Classification Trees

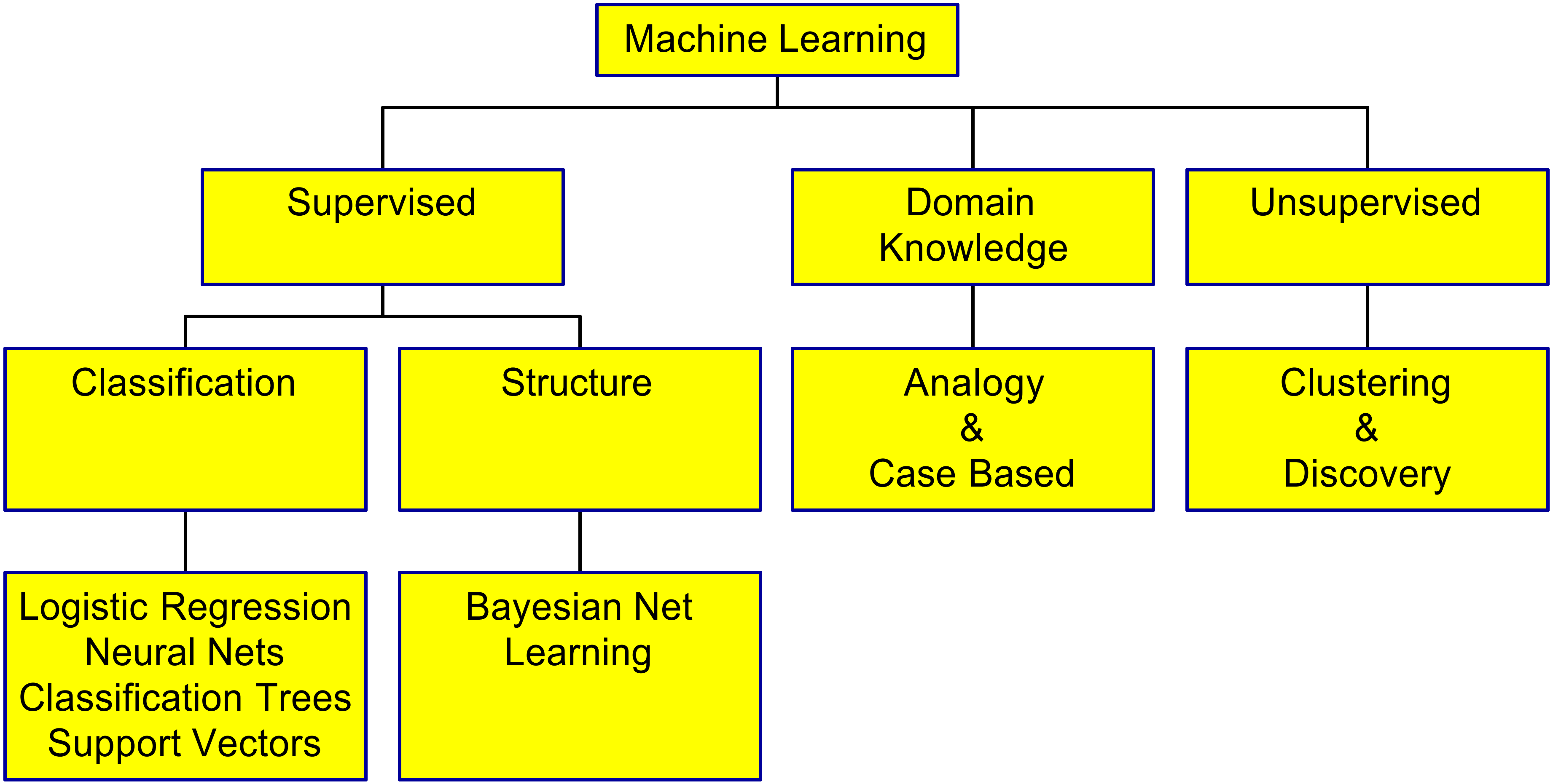
William Long
MIT Lab for Computer Science

