Machine Learning for Language Technology 2015

http://stp.lingfil.uu.se/~santinim/ml/2015/ml4lt_2015.htm

Decision Trees (2)

Entropy, Information Gain, Gain Ratio

Marina Santini
santinim@stp.lingfil.uu.se

Department of Linguistics and Philology
Uppsala University, Uppsala, Sweden

Autumn 2015

Acknowledgements

- Weka’s slides
- Wikipedia and other websites
- Witten et al. (2011: 99-108; 195-203; 192-203)

Outline

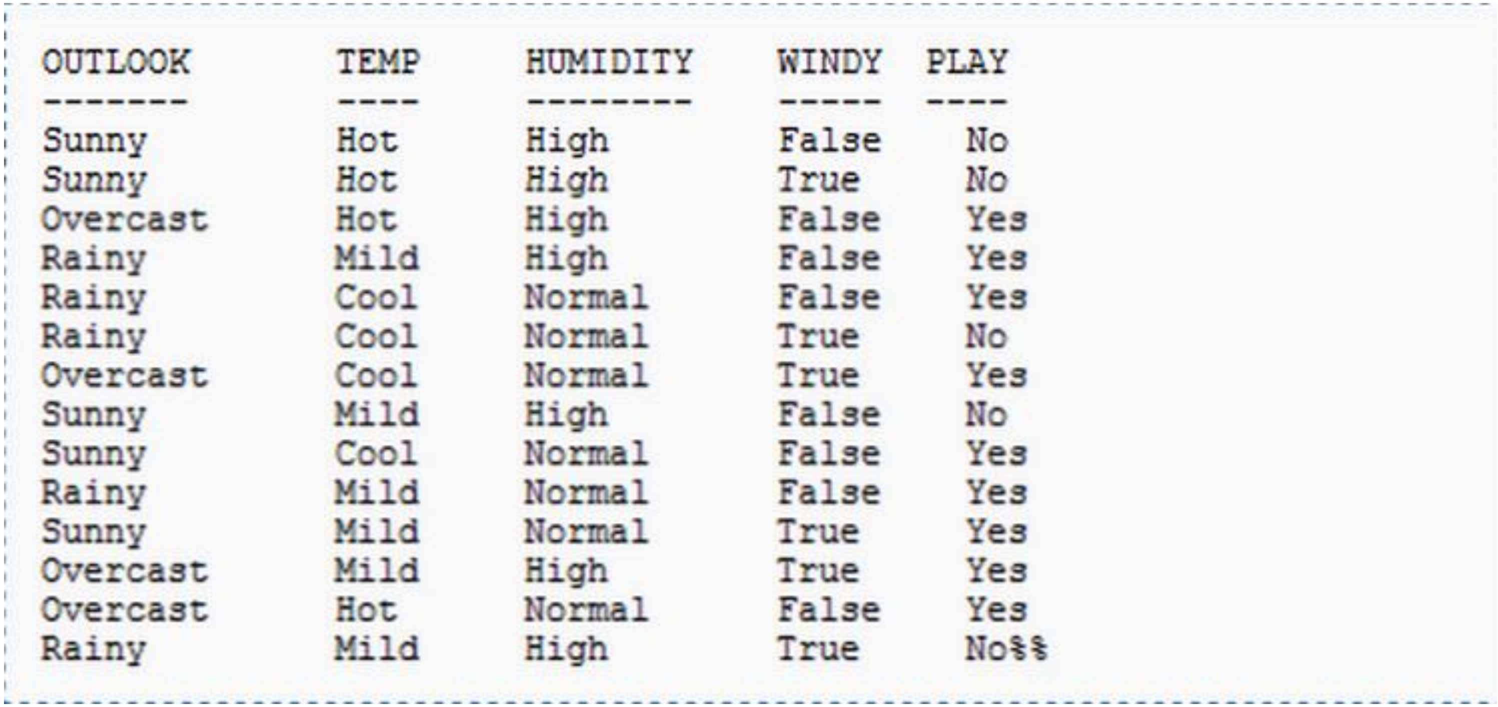
- Attribute selection
- Entropy
- Suprisal
- Information Gain
- Gain Ratio
- Pruning
- Rules

Constructing decision trees

- Strategy: top down
  Recursive *divide-and-conquer* fashion
  - First: select attribute for root node
    Create branch for each possible attribute value
  - Then: split instances into subsets
    One for each branch extending from the node
  - Finally: repeat recursively for each branch, using only instances that reach the branch
- Stop if all instances have the same class

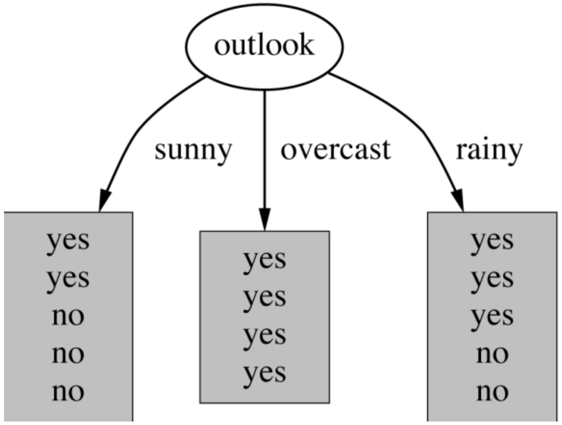
Play or not?

