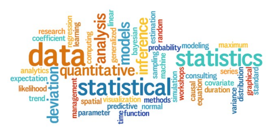
Applied Data Science

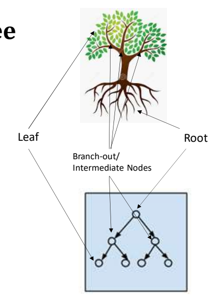
Session 25: Decision Tree

Dr. Soharab Hossain Shaikh



1

Tree



Root node – keeping track of the entire tree/ starting point.

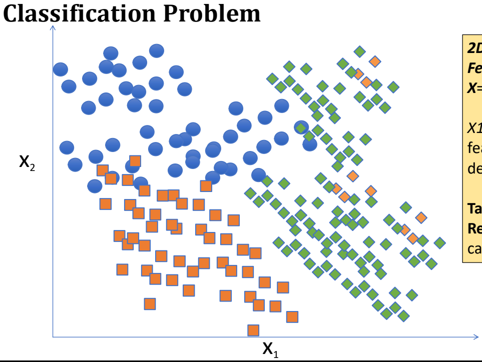
Intermediate node – from here we branch out to different parts of the tree.

Leaf nodes – terminal points – we can not move any further along the tree once a leaf is reached.

A tree could be of arbitrary depth.

2

Classification Problem



2D Feature space
Feature vector,
X*=(X1, X2)*

X1 and *X2* are two features/predictors/independent variables

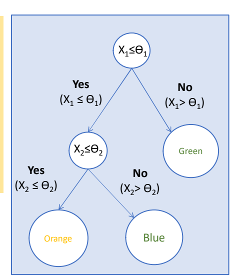
Target/Dependent/ Response: class categories/labels

3

Decision Making with a Tree

Key idea:

Segment the predictor/feature space into sub-regions based on thresholds.

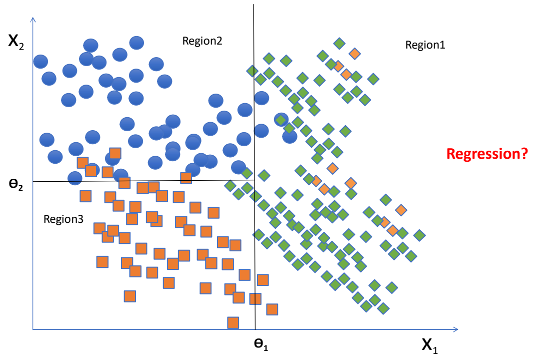


Region1 (Green diamonds)
=> $X_1 > \Theta_1$

Region2 (Blue circles)
=> $X_2 \geq \Theta_2$

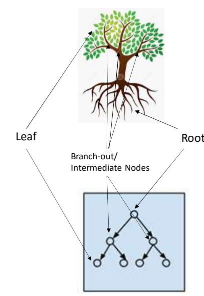
Region3 (Orange squares)
=> $X_2 < \Theta_2$

4



5

Decision Tree



Root node – keeping track of the entire tree/ starting point (making the first decision on some feature- contains all the data points of the training set)

Intermediate node – from here we branch out to different parts of the tree (**decision making based on a feature and a threshold – contains a subset of data points**).

Leaf nodes – terminal points – we can not move any further along the tree once a leaf is reached (**decision/grouping – generally pure subset**).

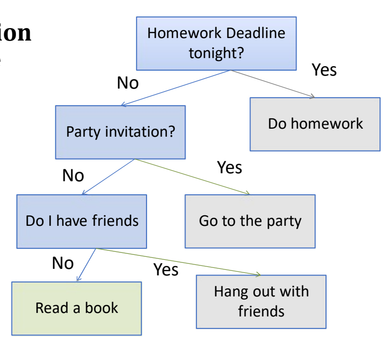
6

Example of Decision Making with Tree

What would you do tonight?

Decide amongst the following:

- Finish homework
- Go to a party
- Read a book
- Hang out with friends



7

Decision Tree

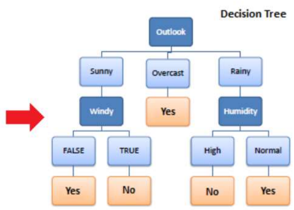
- Decision tree builds classification or regression models in the form of a **tree structure**.
- The tree breaks down a dataset into smaller and smaller subsets by splitting on features.
- The root and other internal nodes represent a decision made on a certain feature.
- A child node is split further based on other features. In the process the decision tree is incrementally developed.
- This process continues in a **top-down, greedy, recursive** manner until a ***terminating condition*** is satisfied.
- The final result is a tree with **decision nodes** and all the data split at the **leaf nodes**.
- Decision trees can handle both categorical and numerical data.