Harvard-MIT Division of Health Sciences and Technology
HST.951J: Medical Decision Support

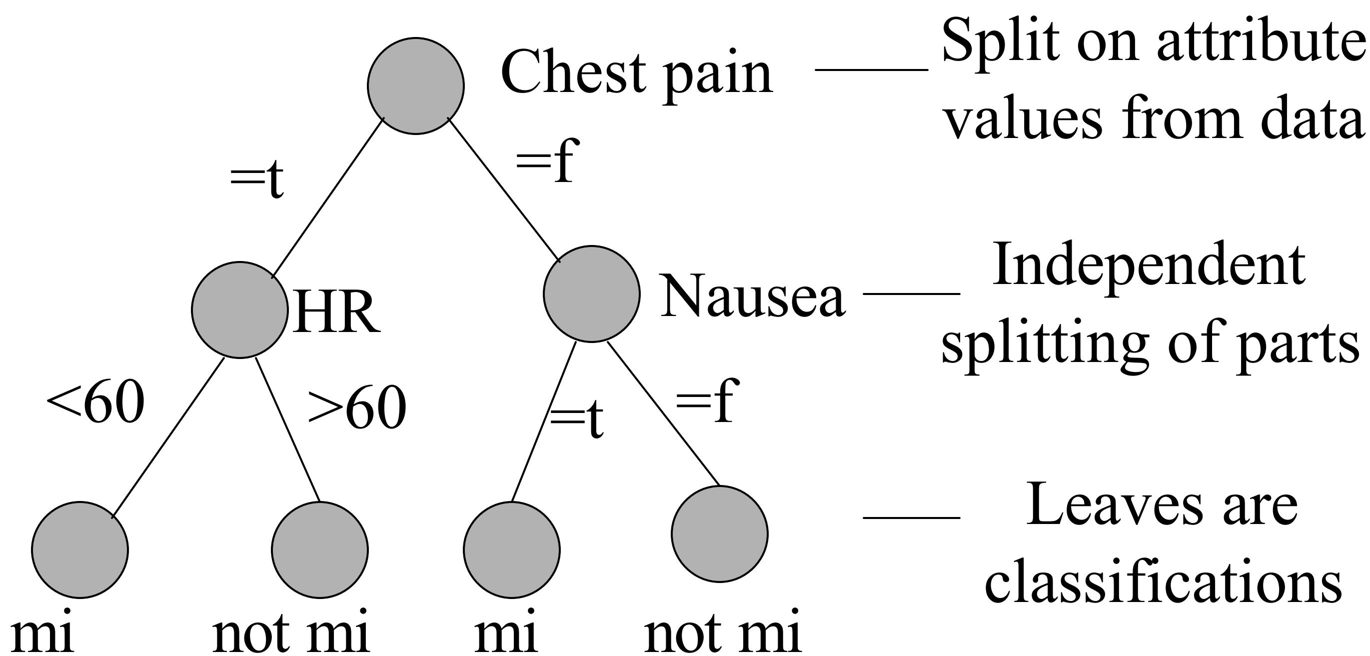
Data Mining

- Prediction vs Knowledge Discovery
- Statistics vs Machine Learning
- Phases:
  - Problem selection
  - Data preparation
  - Data reduction
  - Method application
  - Evaluation of results

Machine Learning



Classification Tree



Classification Trees

- Data consisting of learning set of cases
- Each case consists of a set of attributes with values and has a known class
- Classes are one of a small number of possible values, usually binary
- Attributes may be binary, multivalued, or continuous

Background

- Classification trees were invented twice
- Statistical community: CART
  - Brieman 1984
- Machine Learning community
  - Quinlan and others
  - Originally called “decision trees”

Example

Outlook	Temp	Humidity	Windy?	Class
sunny	75	70	yes	play
sunny	80	90	yes	dont play
sunny	85	85	no	dont play
sunny	72	95	no	dont play
sunny	69	70	no	play
cloudy	72	90	yes	play
cloudy	83	78	no	play
cloudy	64	65	yes	play
cloudy	81	75	no	play
rain	71	80	yes	dont play
rain	65	70	yes	dont play
rain	75	80	no	play
rain	68	80	no	play
rain	70	96	no	play