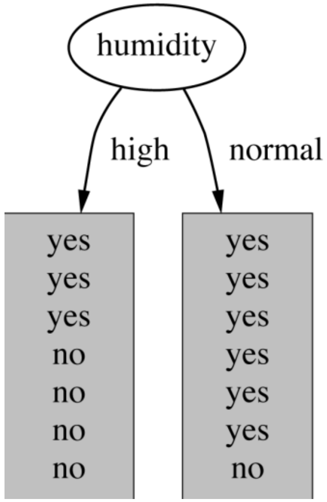
- The weather dataset

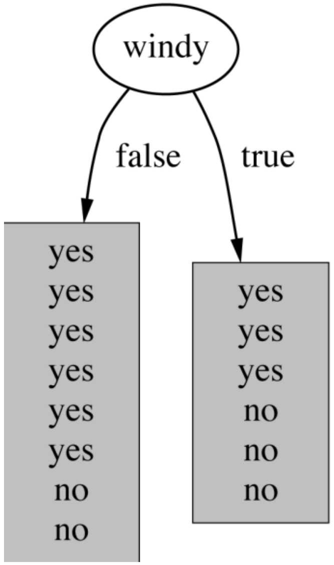


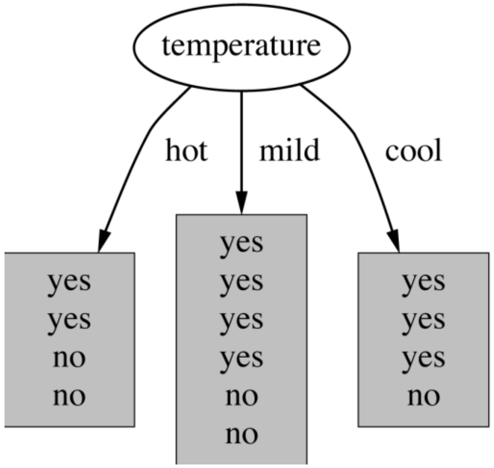
```
OUTLOOK         TEMP      HUMIDITY     WINDY  PLAY
-------         ----      --------     -----  ----
Sunny           Hot       High         False    No
Sunny           Hot       High         True     No
Overcast        Hot       High         False    Yes
Rainy           Mild      High         False    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$$
```

Which attribute to select?







