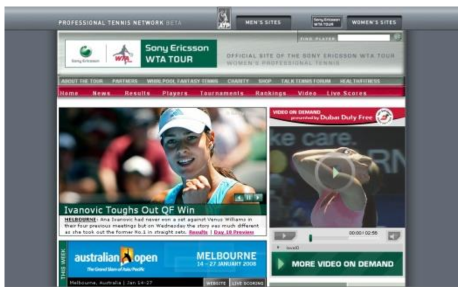
Machine Learning 1

Introduction,Course Overview

Machine Learning

Image Classification


Document Categorization



Speech Recognition   Protein Classification   Spam Detection

Branch Prediction   Fraud Detection   Natural Language Processing

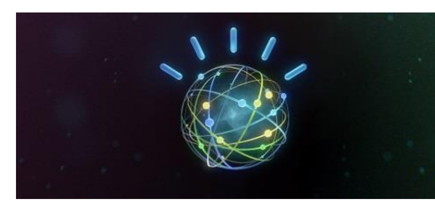
Playing Games   Computational Advertising

Machine Learning is Changing the World

"Machine learning is the hot new thing"
(John Hennessy, President, Stanford)

"A breakthrough in machine learning would be worth ten Microsofts" (Bill Gates, Microsoft)

"Web rankings today are mostly a matter of machine learning"
(Prabhakar Raghavan, VP Engineering at Google)



The COOLEST TOPIC IN SCIENCE

- “A breakthrough in machine learning would be worth ten Microsofts” (Bill Gates, Chairman, Microsoft)
- “Machine learning is the next Internet” (Tony Tether, Director, DARPA)
- Machine learning is the hot new thing” (John Hennessy, President, Stanford)
- “Web rankings today are mostly a matter of machine learning” (Prabhakar Raghavan, Dir. Research, Yahoo)
- “Machine learning is going to result in a real revolution” (Greg Papadopoulos, CTO, Sun)
- “Machine learning is today’s discontinuity” (Jerry Yang, CEO, Yahoo)

<u>This course</u>: introduction to machine learning.

- Cover (some of) the most commonly used machine learning paradigms and algorithms.
  - Sufficient amount of details on their mechanisms: explain why they work, not only how to use them.
  - Applications.

What is Machine Learning?

Examples of important machine learning paradigms.

Supervised Classification

from data to discrete classes

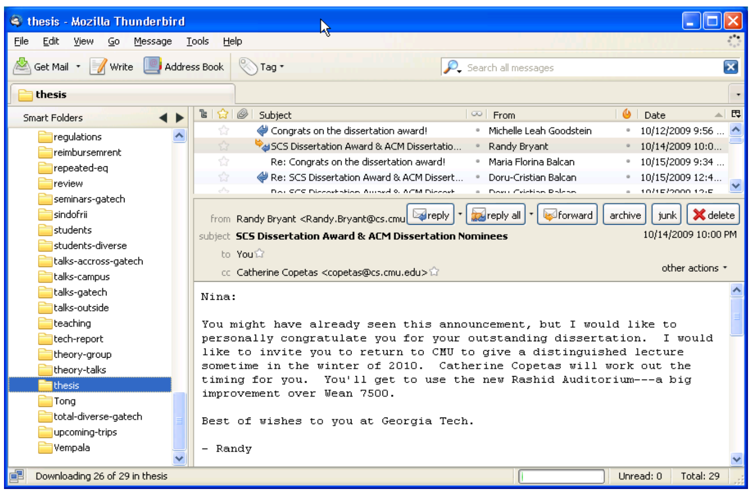
Supervised Classification. Example: Spam Detection