8

Decision Trees

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Predictors: Outlook, Temp, Humidity, Windy — Target: Play Golf



- A decision node (e.g., Outlook) has two or more branches, three in this case, (e.g., Sunny, Overcast and Rainy).
- Leaf node (e.g., Play) represents a classification or decision.
- The topmost decision node in a tree which corresponds to the best predictor called **root node**.

9

Decision Tree : Splitting a Node

- **Which feature to select for partitioning a node?**
- Select the most discriminating feature

 # Classification Problem : The feature that ensures best possible split

>> Highest information gain – feature that generates least impure nodes after splitting. Least impurity means least uncertainty.

>> How to measure that?

>> **Gini Impurity** – a measure of impurity (mixed data items).

>> Decrease in total **entropy** - after splitting compared to the parent node – results in **Information Gain**

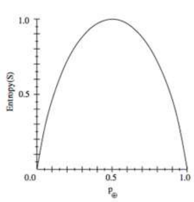
Both Gini index and Entropy is a measure of how heterogeneous the data items are in a node of the tree.

More **homogeneity** implies less mixing of different categories – means less impurity and less uncertainty.

 # Regression Problem: A feature that reduces variance or MSE of the training data the most

10

Entropy



log of a number < 1 is negative, $0 \le p \le 1$, $0 \le entropy \le 1$ for Binary classification

> Entropy measures the uncertainty or randomness present in of a collection of examples.

> In other words it is a measure of heterogeneity/impurity of a collection of items.

> It depends from the distribution of the random variable *p.*

Measured as:

$$\textbf{Entropy (S)} \equiv -\sum_{i=1}^{c} p_i \log_2 p_i$$

Larger value for entropy implies more uncertainty/randomness/impurity.

11

Entropy

- Entropy measures the amount of information in a random variable.
- For a two-class/binary classification (two possible values of a random variable) – more than one class – known as **Cross Entropy**.

If a bag contains a few red and blue balls and we want to randomly pick one ball from the bag (no selection bias), then the output variable is a random variable with two possible discrete outcomes denoted by X = {red, blue}

Entropy of the system can be represented as:

$$H(X) = -p \log_2 p - q \log_2 q$$

p = probability that a red ball will be picked up

q = probability that a blue ball will be picked up

For classification in ***C*** classes

$$H(X) = -\sum_{i=1}^{C} p_i \log_2 p_i = \sum_{i=1}^{C} p_i \log_2 1/p_i \qquad X = \{1, \ldots, C\}$$

Example: rolling a die with 6, equally probable, sides

$$H(X) = -\sum_{i=1}^{6} 1/6 \log_2 1/6 = -\log_2 1/6 = \log_2 6 = 2.58$$

12

Entropy in Binary Classification

- S is a collection of training examples containing two categories *positive(+)* and *negative(-)*
- p the proportion of positive examples in S
- q the proportion of negative examples in S

$Entropy\ (S) \equiv -p\ log_2\ p - q\ log_2\ q$ $\quad [0\ log_2 0 = 0]$

$Entropy\ ([14+, 0-]) = -14/14\ log_2\ (14/14) - 0\ log_2\ (0) = 0$ $\quad [\ log_2 1 = 0]$

$Entropy\ ([9+, 5-]) = -9/14\ log_2\ (9/14) - 5/14\ log_2\ (5/14) = 0.94$

$Entropy\ ([7+, 7-]) = -7/14\ log_2\ (7/14) - 7/14\ log_2\ (7/14) =$

$= 1/2 + 1/2 = 1$ $\quad [log_2 1/2 = -1]$

13

Information Gain as Entropy Reduction

- *Information gain* is the *expected* reduction in entropy caused by partitioning the examples on an attribute.
- The higher the information gain the more effective the attribute in classifying training data.
- Expected reduction in entropy knowing A

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

$Values(A)$ possible values for A

S_v subset of S for which A has value v

14

Example: Information Gain

Windy	Play Golf
False	No
True	No
False	Yes
False	Yes
False	Yes
True	No
True	Yes
False	No
False	Yes
False	Yes
True	Yes
True	Yes
False	Yes
True	No

- Let, $Values(Windy) = \{\text{True, False}\}$
  - $S = [9_{Yes}, 5_{No}]$
  - $S_{True} = [3_{Yes}, 3_{No}]$
  - $S_{False} = [6_{Yes}, 2_{No}]$
- Information gain due to knowing $Wind$:

$Gain(S, Windy)$

$= Entropy(S) - 6/14\ Entropy\ (S_{True}) - 8/14\ Entropy(S_{False})$

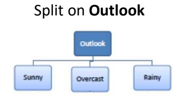
$= 0.94 - 6/14 \times 1 - 8/14 \times 0.811 \ = 0.048$

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

15

Which feature to split on?

