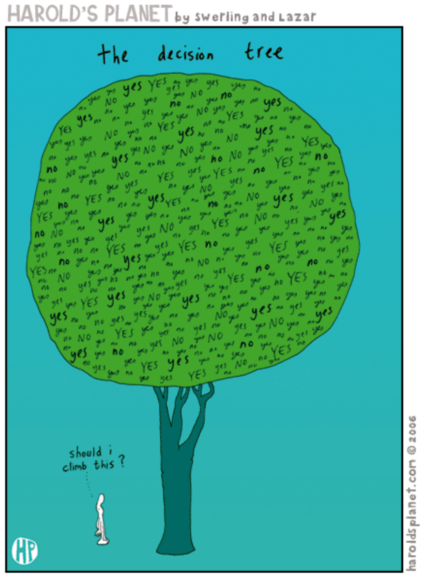
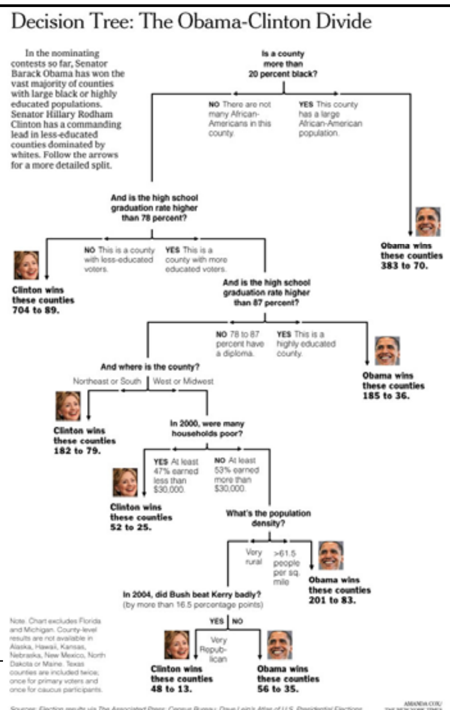
Decision Trees
(Part I: Building the tree)

nanopoulos@ismll.de

1

2


HAROLD'S PLANET by swerling and lazar
the decision tree
should i climb this ?
haroldsplanet.com © 2006

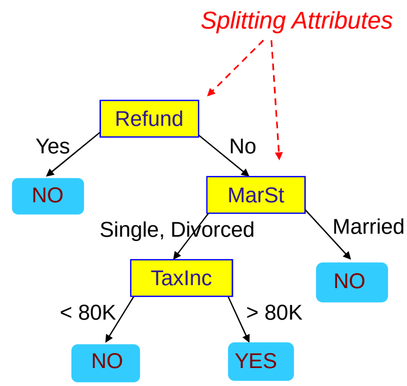

Decision Tree: The Obama-Clinton Divide
In the nominating contests so far, Senator Barack Obama has won the vast majority of counties with large black or highly educated populations. Senator Hillary Rodham Clinton has a commanding lead in less-educated counties dominated by whites. Follow the arrows for a more detailed split.
Is a county more than 20 percent black?
NO There are not many African-Americans in this county.
YES This county has a large African-American population.
Obama wins these counties 383 to 70.
And is the high school graduation rate higher than 78 percent?
NO This is a county with less-educated voters.
YES This is a county with more educated voters.
Clinton wins these counties 704 to 89.
And is the high school graduation rate higher than 87 percent?
NO 78 to 87 percent have a diploma.
YES This is a highly educated county.
Obama wins these counties 185 to 36.
And where is the county?
Northeast or South
West or Midwest
Clinton wins these counties 182 to 79.
In 2000, were many households poor?
YES At least 47% earned less than $30,000.
NO At least 53% earned more than $30,000.
Clinton wins these counties 52 to 25.
What's the population density?
Very rural
>61.5 people per sq. mile
Obama wins these counties 201 to 83.
In 2004, did Bush beat Kerry badly? (by more than 16.5 percentage points)
YES
NO
Very Repub-lican
Clinton wins these counties 48 to 13.
Obama wins these counties 56 to 35.
Note: Chart excludes Florida and Michigan. County-level results are not available in Alaska, Hawaii, Kansas, Nebraska, New Mexico, North Dakota or Maine. Texas counties are included twice; once for primary voters and once for caucus participants.
Sources: Election results via The Associated Press; Census Bureau; Dave Leip's Atlas of U.S. Presidential Elections

