Machine Learning III
Decision Tree Induction

CSE 473

Machine Learning Outline

- Machine learning:
  - √ Function approximation
  - √ Bias
- Supervised learning
  - √ Classifiers & concept learning
  - √ Version Spaces (restriction bias)
  - Decision-trees induction (pref bias)
- Overfitting
- Ensembles of classifiers

Two Strategies for ML

- **Restriction bias**: use prior knowledge to specify a restricted hypothesis space.
  - Version space algorithm over conjunctions.
- **Preference bias**: use a broad hypothesis space, but impose an ordering on the hypotheses.
  - Decision trees.

Decision Trees

- Convenient Representation
  - Developed with learning in mind
  - Deterministic
- Expressive
  - Equivalent to propositional DNF
  - Handles discrete and continuous parameters
- Simple learning algorithm
  - Handles noise well
  - Classify as follows
    - **Constructive (build DT by adding nodes)**
    - **Eager**
    - **Batch (but incremental versions exist)**

Concept Learning

- E.g. Learn concept "Edible mushroom"
  - Target Function has two values: T or F
- Represent concepts as decision trees
- Use ***hill climbing search***
- Thru space of ***decision trees***
  - Start with simple concept
  - Refine it into a complex concept as needed

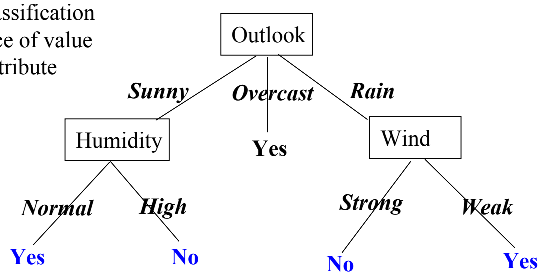
Experience: "Good day for tennis"

Day	Outlook	Temp	Humid	Wind	PlayTennis?
d1	s	h	h	w	n
d2	s	h	h	s	n
d3	o	h	h	w	y
d4	r	m	h	w	y
d5	r	c	n	w	y
d6	r	c	n	s	y
d7	o	c	n	s	y
d8	s	m	h	w	n
d9	s	c	n	w	y
d10	r	m	n	w	y
d11	s	m	n	s	y
d12	o	m	h	s	y
d13	o	h	n	w	y
d14	r	m	h	s	n

Decision Tree Representation

Good day for tennis?

Leaves = classification
Arcs = choice of value for parent attribute

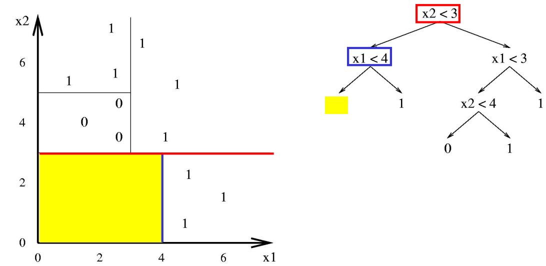


Decision tree is equivalent to logic in disjunctive normal form

G-Day ⇔ (Sunny ∧ Normal) ∨ Overcast ∨ (Rain ∧ Weak)

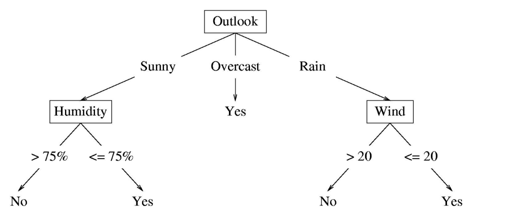
Decision Tree Decision Boundaries

Decision trees divide the feature space into axis-parallel rectangles, and label each rectangle with one of the K classes.



Decision Tree Hypothesis Space

If the features are continuous, internal nodes may test the value of a feature against a threshold.



Decision Trees Provide Variable-Size Hypothesis Space

As the number of nodes (or depth) of tree increases, the hypothesis space grows

- **depth 1** ("decision stump") can represent any boolean function of one feature.
- **depth 2** Any boolean function of two features; some boolean functions involving three features (e.g., $(x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_3)$)
- **etc.**

DT Learning as Search

- Nodes
  Decision Trees
- Operators
  Tree Refinement: Sprouting the tree
- Initial node
  Smallest tree possible: a single leaf
- Heuristic?
  Information Gain
- Goal?
  Best tree possible (???)

What is the Simplest Tree?