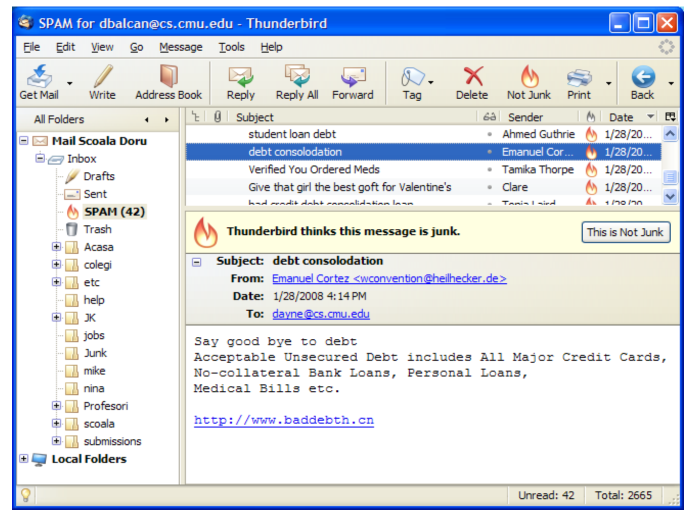
Decide which emails are spam and which are important.

Not spam

Supervised classification

spam





Goal: use emails seen so far to produce good prediction rule for future data.

Supervised Classification. Example: Spam Detection

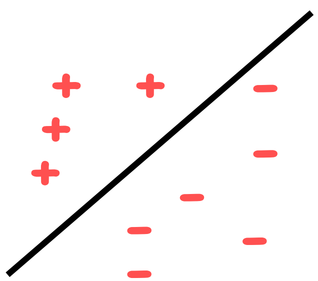
Represent each message by features. (e.g., keywords, spelling, etc.)

	"money"	"pills"	"Mr."	bad spelling	known-sender	spam?	
	Y	N	Y	Y	N	Y	
	N	N	N	Y	Y	N	
	N	Y	N	N	N	Y	
example	Y	N	N	N	Y	N	label
	N	N	Y	N	Y	N	
	Y	N	N	Y	N	Y	
	N	N	Y	N	N	N	

Reasonable RULES:

Predict SPAM if unknown AND (money OR pills)

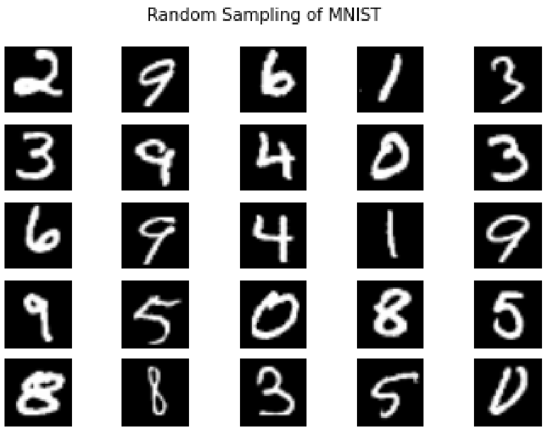
Predict SPAM if 2money + 3pills –5 known > 0



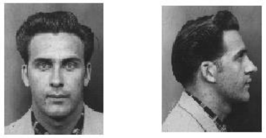
Linearly separable

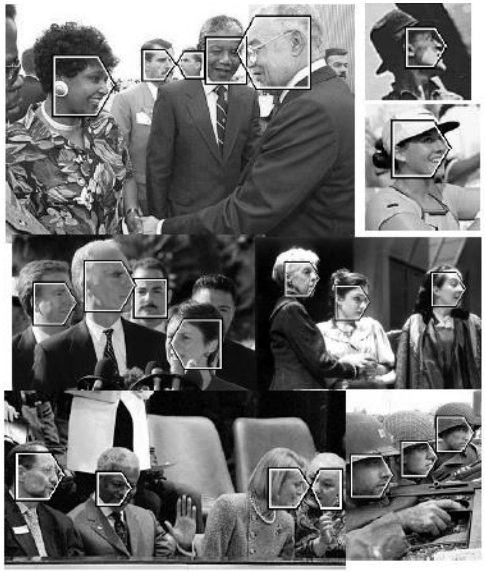
Supervised Classification. Example: Image classification

- Handwritten digit recognition (convert hand-written digits to characters 0..9)