Example: classified

Outlook	Temp	Humidity	Windy?	Class
sunny	75	70	yes	play
sunny	80	90	yes	dont play
sunny	85	85	no	dont play
sunny	72	95	no	dont play
sunny	69	70	no	play
cloudy	72	90	yes	play
cloudy	83	78	no	play
cloudy	64	65	yes	play
cloudy	81	75	no	play
rain	71	80	yes	dont play
rain	65	70	yes	dont play
rain	75	80	no	play
rain	68	80	no	play
rain	70	96	no	play

Tree

- Outlook=sunny
  - Humidity <= 75: play
  - Humidity > 75: don't play
- Outlook=cloudy: play
- Outlook=rain
  - Windy=yes: don't play
  - Windy=no: play

Assumptions

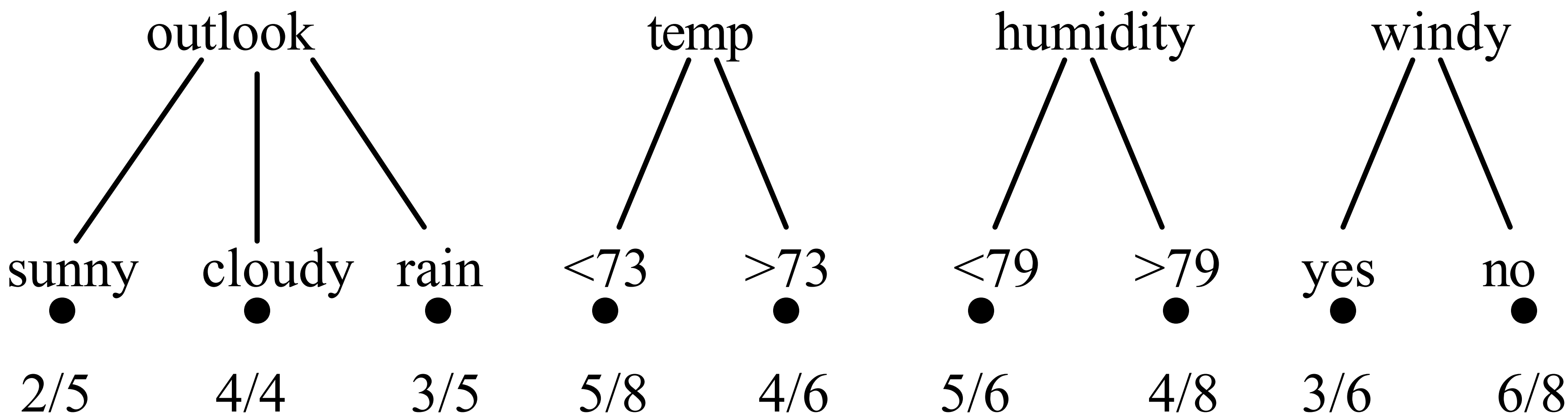
- Independence of partitions
- Branching on individual variables captures behavior
- No linearity assumption
- Classification
  - Although probabilities possible

Data Types

- Binary
- Multiple valued
  - N branches
  - Select subsets of values
- Continuous
  - Find cut point