Example of a Decision Tree

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



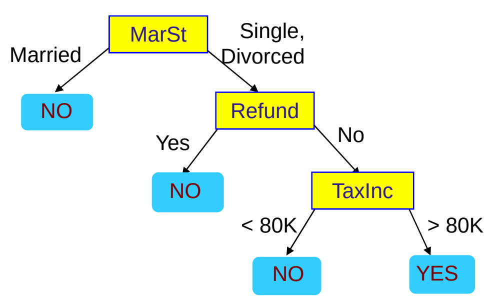
Model: Decision Tree

5

Another Example of Decision Tree

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



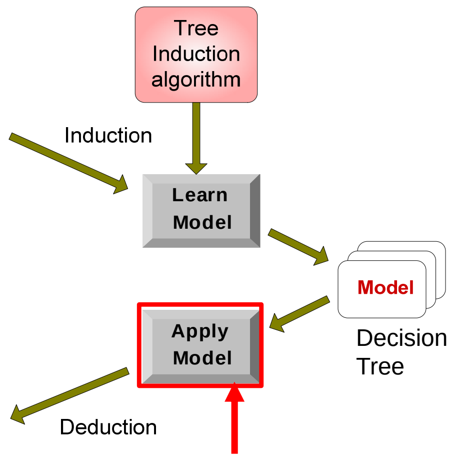
There could be more than one tree that fits the same data!

6

Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

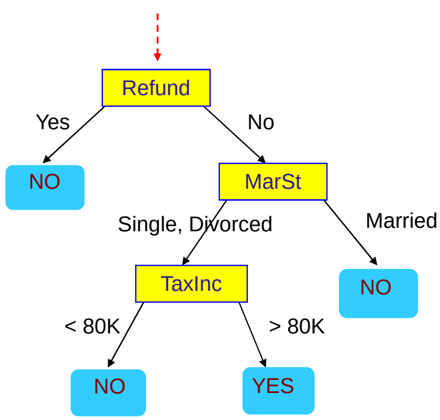
7

Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Start from the root of tree.

