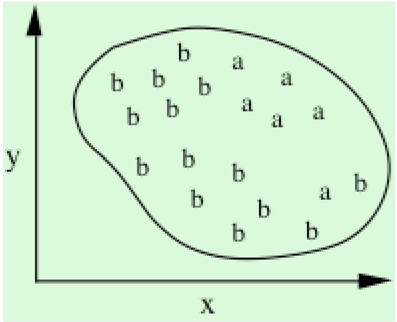
CSCI 347, Data Mining

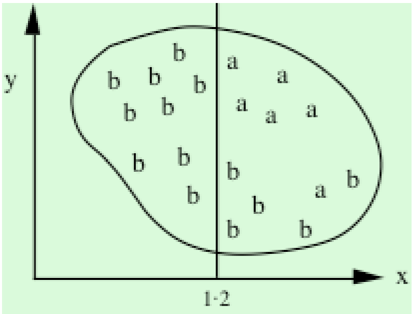
Covering Rules

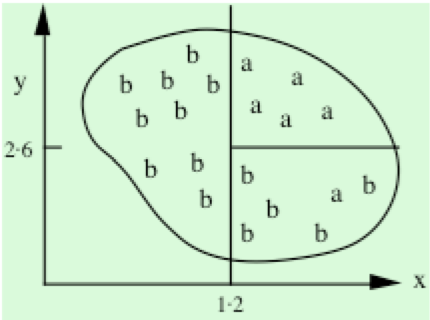
Covering Algorithms

- Rather than looking at what attribute to split on, start with a particular class
- Class by class, develop rules that "cover" the class

Example: Generating a Rule





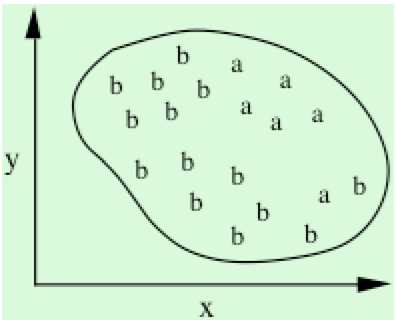


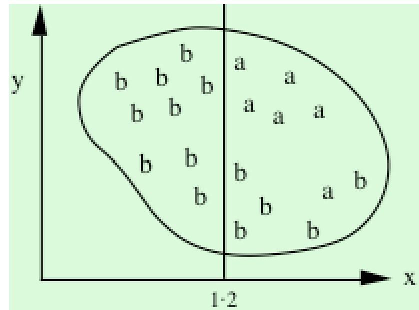
If ???
then class = a

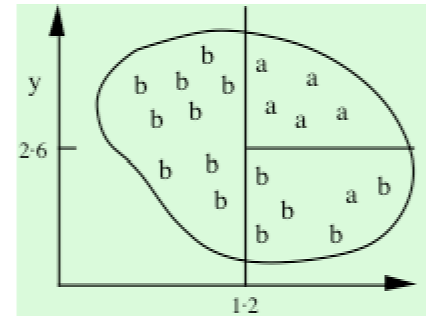
If x > 1.2 and y > 2.6
then class = a

If x > 1.2
then class = a

Example: Generating a Rule








```
If x > 1.2 and y > 2.6
   then class = a
```

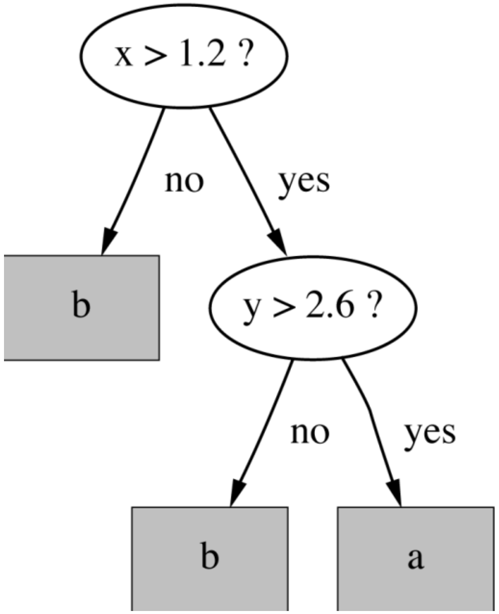
Possible rule set for class “b”:

```
If x ≤ 1.2 then class = b
If x > 1.2 and y ≤ 2.6 then class = b
```