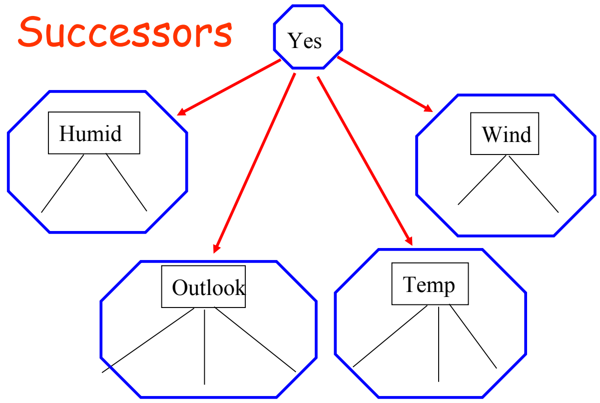
Day	Outlook	Temp	Humid	Wind	Play?
d1	s	h	h	w	n
d2	s	h	h	s	n
d3	o	h	h	w	y
d4	r	m	h	w	y
d5	r	c	n	w	y
d6	r	c	n	s	y
d7	o	c	n	s	y
d8	s	m	h	w	n
d9	s	c	n	w	y
d10	r	m	n	w	y
d11	s	m	n	s	y
d12	o	m	h	s	y
d13	o	h	n	w	y
d14	r	m	h	s	n

How good?

[10+, 4-]

Means:
correct on 10 examples
incorrect on 4 examples

Successors

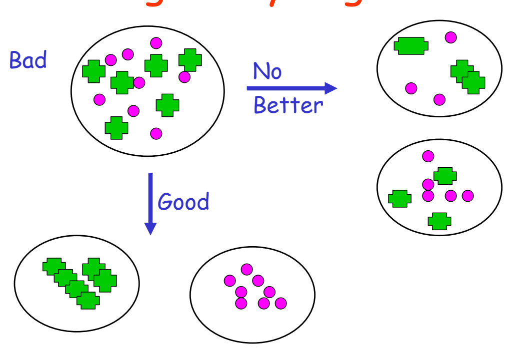


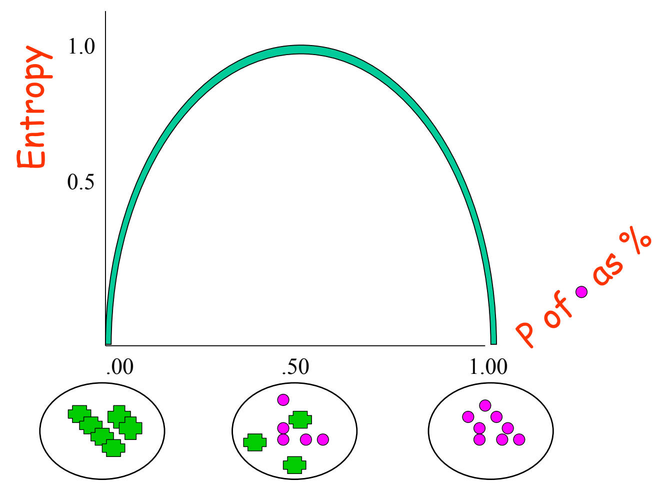
Which attribute should we use to split?

To be decided:

- How to choose best attribute?
  - Information gain
  - Entropy (disorder)
- When to stop growing tree?

Disorder is bad
Homogeneity is good





Entropy (disorder) is bad
Homogeneity is good

- Let S be a set of examples
- Entropy(S) = $-P \log_2(P) - N \log_2(N)$
  - where **P** is proportion of pos example
  - and **N** is proportion of neg examples
  - and 0 log 0 == 0
- Example: S has 10 pos and 4 neg
  - Entropy([10+, 4-]) = $-(10/14) \log_2(10/14) - (4/14)\log_2(4/14)$
  - = 0.863

Information Gain

- ***Measure of expected reduction in entropy***
- **Resulting from splitting along an attribute**

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

Where Entropy(S) = $-P \log_2(P) - N \log_2(N)$

Gain of Splitting on Wind

Values(wind)=weak, strong

S = [10+, 4-]

S_{weak} = [6+, 2-]

S_s = [3+, 3-]

Gain(S, wind)