Divide and Conquer

- 9/14: play



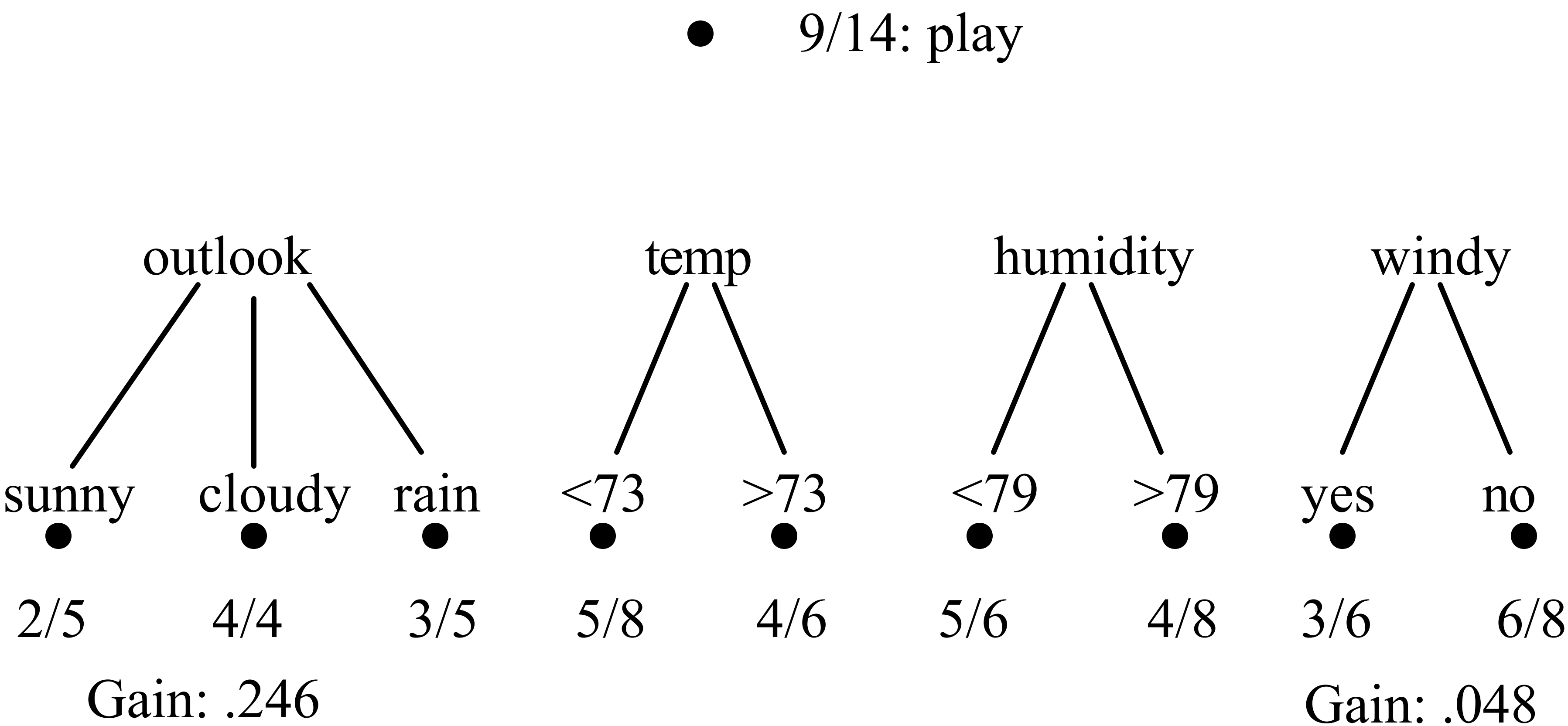
Splitting Criteria

- Information gain
  - gain = $-\Sigma\ p*\log_2 p$
- Gini statistic (weighted average impurity)
  - Gini = $1 - \Sigma\ p^2$
- Information gain ratio
- Others

Information Gain

- gain = $-\Sigma$ p*$\log_2$p
- info() = -9/14*$\log_2$(9/14)-5/14*$\log_2$(5/14)=.940 bits
- info(outlk) = 5/14*(-2/5*$\log_2$(2/5)-3/5*$\log_2$(3/5))
  + 4/14*(-4/4*$\log_2$(4/4)-0/4*$\log_2$(0/4)) +
  5/14*(-3/5*$\log_2$(3/5)-2/5*$\log_2$(2/5))
  = .694 bits
- gain = .246 bits
- vs info(windy) = .892 bits

