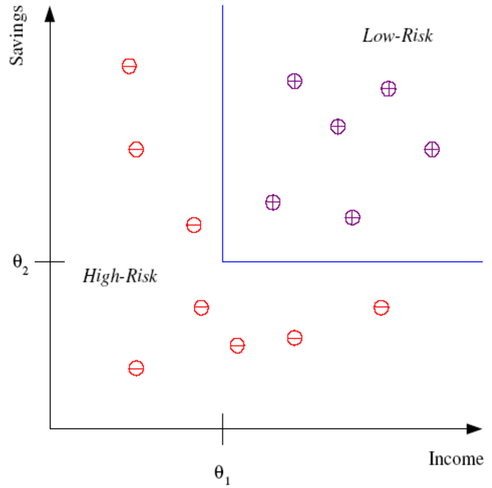
  P (bubble tea | Bud light) = 0.08

- If you bought beer, a retailer could show you an ad for chips
- Probablly a waste of money to send budlight buyer, bubble tea ads

- Of course, you can do many more complex things with more data

Classification : Credit Score

- Differentiating between low-risk and high-risk customers from their *income* and *savings*
- Select some features:
  - Example: income & savings
  - Represent as a vector $x = (x_1, x_2)$
- Learn a function from features to target
  - Use past training data
  - Need to get this data
- Function



Discriminant: IF *income* $> \theta_1$ AND *savings* $> \theta_2$ THEN low-risk ELSE high-risk

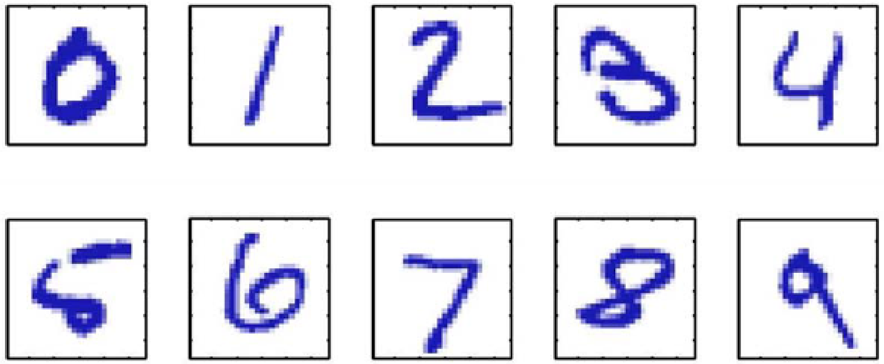
- The function on the right is also an example of a decision tree. If savings are above a line, and then if income is above a line, then the candidate is low-risk.
- Features, target, data, rule

Chihuahua versus Muffin

Can you Classify this? Can your computer?

- How do you classify chihuahua versus muffin?
- Previous case - 2 dim input
- 256x256 input of an image
- Key: build a very deep network
- We will discuss how to do this classifier later

Classification and Expert Rules: Digit Recognition



Images are 28 x 28 pixels

- Problem: Recognize a digit from the image

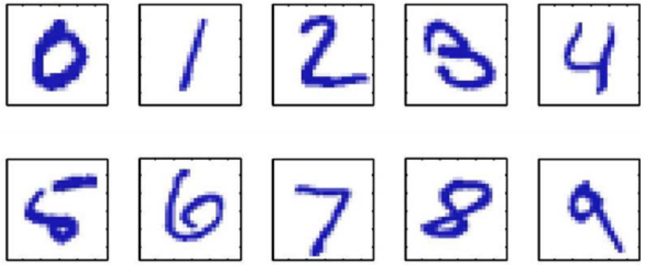
- MNIST dataset challenge
  - Dataset developed in 1990s to spur AI research on a challenging problem for the time
  - Data taken from census forms
  - Became a classic benchmark for machine vision problems
  - We will see this dataset in this class

Classical "Expert" Approach

- Idea: Use your knowledge about digits
  - You are an "expert" since you can do the task
  - So, you construct simple rules and code them

- Expert rule example: "*Image is a digit 7 if…*":
  - There is a single horizontal line, and
  - There is a single vertical line

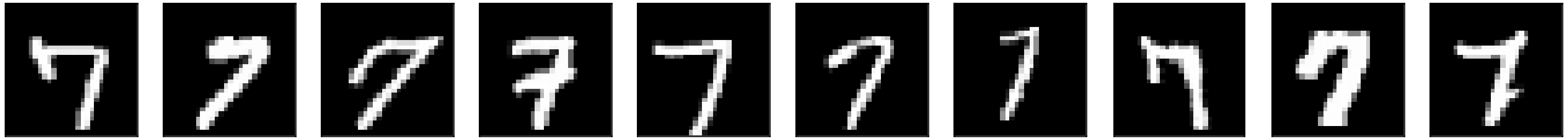
- Rule seems simple and reasonable
- But,…