Could add more rules, get “perfect” rule set

Rules vs. Trees

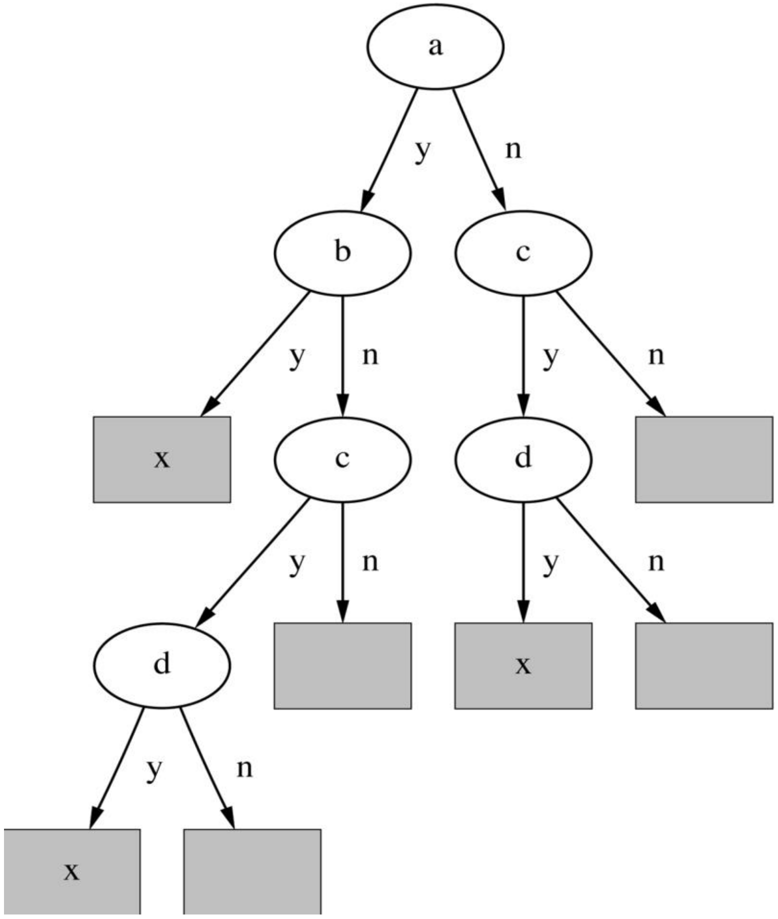
Corresponding decision tree:
(produces exactly the same predictions)



Covering algorithm concentrates on one class value at a time whereas decision tree learner takes all class values into account

Rules vs. Trees

Rules sets *can* be clearer when decision trees suffer from replicated subtrees




```
If a and b then x
If c and d then x
```

Advantages & Disadvantages of Covering Rules

Pros	Cons
Simple to understand (can be easier to understand than a large tree)	Set of covering rules can be overly complex, overfitting
Can easily handle missing values	
Performance is comparable to decision tress	
Can handle both numeric and nominal data	
Not limited to binary class values. Can handle multi-output problems	
Model can be validated via statistical tests	

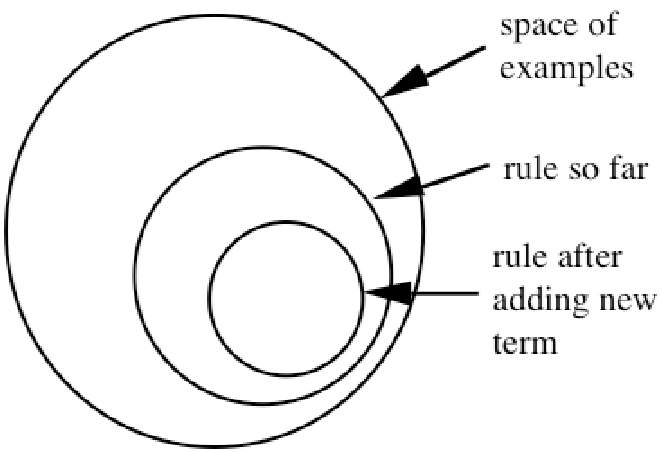
Simple Covering Algorithm

Generates a rule by adding tests that maximize rule’s accuracy

Similar to situation in decision trees: problem of selecting an attribute to split on

- But decision tree inducer maximizes overall purity

Each new test reduces rule’s coverage:



Selecting a Test

Goal: Maximize Accuracy

- t total number of instances to which the rule applies
- p number of instances the rule predict correctly (coverage/support)
- $t - p$ number of errors made by rule

$\Rightarrow$Select test that maximizes the ratio p/t *(confidence/accuracy)*

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

Example: Contact Lens Data

Relation: contact-lenses

No.	age Nominal	spectacle-prescrip Nominal	astigmatism Nominal	tear-prod-rate Nominal	**contact-lenses** Nominal
1	young	myope	no	reduced	none
2	young	myope	no	normal	soft
3	young	myope	yes	reduced	none
4	young	myope	yes	normal	hard
5	young	hypermetrope	no	reduced	none
6	young	hypermetrope	no	normal	soft
7	young	hypermetrope	yes	reduced	none
8	young	hypermetrope	yes	normal	hard
9	pre-presbyopic	myope	no	reduced	none
10	pre-presbyopic	myope	no	normal	soft
11	pre-presbyopic	myope	yes	reduced	none
12	pre-presbyopic	myope	yes	normal	hard
13	pre-presbyopic	hypermetrope	no	reduced	none
14	pre-presbyopic	hypermetrope	no	normal	soft
15	pre-presbyopic	hypermetrope	yes	reduced	none
16	pre-presbyopic	hypermetrope	yes	normal	none
17	presbyopic	myope	no	reduced	none
18	presbyopic	myope	no	normal	none
19	presbyopic	myope	yes	reduced	none
20	presbyopic	myope	yes	normal	hard
21	presbyopic	hypermetrope	no	reduced	none
22	presbyopic	hypermetrope	no	normal	soft
23	presbyopic	hypermetrope	yes	reduced	none
24	presbyopic	hypermetrope	yes	normal	none

Example: Contact Lens 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

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

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

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

Resulting Rule:

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

Instances Not Covered

If there are still instances which have not been handled by a rule, and an acceptable accuracy level has not been reached, repeat the above process beginning with all of the instances which were **not** covered by the previous rule.

Pseudo-Code for PRISM

```
For each class C
  Initialize E to the instance set
  While E contains instances in 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 p)
      Add A = v to R
    Remove the instances covered by R from E
```