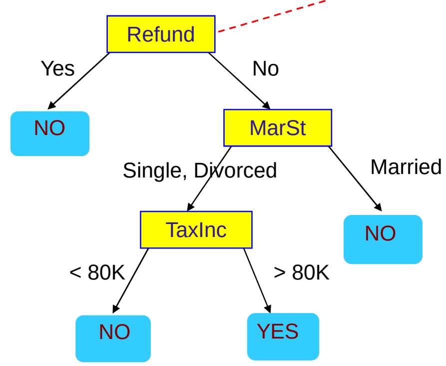


8

Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

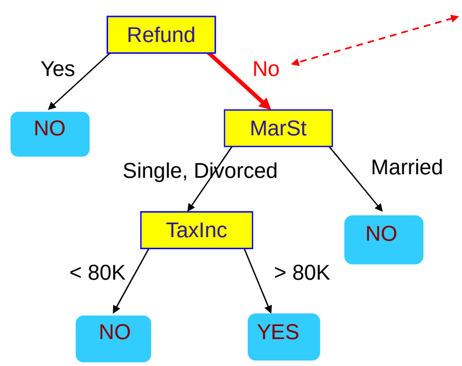


9

Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

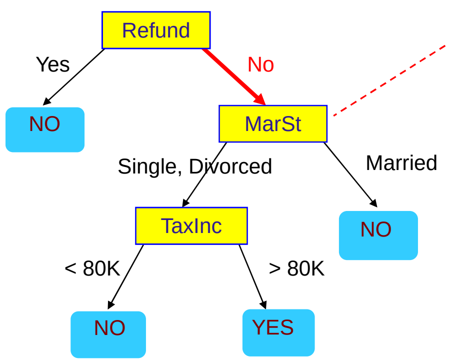


10

Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

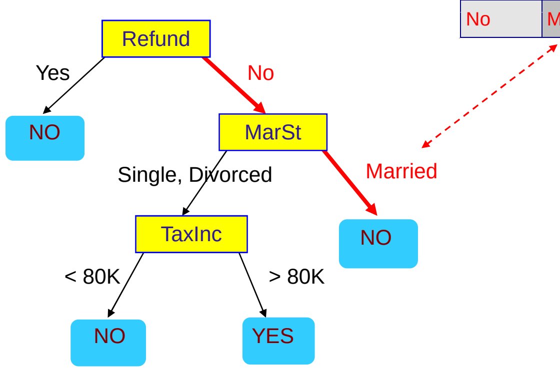


11

Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

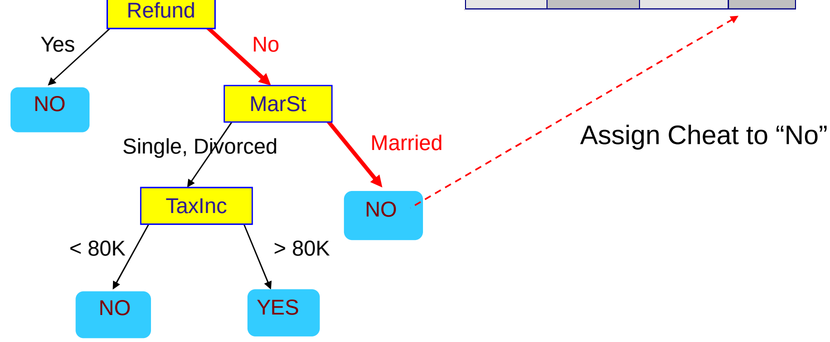


12

Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

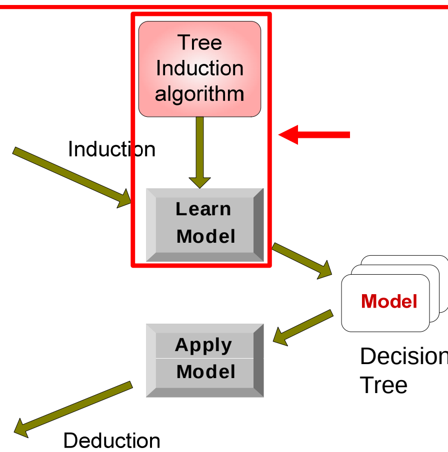


13

Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

14

Decision Tree Induction

Many Algorithms:

- Hunt's Algorithm (one of the earliest)
- CART
- ID3, C4.5
- SLIQ,SPRINT

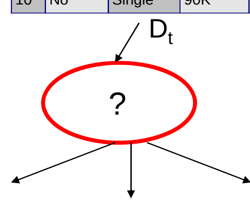
General Structure of Hunt's Algorithm

Let D_t be the set of training records that reach a node t

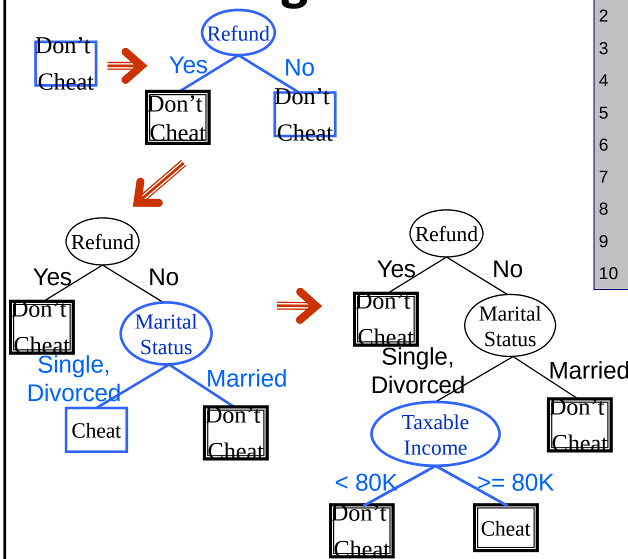
General Procedure:

- If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
- If D_t is an empty set, then t is a leaf node labeled by the default class, y_d
- If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Hunt's Algorithm



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Tree Induction

Greedy strategy.

- Split the records based on an attribute test that optimizes certain criterion.

Issues

- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

Tree Induction

Greedy strategy.

- Split the records based on an attribute test that optimizes certain criterion.

Issues

- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

19

How to Specify Test Condition?

Depends on attribute types

- Nominal
- Ordinal
- Continuous

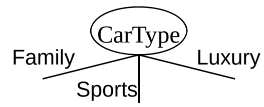
Depends on number of ways to split

- 2-way split
- Multi-way split

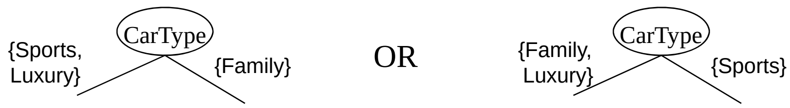
20

Splitting Based on Nominal Attributes

Multi-way split: Use as many partitions as distinct values.



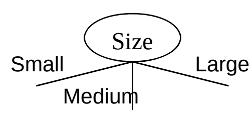
Binary split: Divides values into two subsets. Need to find optimal partitioning.



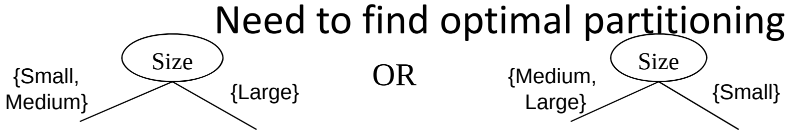
21

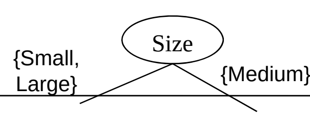
Splitting Based on Ordinal Attributes

Multi-way split: Use as many partitions as distinct values.



Binary split: Divides values into two subsets. Need to find optimal partitioning.





What about this split?

22

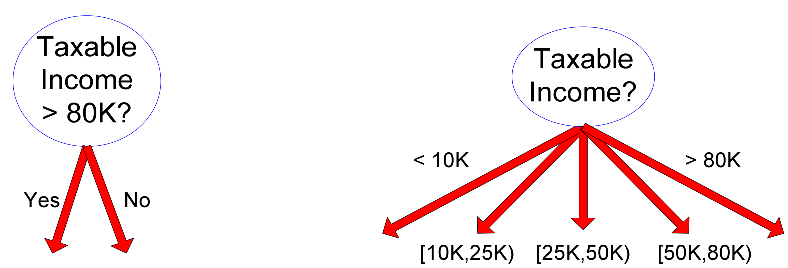
Splitting Based on Continuous Attributes

Different ways of handling

- Discretization to form an ordinal categorical attribute
  - Static – discretize once at the beginning
  - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.
- Binary Decision: (A < v) or (A ≥ v)
  - consider all possible splits and finds the best cut
  - can be more compute intensive

23

Splitting Based on Continuous Attributes



(i) Binary split

(ii) Multi-way split

24

Tree Induction

Greedy strategy.

Split the records based on an attribute test that optimizes certain criterion.