Images are 28 x 28 pixels

```
def count_vert_lines(image):
    ...
def count_horiz_lines(image):
    ...

def classify(image):
    ...
    nv = count_vert_lines(image)
    nh = count_horiz_lines(image)
    ...

    if (nv == 1) and (nh == 1):
        digit = 7
    ...

    return digit
```

Problems with Expert Rules



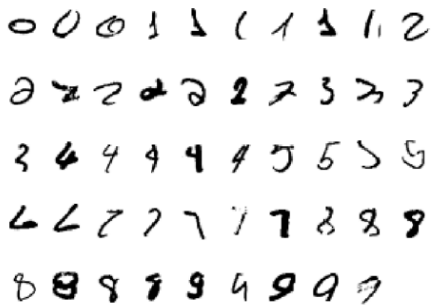
- Simple expert rule breaks down in practice
  - Hard to define a "line" precisely
  - Orientation, length, thickness, …
  - May be multiple lines…

- General problem: We cannot easily code our knowledge
  - We can do the task
  - But, it is hard to translate to simple mathematical formula

```
def count_vert_lines(image):
    ...
def count_horiz_lines(image):
    ...

def classify(image):
    ...
    nv = count_vert_lines(image)
    nh = count_horiz_lines(image)
    ...

    if (nv == 1) and (nh == 1):
        digit = 7
    ...

    return digit
```

ML Approach: Learn from Data

Training inputs images $\boldsymbol{x}_i$ (ex. 5000 ex per class)



?

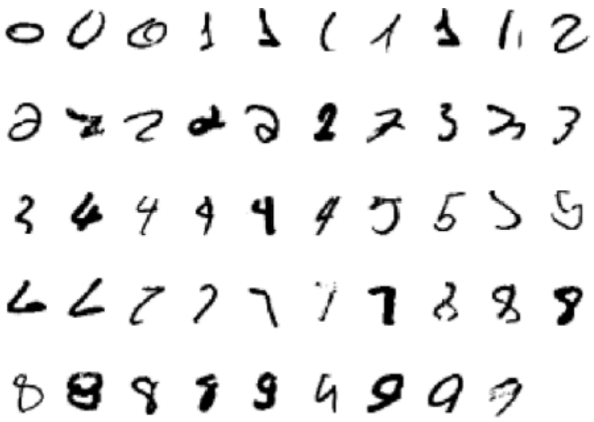
Learned classifier
$f(\boldsymbol{x})$

Training output labels $y_i \in \{0,1,\ldots,9\}$

- Do not use your "expert" knowledge
- Learn the function from data!
- Supervised learning:
  - Get many labeled examples $(\boldsymbol{x}_i, y_i), i = 1, \ldots, N$ (Called the training data)
  - Each example has an input $\boldsymbol{x}_i$ and output y_i
  - Learn a function $f(\boldsymbol{x})$ such that: $f(\boldsymbol{x}_i) = y_i$ for "most" training examples

ML Approach Benefits and Challenges

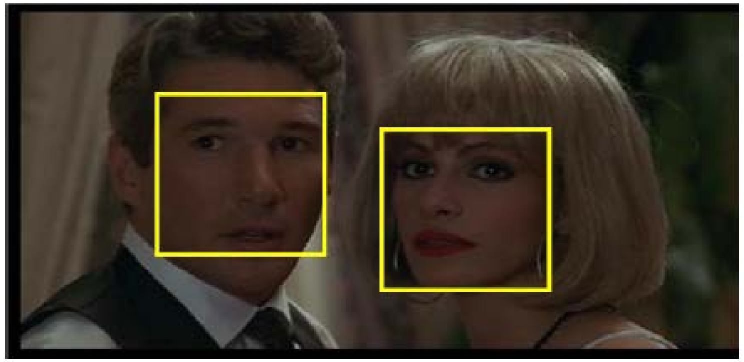
- Learned systems do very well on image recognition problems
  - On MNIST, current systems get <0.21% errors (as of 1/20/2018)
  - Used widely in commercial systems today (e.g. OCR)
  - Cannot match this performance with an expert system



