Machine Learning 10-601

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

August 28, 2017

Today:
- What is machine learning?
- Decision tree learning
- Course logistics

Readings:
- "Machine Learning: Trends, perspectives and prospects" Jordan & Mitchell 2015
- Mitchell, Chapter 3
- Bishop, Chapter 14.4

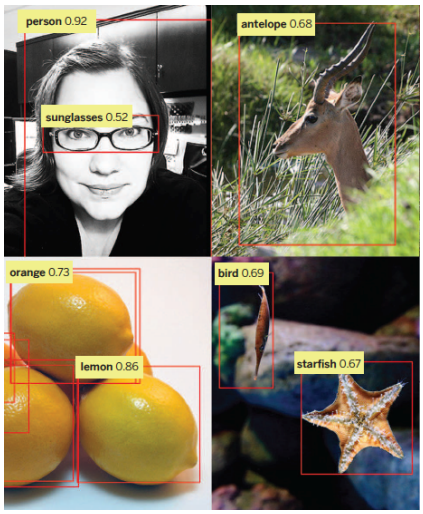
Piazza: https://piazza.com/cmu/fall2017/10601b/home

Machine Learning:

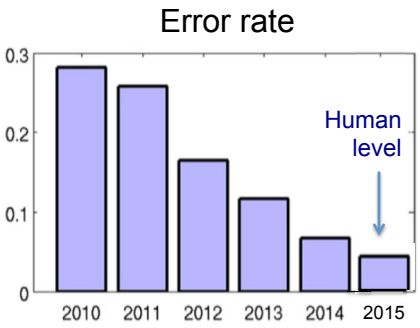
Study of algorithms that
- improve their performance P
- at some task T
- with experience E

well-defined learning task: <P,T,E>

Computer Vision



Imagenet Visual Recognition Challenge



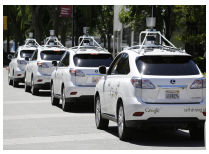
Speech Recognition

October 2016: Microsoft reports reaching human-level accuracy of 94.1% at standard switchboard task.

March 2017: IBM reports 94.5% accuracy

Robots

Factories, Land, Air, Sea, Mines, Homes



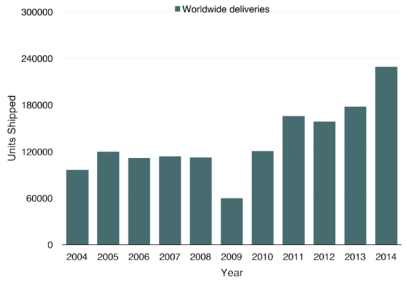



FIGURE 2.4 Worldwide shipping of robots over time. SOURCE: International Federation of Robotics, 2015.

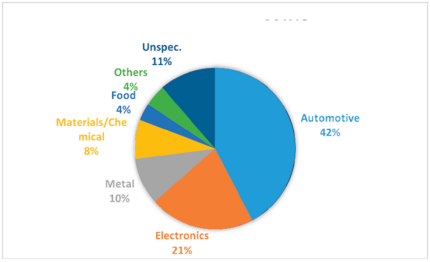
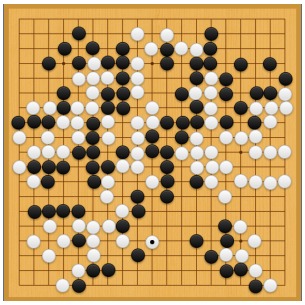


FIGURE 2.5 Robot application areas in 2015. SOURCE: Data from International Federation of Robotics, 2015.

Games and reasoning

Chess

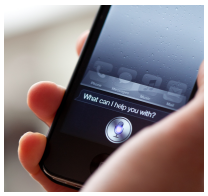


Go

Jeopardy

Poker

The key: Machine Learning

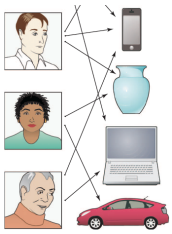


conversational agents

medical diagnosis

fraud detection

translation



recommendations

- Deep neural networks
- Support Vector Machines
- Bayesian networks
- Hidden Markov models
- Decision forests
- Gaussian mixture model
- Expectation maximization
- ....

