Artificial Intelligence

Decision Tree

한양대학교 ERICA
소프트웨어융합대학
COLLEGE OF COMPUTING

인공지능학과
Department of
Artificial Intelligence

정 우 환 (whjung@hanyang.ac.kr)

Fall 2021

Non-neural classification algorithms

- K-nearest neighbor (k-NN) classifier
- Naïve Bayes classifiers
- Decision trees
- Support Vector Machine (SVM)

Information Gain

Information

- Quantity of information

1000 bits

0000000…000000000

Same quantity?

1000 bits

0010001…111001001

0 * 1000

Same quantity?

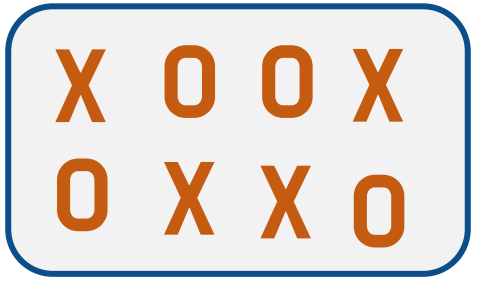
0*2,1*1,0*3…1*3,0*2,1*1,0*2,1*1

(Self) Information $I(x)$

- Roughly speaking, the **minimum number of bits to encode a signal** x
- Definition
  - $I(x) = -\log P(x)$
- Intuition
  - If a pattern is frequent, it can be simply and efficiently encoded/com pressed
  - Example: 0000000...000000000

Entropy

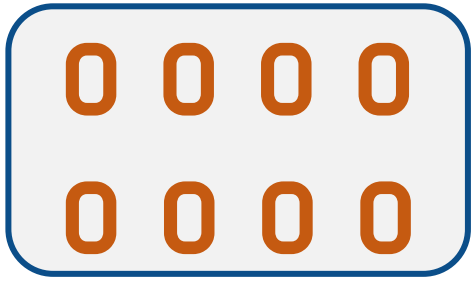
- Entropy (Expected information)
  - $H(X) = E_{X\sim P}[I(x)] = -E_{X\sim P}[\log P(x)]$
  - **Computation:** $H(X) = -\sum_x P(x)\log P(x)$
- Entropy is a measure of impurity



0:4 X:4

$$P(\text{O}) = \frac{1}{2}$$
$$P(\text{X}) = \frac{1}{2}$$

$$H(X) = -P(\text{O})\log P(\text{O}) - P(\text{X})logP(\text{X})$$
$$= -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = \log 2 = 1$$



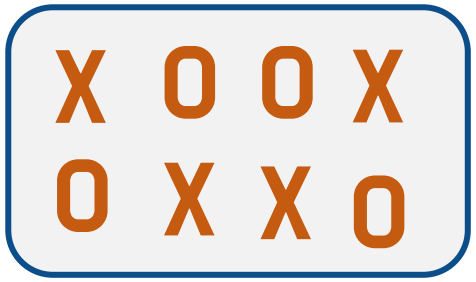
0:8 X:0

$$P(\text{O}) = 1$$
$$P(\text{X}) = 0$$

$$H(X) = -P(\text{O})\log P(\text{O}) - P(\text{X})logP(\text{X})$$
$$= -1\log 1 - 0\log 0 = 0$$

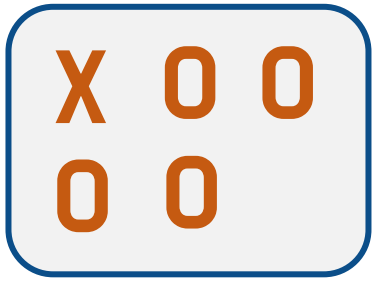
Information Gain

D



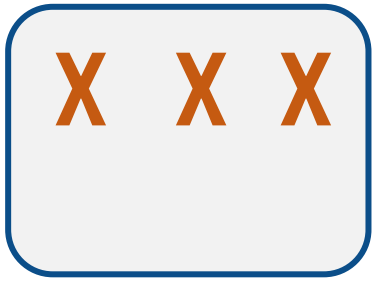
0:4 X:4

D_1



0:4 X:1

D_2



0:0 X:3

Information $I(D) = 8 \cdot H(D) = 8$

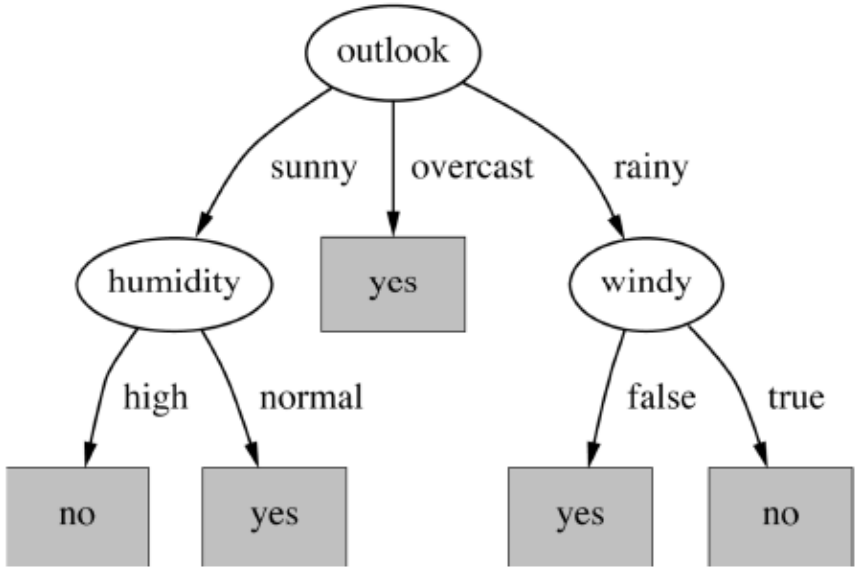
$$I(D_1) = 5 \cdot H(D_1) = -4 \log \frac{4}{5} - \log \frac{1}{5} = 0.7219$$