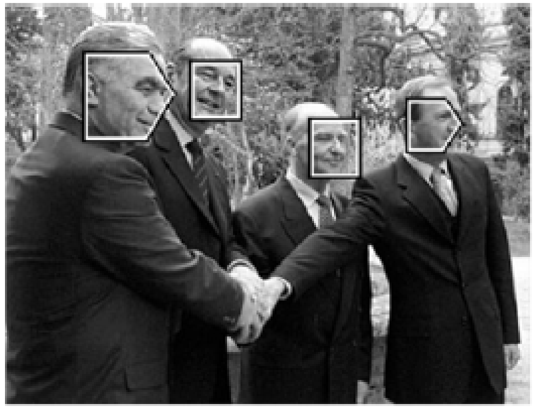
- But, there are challenges:
  - How do we acquire data? Someone has to manually label examples.
  - How do we parametrize a set of functions $f(\boldsymbol{x})$ to search?
  - How do we fit the function to data?
  - If a function works on training example, will it generalize on new data?

- This is what you will learn in this class

Example : Face Detection



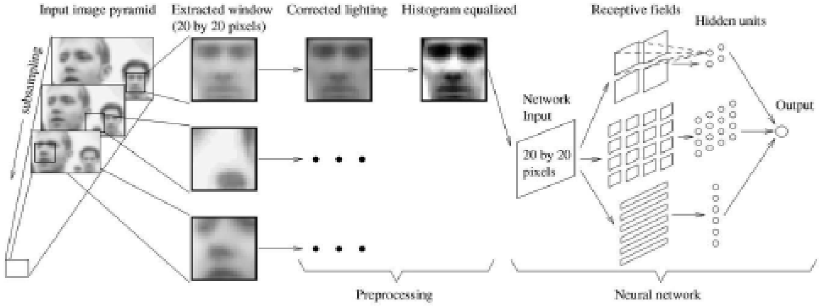
- Problem: For each image region, determine if face or non-face
- More challenging than digit recognition
  - Even harder to describe a face via “rules” in a robust way
- Face Recognition Harder Yet
- MANUAL MEASUREMENTS BY BLEDSOE (1960S)
- INCREASED ACCURACY WITH 21 FACIAL MARKERS (1970S)
- EIGENFACES (LATE 1980S-EARLY 1990S)
  - Low dimensional representation, limited by computer power
- DarPA FERET PROGRAM (1993-2000S)
  - Database of images. Updated in 2003 high-resolution 24-bit color images2,413 still facial images representing 856 people
- SUPER BOWL XXXV (2002)-Law enforcement tried to use it. Fail



UCLA

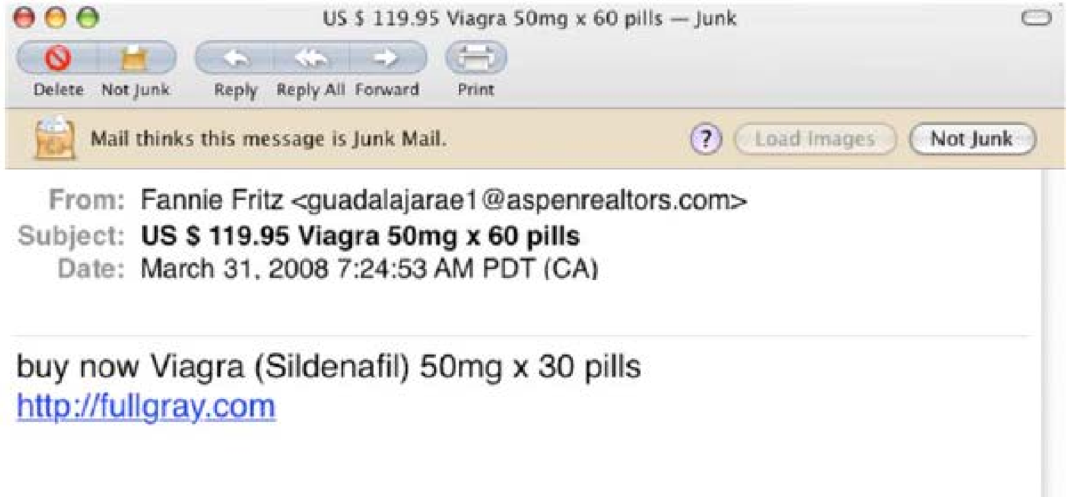
Supervised Learning Approach

- Data: Get large number of face and non-face examples
- Typical early dataset
  - 5000 faces (all near frontal, vary age, race, gender, lighting)
  - 10^8 non faces
  - Faces are normalized (scale, translation)
- Learn a classifier from a class of functions
  - Each function maps image to binary value "face" or "non-face"
  - Select function that works well on training data
  - For good performance, functions may be complex
  - Many parameters
- Many more datasets are available now:
  - See http://www.face-rec.org/databases/
  - You can use this for your project!