Divide and Conquer



Continuous Variable

Temp	Class	Ratio	Gain	Gini
64	play	1/1+8/13	0.048	0.577
65	dont play	1/2+8/12	0.010	0.583
68	play	2/3+7/11	0.000	0.587
69	play	3/4+6/10	0.015	0.582
70	play	4/5+5/9	0.045	0.573
71	dont play	4/6+5/8	0.001	0.586
72	dont play	4/7+5/7	0.016	0.582
72	play	5/8+4/6	0.001	0.586
75	play	6/9+3/5	0.003	0.586
75	play	7/10+2/4	0.025	0.579
80	dont play	7/11+2/3	0.000	0.587
81	play	8/12+1/2	0.010	0.583
83	play	9/13+0/1	0.113	0.555
85	dont play			

Information Gain Ratio

- Attributes with multiple values favored by information gain
- Correction provided by analogous split info
- split info = $-\Sigma T*\log_2 T$
- split info = $-5/14*\log_2(5/14) -4/14*\log_2(4/14)- 5/14*\log_2(5/14)$ = 1.577 bits
- gain ratio = .246/1.577 = .156

Missing Values

- Adjust gain ratio
  - Gain(x) = prob A is known * info(x)
  - Split(x) = $-u*\log_2 u - \Sigma T*\log_2 t$
- Partitioning of training set cases
  - Use weights based on prevalence of values
- Classification
  - Use weights and sum the weighted leaves

Example with missing value

Outlook	Temp	Humidity	Windy?	Class
sunny	75	70	yes	play
sunny	80	90	yes	dont play
sunny	85	85	no	dont play
sunny	72	95	no	dont play
sunny	69	70	no	play
?	72	90	yes	play
cloudy	83	78	no	play
cloudy	64	65	yes	play
cloudy	81	75	no	play
rain	71	80	yes	dont play
rain	65	70	yes	dont play
rain	75	80	no	play
rain	68	80	no	play
rain	70	96	no	play

Frequencies for Outlook

	play	don't play	total
sunny	2	3	5
cloudy	3	0	3
rain	3	2	5
total	8	5	13

Information Gain With Missing

- $info() = -8/13*\log_2(8/13)-5/13*\log_2(5/13)$=.961 bits
- $info(outlk) = 5/13*(-2/5*\log_2(2/5)-3/5*\log_2(3/5))$
  $+ 3/13*(-3/3*\log_2(3/3)-0/3*\log_2(0/3))$ +
  $5/13*(-3/5*\log_2(3/5)-2/5*\log_2(2/5))$
  = .747 bits
- gain = 13/14*(.961-.747) = .199 bits
- $split = -5/14*\log_2(5/14) -3/14*\log_2(3/14) - 5/14*\log_2(5/14) -1/14*\log_2(1/14)$ = 1.809
- gain ratio = .199/1.809 = .110

Dividing Sunny

Outlook	Temp	Humidity	Windy?	Class	Weight
sunny	75	70	yes	play	1
sunny	80	90	yes	dont play	1
sunny	85	85	no	dont play	1
sunny	72	95	no	dont play	1
sunny	69	70	no	play	1
?	72	90	yes	play	5/13

What Next?

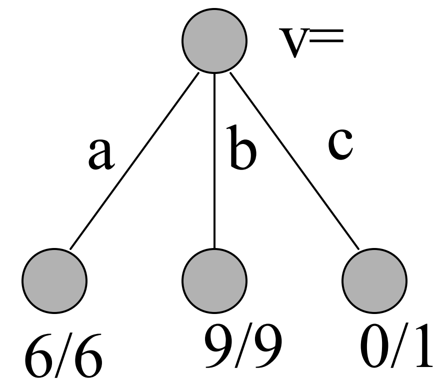
- Most trees are less than perfect
  - Variables don’t completely predict the outcome
  - Data is noisy
  - Data is incomplete (not all cases covered)
- Determine the best tree without overfitting or underfitting the data
  - Stop generating branches appropriately
  - Prune back the branches that aren’t justified

Pruning

- Use a test set for pruning
  - Cost complexity: (CART)
    - » $E/N + \alpha * L(tree)$
  - Reduced error
    - » $E' = \Sigma J + l(s)/2$
    - » $E + 1/2 < e' + se(e')$
- Cross validation
  - Split training set into N parts
  - Generate N trees, each leaving 1 part for validation