Split on **Outlook**



Entropy for c - classes

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
$= -(0.36\ \log_2 0.36) - (0.64\ \log_2 0.64)$
= 0.94

Entropy of PlayGolf before splitting

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

16

Which feature to split on?
Split on Outlook

Information Gain: Difference between the entropy of the parent node (T - PlayGolf) (before splitting) minus the total entropy of the children nodes after splitting on a feature (X - Outlook)

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

17

Which feature to split on?
Split on Temp

Compute Information gain w.r.t every feature in the dataset.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

18

Which feature to split on?
Split on Humidity

Compute Information gain w.r.t every feature in the dataset.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

19

Which feature to split on?
Split on Windy

Compute Information gain w.r.t every feature in the dataset.

Choose the feature that results in the Highest Information Gain.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

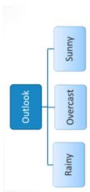
20

Select Outlook

		Play Golf	
★		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

Out of all the four features, **Outlook** results in the Highest Information Gain.

Therefore, it is chosen as the feature to split on at the root.



Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

21

Leaf: Pure Nodes are not Split Further

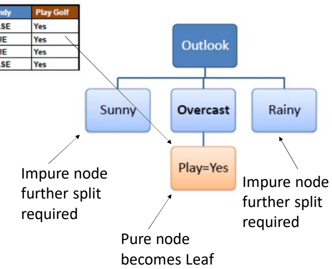
A branch with Entropy 0 is a leaf node.

After splitting on Outlook, three children nodes are created corresponding to the three categorical values of the feature Outlook.

The node **Overcast** contains samples with Play=Yes (pure node). This is a leaf node.

Further splitting can be on other nodes (Sunny and Rainy) as they create impure nodes.

Temp	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes

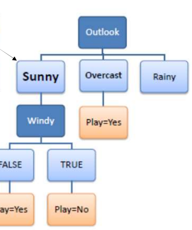


22

Next, Select Windy *from* Sunny

Select the next feature to split on repeating the same process and selecting based on Highest Information gain.

Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



$Values(Windy) = \{\text{True, False}\}$

- $S = [3_{\text{Yes}}, 2_{\text{No}}]$ ← After splitting on ***Sunny***
- $S_{True} = [0_{\text{Yes}}, 2_{\text{No}}]$
- $S_{False} = [3_{\text{Yes}}, 0_{\text{No}}]$

- Information gain due to knowing *Wind* condition:

$Gain(S, Windy)$

$= Entropy(S) - Entropy(S, Windy)$

$= -3/5 \log(3/5) - 2/5\log(2/5) - 2/2 \log(2/2) - 3/3\log(3/3)$

$= 0.966$

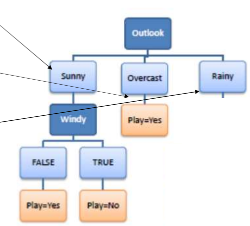
Other features ***Temp***. and ***Humidity*** will result in **impure subset** resulting **less information gain**.

23

Which feature to select *from Rainy?*

What are the options left?

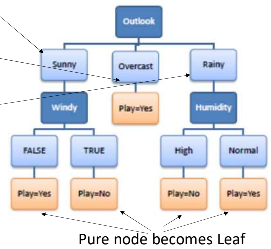
Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes



24

Next, Select **Humidity** *from* ***Rainy***

Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes



Other features ***Temp***. and ***Windy*** will result in **impure subset** resulting **less information gain**.

25

Decision Tree – Decision Rules

A decision tree can easily be transformed to a set of rules by mapping from the root node to the leaf nodes one by one.

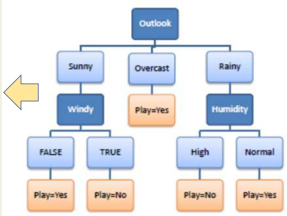
R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R_3: IF (Outlook=Overcast) THEN Play=Yes

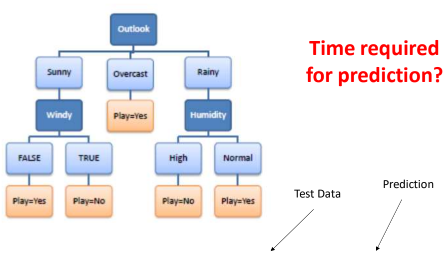
R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes



26

Prediction by Decision Tree



Time required for prediction?