Rowley, Baluja and Kanade, 1998

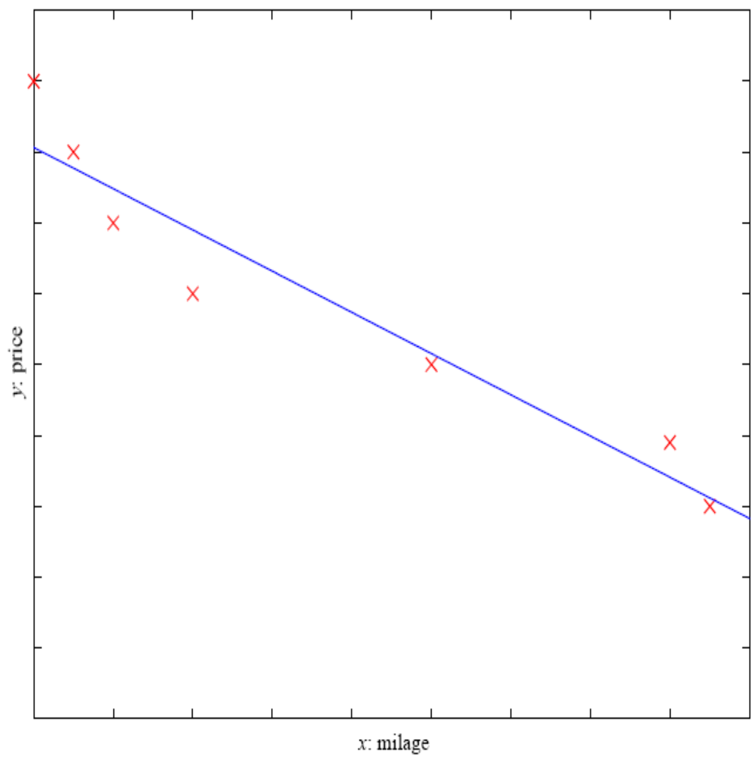
Example 3: Spam Detection



- Classification problem:
  - Is email junk or not junk?
- For ML, must represent email numerically
  - Words bcome vector of numbers
  - Common model: bag of words
  - Enumerate all words, $i = 1, \ldots, N$
  - Represent email via word count
    x_i = num instances of word i
- Challenge:
  - Very high-dimensional vector
  - System must continue to adapt
    (keep up with spammers)

Supervised Learning: Regression

- Target variable y is continuous-valued
- Example:
  - Predict $y =$ price of car
  - From $x =$ mileage, size, horsepower, ..
  - Features can be discrete (e.g. brand)
  - Can use multiple predictors
- Assume some form of the mapping
  - Ex. Linear: $y = \beta_0 + \beta_1 x$
  - Find parameters β_0, β_1 from data



Regression Example

Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Diabetes Data Set

Download: Data Folder, Data Set Description

File Names and format:
(1) Date in MM-DD-YYYY format
(2) Time in XX:YY format
(3) Code
(4) Value

The Code field is deciphered as follows:

33 = Regular insulin dose
34 = NPH insulin dose
35 = UltraLente insulin dose
48 = Unspecified blood glucose measurement
57 = Unspecified blood glucose measurement
58 = Pre-breakfast blood glucose measurement
59 = Post-breakfast blood glucose measurement
60 = Pre-lunch blood glucose measurement
61 = Post-lunch blood glucose measurement
62 = Pre-supper blood glucose measurement
63 = Post-supper blood glucose measurement
64 = Pre-snack blood glucose measurement
65 = Hypoglycemic symptoms
66 = Typical meal ingestion
67 = More-than-usual meal ingestion
68 = Less-than-usual meal ingestion
69 = Typical exercise activity
70 = More-than-usual exercise activity
71 = Less-than-usual exercise activity

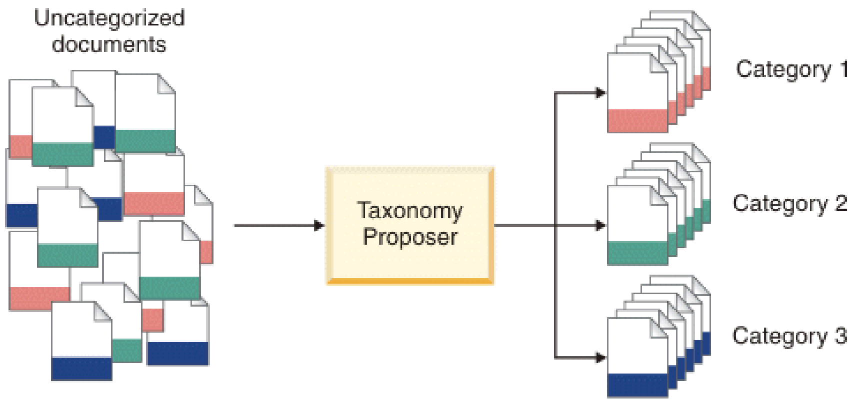
- Predict blood glucose level
- Many possible predictors:
  - Recent past levels
  - Insulin dose
  - Time of last meal
  - …
- Check out data in: https://archive.ics.uci.edu/ml/datasets/Diabetes