Pessimistic Pruning: (C4.5)

- Estimate errors: $\sum N * U_{CF}(E,N)$
- Example:
  - v=a: T (6) $U_{25\%}(0,6)=.206$
  - v=b: T (9) $U_{25\%}(0,9)=.143$
  - v=c: F (1) $U_{25\%}(0,1)=.750$
  - $6*.206+9*.143+1*.750=3.273$
  - vs $16*\ U_{25\%}(1,16)=16*.157=2.512$
  - => eliminate subtree



Developing a Tree for Ischemia

- Data:
  - learning set 3453 cases
  - test set 2320 cases
- Attributes: 52
- Types: dichotomous (chest pain), multiple (primary symptom), continuous (heart rate)
- Related attributes
- Missing values

Concerns

- Probability rather than classification
- Compare to other methods (LR, NN)
- Clinical usefulness

Probability of Disease

- Fraction at leaf estimates probability
- Small leaves give poor estimates
- Correction: $\frac{i(n'-i')+i'}{n(n'-i')+n'}$

Tree for Ischemia

```
STCHANGE = 1: ISCHEMIA (166.0/57.3)
STCHANGE = 6: ISCHEMIA (273.0/43.2)
STCHANGE = 0:
|  NCPNITRO = 2: NO-ISCHEMIA (1613.0/219.1)
|  NCPNITRO = 1:
|  |  SYMPTOM1 = 2: NO-ISCHEMIA (6.1/4.8)
|  |  SYMPTOM1 = 4: NO-ISCHEMIA (6.1/4.0)
|  |  SYMPTOM1 = 7: ISCHEMIA (3.0/2.4)
|  |  SYMPTOM1 = 8: ISCHEMIA (17.2/9.3)
|  |  SYMPTOM1 = 9: NO-ISCHEMIA (52.5/16.8)
|  |  SYMPTOM1 = 1:
|  |  |  SEX = 1: NO-ISCHEMIA (10.1/3.4)
|  |  |  SEX = 2: ISCHEMIA (8.1/4.4)
|  |  SYMPTOM1 = 3:
|  |  |  AGE <= 63 : ISCHEMIA (7.0/4.2)
|  |  |  AGE > 63 : NO-ISCHEMIA (7.1/3.2)
|  |  SYMPTOM1 = 10:
|  |  |  SEX = 2: NO-ISCHEMIA (135.5/55.8)
|  |  |  SEX = 1:
|  |  |  |  TWAVES = 1: NO-ISCHEMIA (1.0/0.9)
|  |  |  |  TWAVES = 2: ISCHEMIA (46.0/15.6)
|  |  |  |  TWAVES = 4: ISCHEMIA (10.0/6.4)
|  |  |  |  TWAVES = 0:
|  |  |  |  |  AGE > 76 : NO-ISCHEMIA (12.7/4.7)
|  |  |  |  |  AGE <= 76 :
|  |  |  |  |  |  SYSBP > 178 : ISCHEMIA (10.2/4.7)
|  |  |  |  |  |  SYSBP <= 178 :
|  |  |  |  |  |  |  AGE <= 52 : NO-ISCHEMIA (19.0/10.3)
|  |  |  |  |  |  |  AGE > 52 :
|  |  |  |  |  |  |  |  AGE <= 61 : ISCHEMIA (27.6/12.4)
|  |  |  |  |  |  |  |  AGE > 61 :
|  |  |  |  |  |  |  |  |  AGE <= 66 : NO-ISCHEMIA (13.0/5.8)
|  |  |  |  |  |  |  |  |  AGE > 66 : ISCHEMIA (12.9/7.7)
|  |  |  |  TWAVES = 3:
|  |  |  |  |  SYSBP <= 126 : NO-ISCHEMIA (6.0/4.0)
|  |  |  |  |  SYSBP > 126 : ISCHEMIA (17.0/7.1)
STCHANGE = 2:
|  SYMPTOM1 = 1: NO-ISCHEMIA (12.2/3.7)
|  SYMPTOM1 = 2: NO-ISCHEMIA (1.0/0.9)
|  SYMPTOM1 = 4: NO-ISCHEMIA (10.1/2.2)
|  SYMPTOM1 = 6: ISCHEMIA (1.0/0.9)
|  SYMPTOM1 = 7: NO-ISCHEMIA (3.0/2.4)
|  SYMPTOM1 = 8: ISCHEMIA (10.1/2.1)
|  SYMPTOM1 = 10: ISCHEMIA (163.2/62.0)
|  SYMPTOM1 = 3:
|  |  AGE <= 67 : ISCHEMIA (9.1/5.5)
|  |  AGE > 67 : NO-ISCHEMIA (13.1/4.9)
|  SYMPTOM1 = 9:
|  |  AGE > 75 : NO-ISCHEMIA (27.0/6.3)
|  |  AGE <= 75 :
|  |  |  AGE <= 70 : NO-ISCHEMIA (37.8/11.6)
|  |  |  AGE > 70 : ISCHEMIA (10.3/4.5)
```