$\langle$ *Outlook=Sunny, Temp=Hot, Humidity=Normal, Windy=True, PlayGolf=No* $\rangle$

27

Gini – Measure of Impurity of a Set

Gini impurity for a set of items with $\boldsymbol{J}$ classes represented by index $\boldsymbol{i} \in \{1,2,..., J\}$ and let $\boldsymbol{p_i}$ be the fraction of items labeled with class label $\boldsymbol{i}$ in the set.

Gini impurity

$$G = \sum_{i=1}^{J} p_i(1-p_i) = 1 - \sum_{i=1}^{J} {p_i}^2$$

> *Impure set* – different types/categories of items/objects are mixed together.

> G takes small values when p_i is small or close to 1.

> Gini index is **zero** for a set containing a single type/category of item (no mixing). In such a case $\boldsymbol{p_i}$ is 1.

28

Gini Index

Example: A bag contains a total of 10 balls - 5 red, 2 blue and 3 green

Compute Gini Impurity:

red:

5/10*(1-5/10) = 0.25

blue:

2/10*(1-2/10) = 0.16

green:

3/10*(1-3/10) = 0.21

G=0.62 (impurity of the collection/bag)

29

Misclassification

Example:

5 red, 2 blue and 3 green

Proportions p_k for each class k are: 5/10, 2/10, 3/10

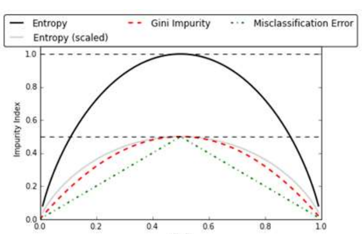
$\max_k(p_k)$ = 5/10 = 1/2

Classification error

E = 1-1/2=1/2

30

Gini Impurity, Entropy, Misclassification Error



Note: Misclassification Error is not used in Decision Trees

Graph valid only for **Binary Classification**

31

Decision Tree: Stopping Condition

- **When to stop building the tree?**

- Keep splitting impure nodes as long as further splitting improves the criteria (reduces gini impurity or reduces entropy and increases information gain).
- Stop splitting a node (even if it is not pure) if further splitting does not improve Information gain.
- Grow the tree until all the leaf nodes become pure (no mixing of different categories of items) – unconstrained. Till all the samples are correctly classified (classification) or the total MSE reduces below a predetermined threshold (regression).
- Grow the tree till a specified maximum depth.
- Till any leaf node contains a minimum number of samples (stop when not satisfied even if the leaf is still impure).

32

Dealing with Continuous-valued Attributes

- So far discrete values for attributes and for outcome.
- Given a continuous-valued attribute *A*, dynamically create a new attribute A_c
  A_c = True *if* $A < c$, False *otherwise*
- How to determine threshold value *c* ?
- Example. *Temperature* in the *PlayGolf* example
  - Sort the examples according to *Temperature*

Temperature	40	48	|	60	72	80	|	90
PlayGolf	No	No	*54*	Yes	Yes	Yes	*85*	No

  - Determine candidate thresholds by averaging consecutive values where there is a change in classification: (48+60)/2=54 and (80+90)/2=85
  - Evaluate candidate thresholds (attributes) according to information gain. The best is $Temperature_{>54}$. The new attribute competes with the other ones.

33

Decision Tree

Advantages

1. Simple, intuitive and fast in processing and effective – less memory required to save the tree after training compared to the no. of features and volume of training data.
2. Does well with noisy data and missing data (noise has zero information gain) – automatic feature selection therefore performs well if there are irrelevant and redundant features.
3. Handles numeric and categorical variables both
4. Explicit feature scaling is, generally, not required.
5. Interpretable results - does not required mathematical or statistical knowledge
6. No explicit assumption of a particular form of relationship between the independent and dependent variables unlike linear models (e.g. linear relationship between predictors and response variable etc.)

