Types of machine learning problems (we'll cover first two).

⊖ Supervised learning :-

The training examples include the desired outputs.

Examples :-

a) SPAM vs HAM, when training set contains emails with the label $\in$ {SPAM, HAM}.

b) Digital images of digits (from 0,... 9) with which digits they are. perhaps labeled by humans.

a) Test :- new, unlabeled examples, guess the output.

- Classification, regression, ...

⊖ unsupervised learning :-

Training data has no desired outputs, and the goal is to find interesting structures in data.

Examples :-

a) Cluster images on the web,
Given new image, find some similar images. Say you saw an (skyline) image of a friend on social network, and don't know where it is. Don't want to appear dumb.

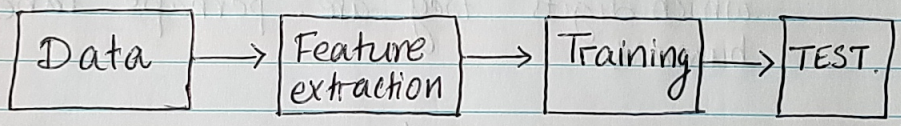
b) Dimensionality reduction / feature extraction / clustering, ...

- Reinforcement learning :-

Players take steps (actions), and obtain rewards. Try to learn ~~games~~ best strategies.

a) Learn to play Backgammon / Chess ....

The ML pipeline :-



F. E. / Processing.

(Step 1) :- very important since data is usually unstructured & messy. Need to decide what is important what is not.

We will start with second step, namely data / examples are given as a set of features.

* MNIST database (search on wikipedia) your friend.

A little more technically:-

Given :- $\langle x_1, f(x_1) \rangle, \ldots, \langle x_n, f(x_n) \rangle$, 'n' labeled examp

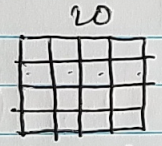
Goal :- Learn what 'f' might be.

> Goal :- Given a new data point x_{n+1}, estimate $\hat{f}(x_{n+1})$.

Test most imptt.

Examples :-

① MNIST dataset :-
curated so each x (features) is a 20×20 = 400 dimensional gray-scale image.



each value $\in \{0, 1, \ldots, 255\}$.

$f(x) \in \{0, \ldots, 9\}$. (60K training & 10K test examples).

② Disease diagnonis :-
x : symptoms, lab-tests (bp, blood-test, etc).
$f(x)$:- which disease...

③

...

Decision Trees:-

- One of the "simplest" models of learning.
  - widely used,
  - practical.
- Random Forests (~ week 8/9) build on such trees
  - winning entry of the NETFLIX prize.

Look:- (-) Chapter 3 of "Machine Learning", Tom Mitchell
(-) CIML - chapter 1,2.

What are decision trees?

- Each internal node test something about an "attribute"
- Each branch corresponds to attribute value
- leaf nodes assign a classification.

Prediction starts @ root node.

$\langle \text{features, value} \rangle = \langle x, f(x) \rangle$.

x - feature vector.

Example:- Play tennis (Mitchell).

Ctrl+.

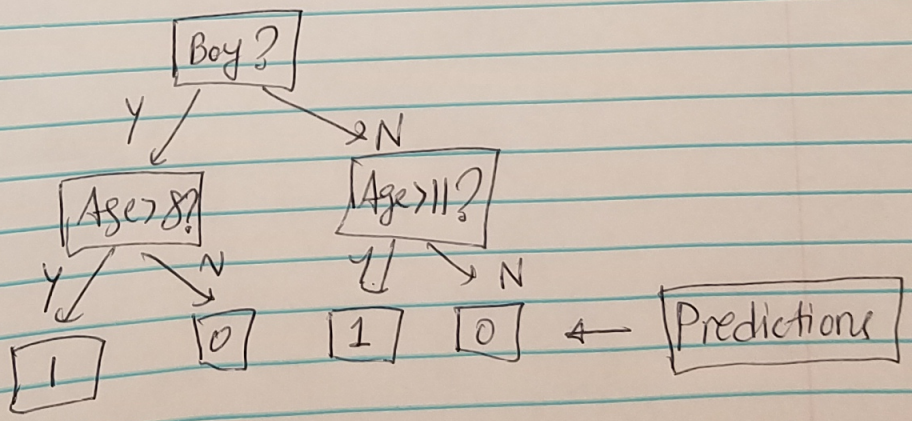
Decision Trees

We looked/introduced DT's with the following example:-

Goal:- predict if the height of a child is at least 55".