Supervised Learning in General

- Labelled Data: Use the labelled training data
- Prediction of future cases: Use the rule to predict the output for future inputs
- Knowledge extraction: Find a relation between predictor and target
- Compression: The rule is simpler than the data it explains
- Outlier detection: Exceptions that are not covered by the rule, e.g., fraud

Unsupervised Learning

- Unsupervised learning: Requires data, but no labels
- Question: When and why would we want to do this?
  - Don't know what we are looking for
  - Automatically organizing data
  - Understanding hidden structure in some data
  - Representing high-dimensional data in a low-dimensional space
- Examples:
  - Customers shopping patterns & regionalities
  - Genes according to expression profile
- Clustering: Finds groups / clusters in data
  - Automatically segment data into groups of similar points
  - Customer segmentation
  - Image compression: Color quantization
  - Search results according to topic
  - A museum catalog according to image similarity

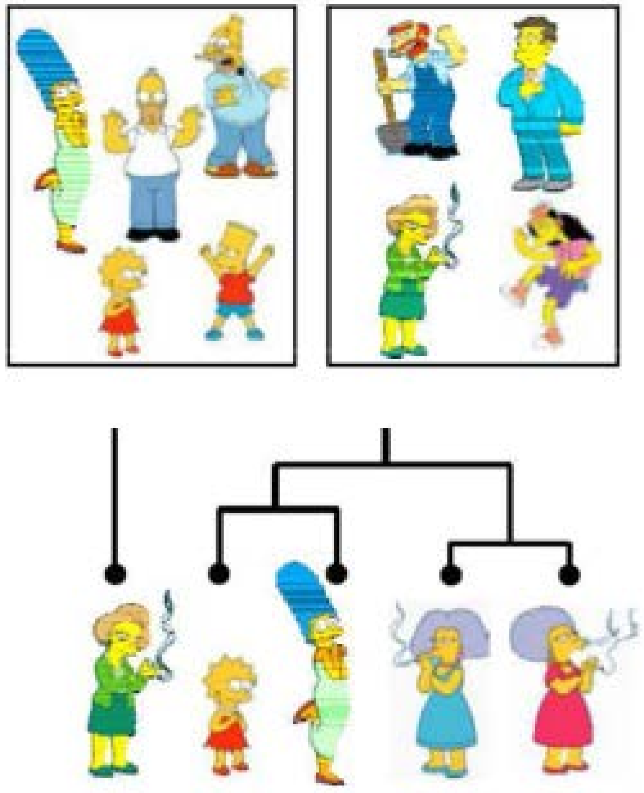


Example: Document classification
http://www.ibm.com/support/knowledgecenter
/SSBRAM_8.7.0/com.ibm.classify.ccenter.doc/
c_WBG_Taxonomy_Proposer.htm

See also Google News

Unsupervised Learning Example: Clustering Algorithms

- Partition algorithms (flat)
  - No hierarchy
  - Simpsons family versus on family
- Hierarchical clustering
  - Bottom up: Agglomerative
  - Top down: Divisive
    - Women versus men
    - Simpsons women versus nonSimpson women
    - Homer family versus nonimmediate Homer Simpson family
    - Or smoking or nonsmoking