34

Decision Tree

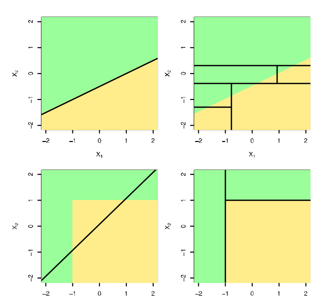
Disadvantages

1. Often biased towards splits or features that have large number of levels (information gain favours features with many values).
2. May not be optimum as modelling some relations on axis parallel basis is not optimal
3. The search process is Greedy – optimal solution not guaranteed.
4. Instability due to small changes in training data can result in large changes to the logic/decision boundaries
5. Large trees can be difficult to interpret (problem of overfitting)
6. Difficulty in approximating non-axis aligned boundaries (for numeric data).

35

Comparison to a Linear Model

- Axis aligned partition of the feature space.
- Takes many steps to approximate a linear decision boundary.
- There are chances of overfitting with every new step.



36

Regularizing Decision Tree to control Overfitting

37

Bias Variance Decomposition

Bias: part of the error caused by bad model (assumptions)

Variance: part of the error caused by the data sample (too much dependence on the training data)

Bias-Variance Trade-off: algorithms that can easily adapt to any given decision boundary are very sensitive to small variations in the data and vice versa

Models with a low bias often have a high variance - e.g., nearest neighbor, unpruned decision trees

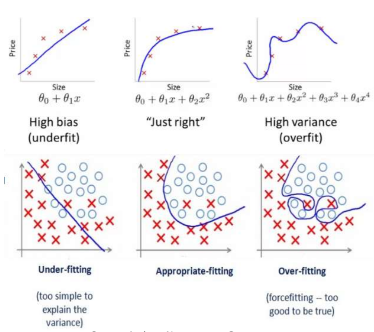
Models with a low variance often have a high bias - e.g., decision stump, linear model

38

Model Complexity

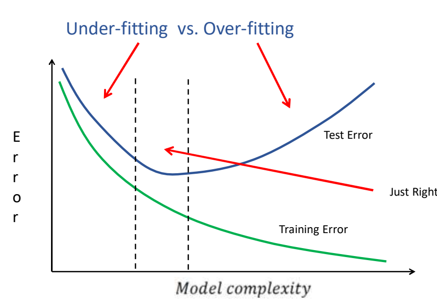
Bias-Variance Tradeoff

Over-fitting and Under-fitting



39

Model Complexity



40

Splitting Data: Importance of Validation Set

- Training Set – data at out disposal to make use of. (80/70%)
- Validation Set – Detect Overfitting & Hyper-parameter tuning (part of training data kept aside)
- Test Set – make prediction – check for model performance (20/30%)

41

Overfitting

- Building trees that "adapt too much" to the training examples may lead to "overfitting".
- Consider error of hypothesis *h* over
  - training data: $error_D(h)$ [empirical error]
  - entire distribution *X* of data: $error_X(h)$ [expected error]
- Hypothesis *h* *overfits* training data if there is an alternative hypothesis $h' \in H$ such that

  $error_D(h) < error_D(h')$ and

  $error_X(h') < error_X(h)$

  i.e. h' behaves better over unseen data

42

Prefer Shorter Hypotheses: Occam's Razor

- Why prefer shorter hypotheses?
- Arguments in favor:
  - If a short hypothesis fits data unlikely to be a coincidence
- Occam's Razor: "*The simplest explanation is usually the best one.*"
  - a principle usually attributed to the 14th-century English Logician William of Ockham.
  - The term razor refers to the act of *shaving away* unnecessary assumptions to get to the simplest explanation.

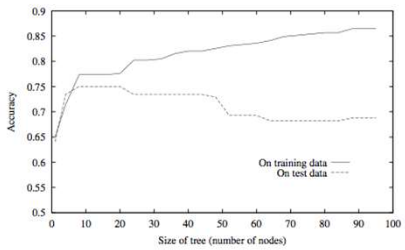
43

Prefer Shorter Hypotheses: Occam's Razor

- Why prefer shorter hypotheses?
- Arguments in favor:
  - There are fewer short hypotheses than long ones
  - If a short hypothesis fits data unlikely to be a coincidence
  - Elegance and aesthetics
- Arguments against:
  - Not every short hypothesis is a reasonable one.
- Occam's Razor: "***The simplest explanation is usually the best one.***"
  - a principle usually attributed to the 14th-century English Logician William of Ockham.
  - The term razor refers to the act of *shaving away* unnecessary assumptions to get to the simplest explanation.