$I(D_2) = 3 \cdot H(D_2) = 0$

Information gain $I(D) - \big(I(D_1) + I(D_2)\big) = 8 - (0.7219 + 0) = 7.2781$

Decision Tree

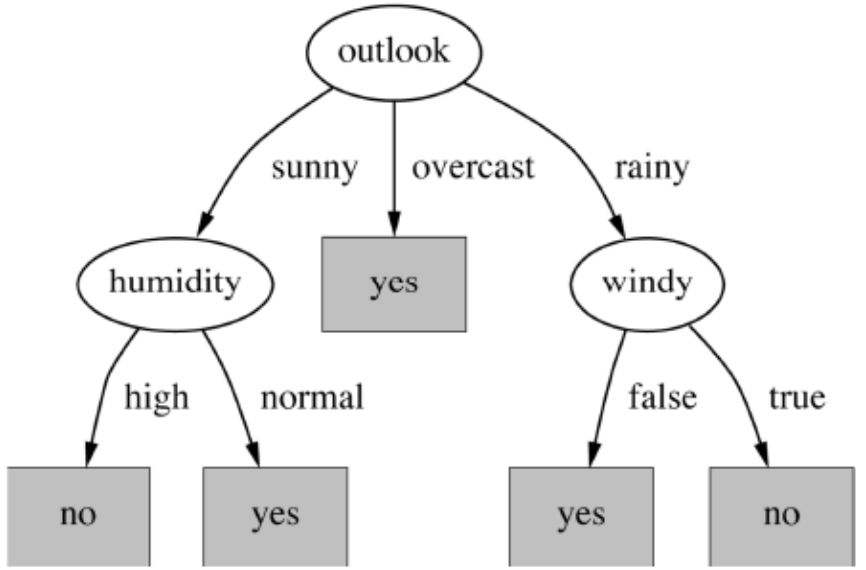
Decision Tree Induction: An Example



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

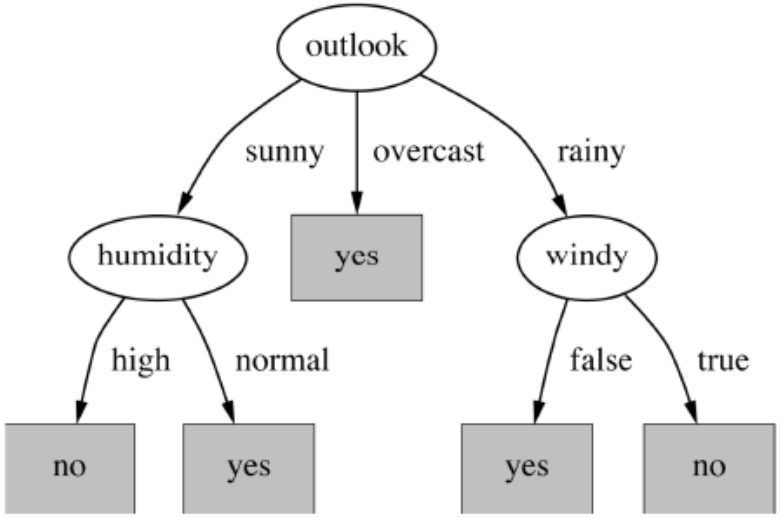
Decision Tree Algorithm

- A decision tree is created in two phases:
  - Building Phase
    - Recursively split nodes using best splitting attribute for node until all the examples in each node belong to one class
  - Pruning Phase
    - Prune leaf nodes recursively to prevent over-fitting
    - Smaller imperfect decision tree generally achieves better accuracy



Decision tree learning : Building phase

- Top-down: recursive divide-and-conquer
  - Select attribute for root node
    - Create branch for each possible attribute value
  - Split instances into subsets
    - One for each branch extending from the node
  - Repeat recursively for each branch
    - using only instances that reach the branch
  - Stop
    - if all instances have the same class