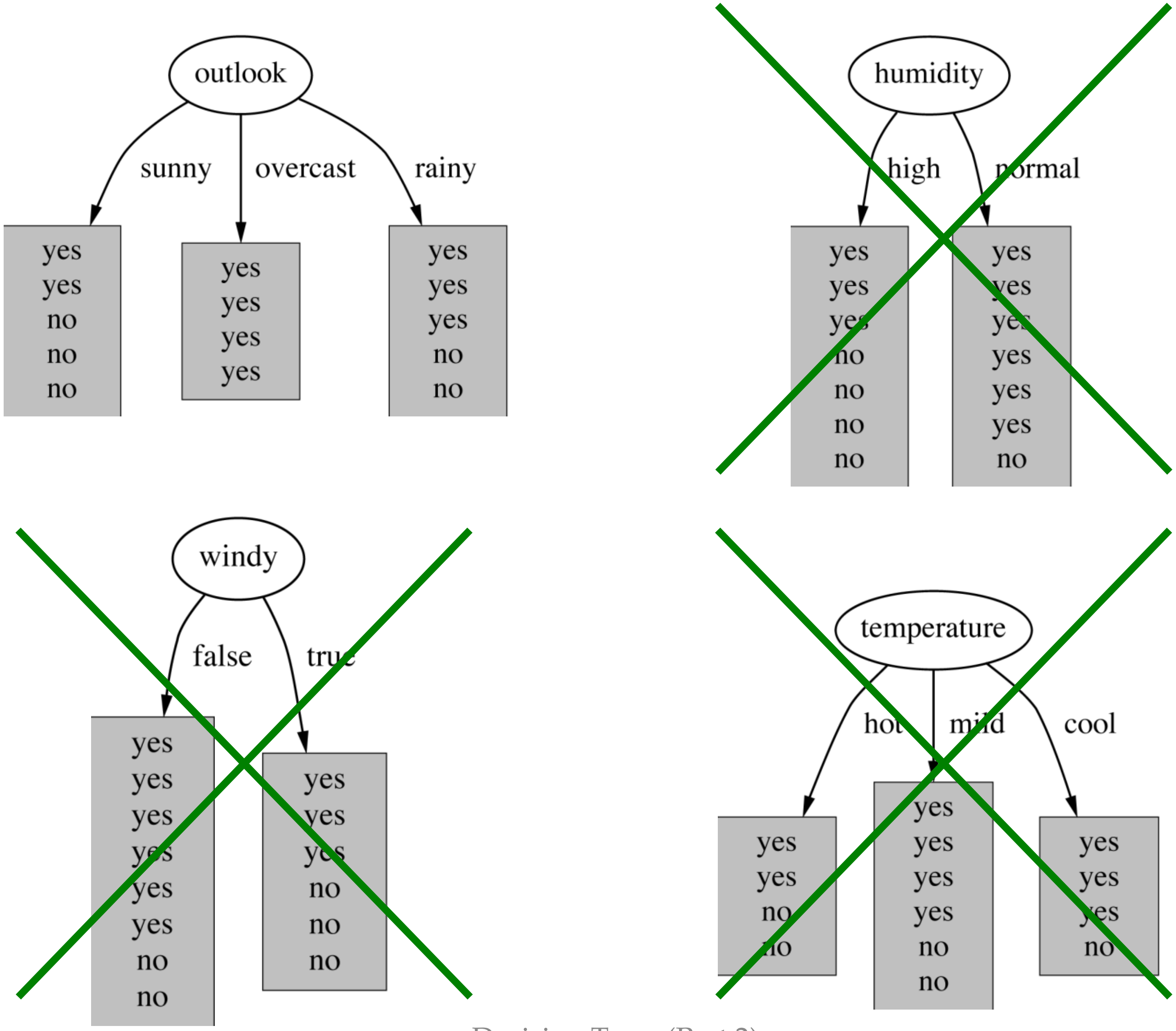


Computing purity: the *information* measure

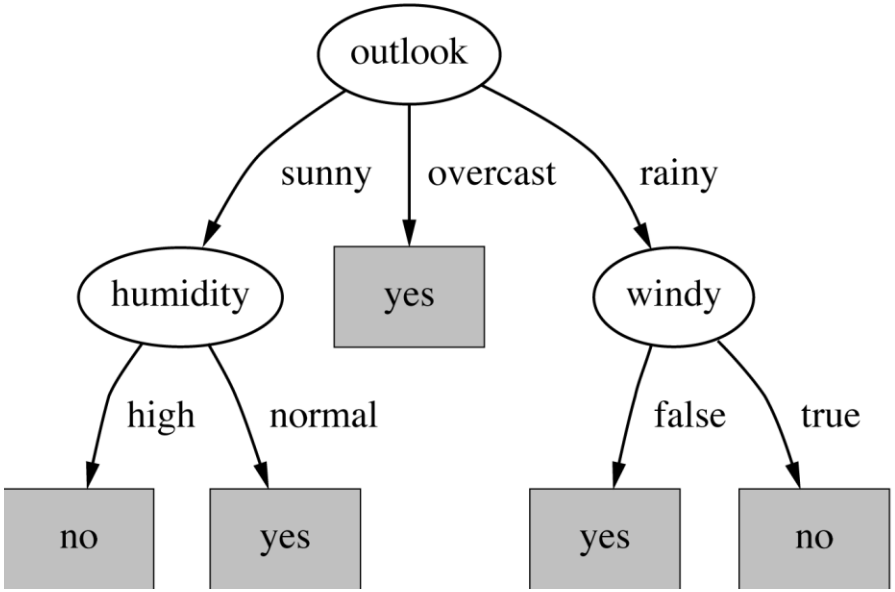
- information is a measure of a reduction of uncertainty

- It represents the expected amount of information that would be needed to “place” a new instance in the branch.

Which attribute to select?



Final decision tree



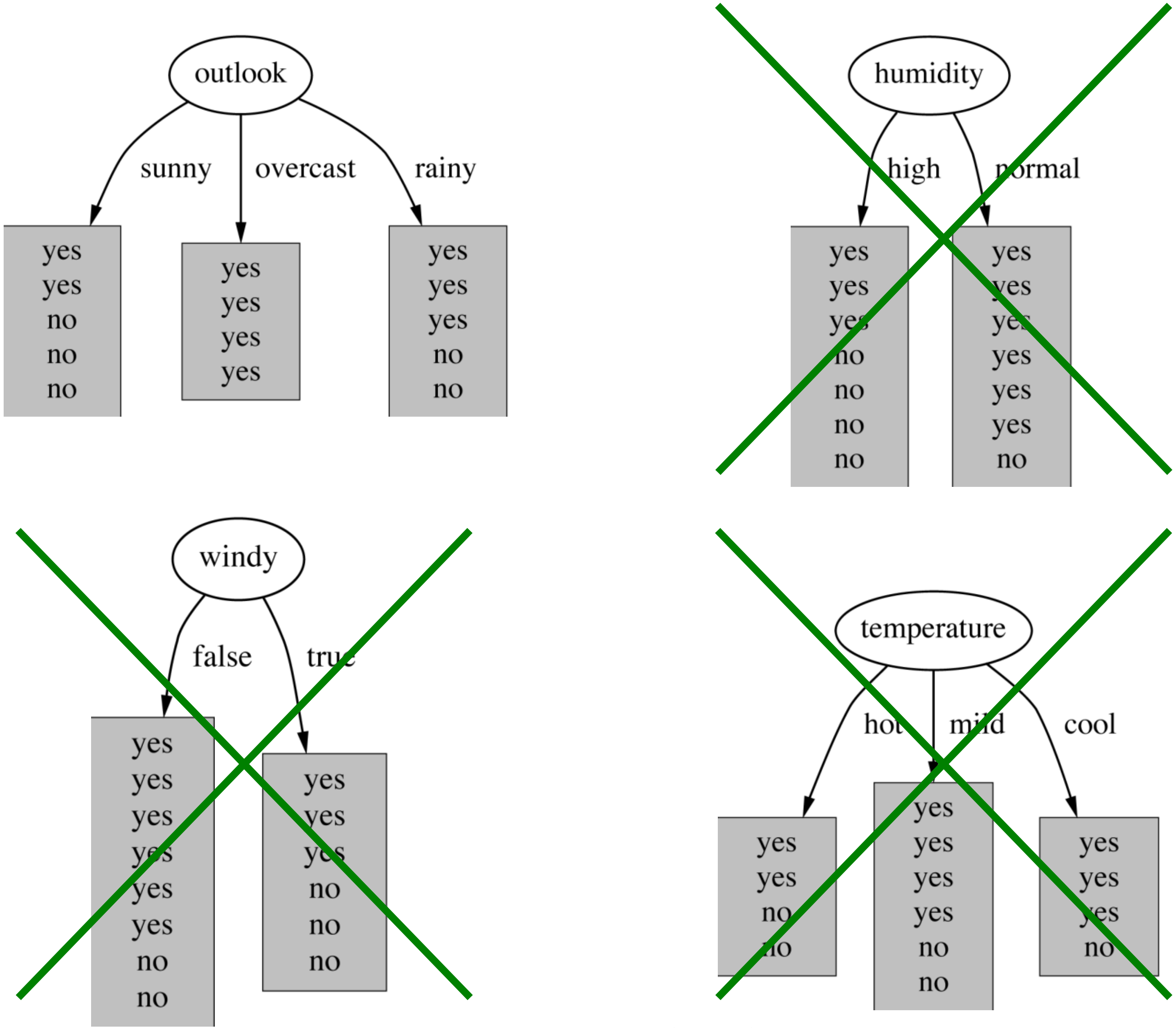
⇒ Splitting stops when data can’t be split any further