Machine Learning - Theory

PAC Learning Theory
(supervised concept learning)

examples (*m*)

representational complexity (*H*)

error rate (ε)

failure probability (δ)

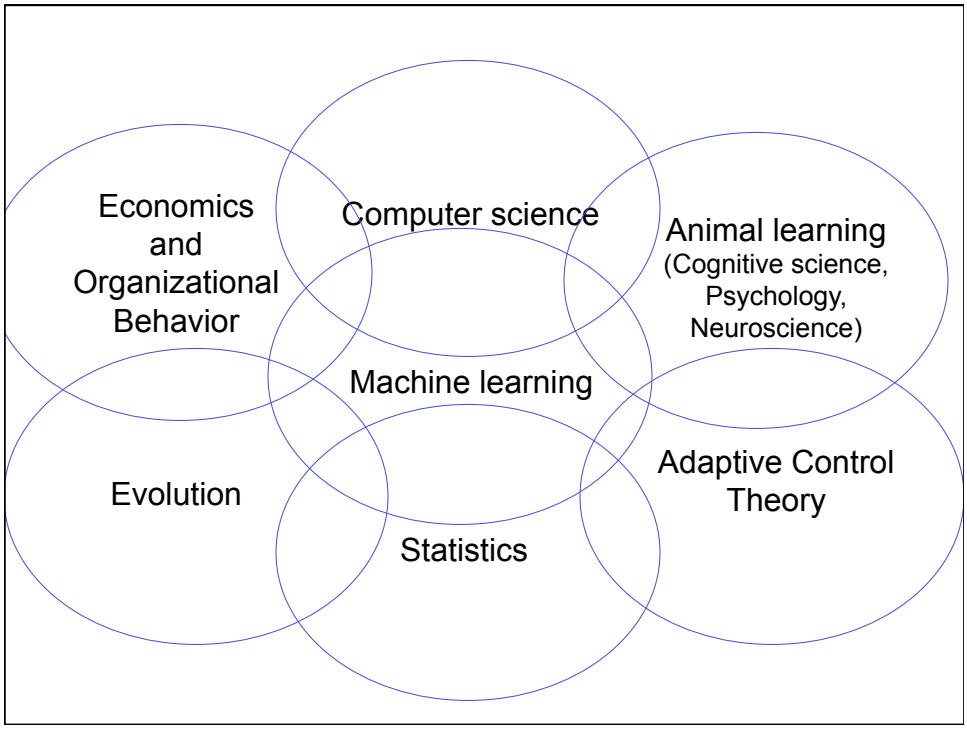
$$m \geq \frac{1}{\epsilon}(\ln |H| + \ln(1/\delta))$$

Other theories for

- Reinforcement skill learning
- Semi-supervised learning
- Active student querying
- …

… also relating:

- # of mistakes during learning
- learner's query strategy
- convergence rate
- computational demands
- asymptotic performance
- bias, variance, Bayesian priors
- VC dimension



We'll cover in this course

Algorithms:

- Decision trees
- Bayes classifiers
- Logistic regression
- Deep neural networks
- Graphical models
- Expectation maximization
- Support Vector Machines
- Kernel regression
- PCA, Matrix factorization
- Markov models
- Reinforcement learning

Concepts:

- Statistical estimation
- Overfitting
- Representation learning
- Probabilistic models
- Maximum margin models
- Probably approximately correct learning
- VC dimension
- Role of unlabeled data
- Optimization

Highlights of Course Logistics

Homework 1
- Out already, due midnight Friday
- PLEASE take this seriously

Grading:
- 30% homeworks
- 35% midterm exam
- 35% final exam

Academic integrity:
- Cheating → You will fail class. Also be expelled from CMU if possible.
- Working together to discuss HW: good
- Writing HW solutions together: No!

Late homework:
- full credit when due
- half credit next 48 hrs
- zero credit after that
- we'll delete your lowest HW score
- must turn in at least n-1 of the n homeworks, even if late

Being present at exams:
- You must be there – plan now.
- Midterm: October 25
- Final: TBA by registrar
- Can we schedule you a separate final? No.