Ian H. Witten's slide

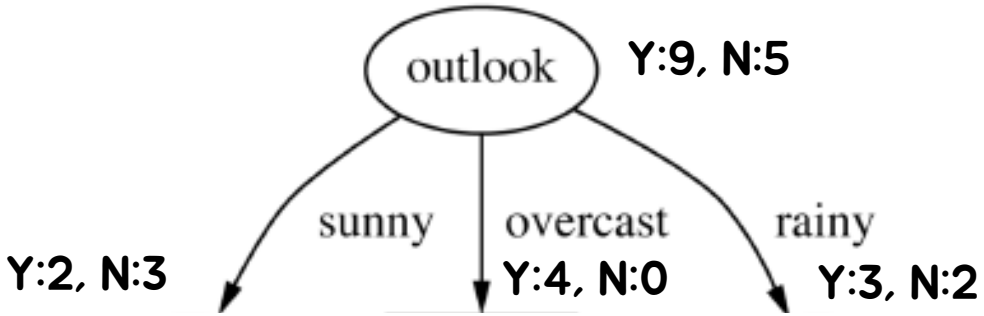
Decision tree learning : Building phase

Y:9, N:5

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

Ian H. Witten's slide

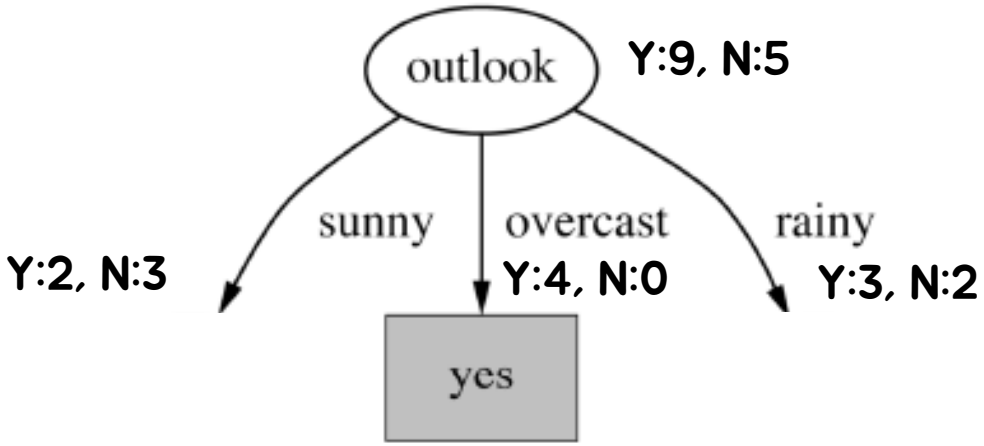
Decision tree learning : Building phase



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

Ian H. Witten's slide

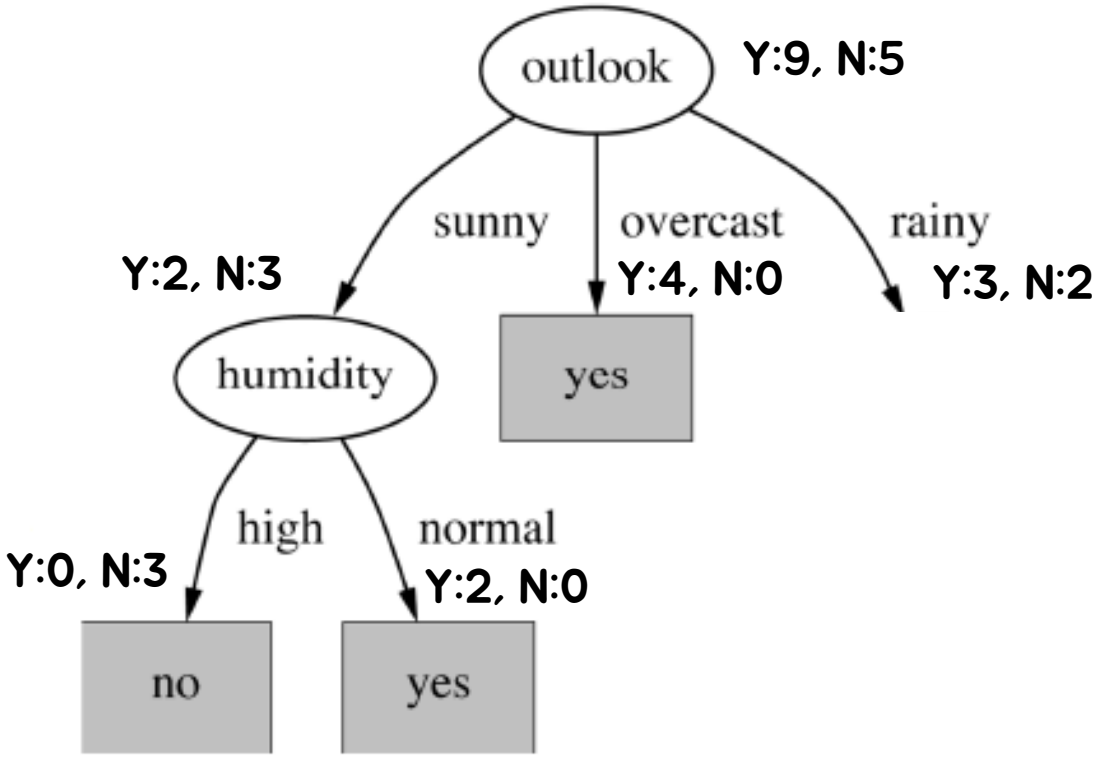
Decision tree learning : Building phase



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