…

Tree for Ischemia: Results

Evaluation on training data (3453 items):

Before Pruning		After Pruning			
Size	Errors	Size	Errors	Estimate	
462	494(14.3%)	74	668(19.3%)	(24.5%)	<<

Evaluation on test data (2320 items):

Before Pruning		After Pruning			
Size	Errors	Size	Errors	Estimate	
462	502(21.6%)	74	426(18.4%)	(24.5%)	<<

(a)	(b)	<-classified as
490	223	(a): class ISCHEMIA
203	1404	(b): class NO-ISCHEMIA

Issues

- Using related attributes in different parts of the tree
  - Use a subset of variables in final tree
- Overfitting: need more severe pruning
  - Adjust confidence level
- Small leaves
  - Set a large minimum leaf size
- Need relative balance of outcomes
  - Enrich outcomes of training set

Treatment of Variables

- Continuous => Ranges
  - When fine distinctions are inappropriate
  - Avoids overfitting
  - Age: 20,30,40,50,60,70,80,90
- Categorical => Continuous
  - When there is some order to the categories
  - Natural subsetting
  - Smoking: never => 0, quit > 5yr => 1, quit 1-5yr => 2, quit < 1yr (or unk) => 3, current => 4

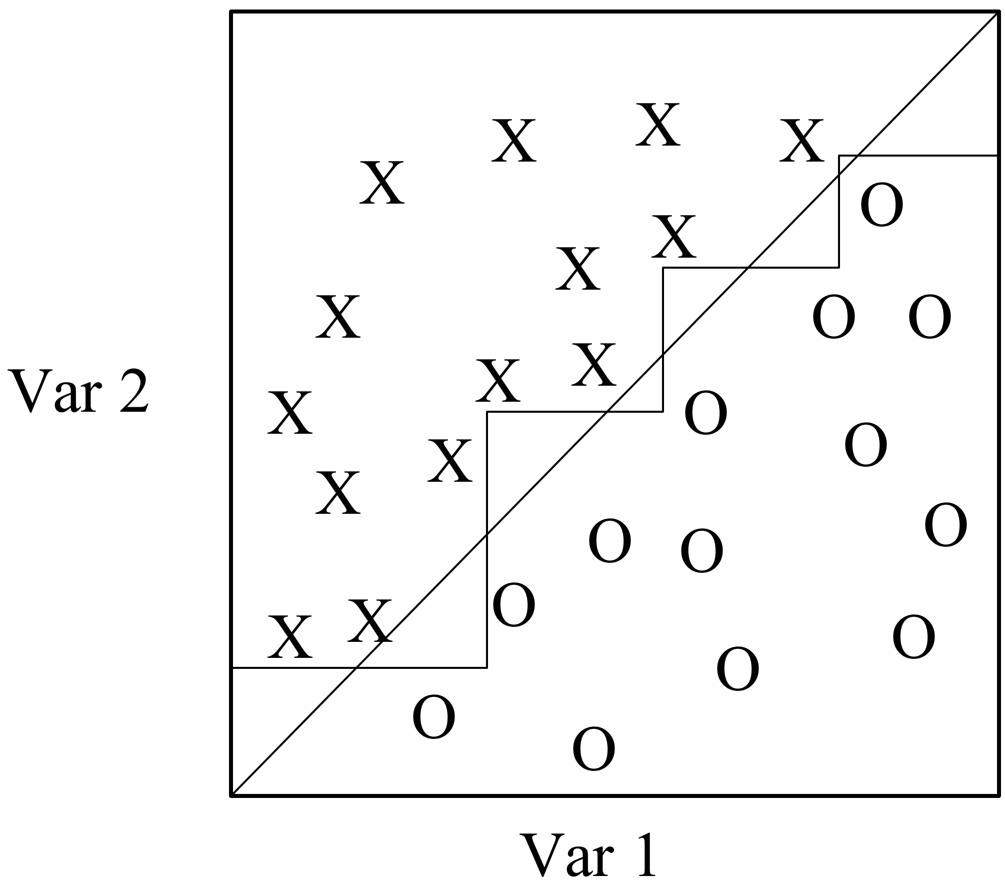
Treatment of Variables

- Specify a value for unknown
  - Stroke: unknown => false
- Combining variables
  - “Or” across drugs by class or implications
- Picking variables on pragmatic grounds
  - Start with many variables and narrow to ones most clinically relevant

Variables Cont’d

- Missing values
  - Force, if likely value different from average of knowns
- Derived values
  - E.g., pulse pressure or product values
  - Combine related variables

Combinations of Variables



Comparison with Logistic Regression

- Trees:
  - Automatic selection
  - Classification
  - Assumes independence of subgroups
  - Handles interactions automatically
  - Handles missing values
  - Linear relationships chopped into categories
  - Handles outliers
- LR:
  - Manual selection
  - Probability
  - Assumes same behavior over all cases
  - Requires interaction variables
  - Requires complete data