44

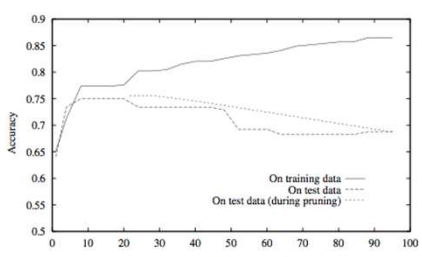
Overfitting in Decision Tree



Avoid Overfitting - Stop before perfect Classification in Decision Trees

- Two strategies:
  1. Stop growing the tree earlier – **Early Stopping**
  2. Allow the tree to *overfit* the data, and then ***post-prune*** the tree
- Training and Validation Set
  - split the training data in two parts (training and validation) and use validation set to assess the utility of *post-pruning* - *Reduced error pruning*
  - *Keep pruning until further post pruning becomes harmful*

Effect of Reduced Error Pruning



Regularization in Decision Tree

1. DT is a **non-parametrized** algorithm unlike linear models where we supply the input parameters.
2. If left unconstrained, they can build tree structures to adapt to the training data leading to **overfitting**
3. To avoid Overfitting, we need to **restrict** the DT's freedom during the tree creation. This is called regularization.
4. The regularization hyperparameters depend on the algorithms used

Regularization in Decision Tree

1. **max_depth** – Is the maximum length of a path from root to leaf (in terms of number of decision points. The leaf node is not split further. It could lead to a tree with leaf node containing many observations on one side of the tree, whereas on the other side, nodes containing much less observations get further split
2. **min_sample_split** - A limit to stop further splitting of nodes when the number of observations in the node is lower than this value
3. **min_sample_leaf** – Minimum number of samples a leaf node must have. When a leaf contains too few observations, further splitting will result in overfitting (modeling of noise in the data).
4. **min_weight_fraction_leaf** – Same as min_sample_leaf but expressed in fraction of total number of weighted instances
5. **max_feature_size** - max number of features that are evaluated for splitting each node

49

Types of Trees

- ID3 (Iterative Dichotomiser 3) was developed in 1986 by Ross Quinlan. The algorithm creates a multiway tree, finding for each node (i.e. in a greedy manner) the categorical feature that will yield the largest information gain for categorical targets. Trees are grown to their maximum size and then a pruning step is usually applied to improve the ability of the tree to generalize to unseen data.
- C4.5 is the successor to ID3 and removed the restriction that features must be categorical by dynamically defining a discrete attribute (based on numerical variables) that partitions the continuous attribute value into a discrete set of intervals. C4.5 converts the trained trees (i.e. the output of the ID3 algorithm) into sets of if-then rules. These accuracy of each rule is then evaluated to determine the order in which they should be applied. Pruning is done by removing a rule's precondition if the accuracy of the rule improves without it. [Quinlan, 1993]
- C5.0 is Quinlan's latest version release under a proprietary license. It uses less memory and builds smaller rulesets than C4.5 while being more accurate.
- CART (Classification and Regression Trees) is very similar to C4.5, but it differs in that it supports numerical target variables (regression) and does not compute rule sets. CART constructs binary trees using the feature and threshold that yield the largest information gain at each node. [L. Breiman, J. Friedman, R. Olshen, 1984]
- **scikit-learn** uses an optimised version of the CART algorithm

50

51

Case-study

52

Wine Quality Prediction

Context

- This datasets is related to red variants of the Portuguese "Vinho Verde" wine. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).
- These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones).

Problem Statement:

- Wine Quality Prediction- Here, we will apply a method of assessing wine quality using a decision tree and test it against the wine-quality dataset from the UC Irvine Machine Learning Repository. The wine dataset is a classic and very easy multi-class classification dataset.

53

53

Description of Attributes:

- 1 - fixed acidity: most acids involved with wine or fixed or nonvolatile (do not evaporate readily)
- 2 - volatile acidity: the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste
- 3 - citric acid: found in small quantities, citric acid can add 'freshness' and flavor to wines
- 4 - residual sugar: the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet
- 5 - chlorides: the amount of salt in the wine
- 6 - free sulfur dioxide: the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine

54

54

Attribute Information Contd.

- 7 - total sulfur dioxide: amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine
- 8 - density: the density of water is close to that of water depending on the percent alcohol and sugar content
- 9 - pH: describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale
- 10 - sulphates: a wine additive which can contribute to sulfur dioxide gas (S02) levels, wich acts as an antimicrobial and antioxidant
- 11 - alcohol: the percent alcohol content of the wine
- **Output variable** (based on sensory data):
- 12 - quality (score between 0 and 10)

55

55

Let's go to the Coding Demo…

56

To be continued in the next session…..