Ian H. Witten's slide

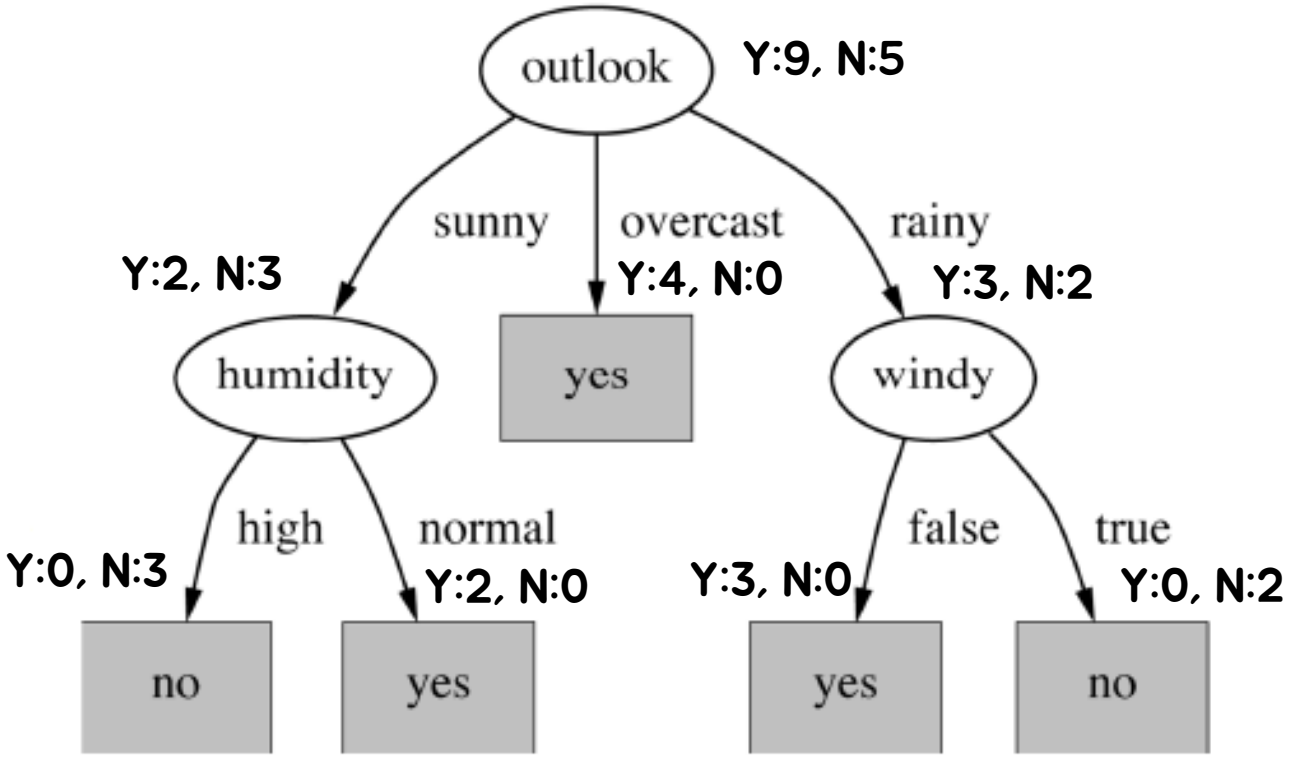
Decision tree learning : Building phase



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

Ian H. Witten's slide

Decision tree learning : Building phase

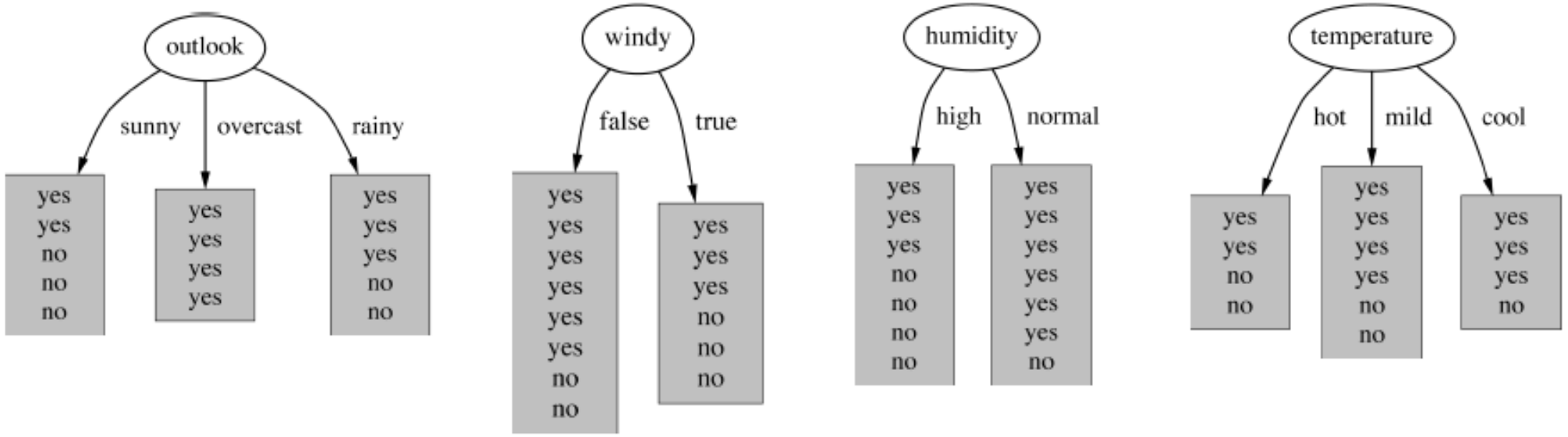


Outlook	Temp	Humidity	Wind	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny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

Ian H. Witten's slide

Decision tree learning : Building phase

Which attribute to select?