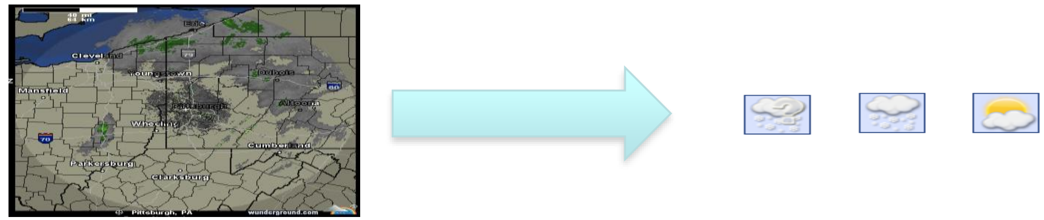
- Face Detection and Recognition





Supervised Classification. Many other examples

- Weather prediction



- Medicine:
  - diagnose a disease
    - input: from symptoms, lab measurements, test results, DNA tests, …
    - output: one of set of possible diseases, or “none of the above”
    - examples: audiology, thyroid cancer, diabetes, …
      - or: response to chemo drug X
      - or: will patient be re-admitted soon?

- Computational Economics:
  - predict if a stock will rise or fall
  - predict if a user will click on an ad or not
    - in order to decide which ad to show

Regression. Predicting a numeric value

Stock market

Weather prediction

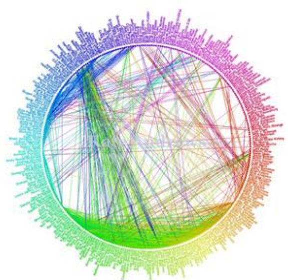
Temperature
72° F

Predict the temperature at any given location

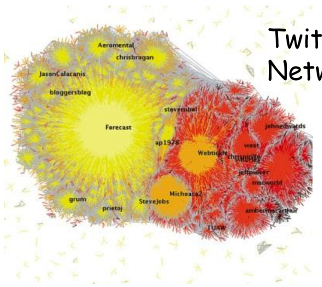
Other Machine Learning Paradigm

Clustering: discovering structure in data (only unlabeled data)

- E.g, cluster users of social networks by interest (community detection).