Issues

Determine how to split the records

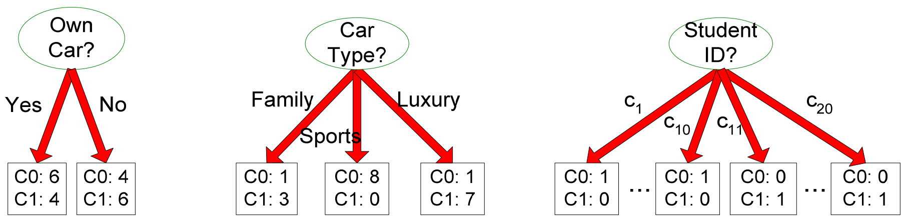
How to specify the attribute test condition?

How to determine the best split?

Determine when to stop splitting

How to determine the Best Split

Before Splitting: 10 records of class 0,
10 records of class 1



Which test condition is the best?

How to determine the Best Split

Greedy approach:

- Nodes with homogeneous class distribution are preferred

Need a measure of node impurity:

C0: 5	C0: 9
C1: 5	C1: 1
Non-homogeneous, High degree of impurity	Homogeneous, Low degree of impurity

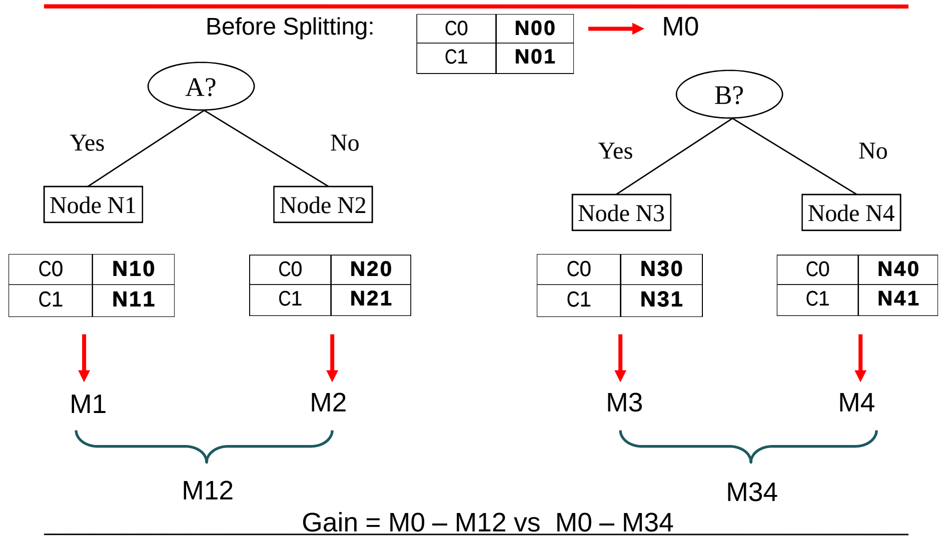
Measures of Node Impurity

Gini Index

Entropy

Misclassification error

How to Find the Best Split



29

Measure of Impurity: GINI

Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j\,|\,t)]^2$$

(NOTE: $p(j\,|\,t)$ is the relative frequency of class j at node t).

Maximum ($1 - 1/n_c$) when records are equally distributed among all classes, implying least interesting information

Minimum (0.0) when all records belong to one class, implying most interesting information

C1	0
C2	6
Gini=0.000	

C1	1
C2	5
Gini=0.278	

C1	2
C2	4
Gini=0.444	

C1	3
C2	3
Gini=0.500	

30

Examples for computing GINI

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Gini = 1 – $P(C1)^2$ – $P(C2)^2$ = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Gini = 1 – $(1/6)^2$ – $(5/6)^2$ = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

Gini = 1 – $(2/6)^2$ – $(4/6)^2$ = 0.444

Splitting Based on GINI

- **Used in CART, SLIQ, SPRINT.**
- **When a node p is split into k partitions (children), the quality of split is computed as,**

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

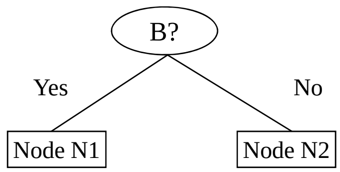
where, **n_i = number of records at child i,**

n = number of records at node p.

Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for.

	Parent
C1	**6**
C2	**6**
Gini = 0.500	



Gini(N1)
$= 1 - (5/7)^2 - (2/7)^2$
$= 0.408$

Gini(N2)
$= 1 - (1/5)^2 - (4/5)^2$
$= 0.32$

	N1	N2
C1	**5**	**1**
C2	**2**	**4**
Gini=0.371		

Gini(Children)
$= 7/12 * 0.408 +$
$5/12 * 0.32$
$= 0.371$

Categorical Attributes: Computing Gini Index

For each distinct value, gather counts for each class in the dataset

Use the count matrix to make decisions

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

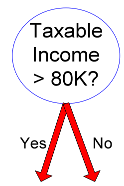
	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Continuous Attributes: Computing Gini Index

- **Use Binary Decisions based on one value**
- **Several Choices for the splitting value**
  - Number of possible splitting values = Number of distinct values
- **Each splitting value has a count matrix associated with it**
  - Class counts in each of the partitions, A < v and A ≥ v
- **Simple method to choose best v**
  - For each v, scan the database to gather count matrix and compute its Gini index
  - Computationally Inefficient! Repetition of work.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Continuous Attributes: Computing Gini Index...

For efficient computation: for each attribute,

Sort the attribute on values

Linearly scan these values, each time updating the count matrix and computing gini index

Choose the split position that has the least gini index

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
	Taxable Income									
Sorted Values →	60	70	75	85	90	95	100	120	125	220

Split Positions →	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		***0.300***		0.343		0.375		0.400		0.420	

Alternative Splitting Criteria based on INFO

Entropy at a given node t:

$$Entropy(t) = -\sum_j p(j\,|\,t) \log p(j\,|\,t)$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- Measures homogeneity of a node.
  - Maximum (log n_c) when records are equally distributed among all classes implying least information
  - Minimum (0.0) when all records belong to one class, implying most information
- Entropy based computations are similar to the GINI index computations

37

Examples for computing Entropy

$$Entropy(t) = -\sum_j p(j\,|\,t) \log_2 p(j\,|\,t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

38

Splitting Based on INFO...

- **Information Gain:**

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Parent Node, p is split into k partitions;
n_i is number of records in partition i

- Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
- Used in ID3 and C4.5
- Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

39

Splitting Based on INFO...

- **Gain Ratio:**

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions
n_i is the number of records in partition i

- Adjusts Information Gain by the entropy of the partitioning (SplitINFO). Higher entropy partitioning (large number of small partitions) is penalized!
- Used in C4.5
- Designed to overcome the disadvantage of Information Gain

40

Splitting Criteria based on Classification Error

Classification error at a node t :

$$Error(t) = 1 - \max_i P(i \mid t)$$

Measures misclassification error made by a node.

- Maximum (1 - 1/n_c) when records are equally distributed among all classes, implying least interesting information
- Minimum (0.0) when all records belong to one class, implying most interesting information

41

Examples for Computing Error

$$Error(t) = 1 - \max_i P(i \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