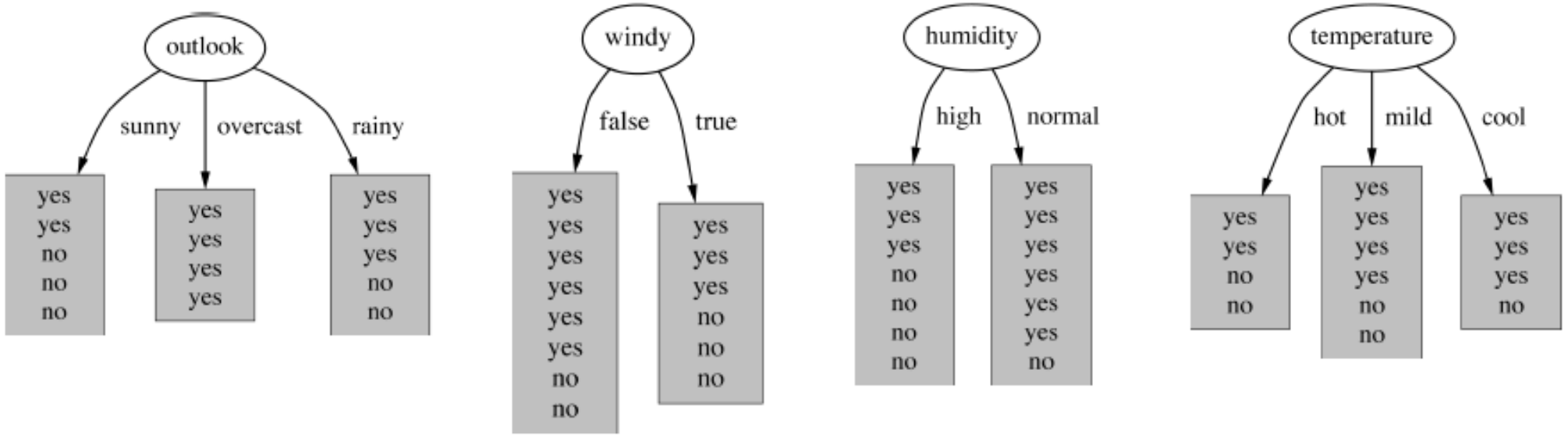


Ian H. Witten's slide

Decision tree learning : Building phase

Which attribute to select?

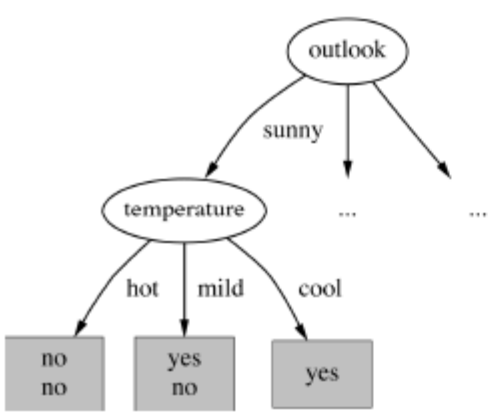
Information gain!

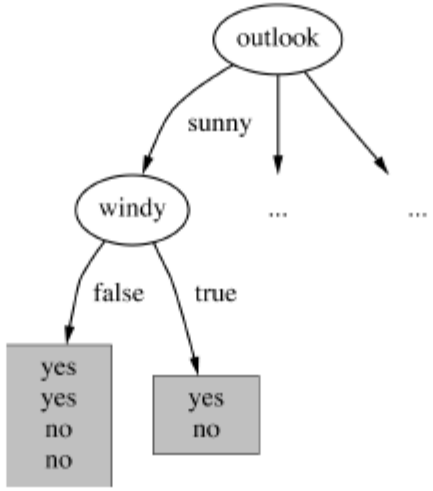


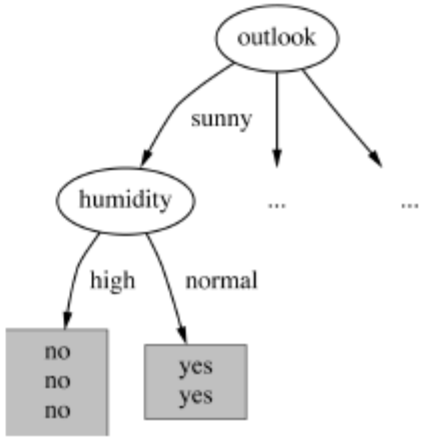
Ian H. Witten's slide

Decision Trees

Continue to split ...







gain(*temperature*) = 0.571 bits
gain(*windy*) = 0.020 bits
gain(*humidity*) = 0.971 bits