Facebook network



Twitter Network

Semi-Supervised Learning: learning with labeled & unlabeled data

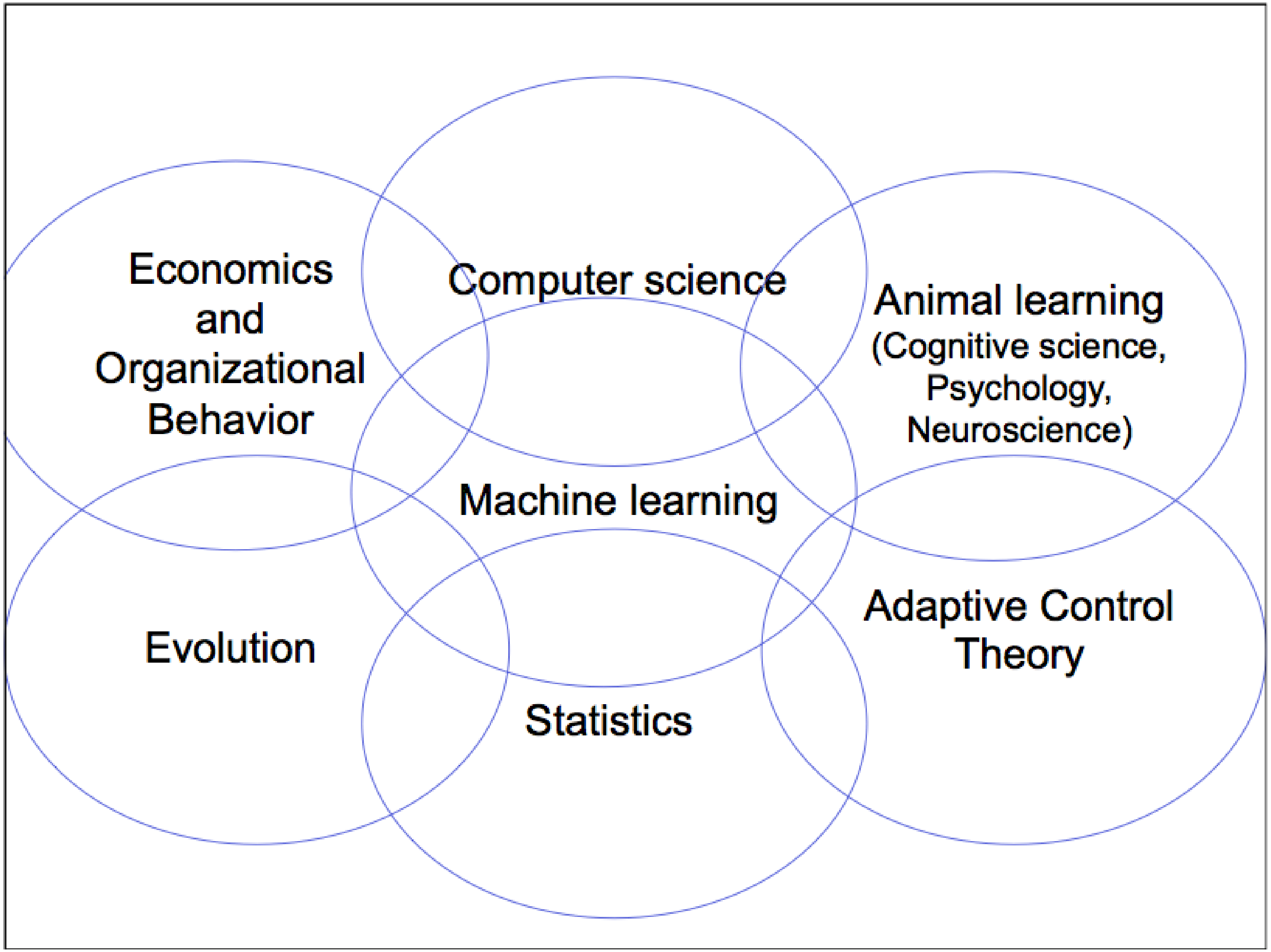
Active Learning: learns pick informative examples to be labeled

Reinforcement Learning (acommodates indirect or delayed feedback)

Dimensionality Reduction

Collaborative Filtering (Matrix Completion), …

Many communities relate to ML



Prerequisites. What do you need to know now?

- You should know how to do math and how to program:
  - Calculus (multivariate)
  - Probability/statistics
  - Algorithms. Big O notation.
  - Linear algebra (matrices and vectors)
  - Programming:
    - You will implement some of the algorithms and apply them to datasets
    - Assignments will be in Python

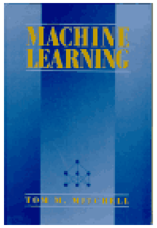
- We may review these things but we will **not** teach them

Source Materials

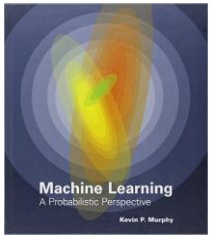
No textbook required. Will point to slides and freely available online material.

Useful textbooks:

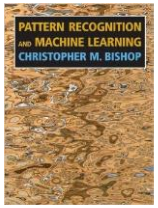
Machine Learning, Tom Mitchell, McGraw Hill, 1997.



Machine Learning: a Probabilistic Perspective, K. Murphy, MIT Press, 2012



Pattern Recognition and Machine Learning
Christopher Bishop, Springer-Verlag 2006



Grading

- 30% for homeworks.

- 25% for midterm
- 40% for final
- 5% for class participation.

- Homeworks:
  - Theory/math handouts
  - Programming exercises; applying/evaluating existing learners
  - Late assignments:
    - Up to 50% credit if it's less than 48 hrs late

Collaboration policy (see syllabus)

- Discussion of anything is ok…
- …but the goal should be to *understand* better, not save work.