Function Approximation and Decision tree learning

Function approximation

Problem Setting:

- Set of possible instances X
- Unknown target function $f : X \rightarrow Y$
- Set of candidate hypotheses $H=\{ h \mid h : X \rightarrow Y \}$

superscript: i^{th} training example

Input:

- Training examples $\{<x^{(i)}, y^{(i)}>\}$ of unknown target function f

Output:

- Hypothesis $h \in H$ that best approximates target function f

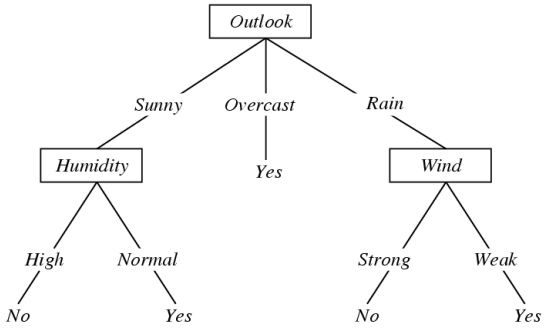
Simple Training Data Set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
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

A Decision tree for

f: <Outlook, Temperature, Humidity, Wind> → PlayTennis?

< X_1 X_2 X_3 X_4 > → Y

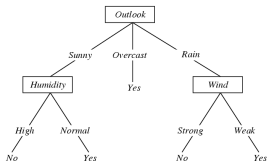


Each internal node: test one discrete-valued attribute X_i

Each branch from a node: selects one value for X_i

Each leaf node: predict Y (or P(Y|X ∈ leaf))

Decision Tree Learning



Problem Setting:

- Set of possible instances X
  - each instance x in X is vector of discrete-valued features $x = < x_1, x_2 \ldots x_n >$
- Unknown target function $f : X \rightarrow Y$
  - Y is discrete-valued
- Set of function hypotheses $H = \{ h \mid h : X \rightarrow Y \}$
  - each hypothesis h is a decision tree

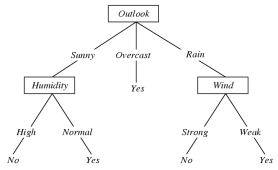
Input:

- Training examples $\{<x^{(i)}, y^{(i)}>\}$ of unknown target function f

Output:

- Hypothesis $h \in H$ that best approximates target function f

Decision Trees



Suppose $X = <X_1, \ldots X_n>$
where X_i are boolean-valued variables

How would you represent $Y = X_2 X_5$? $Y = X_2 \vee X_5$

How would you represent $X_2 X_5 \vee X_3 X_4 (\neg X_1)$

A Tree to Predict C-Section Risk

[Sims et al., 2000]

Learned from medical records of 1000 women
Negative examples are C-sections

```
[833+,167-] .83+ .17-
Fetal_Presentation = 1: [822+,116-] .88+ .12-
| Previous_Csection = 0: [767+,81-] .90+ .10-
| | Primiparous = 0: [399+,13-] .97+ .03-
| | Primiparous = 1: [368+,68-] .84+ .16-
| | | Fetal_Distress = 0: [334+,47-] .88+ .12-
| | | | Birth_Weight < 3349: [201+,10.6-] .95+
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+
| | | Fetal_Distress = 1: [34+,21-] .62+ .38-
| Previous_Csection = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-
```

Top-Down Induction of Decision Trees

[ID3, C4.5, Quinlan]

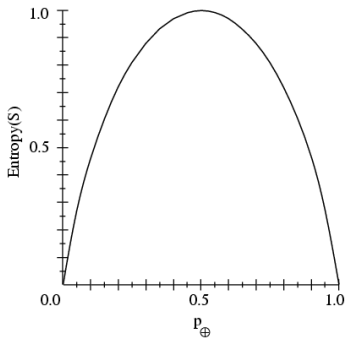
node = Root

Main loop:

1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?

Sample Entropy



- S is a sample of training examples
- $p_{\oplus}$ is the proportion of positive examples in S
- $p_{\ominus}$ is the proportion of negative examples in S
- Entropy measures the impurity of S

$$Entropy(S) \equiv H(S) \equiv -p_{\oplus} \log_2 p_{\oplus} \; - p_{\ominus} \log_2 p_{\ominus}$$

Entropy

Entropy $H(X)$ of a random variable X

# of possible values for X

$$H(X) = -\sum_{i=1}^{n} P(X=i)\log_2 P(X=i)$$

$H(X)$ is the expected number of bits needed to encode a randomly drawn value of X (under most efficient code)

Why? Information theory:

- Most efficient possible code assigns $-\log_2 P(X=i)$ bits to encode the message $X=i$
- So, expected number of bits to code one random X is:

$$\sum_{i=1}^{n} P(X=i)(-\log_2 P(X=i))$$

Entropy

Entropy $H(X)$ of a random variable X

$$H(X) = -\sum_{i=1}^{n} P(X=i)\log_2 P(X=i)$$

Specific conditional entropy $H(X|Y=v)$ of X given $Y=v$:

$$H(X|Y=v) = -\sum_{i=1}^{n} P(X=i|Y=v)\log_2 P(X=i|Y=v)$$

Conditional entropy $H(X|Y)$ of X given Y :

$$H(X|Y) = \sum_{v \in values(Y)} P(Y=v)H(X|Y=v)$$

Mutual information (aka Information Gain) of X and Y :

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Information Gain is the mutual information *I(A,Y)* between input attribute A and target variable Y

Information Gain is the expected reduction in entropy of target variable Y for data sample S, due to sorting on variable A

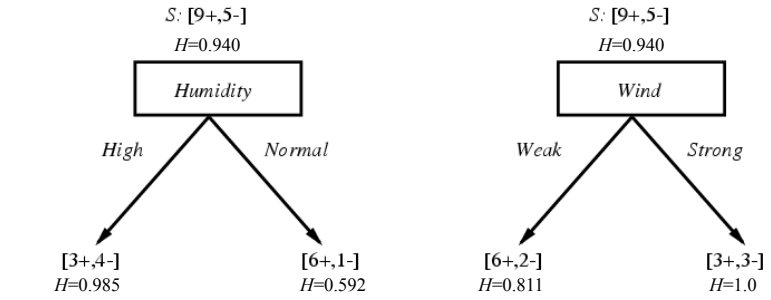
$$Gain(S, A) = I_S(A, Y) = H_S(Y) - H_S(Y|A)$$

Simple Training Data Set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
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

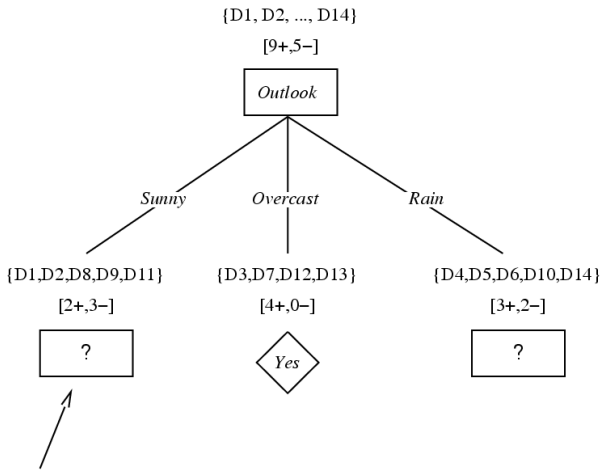
Selecting the Next Attribute

Which attribute is the best classifier?



Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048



Which attribute should be tested here?

S_{sunny} = {D1,D2,D8,D9,D11}

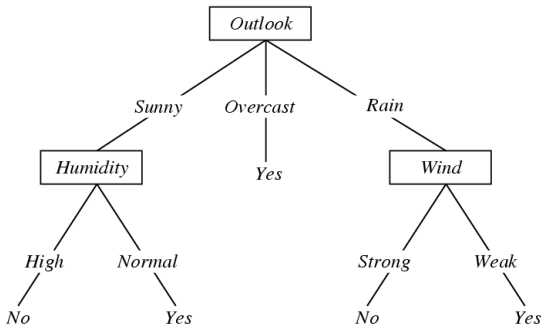
Gain (S_{sunny}, *Humidity*) = .970 − (3/5) 0.0 − (2/5) 0.0 = .970

Gain (S_{sunny}, *Temperature*) = .970 − (2/5) 0.0 − (2/5) 1.0 − (1/5) 0.0 = .570

Gain (S_{sunny}, *Wind*) = .970 − (2/5) 1.0 − (3/5) .918 = .019

Final Decision Tree for

f: <Outlook, Temperature, Humidity, Wind> → PlayTennis?