Ian H. Witten's slide

Attribute Selection: Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

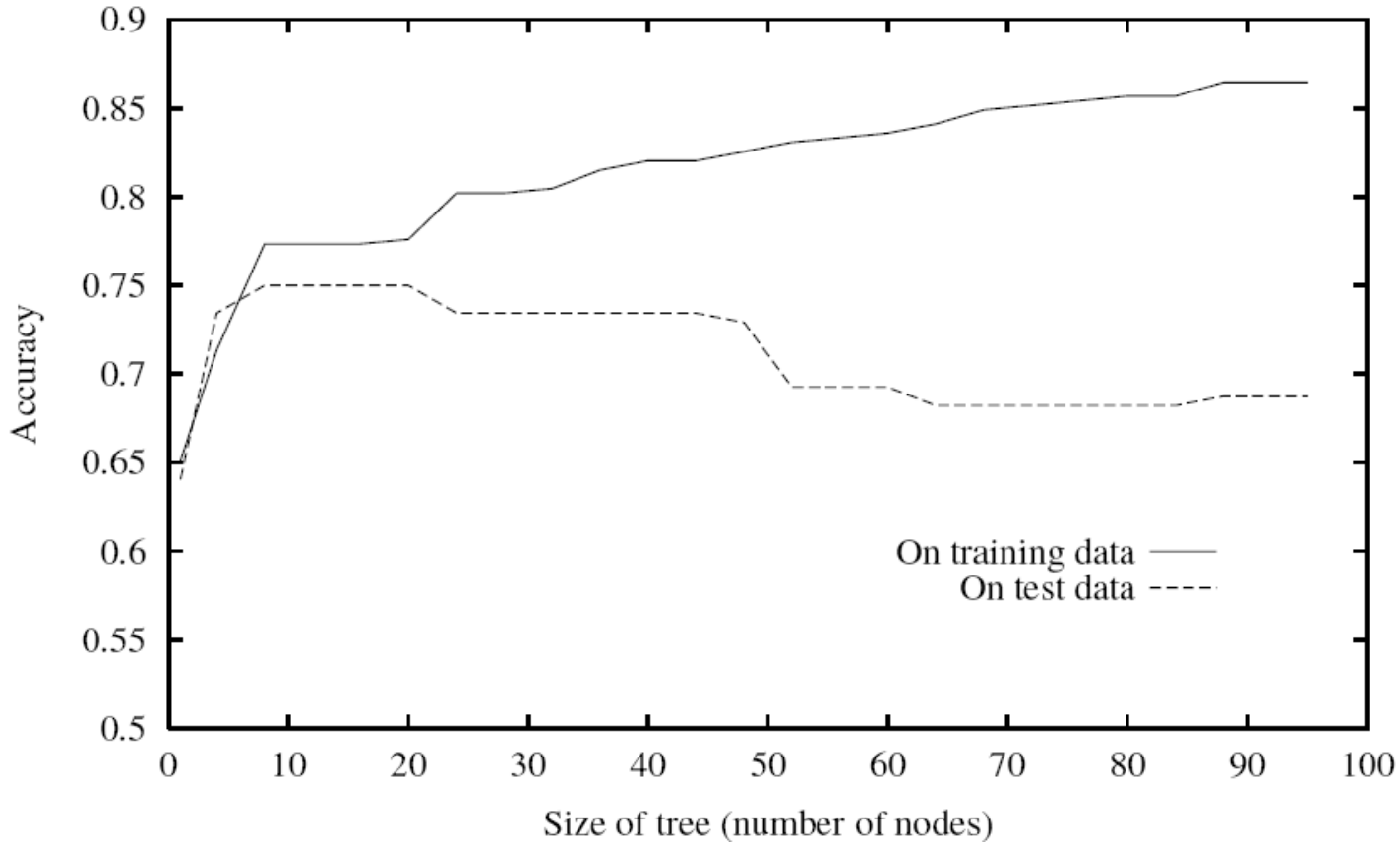
$\frac{5}{14}I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes's and 3 no's. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