- So:
  - *no notes* of the discussion are allowed…the only thing you can take away is whatever's in your brain.

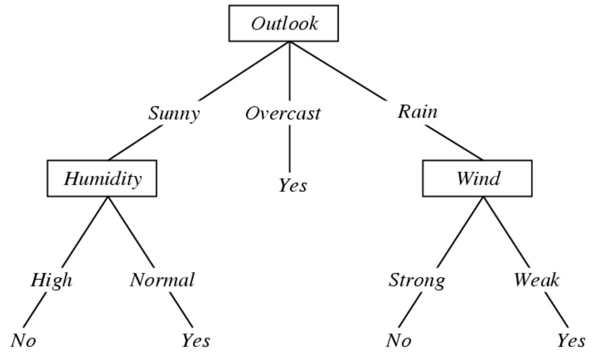
Learning Decision Trees.

Supervised Classification.

Useful Readings:

- Mitchell, Chapter 3
- Bishop, Chapter 14.4

DT learning: Method for learning discrete-valued target functions in which the function to be learned is represented by a decision tree.



Supervised Classification: Decision Tree Learning

Example: learn concept **PlayTennis** (i.e., decide whether our friend will play tennis or not in a given day)

Simple Training Data Set

example

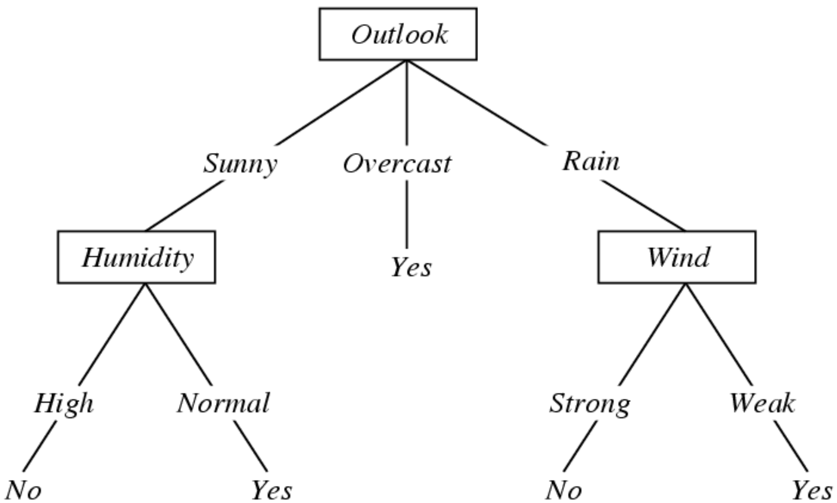
Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

label

Supervised Classification: Decision Tree Learning

- Each internal node: test one (discrete-valued) attribute X_i
- Each branch from a node: corresponds to one possible values for X_i
- Each leaf node: predict Y (or $P(Y=1|x \in leaf)$)

Example: A Decision tree for
f: <Outlook, Temperature, Humidity, Wind> → PlayTennis?



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

E.g., x=(Outlook=sunny, Temperature-Hot, Humidity=Normal,Wind=High), f(x)=Yes.

Supervised Classification: Problem Setting

Input: Training labeled examples $\{(x^{(i)},y^{(i)})\}$ of unknown target function f

- Examples described by their values on some set of features or attributes

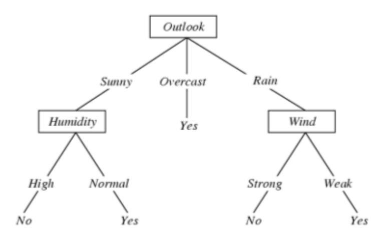
Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

  - E.g. 4 attributes: *Humidity*, *Wind*, *Outlook*, *Temp*
    - e.g., <*Humidity=High*, *Wind=weak*, *Outlook=rain*, *Temp=Mild*>
  - Set of possible instances X (a.k.a instance space)