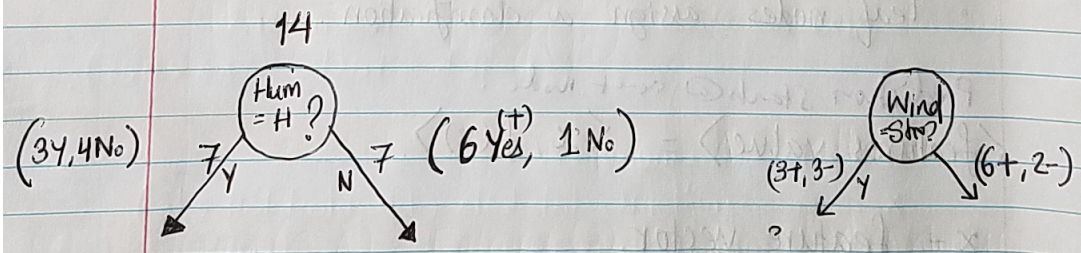
Name	Age	Boy?	Height > 55"
A	14	0	1
B	10	1	1
C	13	0	1
D	8	1	0
E	11	0	0
F	9	1	1
G	10	0	0

The decision tree obtained ~~is~~ considered was:-

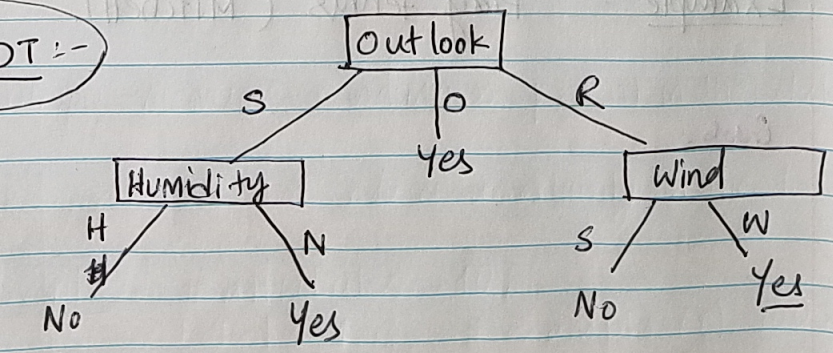


Training Examples :-

Day	Outlook	Temperature	Humidity	Wind	Tennis
1	S	H	H	W	N
2	S	H	H	S	N
3	O	H	H	W	Y
4	R	M	H	W	Y
5	R	C	N	W	Y
6	R	C	N	S	N
7	O	C	N	S	Y
8	S	M	H	W	N
9	S	C	N	W	Y
10	R	M	N	W	Y
11	S	M	N	S	Y
12	O	M	H	S	Y
13	O	H	N	W	Y
14	R	M	H	S	N



Example DT :-



* how to incorporate continuous values?

- Can incorporate continuous values ~~with~~ also.
  eg. Humidity ≥ 75 (N) Humidity < 75 (Y).
  is very common.

⊖ How many decision trees are possible? with 'n' binary attributes.

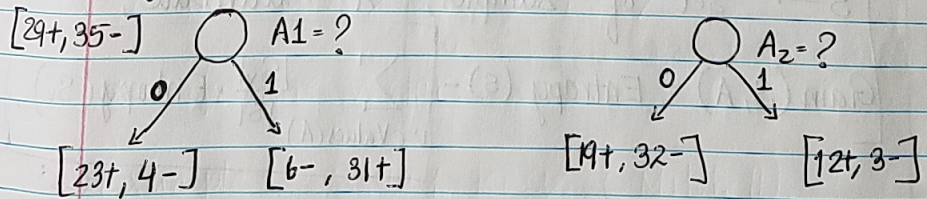
⊕ What is a good-decision tree?

Decision Tree Approach :-

LOOP :-

1. A ← the "best" decision attribute for next node.
2. Assign A as decision attribute for node
3. For each value of A, create new descendant of node
4. Sort examples to leaf nodes
5. If training examples perfectly classified, STOP, else iterate over leaf nodes.

Binary attributes :-



⊗ Which attribute is better you think?

⊗ When to declare a node as leaf?

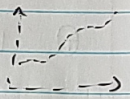
⊗ small, large, balanced tree?

best attribute
Principles underlying the tree, criterion :-

- Each attribute divides data into subsets
- - Make each subset as pure as possible.
- Prefer simple trees.