$= Entropy(S) - \sum_{v \in \{weak, s\}} (|S_v| / |S|) \, Entropy(S_v)$

$= Entropy(S) - 8/14 \, Entropy(S_{weak}) - 6/14 \, Entropy(S_s)$

= 0.863 - (8/14) 0.811 - (6/14) 1.00

= -0.029

Day	Wind	Tennis?
d1	weak	n
d2	**s**	n
d3	weak	yes
d4	weak	yes
d5	weak	yes
d6	**s**	yes
d7	**s**	yes
d8	weak	n
d9	weak	yes
d10	weak	yes
d11	**s**	yes
d12	**s**	yes
d13	weak	yes
d14	**s**	n

Gain of Split on Humidity

Day	Outlook	Temp	Humid	Wind	Play?
d1	s	h	**h**	w	n
d2	s	h	**h**	s	n
d3	o	h	**h**	w	*y*
d4	r	m	**h**	w	*y*
d5	r	c	*n*	w	*y*
d6	r	c	*n*	s	*y*
d7	o	c	*n*	s	*y*
d8	s	m	**h**	w	n
d9	s	c	*n*	w	*y*
d10	r	m	*n*	w	*y*
d11	s	m	*n*	s	*y*
d12	o	m	**h**	s	*y*
d13	o	h	*n*	w	*y*
d14	r	m	**h**	s	n

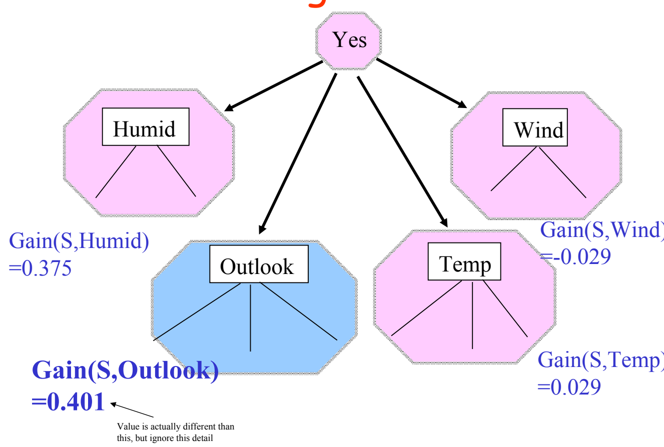
$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} (|S_v| / |S|) \, Entropy(S_v)$$

Where $Entropy(S) = -P \log_2(P) - N \log_2(N)$

Is…

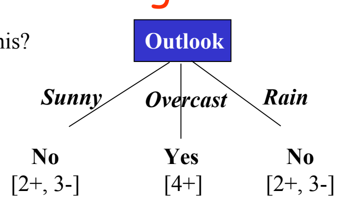
- Entropy([4+,3-]) = .985
- Entropy([7,0]) =
- Gain = 0.863- .985/2 = 0.375

Evaluating Attributes

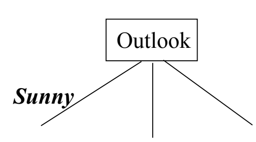


Resulting Tree ….

Good day for tennis?

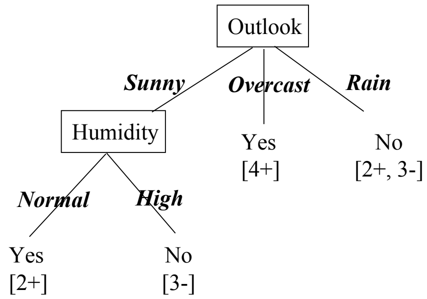


Recurse!



Day	Temp	Humid	Wind	Tennis?
d1	h	h	weak	n
d2	h	h	s	n
d8	m	h	weak	n
d9	c	n	weak	yes
d11	m	n	s	yes

One Step Later…



Decision Tree Algorithm

BuildTree(TraingData)
    Split(TrainingData)

Split(D)
    If (all points in D are of the same class)
        Then Return
    For each attribute A
        Evaluate splits on attribute A
    Use best split to partition D into D1, D2
    Split (D1)
    Split (D2)

Movie Recommendation

- Features?