- Unknown target function $f : X \rightarrow Y$
  - e.g., $Y=\{0,1\}$ label space
  - e.g., 1 if we play tennis on this day, else 0

Output: Hypothesis $h \in H$ that (best) approximates target function f

- Set of function hypotheses $H=\{\ h \mid h : X \rightarrow Y\ \}$
  - each hypothesis h is a decision tree

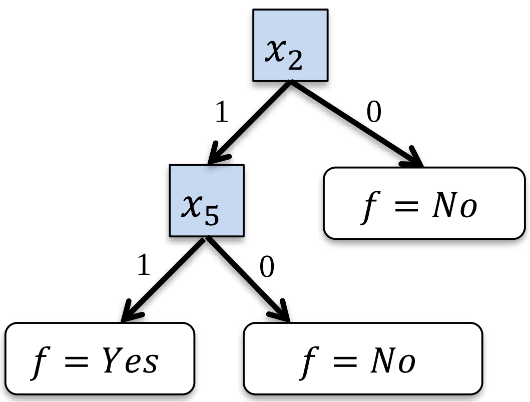


Supervised Classification: Decision Trees

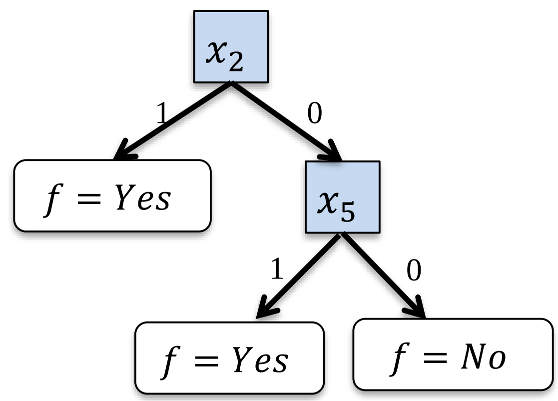
Suppose $X = <x_1, \ldots\ x_n>$
where x_i are boolean-valued variables

How would you represent the following as DTs?

$f(x) = x_2\ AND\ x_5$?



$f(x) = x_2\ OR\ x_5$



Hwk: How would you represent $X_2 X_5 \vee X_3 X_4 (\neg X_1)$?

Supervised Classification: Problem Setting

Input: Training labeled examples $\{(x^{(i)},y^{(i)})\}$ of unknown target function f

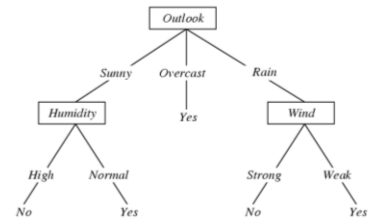
Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- Examples described by their values on some set of features or attributes
  - E.g. 4 attributes: *Humidity*, *Wind*, *Outlook*, *Temp*
    - e.g., <*Humidity=High*, *Wind=weak*, *Outlook=rain*, *Temp=Mild*>
  - Set of possible instances X (a.k.a instance space)

- Unknown target function $f : X \rightarrow Y$
  - e.g., Y={0,1} label space
  - e.g., 1 if we play tennis on this day, else 0

Output: Hypothesis $h \in H$ that (best) approximates target function f

- Set of function hypotheses $H=\{ h \mid h : X \rightarrow Y \}$
  - each hypothesis h is a decision tree



Core Aspects in Decision Tree & Supervised Learning

How to automatically find a good hypothesis for training data?

- This is an **algorithmic** question, the main topic of computer science

When do we generalize and do well on unseen data?

- **Learning theory** quantifies ability to ***generalize*** as a function of the amount of training data and the hypothesis space
- **Occam's razor:** use the *simplest* hypothesis consistent with data!

Fewer short hypotheses than long ones

- a short hypothesis that fits the data is less likely to be a statistical coincidence
- highly probable that a sufficiently complex hypothesis will fit the data

Core Aspects in Decision Tree & Supervised Learning

How to automatically find a good hypothesis for training data?

- This is an **algorithmic** question, the main topic of computer science

When do we generalize and do well on unseen data?

