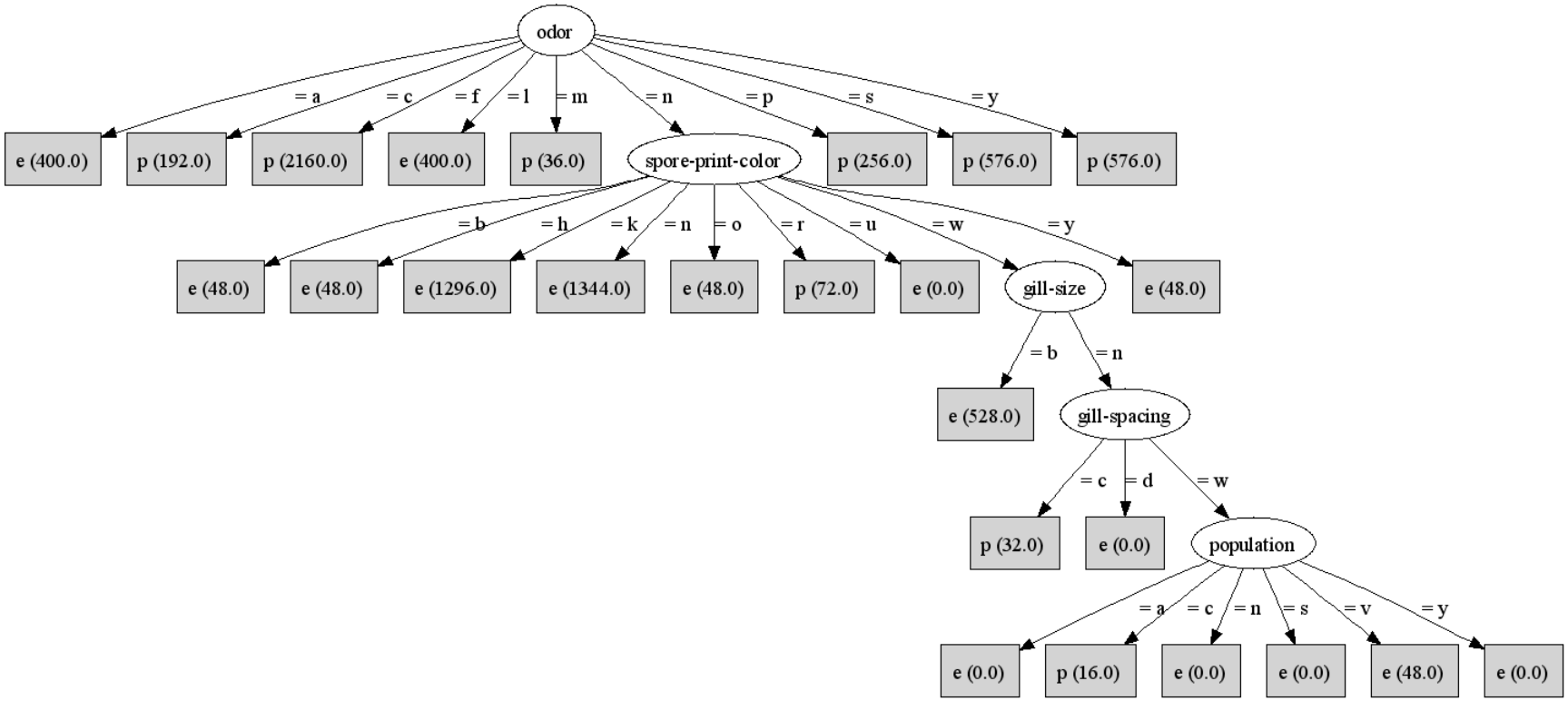
Decision Tree

Dae-Ki Kang

Definition

- Definition #1
  - A hierarchy of if-then's
  - Node – test
  - Edge – direction of control
- Definition #2
  - A tree that represents compression of data based on class
- Manually generated decision tree is not interesting at all!