Criterion for attribute selection

- Which is the best attribute?
  - Want to get the smallest tree
  - Heuristic: choose the attribute that produces the "purest" nodes

-- Information gain: increases with the average purity of the subsets

-- Strategy: choose attribute that gives greatest information gain

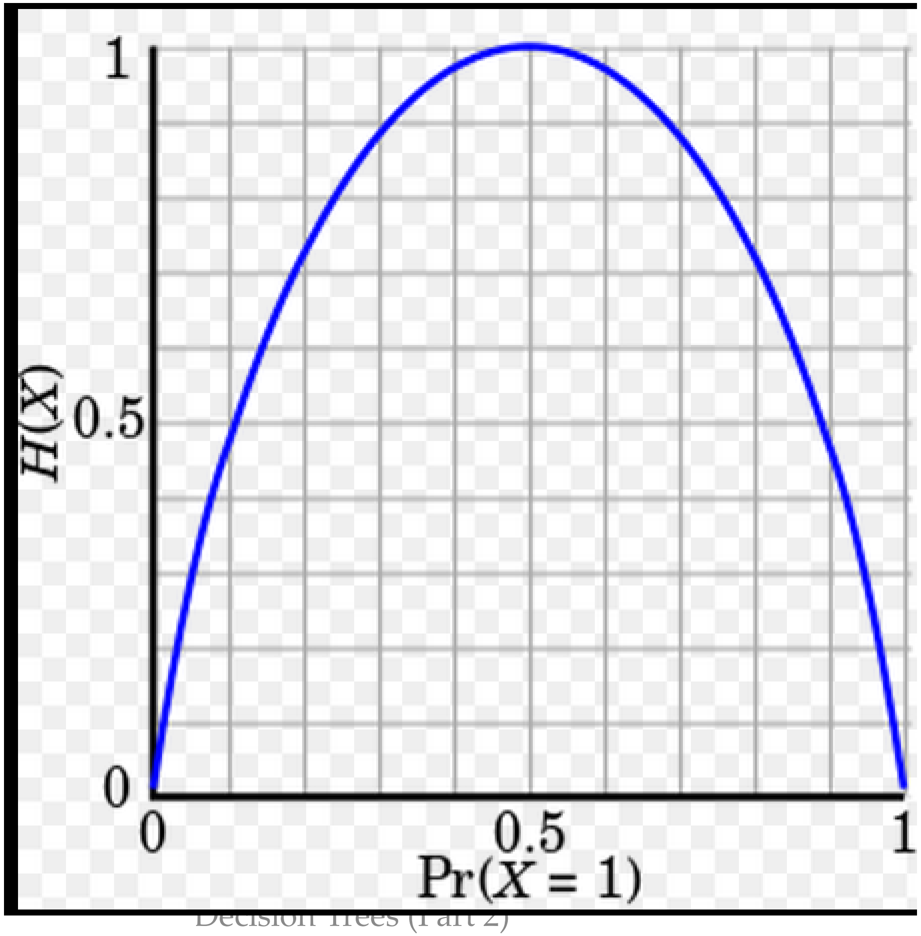


How to compute Informaton Gain: Entropy

1. When the number of either yes OR no is zero (that is the node is pure) the information is zero.
2. When the number of yes and no is equal, the information reaches its maximum because we are very uncertain about the outcome.
3. Complex scenarios: the measure should be applicable to a multiclass situation, where a multi-staged decision must be made.

Entropy

- Entropy (aka expected surprisal)



Suprisal: Definition

- Surprisal (aka self-information) is a measure of the information content associated with an event in a probability space.

- The smaller its probability of an event, the larger the surprisal associated with the information that the event occur.

- By definition, the measure of surprisal is positive and additive. If an event C is the *intersection* of two independent events A and B, then the amount of information knowing that C has happened, equals the sum of the amounts of information of event A and event B respectively:

$$I(A \cap B)=I(A)+I(B)$$

Surprisal: Formula

- The surprisal “I” of an event is:

$$I(\omega_n) = \log\left(\frac{1}{P(\omega_n)}\right) = -\log(P(\omega_n))$$

Entropy: Formulas

- Formulas for computing entropy:

$$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$$

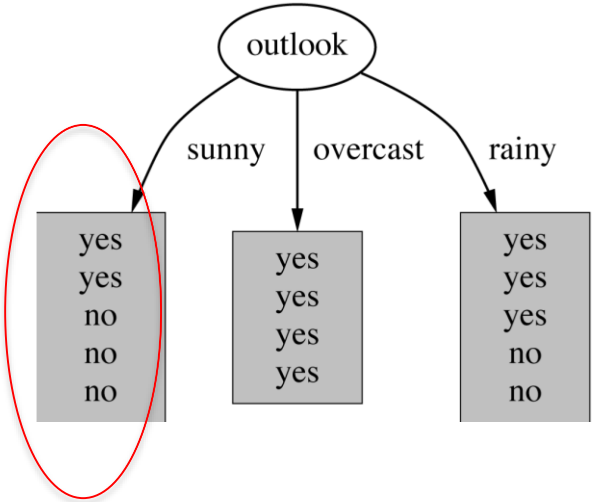
$$H(X) = -\sum_x p(x) \log p(x)$$

Entropy: Outlook, sunny

- Formulae for computing the entropy:

$$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$$

$$H(X) = -\sum_x p(x) \log p(x)$$



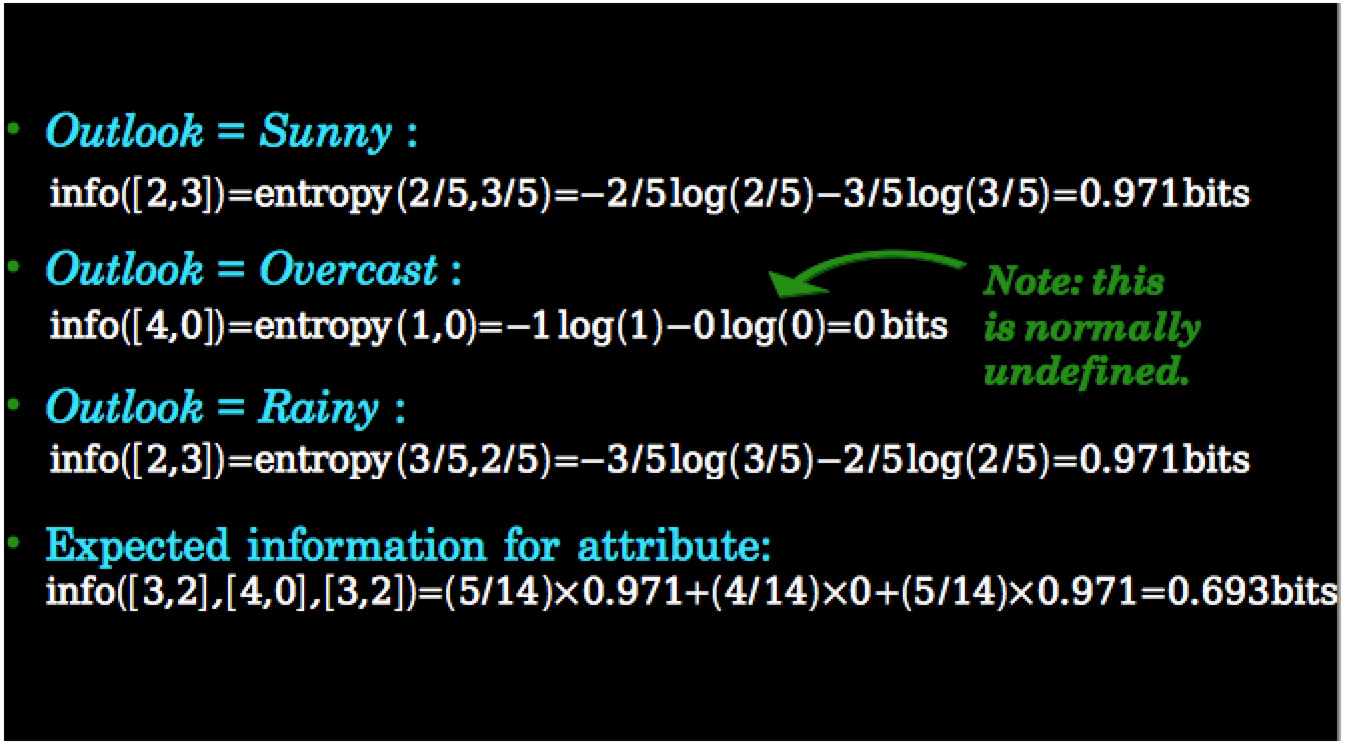
$$\text{info}([2,3]) = -2/5 \times \log 2/5 - 3/5 \times \log 3/5 = 0.971 \text{ bits,}$$

= (((-2) / 5) log2(2 / 5)) + (((-3) / 5) x log2(3 / 5)) = 0.97095059445

Measures: *Information* & Entropy

- Watch out: There are many statements in the literature which say that information is the same as entropy.
- **Properly speaking:** *entropy* is a probabilistic measure of uncertainty or ignorance and *information* is a measure of a reduction of uncertainty
- However, in our context we use entropy (ie the quantity of uncertainty) to measure the purity of a node.

Example: *Outlook*



- *Outlook = Sunny* :

  $\text{info}([2,3]) = \text{entropy}(2/5,3/5) = -2/5\log(2/5) - 3/5\log(3/5) = 0.971\,\text{bits}$

- *Outlook = Overcast* :

  $\text{info}([4,0]) = \text{entropy}(1,0) = -1\log(1) - 0\log(0) = 0\,\text{bits}$

  Note: this is normally undefined.