Overfitting in Decision Tree Learning

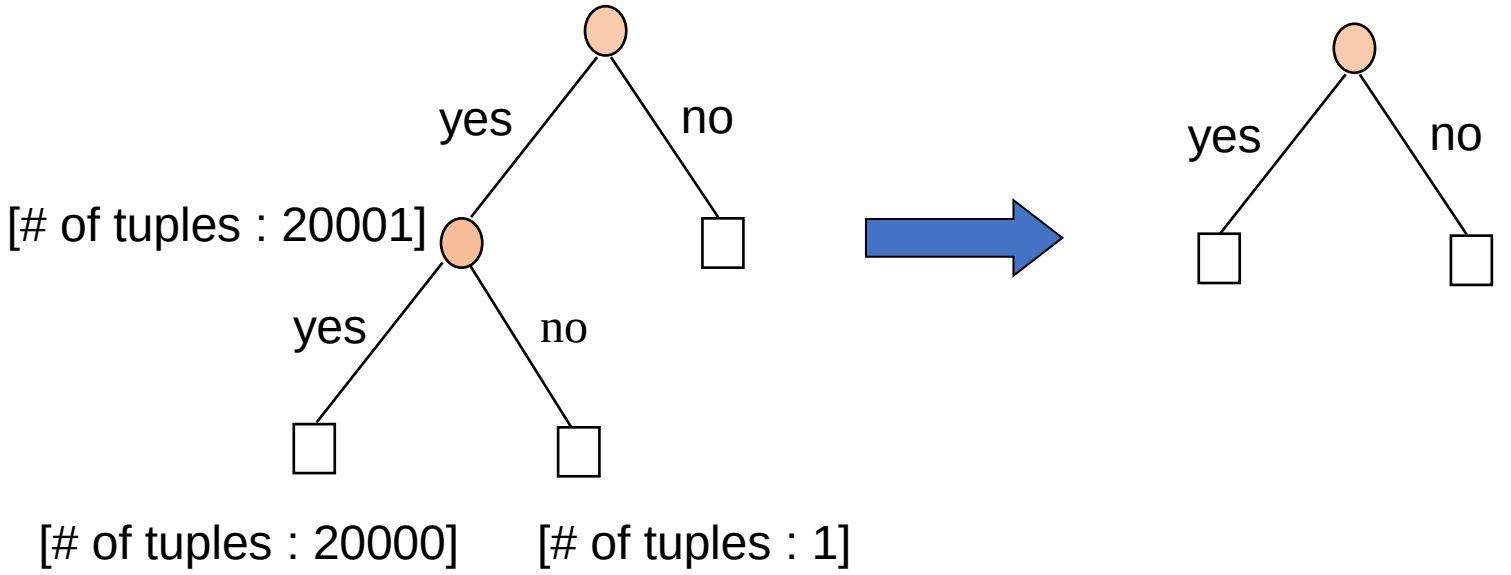


Overfitting and Tree Pruning

- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - **Postpruning*** (preferred in practice) - take a fully-grown decision tree and discard unreliable parts
  - Prepruning - stop growing a branch when information becomes unreliable

Pruning Phase

- Smaller imperfect decision tree generally achieves better accuracy
- Prune leaf nodes recursively to prevent over-fitting



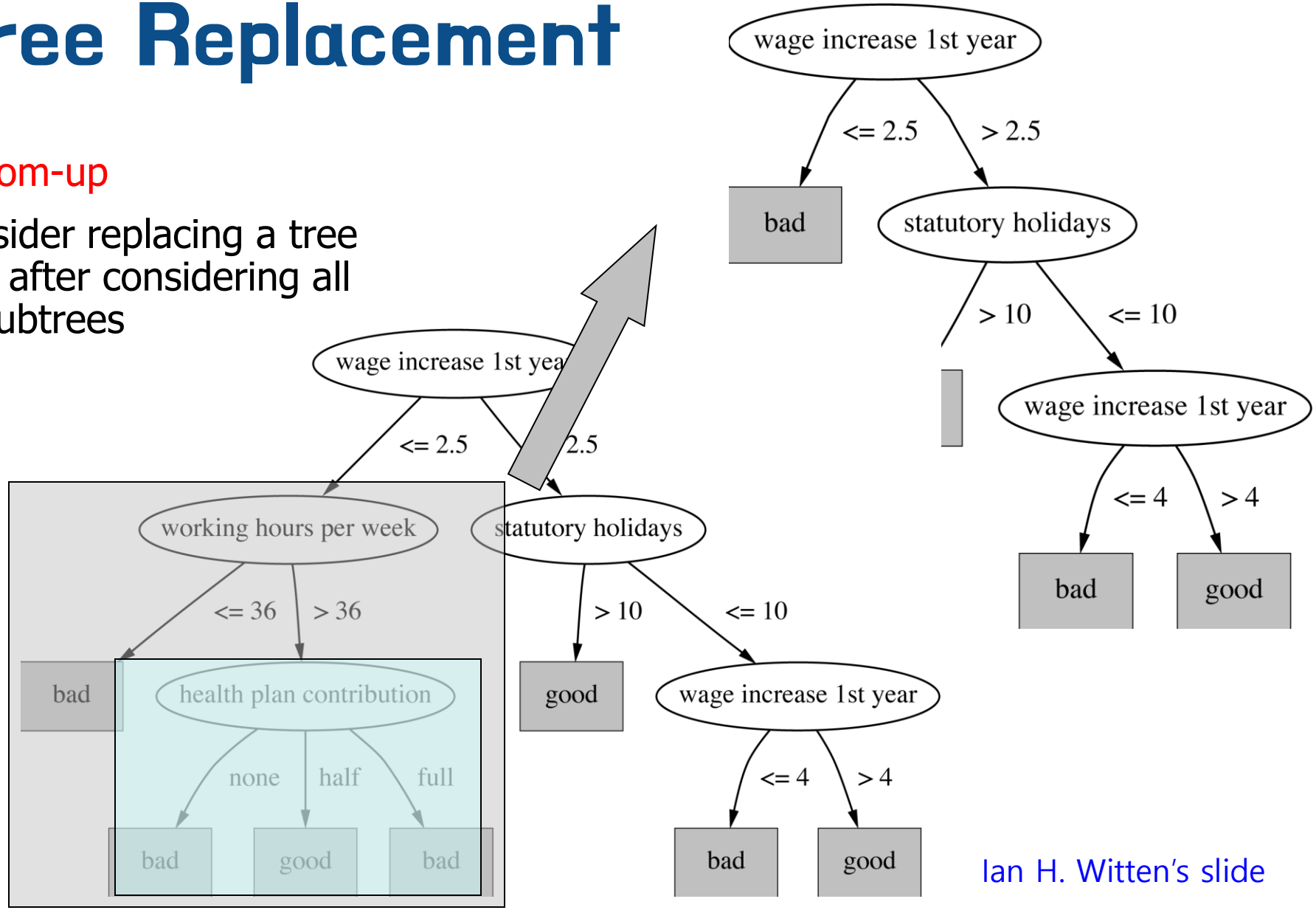
Post-pruning

- Split data into training and validation set
- Build full tree using training dataset
- Do until further pruning is harmful:
  - 1. Evaluate impact on validation set of pruning each possible node (plus those below it)
  - 2. Greedily remove the one whose removal most increases validation set accuracy