(np?) (William of Occam)

Occam's Razor :- Simplest model that explains should be preferred.
(Overfitting is a problem).



Criterion for choosing attributes :-

⊖ Information gain. (ID3 algorithm) (Quinlan' 1975). || King Ronald ∧ King Kong M.

Based on entropy / information theory.

what is entropy?

$$H(p) = \sum_i p_i \log\left(\frac{1}{p_i}\right) \longrightarrow \text{(example w/ labels).}$$

$S \rightarrow$ set of samples, $A \rightarrow$ attribute,

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)$$

(LS)

Entropy measures randomness present in the data.

Choose attribute that reduces uncertainty in the labels.

Example :- # attributes = 10,
Label $\in \{0, 1\}$, # examples $(n) = 100$.

Assume all attributes are binary (take two values).
If one of the attributes perfectly classifies the positive and negative examples, just use it (why?)

⊛ In the discussion session:-
Review of probability, and entropy (information).

⊛ Chapter 3, Mitchell, works this example out.

⊖ Gini index: another measure of randomness.

$$Gini(p) = 1 - \sum_i p_i^2$$

Back to playing tennis:-

$$H(S) = \frac{5}{14}\log\frac{14}{5} + \frac{9}{14}\log\frac{14}{9} \approx 0.940$$

$$Gain(S, Humidity) = 0.940 - \frac{7}{14}\cdot\left(\frac{3}{7}\log\frac{7}{3} + \frac{4}{7}\log\frac{7}{4}\right) - \frac{7}{14}\left(\frac{1}{7}\log 7 + \frac{6}{7}\log\frac{7}{6}\right)$$

$$= 0.151$$

$$Gain(S, Wind) = 0.940 - \frac{8}{14}\left(\frac{2}{8}\log 4 + \frac{6}{8}\log\frac{8}{6}\right) - \frac{6}{14}\left(\frac{3}{6}\log\frac{6}{3} + \frac{3}{6}\log\frac{6}{3}\right)$$

$$= 0.048.$$

$$Gain(S, Outlook) = 0.246.$$

$$Gain(S, Temp) = 0.029.$$

...

Suppose there are many decision trees consistent with the examples. Which of them is picked up by the ID3 algorithm?

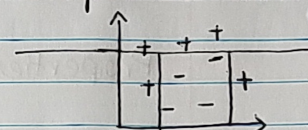
- shorter trees are preferred.
- more info near root.

Lecture 8

(-) Remind Piazza sign-up / wait list announcement. (tennis example up)

Recap :- Decision trees, simplest supervised learning.

(-) Straight lines.



Key steps :- LOOP ON :-

- Pick the best attribute for splitting.

(-) NOTE about continuous variables

until all training data is perfectly classified.

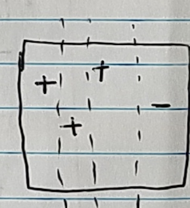
* * *

- How to find the best attribute?
- What is overfitting in decision trees?

Principles underlying best attribute criterion.

(go to page 10).

Impurity Criterion :-



① Simplest :- 0/1 loss function.

② Bernoulli Variance :-

③ Information gain :- (ID3, Quinlan 1975).

- go to page (19).
- Tennis example from (p 8) write on board.
- do computation of page (11), back to play tennis. ($11\frac{1}{2} \rightarrow$ end).

- Gini index. $G(P) = \sum p_i^2$.

Properties of greedy attribute selection:-

① Every intermediate step is a decision tree.
② Cannot reconsider different higher order splits.

→ if a tree has not achieved perfect classification, assign new labels by:-
① assign to majority of labels at the leaf nodes
② randomize to the distribution of labels at the leaf

Issues ① missing values
② Overfitting in DT:-

- Stopping criterion:- all nodes are "pure".
  - what about noisy data?
    if all features are same but labels different?
    (nothing can be done)

+

Prove:- if this is not the case, show that
- there is a decision tree that yields the perfect classification.

So, we do not stop until perfect classification

. might have one example at each node.

- looking up @ examples - bad at generalization.

keeps fitting better on training data, poor on test.
(pic @ P12)

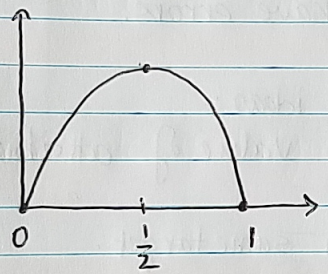
- Entropy of a distribution.

