Rule-Based Classifiers

Rule-Based Classifier

- Classify records by using a collection of "if…then…" rules

Examples:

(Blood Type=Warm) $\wedge$ (Lay Eggs=Yes) $\rightarrow$ Birds

(Taxable Income < 50K) $\wedge$ (Refund=Yes) $\rightarrow$ (Evade=No)

- Rule: $(\mathit{Condition}) \rightarrow y$
  - *Condition* is a conjunction of attributes
  - y is the class label

- *LHS*: rule antecedent or condition
- *RHS*: rule consequent

(Example)

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fish

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Application of Rule-Based Classifier

- A rule *r* covers an instance **x** if the attributes of the instance satisfy the condition of the rule

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers a hawk => Bird

The rule R3 covers the grizzly bear => Mammal

Rule Coverage and Accuracy

- **Coverage** of a rule:
  – Fraction of records that satisfy the antecedent of a rule
- **Accuracy** of a rule:
  – Fraction of records that satisfy both the antecedent and consequent of a rule (over those that satisfy the antecedent)

Tid	Refund	Marital Status	Taxable Income	Class
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

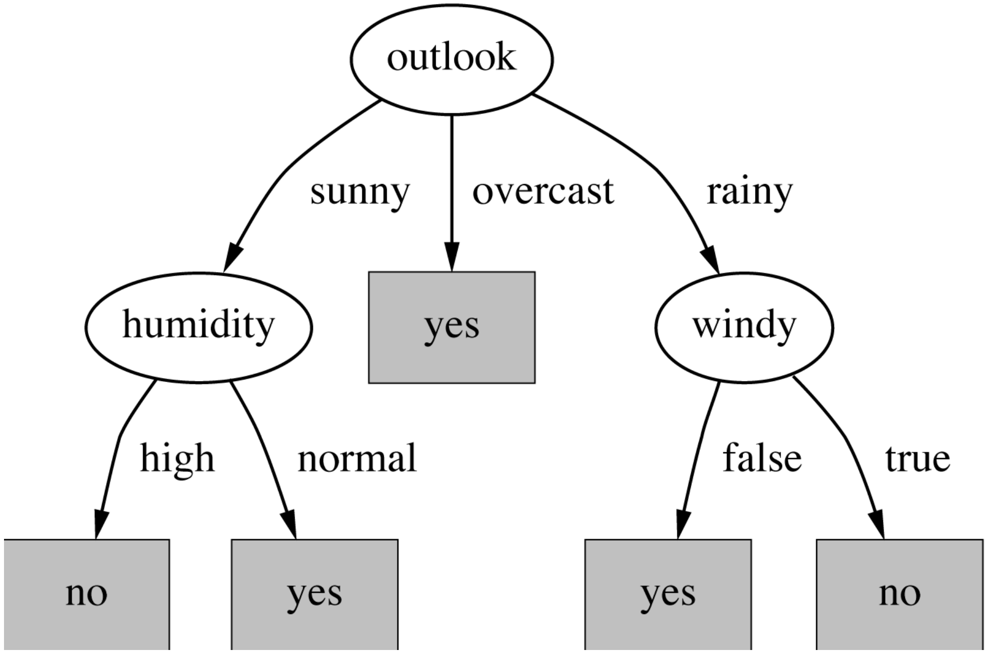
(Status=Single) → No

Coverage = 40%, Accuracy = 50%

Decision Trees vs. rules

From trees to rules.

- Easy: converting a tree into a set of rules
  - One rule for each leaf:
  - Antecedent contains a condition for every node on the path from the root to the leaf
  - Consequent is the class assigned by the leaf
  - Straightforward, but rule set might be overly complex