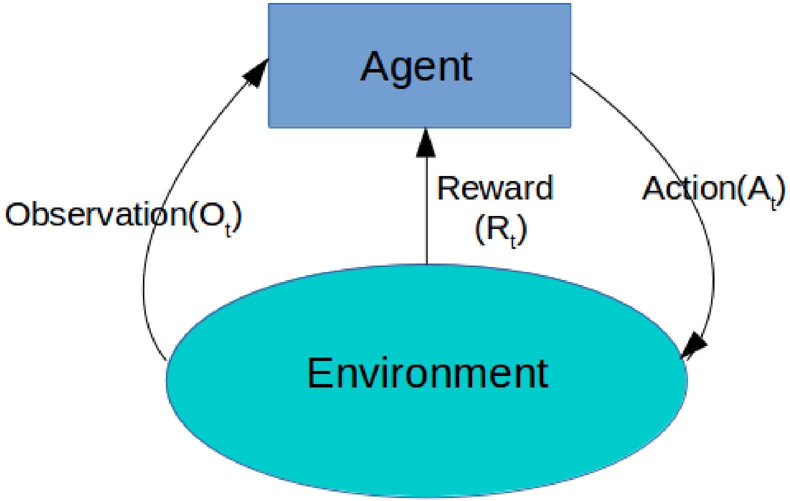
Example: Image Segmentation

- Goal: Group similar pixels together
  - Break up image into clusters
  - Meaningful or perceptually similar regions
  - E.g. proximity and color
  - Useful for inpainting
  - Photoshop -- only process "clouds"
  - Measurement - amount of clouds

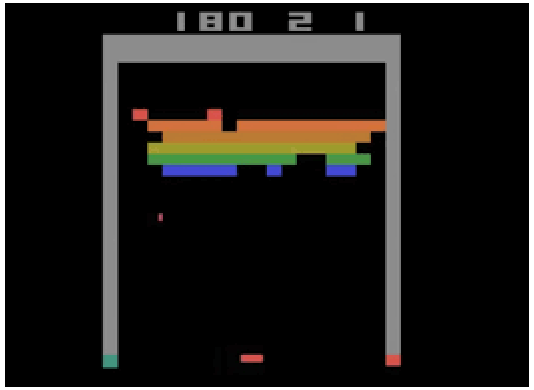
[Slide from James Hayes]

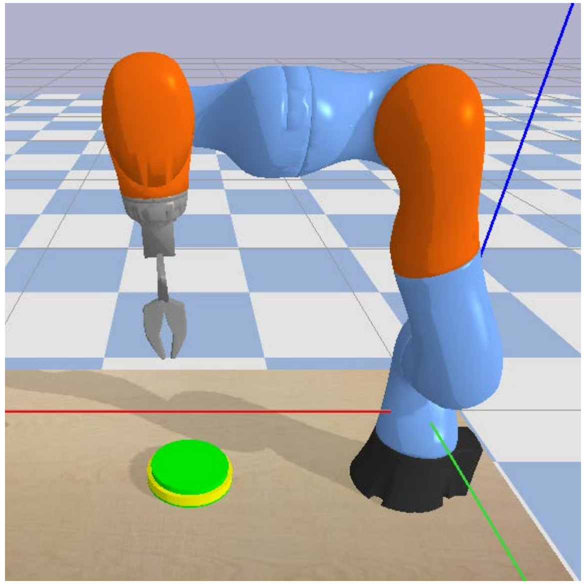
Reinforcement Learning

- Problem:
  - Get a sequence of observations
  - Learn a sequence of actions
- Key challenge: No supervised output but delayed rewa
  - Effect of actions now only available in future
  - Called the credit assignment problem
  - Which past actions do we assign credit for a reward?
- Learn a game of chess:
  - Observations: opponent's moves.
  - Actions: Your move.
  - Reward: you win or lose, but reward is delayed



Reinforcement Learning Examples Today





- Game playing by Google Deepmind
  - Atari
  - AlphaGo: First computer to beat Go Master
- Robotics

- Key features
  - Learn a sequence of actions
  - Reward arrives in future (e.g. at the end of the game)

What ML is Doing Today?

- Autonomous driving
- Jeopardy
- Machine translation

- Many, many others…

Why Now?

- Machine learning is an old field
  - Much of the pioneering statistical work dates to the 1950s
- So what is new now?
- Big Data:
  - Massive storage. Large data centers
  - Massive connectivity
  - Sources of data from Internet and elsewhere
- Computational advances
  - Distributed machines, clusters
  - GPUs and hardware

Google Tensor Processing Unit (TPU)