Decision tree for mushroom data



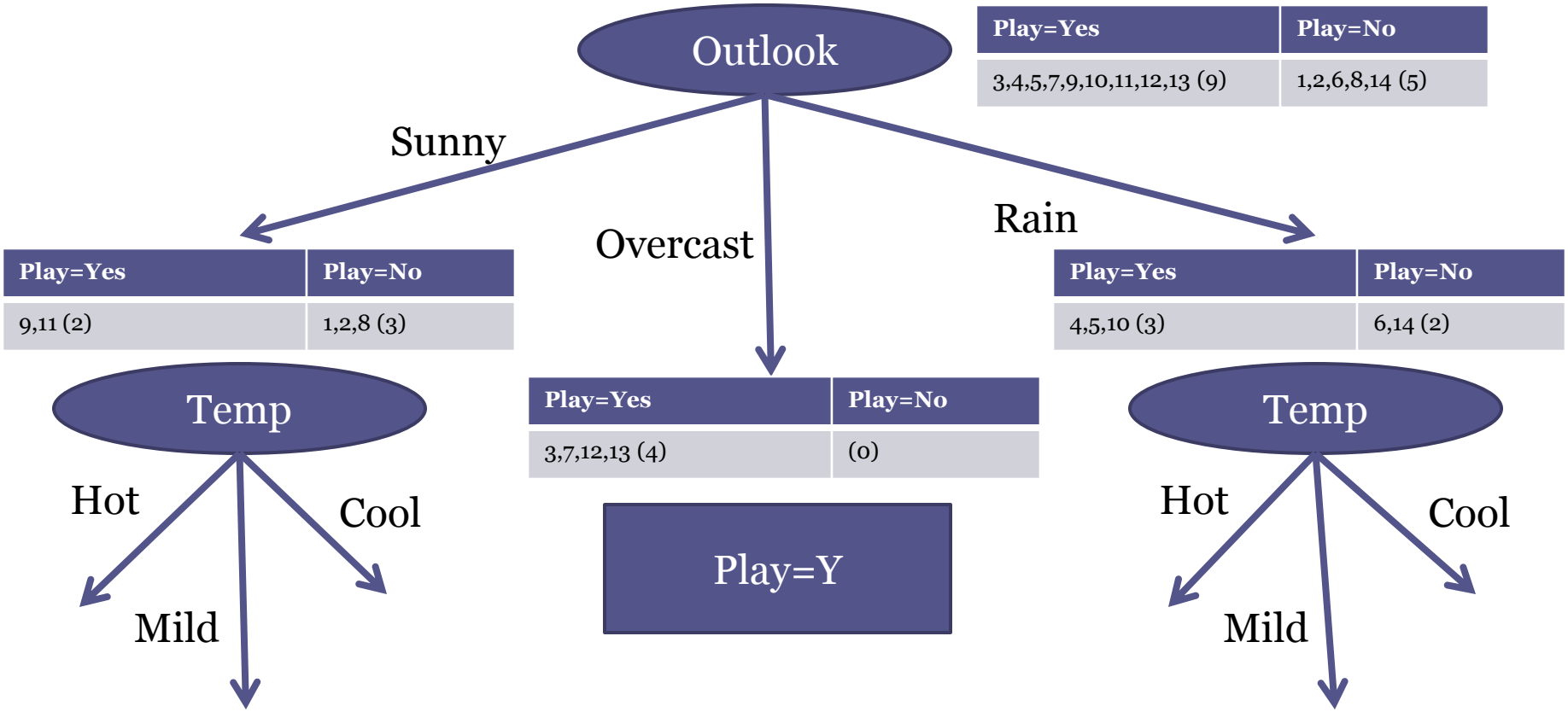
Algorithms

- ID3
  - Information gain
- C4.5 (=J48 in WEKA) (and See5/C5.0)
  - Information gain ratio
- Classification and regression tree (CART)
  - Gini gain
- Chi-squared automatic interaction detection (CHAID)

Day	Outlook	Temp	Humidty	Wind	Play?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example from Tom Mitchell's book

Naïve strategy of choosing attributes (i.e. choose the next available attribute)



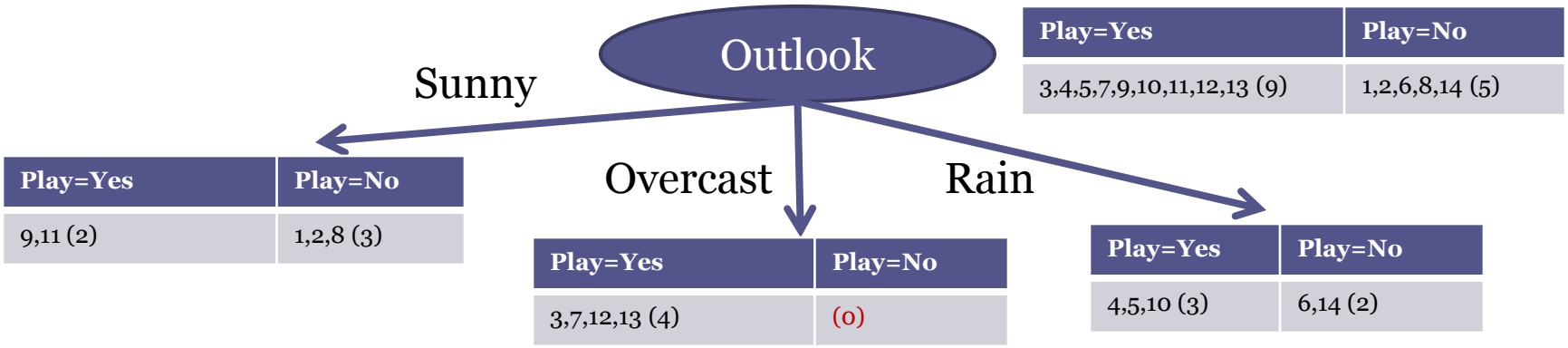
How to generate decision trees?

