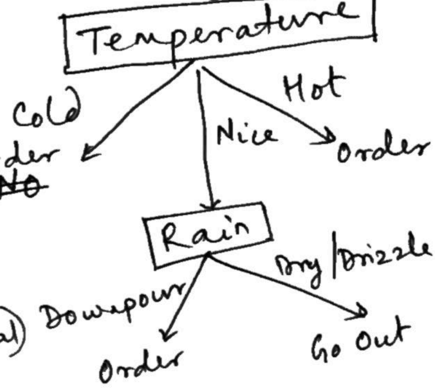
DECISION TREES

Represents hierarchical decision making process.

Dinner: Go Out or Order?



Each node represents an <u>attribute</u> or feature & each branch represents possible <u>value</u> for that attribute (can be non-numerical)

<u>How to decide hierarchy?</u>

Nodes with <u>lowest entropy</u> (best classification) is at the top/higher level.

A: Attribute S: Total collection of examples

SA_v: Subset of S for which Attribute A has value 'v'.

H_S: Entropy of S

H_{ASv}: Entropy of SA_v.

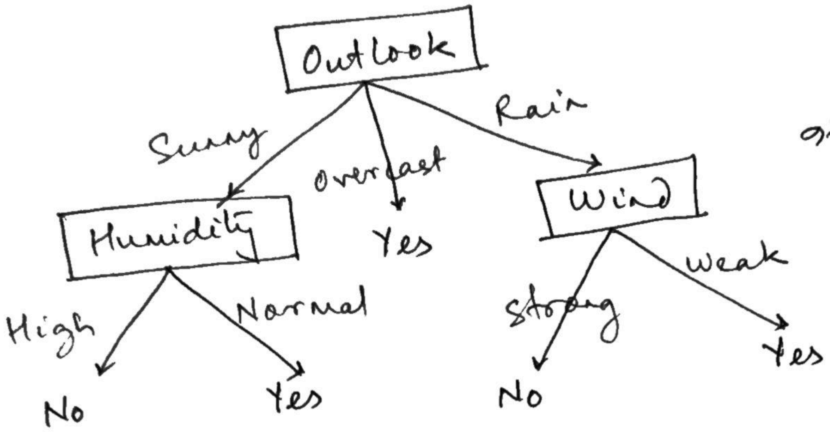
H_A: Entropy of A.

$$H_{SA_v} = \sum - p_i \log_2 p_i$$

$$H_A = \sum_{v \in A} \frac{|SA_v|}{|S|} H_{SA_v}$$

<u>Information Gain</u>: $G(S,A) = H_S - H_A$

ML by Tom Mitchell. [Play Tennis]



ID3:
Iterative Dichotomiser 3

Gini Index/Impurity:

$$G_{AS_v} = 1 - \sum p_i^2$$

$$G_A = \sum \frac{|SA_v|}{|S|} G_{AS_v}$$

Works almost similar to entropy measure but faster to calculate.

Overfitting

- Pre-pruning: stop split beyond a significance threshold
- Post-pruning: based on validation set [better than pre-prune since hard to decide threshold]

Missing Features

Assign most common value or using some probability measure.

ADVANTAGE OF TREES

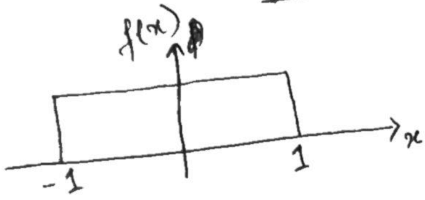
- Easy to interpret
- Easily handle mix data [Symbols | Numbers]
- Robust to outliers
- Scales well to large datasets
-

DISADV.

- Low bias & high variance
- Unstable due to hierarchical nature [small change in data can lead to large change in tree]

Addressed by Random Forest.

RANDOM FOREST



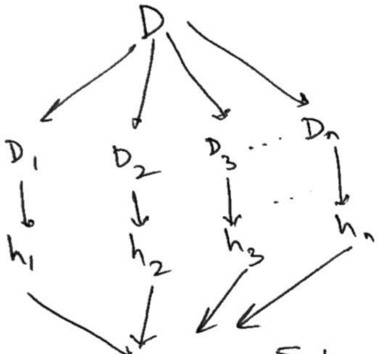
$-1 \le x \le 1 \qquad \mathbb{E}[X] = 0$

$X_1, X_2, X_3, \ldots, X_n$ iid rv.

$Y = \frac{X_1 + X_2 + \ldots + X_n}{n} \longrightarrow \mathbb{E}[X]$ as $n \to \infty$

Law of Large Numbers

standard method of reducing variance in prediction.



$h = \frac{1}{n} \sum_i h_i$

→ Bootstrap Aggregating

BAGGING : very effective way to reduce variance.

Each D_i has same size as D
[drawn randomly, can have same data point ~~to~~ multiple times]

for each split, use a subset of features.

RANDOM FOREST : Bagging + Decision Trees.

~~For each Di, also choose features randomly.~~

Very resilient to "curse of dimensionality"
gets slower but still works very well.

only two hyper-parameters
n & $\sqrt{D}$.

But looses interpretability.

Each decision tree is a weak learner
Bagging makes it a strong learner.

ENSEMBLE LEARNING ~~Loss~~

For ML methods with ~~poor~~ high bias, use BOOSTING.

STACKING

Bagging runs in parallel
Boosting is serial or sequential
[use training accuracy to assign higher weight to misclassified data & re-train. Do in loop for few times]

Logistic Regression vs. SVM vs. Random Forest

LR	SVM	RF
① works best with linearly separable data	Can handle nonlinearity by using kernels.	Inherently nonlinear
② Easier to interpret results. Also gives probability estimates	Hard to interpret results no prob.	~~Same as SVM~~ hard to interpret but gives prob.
③ Usually fast	faster than LR	Usually very slow
④ works better with balanced data		Can easily handle unbalanced data
⑤ Has problems with high dimensional data (lots of features)	same as LR	Easily handles curse of dimensionality (feature subset used)
⑥ can't handle outliers	Handles outliers better	Can handle noisy data
⑦ better for 2-class	better for 2-class	Better for multi-class classification

BEST PRACTICE — Try all 3, tune hyper-parameters & choose the one that works best.

DECISION TREES

Data fragmentation: Continuous partitioning leaves less data for lower-level nodes leading to weak statistical support.

→ Can lead to multiple mis-classifications.

→ General problem in rule based learning.

NP - Complete:

A problem G in NP is NP-complete if every other problem in NP can be transformed (or reduced) to G in polynomial time.

obviously, G is also NP-hard.
but NP-hard does not imply NP-complete (eg. halting problem)

→ Decision Trees are NP-complete.
If you can solve Decision Tree problem in polynomial time, it would prove P=NP.

→ ID3 is a greedy heuristic algorithm
does best fit search for locally optimal entropy values.
(no looking back) approximates the globally optimal solution.

→ Decision Tree : No real learning } should it be called
purely rule based } ML/AI?

→ In RF, choose subset of features in each tree, to avoid correlations between trees.

→ ~~IB~~ ID3 was originally for categorical values.
C4.5 allowed numerical values.
CART has binary ~~classification~~ tree & allows regression.

Gini represent the expected error resulting from labelling instances in the leaf randomly.

$p(1-p)$
↳ coin toss variance.

Thresholding & Discretization.

Unsupervised thresholding : mean/median
Supervised " : lower convex hull
need to know number of 'bins' apriori
Requires careful data analysis
(mean/median may not be good classifiers)