Outline

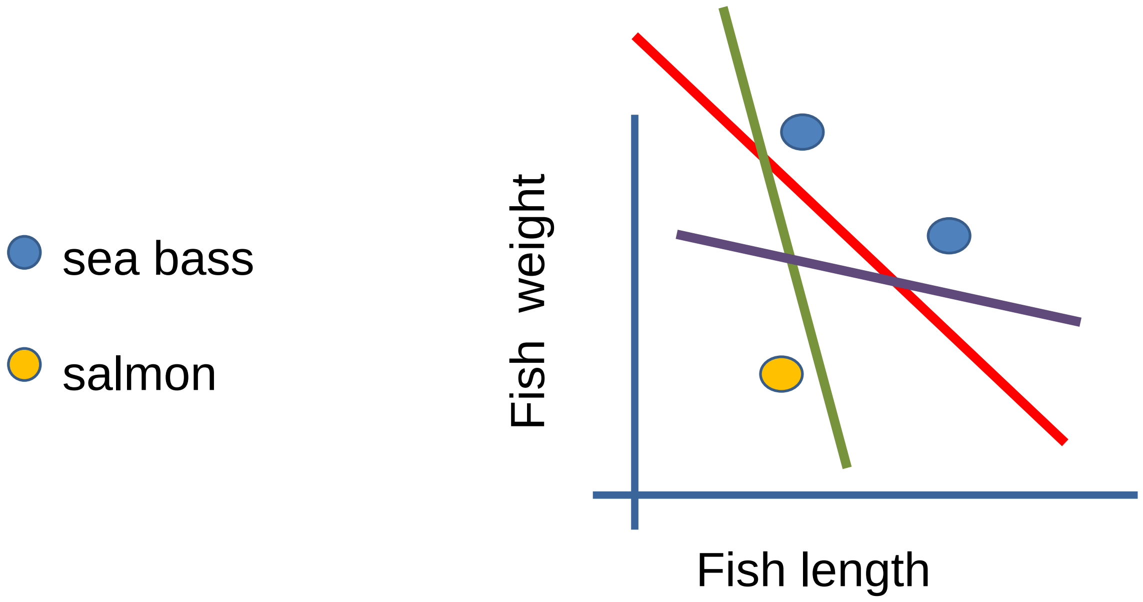
- Course Information and Details
- What and why Machine Learning?
  - Machine learning vs. expert knowledge
  - Classification, regression, unsupervised, reinforcement learning
  - Why machine learning today?
- Principles of Supervised Learning
  - Model Selection and Generalization
  - Overfitting and Underfitting
- Decision Theory (1.5 Bishop and Ch3 Alpaydin)
  - Classification, Maximum Likelihood and Log likelihood
  - Bayes Methods: MAP and Bayes Risk

Memorization vs. Generalization

- Two key concepts in ML:
  - Memorization: Finding an algorithm that fits training data well
  - Generalization: Gives good results on data not yet seen. Prediction.
- Example: Suppose we have only three samples of fish
  - Can we learn a classification rule? Sure

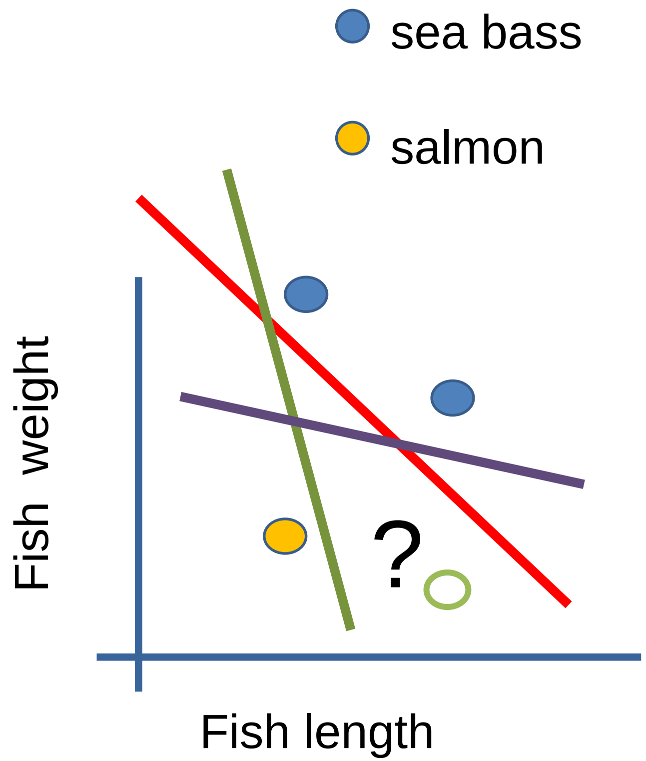
Memorization vs. Generalization

- Many possible classifier fit training data
  - Easy to memorize the data set, but need to generalize to new data
  - All three classifiers below (Classifier 1, C2, and C3) fit data
- But, which one will predict new sample correctly?