- **Occam's razor:** use the *simplest* hypothesis consistent with data!

- Decision trees: if we were able to find a small decision tree that explains data well, then good generalization guarantees.
  - NP-hard [Hyafil-Rivest'76]: unlikely to have a poly time algorithm

- Very nice practical heuristics; top down algorithms, e.g, ID3

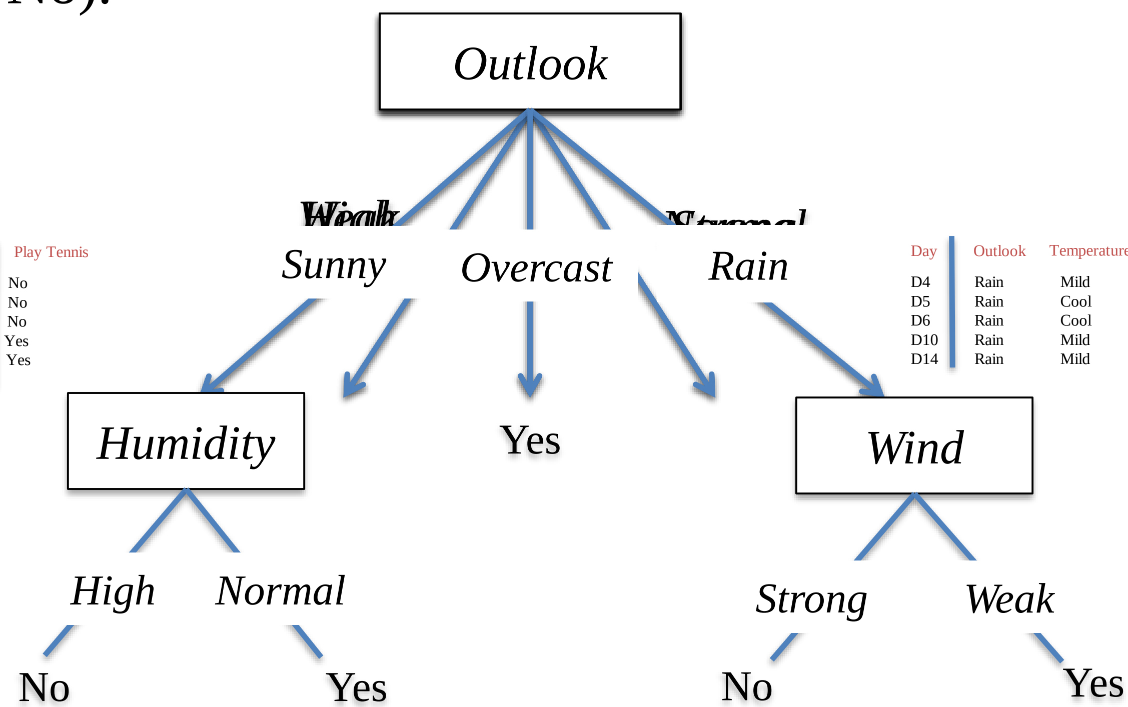
Top-Down Induction of Decision Trees [ID3, C4.5, Quinlan]

ID3: Natural greedy approach to growing a decision tree top-down (from the root to the leaves by repeatedly replacing an existing leaf with an internal node.).

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Algorithm:

- Pick "best" attribute to split at the root based on training data.
- Recurse on children that are impure (e.g, have both Yes and No).



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Key Issues in Machine Learning

- How can we gauge the accuracy of a hypothesis on unseen data?
  - **Occam's razor:** use the *simplest* hypothesis consistent with data! This will help us avoid overfitting.
  - ***Learning theory*** will help us quantify our ability to ***generalize*** as a function of the amount of training data and the hypothesis space
- How do we find the best hypothesis?
  - This is an **algorithmic** question, the main topic of computer science

- How do we choose a hypothesis space?
  - Often we use **prior knowledge** to guide this choice
- How to model applications as machine learning problems? (engineering challenge)