Ian H. Witten's slide

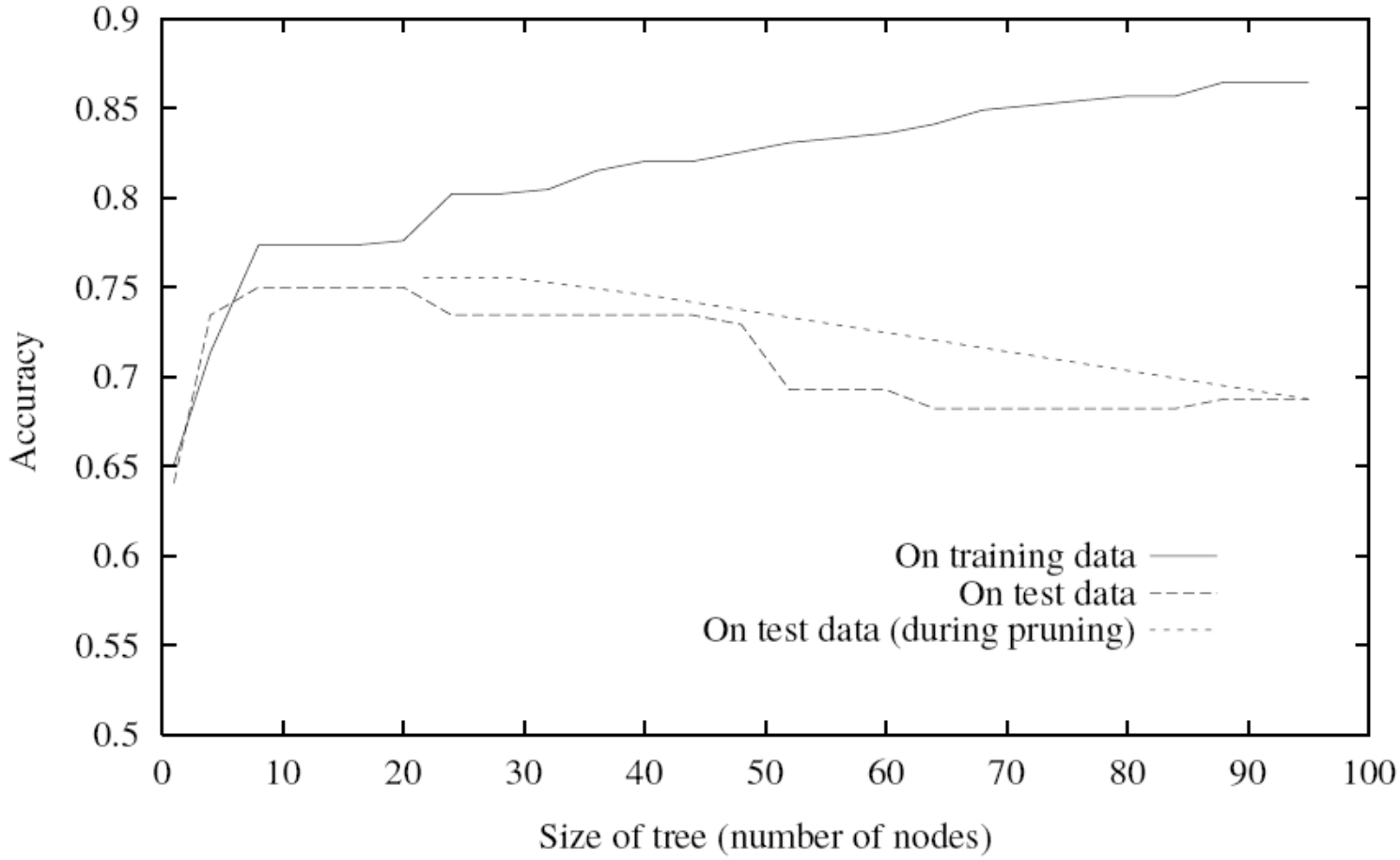
Subtree Replacement

- Bottom-up
- Consider replacing a tree only after considering all its subtrees



Ian H. Witten's slide

Effect of Reduced-Error Pruning



Decision trees for continuous data

- https://lovit.github.io/machine%20learning/2018/04/30/decision_tree/

Decision trees for continuous data

- Selection of the split point
  - There are many possible split points for a continuous attribute
  - Step 1: sort records by the attribute values
  - Step 2: evaluate each split point by using the information gain

salary	label	rid
10000	reject	0
40000	accept	1
15000	reject	2
75000	accept	3
18000	accept	4

salary	label	rid
10000	reject	0
15000	reject	2
18000	accept	4
40000	accept	1
75000	accept	3

Decision Boundaries