- *Outlook = Rainy* :

  $\text{info}([2,3]) = \text{entropy}(3/5,2/5) = -3/5\log(3/5) - 2/5\log(2/5) = 0.971\,\text{bits}$

- Expected information for attribute:

  $\text{info}([3,2],[4,0],[3,2]) = (5/14)\times 0.971 + (4/14)\times 0 + (5/14)\times 0.971 = 0.693\text{bits}$

Computing Information Gain

- Information gain: information before splitting – information after splitting

gain(*Outlook*) = info([9,5]) – info([2,3],[4,0],[3,2])
= 0.940 – 0.693
= 0.247 bits

- Information gain for attributes from weather data:

gain(*Outlook*)	= 0.247 bits
gain(*Temperature*)	= 0.029 bits
gain(*Humidity*)	= 0.152 bits
gain(*Windy*)	= 0.048 bits

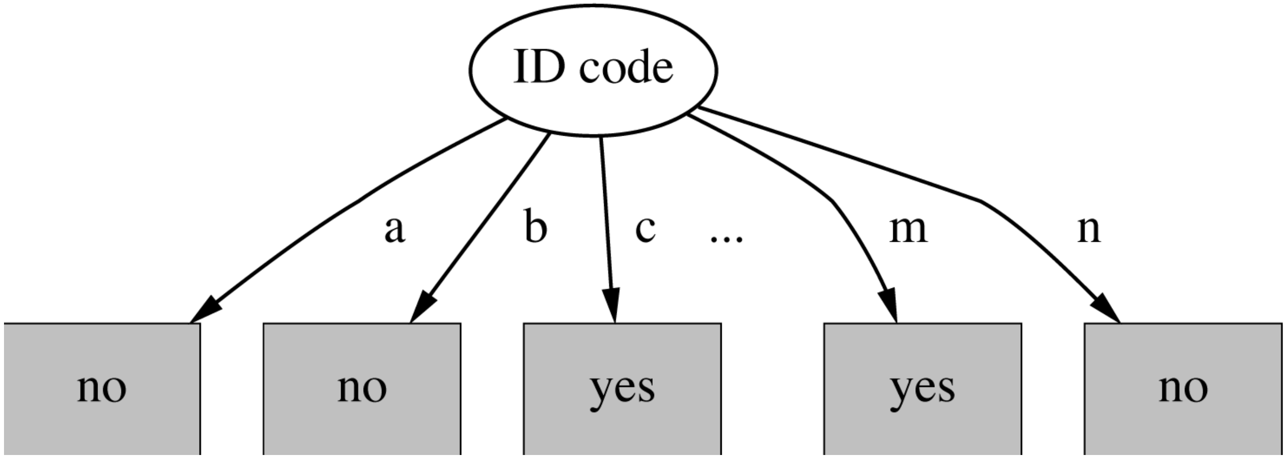
Information Gain Drawbacks

- Problematic: attributes with a large number of values (extreme case: ID code)

Weather data with *ID code*

ID code	Outlook	Temp.	Humidity	Windy	Play
A	Sunny	Hot	High	False	No
B	Sunny	Hot	High	True	No
C	Overcast	Hot	High	False	Yes
D	Rainy	Mild	High	False	Yes
E	Rainy	Cool	Normal	False	Yes
F	Rainy	Cool	Normal	True	No
G	Overcast	Cool	Normal	True	Yes
H	Sunny	Mild	High	False	No
I	Sunny	Cool	Normal	False	Yes
J	Rainy	Mild	Normal	False	Yes
K	Sunny	Mild	Normal	True	Yes
L	Overcast	Mild	High	True	Yes
M	Overcast	Hot	Normal	False	Yes
N	Rainy	Mild	High	True	No

Tree stump for *ID code* attribute



- Entropy of split (see Weka book 2011: 105-108):

$$\text{info}([0,1]) + \text{info}([0,1]) + \text{info}([1,0]) + \ldots + \text{info}([1,0]) + \text{info}([0,1])$$

⇒ Information gain is maximal for ID code (namely 0.940 bits)

Information Gain Limitations

- Problematic: attributes with a large number of values (extreme case: ID code)
- Subsets are more likely to be pure if there is a large number of values
  - ⇒ Information gain is biased towards choosing attributes with a large number of values
  - ⇒ This may result in *overfitting* (selection of an attribute that is non-optimal for prediction)
- (Another problem: *fragmentation)*

Gain ratio

- *Gain ratio*: a modification of the information gain that reduces its bias
- Gain ratio takes number and size of branches into account when choosing an attribute
    - It corrects the information gain by taking the *intrinsic information* of a split into account
- Intrinsic information: information about the class is disregarded.

Gain ratios for weather data

Outlook		Temperature	
Info:	0.693	Info:	0.911
Gain: 0.940-0.693	0.247	Gain: 0.940-0.911	0.029
Split info: info([5,4,5])	1.577	Split info: info([4,6,4])	1.557
Gain ratio: 0.247/1.577	0.157	Gain ratio: 0.029/1.557	0.019
Humidity		Windy	
Info:	0.788	Info:	0.892
Gain: 0.940-0.788	0.152	Gain: 0.940-0.892	0.048
Split info: info([7,7])	1.000	Split info: info([8,6])	0.985
Gain ratio: 0.152/1	0.152	Gain ratio: 0.048/0.985	0.049

More on the gain ratio

- “Outlook” still comes out top
- However: “ID code” has greater gain ratio
  - Standard fix: *ad hoc* test to prevent splitting on that type of attribute
- Problem with gain ratio: it may overcompensate
  - May choose an attribute just because its intrinsic information is very low
  - Standard fix: only consider attributes with greater than average information gain

Interim Summary

- Top-down induction of decision trees: ID3, algorithm developed by Ross Quinlan
    - Gain ratio just one modification of this basic algorithm
    - ⇒ C4.5: deals with numeric attributes, missing values, noisy data
- Similar approach: CART
- There are many other attribute selection criteria!
  (But little difference in accuracy of result)