  - Handles linear relationships
  - Sensitive to outliers

Multiple Trees

- Weakness: Limited number of categories (leaf nodes) in optimal tree – there is only one way to categorize a case
- Strategy: Generate several different trees and use them to vote on a classification
- Advantage: Allows multiple ways of categorizing a case
- Disadvantage: Makes it much harder to explain the classification of a case

Generating Multiple Trees

- Use different subsets of the learning set
  - Bagging: uniformly sampling *m* cases with replacement for each tree
  - Divide set into 10 parts and use each 9 to generate a tree
- Adapt the learning set
  - Boosting: after generating each tree, increase the weight of cases misclassified by the tree

Voting on a Classification

- Equal votes
- Votes in proportion to the size of the leaves
- Votes weighted by the α used to reweigh the cases (standard for boosting)

Boosting

- C_1 constructed from training & e_1=error rate
- $W(c) = w(c) / \begin{cases} 2e & \text{if case misclassified} \\ 2(1-e) & \text{otherwise} \end{cases}$
- Composite classifier obtained by voting
  - $Weight(C_i) = \log((1-e_i)/e_i)$

Boosting

- Adaboost: Freund & Schapire, 1997
  - many classifiers: 25, 100, 1000
- Miniboost: Quinlan 1998
  - 3 classifiers and take majority vote
  - allows simplifications
  - computationally efficient

MiniBoosting

- Performance is improved
- Combined trees are possible but very complex
- Even the leafless branches of combined trees contribute to the performance improvement

Empirical Comparison

- Bauer & Kohavi, Mach Learn 36:105, ’99
- Bagging, AdaBoost, Arc (bag+reweigh)
  - AdaBoost & Arc better than Bagging on avg
  - AdaBoost had problems with noisy datasets
  - Reweighing can be unstable when error rates are small
  - Not pruning decreased errors for bagging and increased them for AdaBoost

Literature

- Breiman et al., Classification and Regression Trees
- Quinlan, C4.5 Programs for Machine Learning
- Resources: http://www.kdnuggets.com/