Memorization vs. Generalization

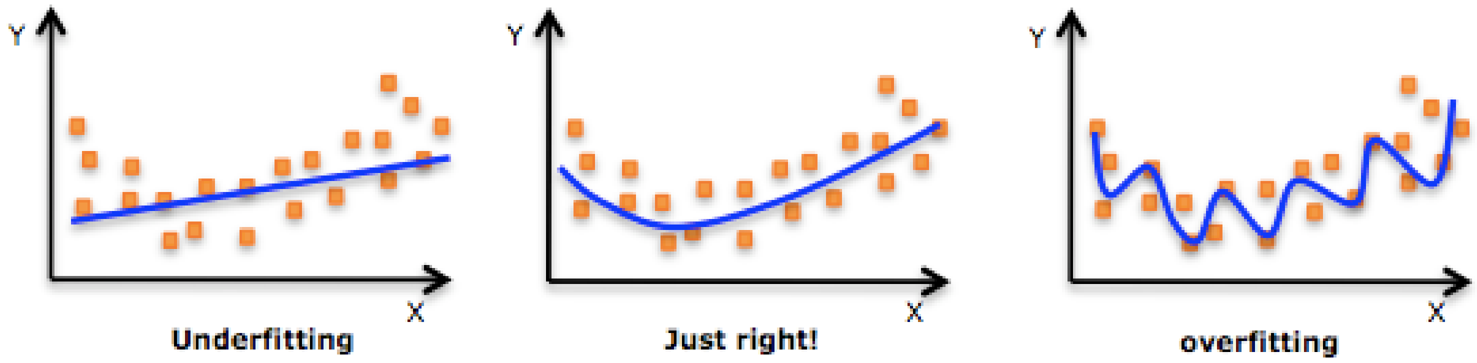
- Which classifier predicts new sample correctly?
  - Classifier 1 predicts salmon
  - Classifier 2 predicts salmon
  - Classifier 3 predicts sea bass
- We do not know which one is right:
  - Not enough training data
  - Need more samples to generalize



Basic Tradeoff

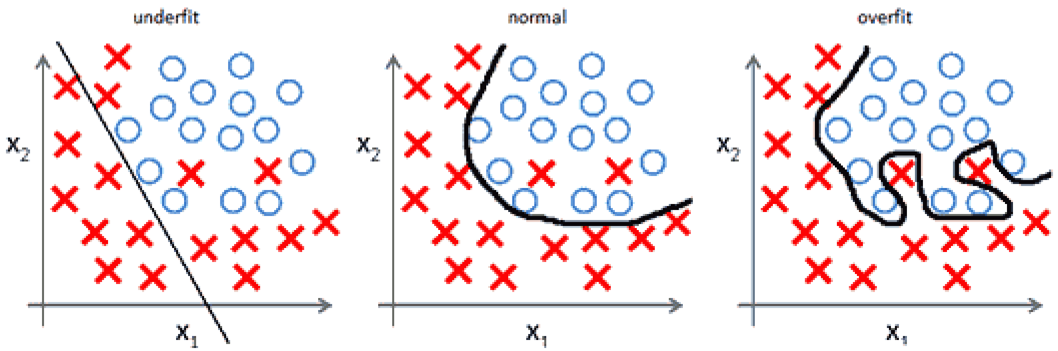
- Generalization requires assumptions
- ML uses a model
- Basic tradeoff between three factors:
  - Model complexity: Allows to fit complex relationships
  - Sample complexity: Amount of training data
  - Generalization error: How model fits new samples
- This class: Provides a principled ways to:
  - Formulate models that can capture complex behavior
  - Analyze how well they perform under statistical assumptions

Generalization: Underfitting and Overfitting



- Example: Consider fitting a polynomial
- Assume a low-order polynomial
  - Easy to train. Less parameters to estimate
  - But model does not capture full relation. Underfitting
- Assume too high a polynomial
  - Fits complex behavior
  - But, sensitive to noise. Needs many samples. Overfitting
- This course:
  - How to rigorously quantify model selection and algorithm performance

Generalization: Underfitting and Overfitting



- Similar example for a classification problem
  - Learn a polynomial classification boundary
- Assume a low-order polynomial
  - Easy to train. Less parameters to estimate
  - But model does not capture full relation. Underfitting
- Assume too high a polynomial
  - Fits complex behavior
  - But, sensitive to noise. Needs many samples. Overfitting

Ingredients in Supervised Learning

- Select a model: $\hat{y} = g(x, \theta)$
  - Describes how we predict target y from features x
  - Has parameters θ
- Get training data: $(x_i, y_i), i = 1, \ldots, n$
- Select a loss function $L(y_i, \hat{y}_i)$
  - How well prediction matches true value on the training data
- Design algorithm to try to minimize loss:

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

- The art: principled methods to develop models and algorithms for often intractable loss functions and complex large data sets is what machine learning is really all about.