- Optimal one
  - Equal to (or harder than) NP-Hard
- Greedy one
    - Greedy means big questions first
    - Strategy – divide and conquer
  - Choose *an easy-to-understand test* such that divided sub-data sets by the chosen test are *the easiest to deal with*
    - Usually choose an attribute as a test
    - Usually adopt impurity measure to see how easy to deal with the sub-data sets
- Are there any other approaches? – there are many and open

Impurity criteria

- Entropy → Information Gain, Information Gain Ratio
  - Most popular
  - Entropy – Sum of -plogp
  - IG – Entropy(S) - Sum of Entropy(sub-data t) * |t|/|S|
  - IG favors Social Security Number or ID
  - Information Gain Ratio
- Gini index → Gini Gain (used in CART)
  - Related with Area Under the Curve
  - GG – 1 - Sum of fractions^2
- Misclassification rate
  - (misclassified instances)/(all instances)
  - Problematic – lead to many indistinguishable splits (where other splits are more desirable)

Using IG for choosing attributes



IG(S) = Entropy(S) – Sum(|S_i|/|S|*Entropy(S_i))

IG(Outlook)= Entropy(Outlook)
-|Sunny|/|Outlook|*Entropy(Sunny)
-|Overcast|/|Outlook|*Entropy(Overcast)
-|Rain|/|Outlook|*Entropy(Rain)

Entropy(Outlook) =-(9/14)*log(9/14)-(5/14)*log(5/14)
|Sunny|/|Outlook|*Entropy(Sunny) = 5/14*(-(2/5)*log(2/5)-(3/5)*log(3/5))
|Overcast|/|Outlook|*Entropy(Overcast) = 4/14*(-(4/4)*log(4/4)-(0/4)*log(0/4))
|Rain|/|Outlook|*Entropy(Rain) = 5/14*(-(3/5)*log(3/5)-(2/5)*log(2/5))

Overfitting

- Training set error
  - Error of the classifier on the training data
  - It is a bad idea to use up all data for training. → You will be out of data to evaluate the learning algorithm.
- Test set error
  - Error of the classifier on the test data
  - Jackknife – Use n-1 examples to learn and 1 to test. Repeat n times.
  - x-folds stratified cross-validation – Divide data into x-folds with the same proportion of class. x-1 folds to train and 1 fold to test. Repeat x times.
- Overfitting
  - The input data is incomplete (Quine)
  - The input data do not reflect all possible cases.
  - The input data can include noise.
  - I.e. fit the classifier tightly to the input data is a bad idea.
- Occam's razor
  - Old axiom used to prove the existence of God.
  - "plurality should not be posited without necessity"

Pruning

- Prepruning (=forward pruning)
- Postpruning (=backward pruning)
  - Subtree raising
  - Subtree replancement
  - Reduced error pruning

Pros and Cons

- Pros
  - Easy to understand
  - Fast learning algorithms (because they are greedy)
  - Robust to noise
  - Good accuracy
- Cons
  - Unstable
  - Hard to represent some functions (Parity, XOR, etc.)
  - Duplication in subtrees
  - Cannot be used to express all first order logic because the test cannot refer to two or more different objects

Generation of data from a decision tree (based on the definition #2)

- Decision tree with support for each node → Rule set
  - support = # of training instances assigned for a node
- Rule set → Instances
- In this way, one can combine multiple decision trees by combining rule sets

- cf. Bayesian classifiers → Fractional instances

Extensions and further considerations

- Extensions
  - Alternating decision tree
  - Naïve Bayes Tree
  - Attribute Value Taxonomy guided Decision Tree
  - Recursive Naïve Bayes
  - and many more
- Further Researches
  - Decision graph
  - Bottom up generation of decision tree
  - Evolutionary construction of decision tree
  - Integrating two decision trees
  - and many more