Decision Trees vs. rules

From rules to trees

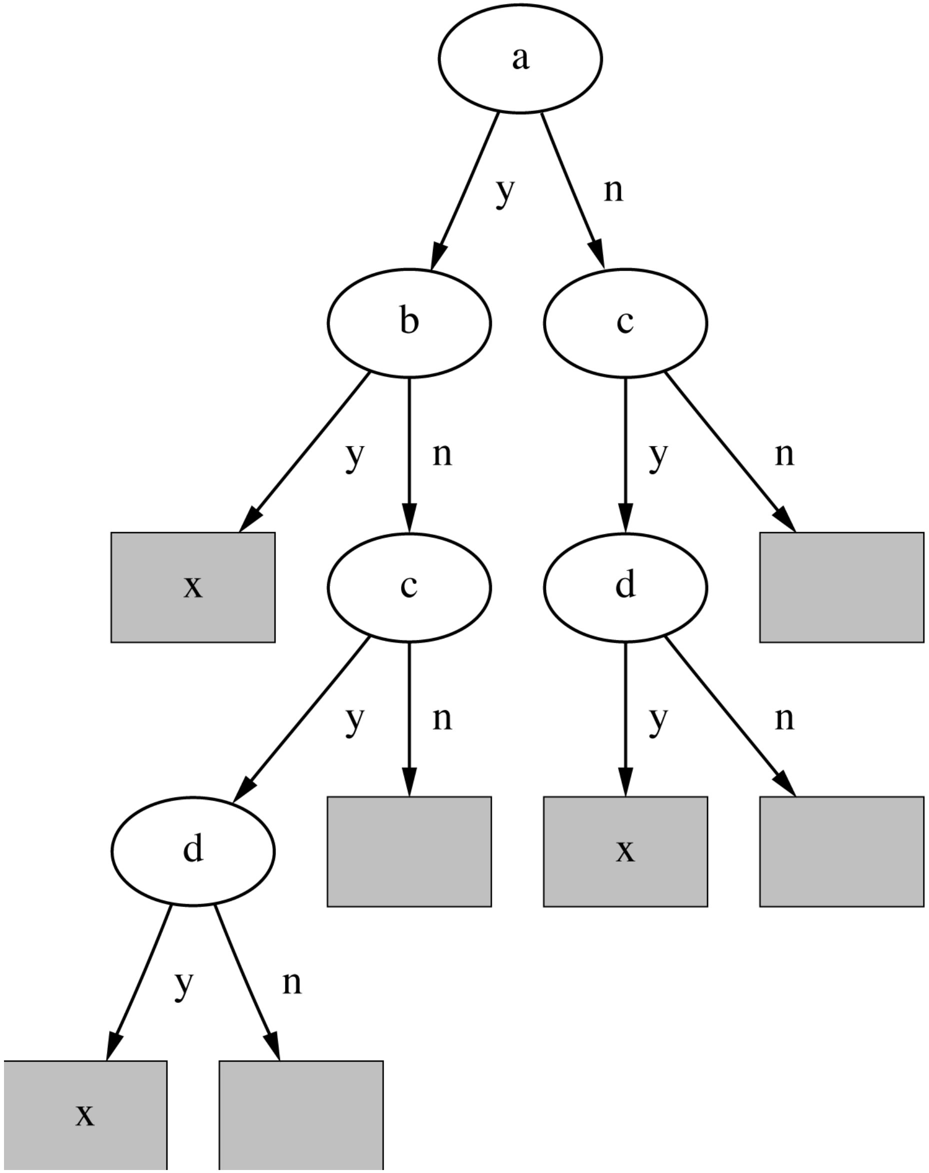
- More difficult: transforming a rule set into a tree
  - Tree cannot easily express disjunction between rules
- Example:

If a and b then x

If c and d then x

  - Corresponding tree contains identical subtrees ($\Rightarrow$"replicated subtree problem")

A tree for a simple disjunction



How does Rule-based Classifier Work?

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fish

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule R3, so it is classified as a mammal

A turtle triggers both R4 and R5

A dogfish shark triggers none of the rules

Desiderata for Rule-Based Classifier

- Ensure that every record is covered by **at most** one rule.

  Solution:

  - Order rules based on their quality

- Ensure that every record is covered by **at least** one rule.

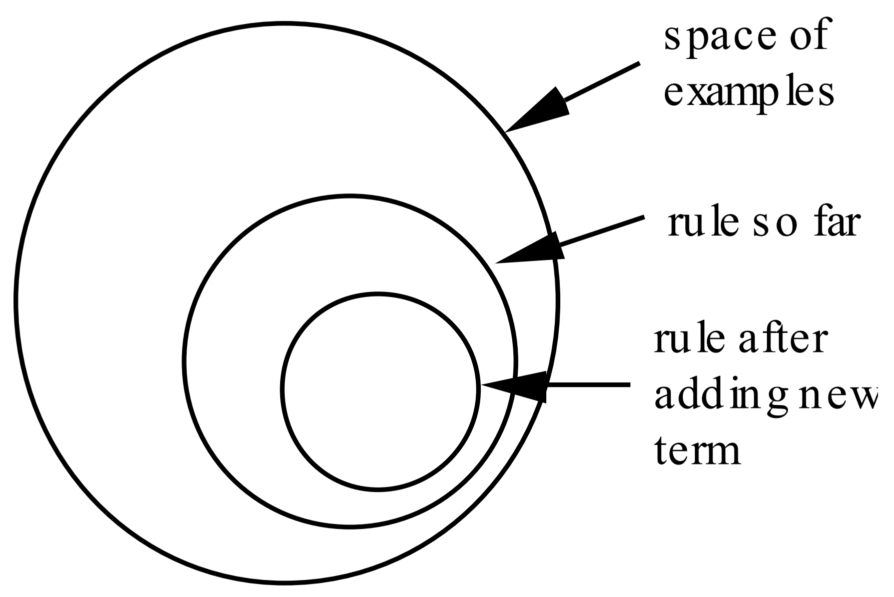
  Solution:

  - Use a default class

- When a new record is presented to the classifier
  - It is assigned to the class of the highest ranked rule it triggers
  - If none of the rules is fired, it is assigned to the default class

A simple covering algorithm

- Generate a rule by adding tests that maximize rule's accuracy
- Each new test (growing the rule) reduces rule's coverage, but it increases rule's accuracy.



Selecting a test

- Goal: maximize accuracy
  - **t**: total number of instances covered by rule
  - **p**: positive (correct) examples of the class covered by rule

$\Rightarrow$ Select test that maximizes the accuracy **p/t**

- We are finished when **p/t** = 1 or the set of instances can't be split any further

Example: contact lenses data

age	spectacle-prescrip	astigmatism	tear-prod-rate	contact-lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

Example: contact lenses data

- Rule we seek:

```
If ?
    then recommendation = hard
```

- Possible tests:

Age = Young	2/8
Age = Pre-presbyopic	1/8
Age = Presbyopic	1/8
Spectacle prescription = Myope	3/12
Spectacle prescription = Hypermetrope	1/12
Astigmatism = no	0/12
Astigmatism = yes	4/12
Tear production rate = Reduced	0/12
Tear production rate = Normal	4/12

The numbers on the right show the fraction of “correct” instances in the set singled out by that choice.
In this case, correct means that their recommendation is “hard.”

Modified rule and resulting data

- Rule with best test added:

```
If astigmatism = yes
   then recommendation = hard
```

- Instances covered by modified rule:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

The rule isn't very accurate, getting only 4 out of 12 that it covers. So, it needs further refinement.

Further refinement

- Current state:

```
If astigmatism = yes
    and ?
  then recommendation = hard
```

- Possible tests:

```
Age = Young                                  2/4
Age = Pre-presbyopic                         1/4
Age = Presbyopic                             1/4
Spectacle prescription = Myope               3/6
Spectacle prescription = Hypermetrope        1/6
Tear production rate = Reduced               0/6
Tear production rate = Normal                4/6
```

Modified rule and resulting data

- Rule with best test added:

```
If astigmatism = yes
     and tear production rate =
normal
   then recommendation = hard
```

- Instances covered by modified rule:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Normal	None

Should we stop here? Perhaps. But let's say we are going for exact rules, no matter how complex they become.
So, let's refine further.

Further refinement

- Current state:

```
If astigmatism = yes
     and tear production rate = normal
     and ?
   then recommendation = hard
```

- Possible tests:

Age = Young	2/2
Age = Pre-presbyopic	1/2
Age = Presbyopic	1/2
Spectacle prescription = Myope	3/3
Spectacle prescription = Hypermetrope	1/3

- Tie between the first and the fourth test
  - We choose the one with greater coverage

The result

- Final rule:

```
If astigmatism = yes
   and tear production rate = normal
   and spectacle prescription = myope
   then recommendation = hard
```

- Second rule for recommending "hard lenses":
  (built from instances not covered by first rule)

```
If age = young and astigmatism = yes
   and tear production rate = normal
   then recommendation = hard
```

- These two rules cover all "hard lenses":
  - Process is repeated with other two classes

Pseudo-code for PRISM

Heuristic: order C in ascending order of occurrence.

For each class C

  Initialize E to the instance set

  While E contains instances of class C

    Create a rule R with an empty left-hand side that predicts class C

    Until R is perfect (or there are no more attributes to use) do

      For each attribute A not mentioned in R, and each value v,

        Consider adding the condition A = v to the left-hand side of R

        Select A and v to maximize the accuracy p/t

        (break ties by choosing the condition with the largest t)

      Add A = v to R

    Remove the instances covered by R from E

RIPPER Algorithm is similar. It uses instead of p/t the info-gain.

Also, it uses tests

- x = value (for discrete attributes)
- x ≤ value (for numerics)
- x > value (for numerics)