Pruning

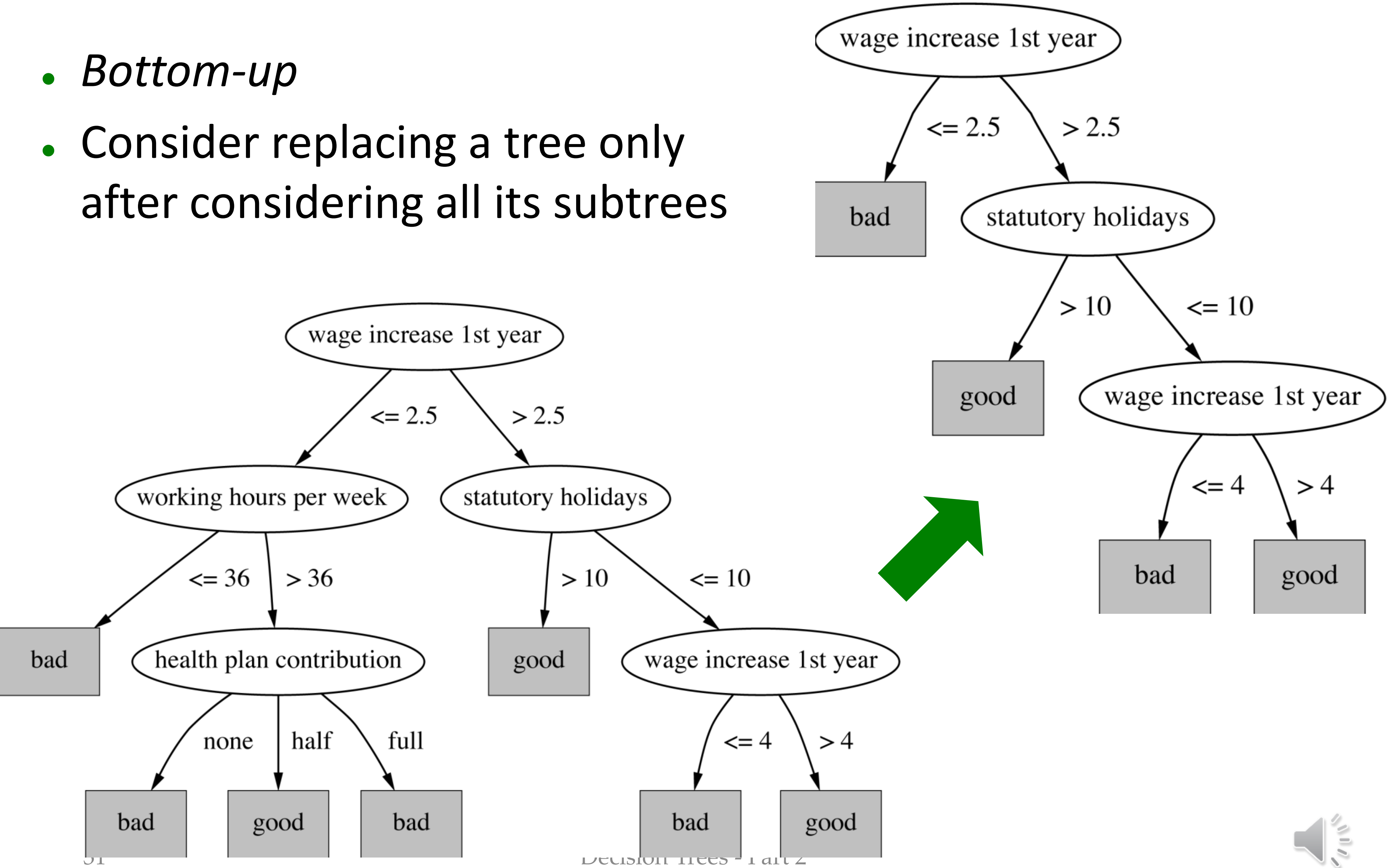
- Prevent overfitting to noise in the data
- “Prune” the decision tree
- Two strategies:
  - *Postpruning*
    take a fully-grown decision tree and discard unreliable parts
  - *Prepruning*
    stop growing a branch when information becomes unreliable
- Postpruning preferred in practice—prepruning can “stop early”

Postpruning

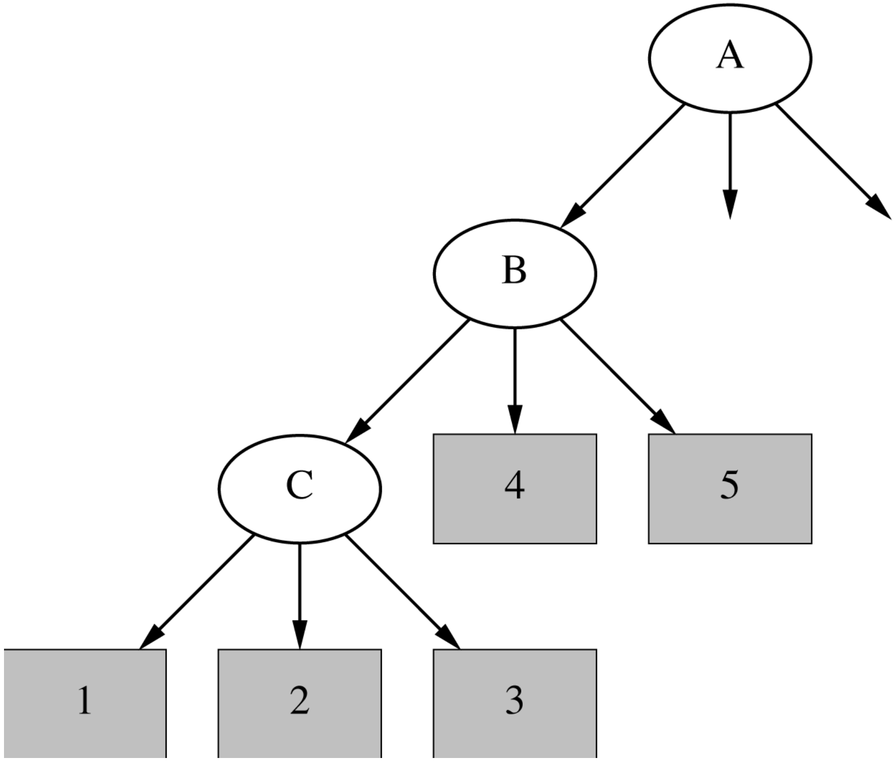
- First, build full tree
- Then, prune it
  - Fully-grown tree shows all attribute interactions
- Problem: some subtrees might be due to chance effects
- Two pruning operations:
  - *Subtree replacement*
  - *Subtree raising*