$P \longrightarrow$ distribution on x

p_x = probability of x.

$$H(P) \triangleq \sum_{x \in X} p_x \log\left(\frac{1}{p_x}\right).$$

- $x = \{-1, 1\}$, binary distributions. $H(P) = h(p) = p \log\frac{1}{p} + \bar{p} \log\frac{1}{\bar{p}}$



- Entropy of ~~examples~~/ labels

Recall that we wanted to reduce the uncertainty of labels.

S = a set of examples, A = an attribute

S_+ = +ve examples in S, (eg, is age > 10). (age $\in [0,10], [0,\infty], \ldots$)

$a_1, a_2, \ldots, a_K \leftarrow$ possible answers to 'A'.

$$H(S) \triangleq \frac{|S_+|}{|S|} \log \frac{|S|}{|S_+|} + \frac{|S_-|}{|S|} \log \frac{|S|}{|S_-|} \quad \text{—①}$$

$S_{a_1} = \{$samples in S with attribute value $a_1\}, \ldots$

$$IG(S,A) = \underbrace{H(S)}_{\text{entropy of data}} - \underbrace{\sum_{a_i} \frac{|S_{a_i}|}{|S|} H(S_{a_i})}_{\text{entropy of groups.}}.$$

<choose attribute with highest information gain>

Avoiding overfitting :- Most

① Stop splitting when few samples.

② Grow and then post-prune.

(-) validation test :- can be used to prevent overfitting.
↓
introduce here.
(hold-out sets)

⊖ It is ok for trees to have errors.

③ Missing values :- Most used ideas :-
(i) Assign most common value of attribute
(ii) " " " " "
among examples with same target.
(iii)

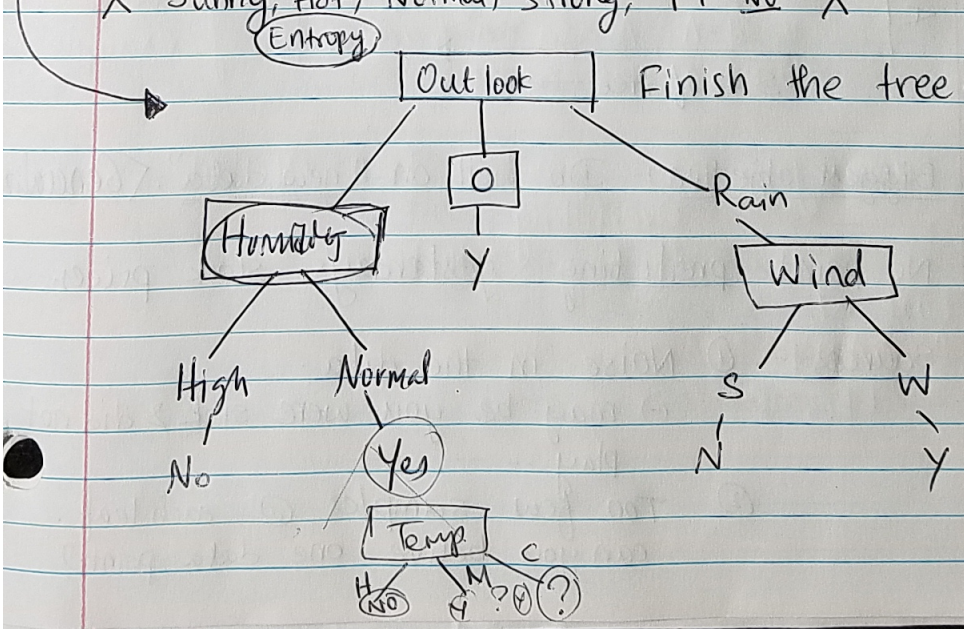
Logistics: ① Class Open.
② Register on Piazza
③ CMS- we are learning.
(④ Self grading).
⑤ Mert Kaggle, Fri 10AM @ 300 Rhodes.

Recap: Decision Trees → "recursively divides samples".
(-) how to find best attributes?
(-) Attributes that make nodes pure.
• Entropy.

Issues in decision trees learning

- Overfitting
  (-) how to get shorter trees?
- Continuous / not
- Missing values.
- Noisy data.

X Sunny, Hot, Normal, Strong, PT = NO X



Why overfitting happens :-

Too much emphasis on doing well on the training examples.