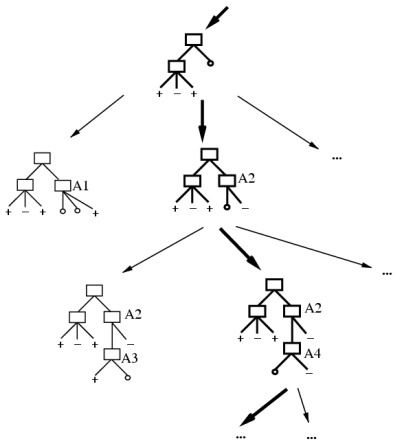


Each internal node: test one discrete-valued attribute X_i

Each branch from a node: selects one value for X_i

Each leaf node: predict Y

Which Tree Should We Output?



- ID3 performs heuristic search through space of decision trees
- It stops at smallest acceptable tree. Why?

Occam's razor: prefer the simplest hypothesis that fits the data

Why Prefer Short Hypotheses? (Occam's Razor)

Arguments in favor:

Arguments opposed:

Why Prefer Short Hypotheses? (Occam's Razor)

Argument in favor:

- Fewer short hypotheses than long ones

→ a short hypothesis that fits the data is less likely to be a statistical coincidence

→ highly probable that *some* sufficiently complex hypothesis will fit the data

Argument opposed:

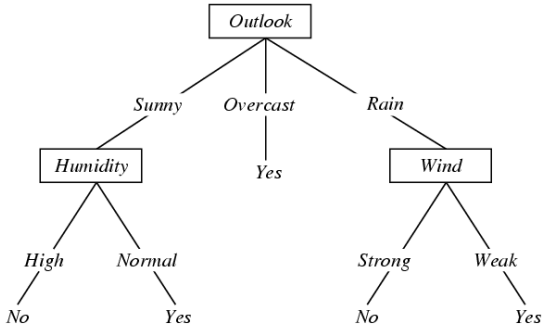
- Also fewer hypotheses with prime number of nodes and attributes beginning with "Z"
- What's so special about "short" hypotheses?

Overfitting in Decision Trees

Consider adding noisy training example #15:

$Sunny,\ Hot,\ Normal,\ Strong,\ PlayTennis = No$

What effect on earlier tree?

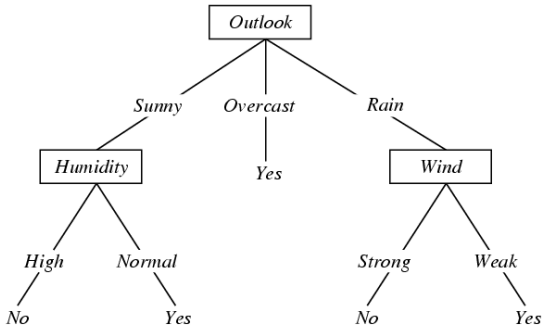


Overfitting in Decision Trees

Consider adding noisy training example #15:

$Sunny,\ Hot,\ Normal,\ Strong,\ PlayTennis = No$

What effect on earlier tree?



Overfitting

Consider a hypothesis h and its

- Error rate over training data: $error_{train}(h)$
- True error rate over all data: $error_{true}(h)$

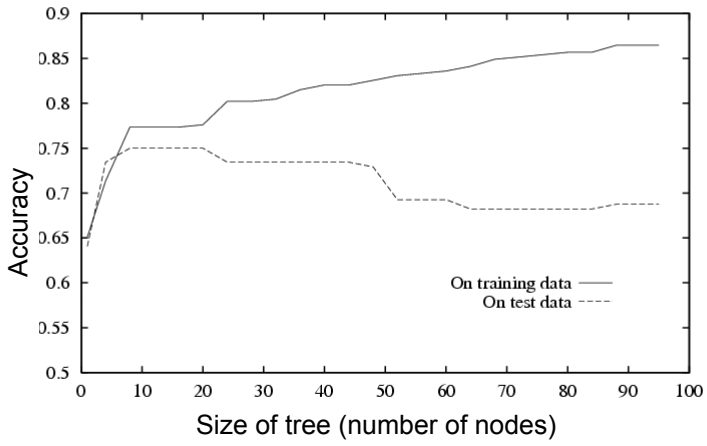
We say h overfits the training data if

$$error_{true}(h) > error_{train}(h)$$

Amount of overfitting =

$$error_{true}(h) - error_{train}(h)$$

Overfitting in Decision Tree Learning



Avoiding Overfitting

How can we avoid overfitting?