Subtree replacement

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

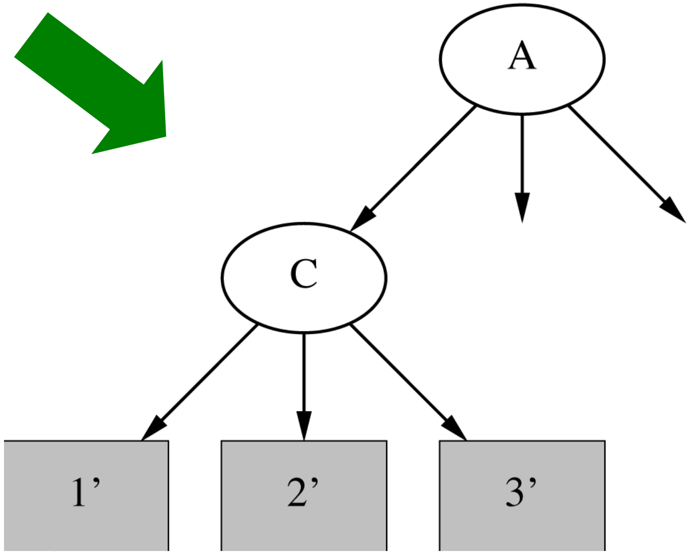


Subtree raising



- Delete node
- Redistribute instances
- Slower than subtree replacement

  (Worthwhile?)



Prepruning

- Based on statistical significance test
  - Stop growing the tree when there is no *statistically significant* association between any attribute and the class at a particular node
- Most popular test: *chi-squared test*
- ID3 used chi-squared test in addition to information gain
  - Only statistically significant attributes were allowed to be selected by information gain procedure

From trees to rules

- Easy: converting a tree into a set of rules
    - One rule for each leaf:
- Produces rules that are unambiguous
    - Doesn’t matter in which order they are executed
- But: resulting rules are unnecessarily complex
    - Pruning to remove redundant tests/rules

From rules to trees

- More difficult: transforming a rule set into a tree
  - Tree cannot easily express disjunction between rules

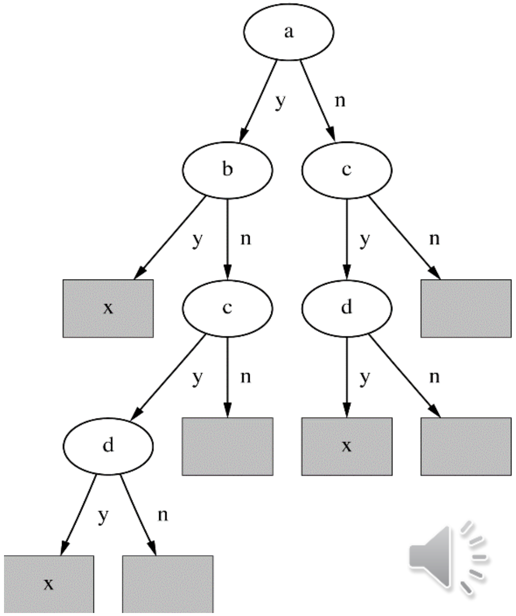
From rules to trees: Example

- Example: rules which test different attributes

```
If a and b then x
If c and d then x
```

Symmetry needs to be broken

- Corresponding tree contains identical subtrees (⇒ “replicated subtree problem”)



Topic Summary

- Attribute selection
- Entropy
- Suprisal
- Information Gain
- Gain Ratio
- Pruning
- Rules

- Quizzes are naively tricky, just to double check that your attention is still with me ☺

Quiz 1: Regression and Classification

Which of these statement is correct in the context of machine learning?

1. Classification is is the process of computing model that predicts a numeric quantity.
2. Regression and Classification mean the same.
3. Regression is the process of computing model that predicts a numeric quantity.

Quiz 2: Information Gain

What is the main drawback of the IG metric in certain contexts?

1. It is biassed towards attributes that have many values.
2. It is based on entropy rather than suprisal.
3. None of the above.

Quiz 3: Gain Ratio

What is the main difference between IG and GR?

1. GR disregards the information about the class, and IG takes the class into account.
2. IG disregards the information about the class and GR takes the class into account.
3. None of the above.

Quiz 4: Pruning

Which pruning strategy is commonly recommended?

1. Prepruning
2. Postpruning
3. Subtree raising

The End