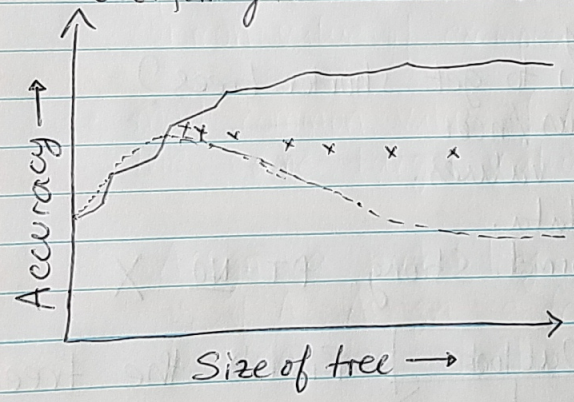
Def^n of overfitting. (in ML not only DT's) :-

We say that 'h' is overfitting if there is some 'h'' such that :-

① On training data error(h) < error(h'),

② On the test data error(h') < error(h).

- overfitting in decision trees :-



Biggest objective :- Do well on "new data". (Generalize)

No point predicting yesterday's stock prices.

Sources :- ① Noise in the data.
(-) may be you were sick & did not play ...
② Too few samples @ each leaf.
can you belive one data point?

How to avoid overfitting

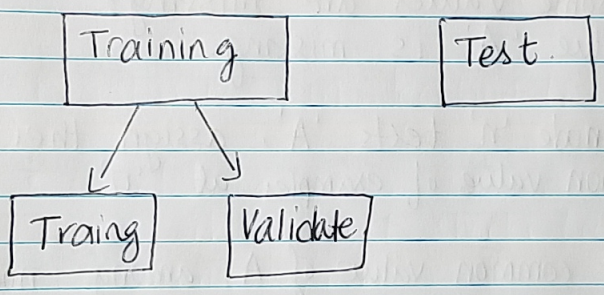
① In general.
② In decision trees.

Two approaches to ~~stop~~ overfitting:-

① Stop growing the tree earlier (~~than~~ b4 pure leaves).

② Grow the tree fully & then post-prune.

- Validation / Hold-out set.



Growing.

① Stop growing tree when not enough samples at each node (say >4).
② ~~Stop when~~ Keep growing the tree until per formance on validation set goes up.

Pruning.

Grow full tree.

① Reduced error pruning.
- evaluate performance ~~of~~ on validation set of pruning each node (+ those below it)
② - Greedily remove the one's that most improve validation accuracy.

Rule post pruning.

- Get full decision tree.
- Convert the decision to a set of rules.
- Prune each rule by removing conditions that improve ~~estim~~ validation accuracy.
- Sort rules using estimated accuracy.
- classify new instances using sorted sequence.

Missing Values. (Probabilistic values :)

What if some values are missing?
Say value 'A' is missing.

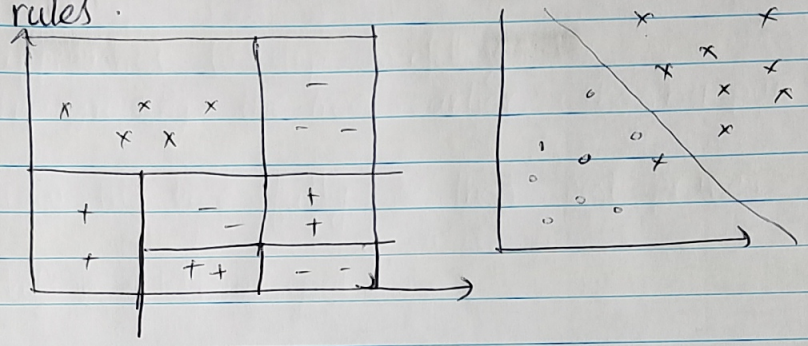
- If node 'n' tests 'A', assign the most common value of examples at 'n'.
- most common value of A, among nodes with the same target value.
- Assign a probability p_i to each value v_i of A. Assign a fraction p_i of example to each descendant.

Summary :- DT, ~~growth~~, best attribute
Overfitting,
missing attributes.

Look at :-
" CART :-
ID3 :-

Generative vs Discriminative classifiers:-

D:- Look at the features, divides points / design rules.



break the "feature space" into regions.

namely, space of all possible features.

Example:- 'A' 'N' $\longrightarrow$ $\underbrace{\{\quad\}}_{\text{Age} \in \mathbb{Z}^+ \{1,\ldots,20\}} \times \{0,1\}$.

Depending on where the test data lies, assign appropriate values.