- stop growing when data split not statistically significant
- grow full tree, then post-prune

Reduced-Error Pruning

Split data into *training* and *validation* set

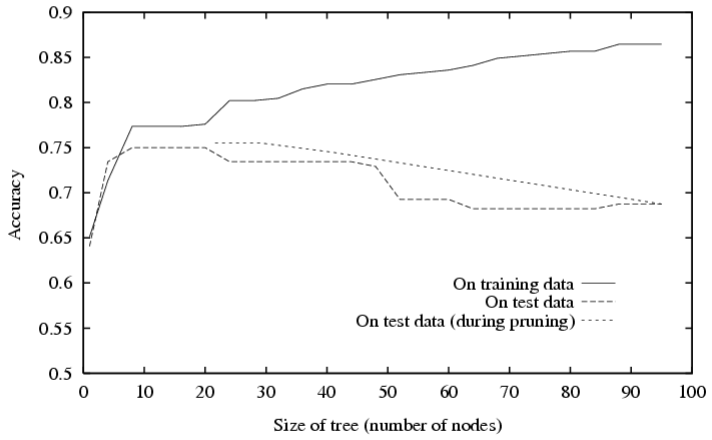
Create tree that classifies *training* set correctly

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

- produces smallest version of most accurate subtree
- What if data is limited?

Effect of Reduced-Error Pruning

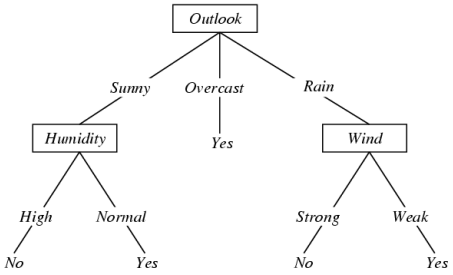


Rule Post-Pruning

1. Convert tree to equivalent set of rules
2. Prune each rule independently of others
3. Sort final rules into desired sequence for use

Perhaps most frequently used method (e.g., C4.5)

Converting A Tree to Rules



Continuous Valued Attributes

Create a discrete attribute to test continuous

- $Temperature = 82.5$
- $(Temperature > 72.3) = t, f$

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

Decision Forests

Key idea: learn a collection of decision trees, then let them vote

- Each tree trained on random subset of training examples and features
- Addresses overfitting!
- Popular method in real-world practice
  - human pose recognition in Microsoft kinect
  - classifying loan applications (accept/reject)
  - classify disease from gene expression data

Decision Forests

Key idea: learn a collection of decision trees, then let them vote

- Each tree trained on random subset of training e
- A
- P
  -
  -
  -

more to come

- later lecture on boosting and ensemble methods…

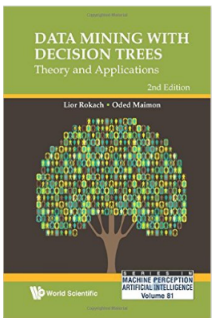
You should know:

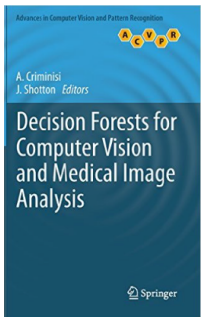
- Well posed function approximation problems:
  - Instance space, X
  - Sample of labeled training data { $<x^{(i)}, y^{(i)}>$ }
  - Hypothesis space, H = { f: X→Y }

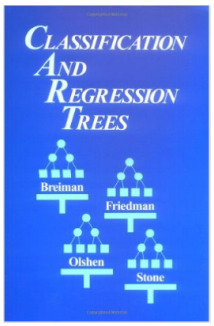
- Learning is a search/optimization problem over H
  - Various objective functions to define the goal
    - minimize training error (0-1 loss)
    - minimize validation error (0-1 loss)
    - among hypotheses that minimize error, select smallest (?)

- Decision tree learning
  - Greedy top-down learning of decision trees (ID3, C4.5, ...)
  - Overfitting and post-pruning
  - Extensions… to continuous values, probabilistic classification
  - Widely used commercially: decision *forests*

Further Reading…







Questions to think about (1)

- Consider target function f: <x1,x2> → y, where x1 and x2 are real-valued, y is boolean. What is the set of decision surfaces describable with decision trees that use each attribute at most once?

Questions to think about (2)

- ID3 and C4.5 are heuristic algorithms that search through the space of decision trees. Why not just do an exhaustive search?

Questions to think about (3)

- Why use Information Gain to select attributes in decision trees? What other criteria seem reasonable, and what are the tradeoffs in making this choice?