C1	**2**
C2	**4**

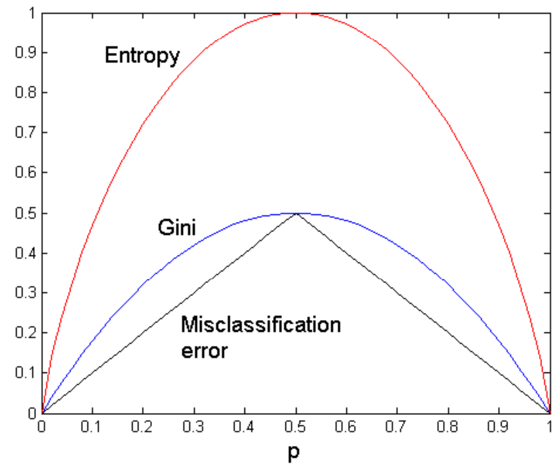
P(C1) = 2/6 P(C2) = 4/6

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

42

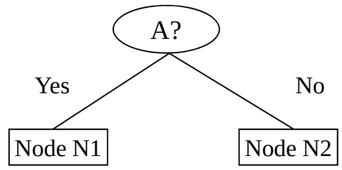
Comparison among Splitting Criteria

For a 2-class problem:



43

Misclassification Error vs Gini



	Parent
C1	**7**
C2	**3**
Gini = 0.42	

Gini(N1)
$= 1 - (3/3)^2 - (0/3)^2$
$= 0$

Gini(N2)
$= 1 - (4/7)^2 - (3/7)^2$
$= 0.489$

	N1	N2
C1	**3**	**4**
C2	**0**	**3**
Gini=0.361		

Gini(Children)
= 3/10 * 0
+ 7/10 * 0.489
= 0.342

Gini improves !!

44

Tree Induction

Greedy strategy.

- Split the records based on an attribute test that optimizes certain criterion.

Issues

- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

Stopping Criteria for Tree Induction

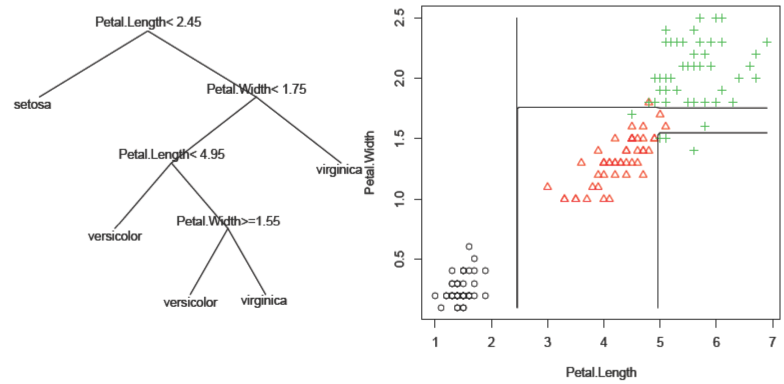
Stop expanding a node when all the records belong to the same class

Stop expanding a node when all the records have similar attribute values

Early termination (to be discussed later)

Characteristics of decision trees

Decision boundaries are rectangular.



47

Advantages

Inexpensive to construct

Extremely fast at classifying unknown records

Easy to interpret for small-sized trees

Accuracy is comparable to other classification techniques for many simple data sets

48

Disadvantages

Decision trees often are used to visually explain models.

Nevertheless, usually there are many candidates for the primary split with very similar values of the quality criterion. So the choice of the primary split shown in the tree is somewhat arbitrary: the split may change, if the data changes a bit. The tree is said to be **instable**.

49

Real implementations

name	ChAID	CART	ID3	C4.5
author	Kass 1980	Breiman et al. 1984	Quinlan 1986	Quinlan 1993
selection measure	χ^2	Gini index, twoing index	information gain	information gain ratio
splits	all	binary nominal, binary quantitative, binary bivariate quantitative	complete	complete, binary nominal, binary quantitative
stopping criterion	χ^2 independence test	minimum number of cases/node	χ^2 independence test	lower bound on selection measure
pruning technique	none	error complexity pruning	pessimistic error pruning	pessimistic error pruning, error based pruning

50

Example: C4.5

Simple depth-first construction.

Uses Information Gain

Sorts Continuous Attributes at each node.

Needs entire data to fit in memory.

Unsuitable for Large Datasets.

- Needs out-of-core sorting.

You can download the software from:

http://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz

51