Rambo								
Matrix								
Rambo 2								
•								
•								
•								

Issues

- Content *vs.* Social
- Non-Boolean Attributes
- Missing Data
- Scaling up

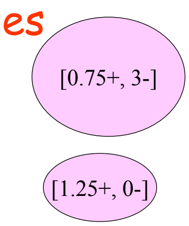
Missing Data 1

Day	Temp	Humid	Wind	Tennis?
d1	h	h	weak	n
d2	h	h	s	n
d8	m	h	weak	n
d9	c		weak	yes
d11	m	n	s	yes

- Don't use this instance for learning?
- Assign attribute …
    most common value at node, or
    most common value, … given classification

Fractional Values

Day	Temp	Humid	Wind	Tennis?
d1	h	h	weak	n
d2	h	h	s	n
d8	m	h	weak	n
d9	c		weak	yes
d11	m	n	s	yes



- 75% h and 25% n
- Use in gain calculations
- Further subdivide if other missing attributes
- Same approach to classify test ex with missing attr
    Classification is most probable classification
    Summing over leaves where it got divided

Non-Boolean Features

- Features with multiple discrete values
  - Construct a multi-way split
  - Test for one value *vs.* all of the others?
  - Group values into two disjoint subsets?
- Real-valued Features
  - Discretize?
  - Consider a threshold split using observed values?

Attributes with many values

Problem:

- If attribute has many values, $Gain$ will select it
- Imagine using $Date = Jun\_3\_1996$ as attribute

- So many values that it
  - Divides examples into tiny sets
  - Each set likely uniform → high info gain
  - But poor predictor…
- Need to penalize these attributes

One approach: Gain ratio

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

$$SplitInformation(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

where S_i is subset of S for which A has value v_i

SplitInfo ≅ entropy of S *wrt* values of A

(Contrast with entropy of S *wrt* **target** value)

⇓ attribs with many uniformly distrib values

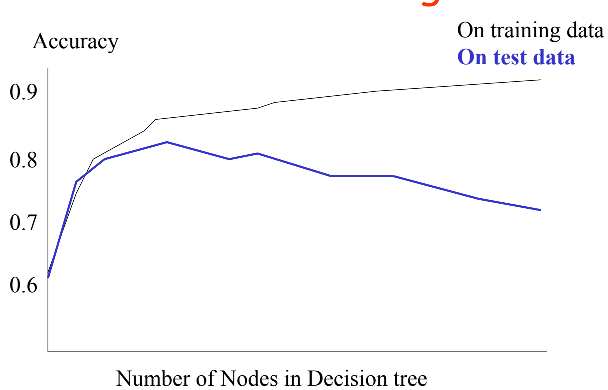
e.g. if A splits S uniformly into n sets

SplitInformation = $\log_2(n)$… = 1 for Boolean

Machine Learning Outline

- Machine learning:
- Supervised learning
- Overfitting
  - What is the problem?
  - Reduced error pruning
- Ensembles of classifiers

Overfitting



Overfitting 2

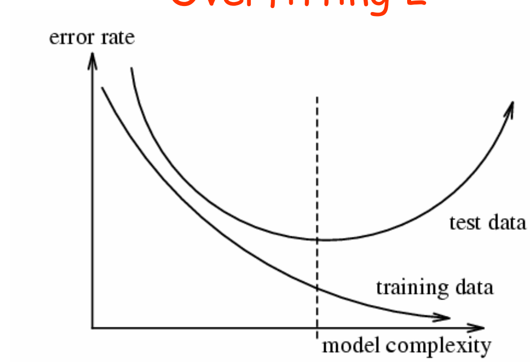


Figure from w.w.cohen

Overfitting…

- DT is *overfit* when exists another DT' and
  - DT has ***smaller*** error on training examples, but
  - DT has ***bigger*** error on test examples
- Causes of overfitting
  - Noisy data, or
  - Training set is too small

Avoiding Overfitting

How can we avoid overfitting?

- Stop growing when data split not statistically significant
- Grow full tree, then post-prune

How to select "best" tree:

- Measure performance over training data
- Measure performance over separate validation data set
- Add complexity penalty to performance measure

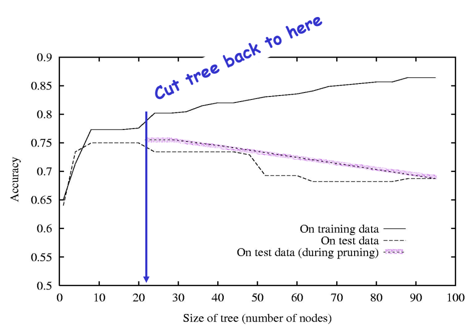
Reduced-Error Pruning

Split data into *training* and *validation* set

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 that most improves *validation* set accuracy

Effect of Reduced-Error Pruning



Machine Learning Outline

- Machine learning:
- Supervised learning
- Overfitting
- Ensembles of classifiers
  - Bagging
  - Cross-validated committees
  - Boosting