Data Mining :: Unit-3

(Classification – Rule Based Classifier)

Er. Dinesh Baniya Kshatri
(Lecturer)

Department of Electronics and Computer Engineering
Institute of Engineering, Thapathali Campus

Classification using Rules – [1]

- A **rule-based classifier** uses a set of IF-THEN rules for classification.
- An IF-THEN rule is an expression of the form:

  IF *condition* THEN *conclusion*.

  – where
    - Condition (or LHS) is rule antecedent/precondition
    - Conclusion (or RHS) is rule consequent

Classification using Rules – [2]

- An example is rule *R*1:

  R1: IF $age = youth$ AND $student = yes$ THEN $buys\_computer = yes$

  – The condition consists of one or more attribute tests that are **logically ANDed**
    - such as *age = youth, and student = yes*

  – The rule's consequent contains a class prediction
    - we are predicting whether a customer will buy a computer

- *R*1 can also be written as

  R1: $(age = youth) \land (student = yes) \Rightarrow (buys\_computer = yes)$

Prepared by: Er. Dinesh Baniya Kshatri 3

Classification using Rules – [3]

- Rule: $(Condition) \rightarrow y$
  - where
    - *Condition* is a conjunction of attributes
    - *y* is the class label
  - *LHS*: rule antecedent or condition
  - *RHS*: rule consequent
- *Rule set:* $R = \{r_1, r_2, \ldots, r_n\}$
- Examples of classification rules
  - (Blood Type=Warm) $\wedge$ (Lay Eggs=Yes) $\rightarrow$ Birds
  - (Taxable Income < 50K) $\wedge$ (Refund=Yes) $\rightarrow$ Cheat=No

Prepared by: Er. Dinesh Baniya Kshatri 4

Classifying Instances with Rules

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

    r: (Age < 35) $\wedge$ (Status = Married) $\rightarrow$ Cheat=No
  - Instances:

    x_1 : (Age=29, Status=Married, Refund=No)

    x_2 : (Age=28, Status=Single, Refund=Yes)

    x_3 : (Age=38, Status=Divorced, Refund=No)
  - Only x_1 is covered by the rule r

Which instances are covered by rule (r) ?

Assessment of Rules – [1]

– **Coverage of a rule:**

- The percentage of instances that satisfy the antecedent of a rule (i.e., whose attribute values hold true for the rule's antecedent).

– **Accuracy of a rule:**

- The percentage of instances that satisfy both the antecedent and consequent of a rule

Assessment of Rules – [2]

- **Rule accuracy and coverage:**

$$coverage(R) = \frac{n_{covers}}{|D|}$$

$$accuracy(R) = \frac{n_{correct}}{n_{covers}}$$

- *D*: class labeled data set
- |D|: number of instances in *D*
- n_{covers} : number of instances covered by R
- $n_{correct}$: number of instances correctly classified by R

Assessment of Rules – [3]

For a rule $r: A \rightarrow y$

- Coverage of a rule:
  - Fraction of records that satisfy the antecedent of a rule
  - $\text{Coverage}(r) = |A| / |D|$
- Accuracy of a rule:
  - Fraction of records that satisfy both the antecedent and consequent of a rule
  - $\text{Accuracy}(r) = |A \cap y| / |A|$

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Example:

Training Dataset

Assessment of Rules – [4]

- The rule R1:

  R1: IF *age* = *youth* AND *student* = *yes* THEN *buys_computer* = *yes*

  - **What is the Coverage and Accuracy of R1?**

  – R1 covers 2 of the 14 instances

  – It can correctly classify both instances

  – *Coverage*(*R*1) = 2/14 = 14.28%

  – *Accuracy*(*R*1) = 2/2 = 100%.

Assessment of Rules – [5]

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%

Example: Vertebrate Dataset

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

Rule Set for Vertebrate Classification

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

Application of Rule-Based Classifier

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

Characteristics of Rule-Based Classifier

- **Mutually Exclusive Rules**
  - Rules are independent of each other
  - No two rules are triggered by the same record
  - Every record is covered by at most one rule
- **Exhaustive Rules**
  - Each record is covered by at least one rule
  - Rules account for every possible combination of attribute values

Prepared by: Er. Dinesh Baniya Kshatri 15

Mutually Exclusive & Exhaustive Rule Set

- **Rule set (R) = {r1, r2, r3}**
- **No two rules are triggered by the same record**
- **There is a rule for each combination of attribute values**

r_1: (Body Temperature = cold-blooded) $\longrightarrow$ Non-mammals
r_2: (Body Temperature = warm-blooded) $\wedge$ (Gives Birth = yes) $\longrightarrow$ Mammals
r_3: (Body Temperature = warm-blooded) $\wedge$ (Gives Birth = no) $\longrightarrow$ Non-mammals

Prepared by: Er. Dinesh Baniya Kshatri 16

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) $\rightarrow$ 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
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

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

A turtle triggers both R4 and R5

A dogfish shark triggers none of the rules

Potential Problems: Using Rules

- Rules that are not mutually exclusive
  - A record may trigger more than one rule
  - Solution? – Conflict resolution
- Rules that are not exhaustive
  - A record may not trigger any rules
  - Solution? - Use a default class (rule)

Conflict Resolution Strategy
(Unordered Rule Set)

- **Allows a single record to trigger multiple classification rules:**
  - Considers the consequent of each rule that got triggered
  - The consequent (class) that received the highest vote is assigned to the record

Conflict Resolution Strategy
(Ordered Rule Set)

- Rules are rank ordered according to their priority
  - An ordered rule set is known as a decision list
- When a test record is presented to the classifier
  - It is assigned to the class label of the highest ranked rule it has triggered
  - If none of the rules fired, it is assigned to the default class

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

The Class is Reptiles

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

Rule-Ordering Schemes
(Rule-Based Ordering)

- **Individual rules are ranked based on their quality**

- **Rules are organized into a priority list, according to some measure of rule quality such as:**
  - Accuracy
  - Coverage
  - By experts

Prepared by: Er. Dinesh Baniya Kshatri 21

Rule-Ordering Schemes
(Class-Based Ordering)

- **Rules that belong to the same class appear together**

- **It is possible to order the rules based upon class frequency**
  - The most frequent class comes first, the rules for the next most frequent class comes next, and so on

Prepared by: Er. Dinesh Baniya Kshatri 22

Rule-Ordering Schemes
(Rule-Based & Class-Based Ordering)

Rule-Based Ordering

(Skin Cover=feathers, Aerial Creature=yes) ==> Birds

(Body temperature=warm-blooded, Gives Birth=yes) ==> Mammals

(Body temperature=warm-blooded, Gives Birth=no) ==> Birds

(Aquatic Creature=semi)) ==> Amphibians

(Skin Cover=scales, Aquatic Creature=no) ==> Reptiles

(Skin Cover=scales, Aquatic Creature=yes) ==> Fishes

(Skin Cover=none) ==> Amphibians

Class-Based Ordering

(Skin Cover=feathers, Aerial Creature=yes) ==> Birds

(Body temperature=warm-blooded, Gives Birth=no) ==> Birds

(Body temperature=warm-blooded, Gives Birth=yes) ==> Mammals

(Aquatic Creature=semi)) ==> Amphibians

(Skin Cover=none) ==> Amphibians

(Skin Cover=scales, Aquatic Creature=no) ==> Reptiles

(Skin Cover=scales, Aquatic Creature=yes) ==> Fishes

Conflict Resolution Strategy
(Default Rule)

- **If no rule is satisfied by *X* :**
  - A default rule can be set up to specify a default class, based on a training set.
  - This may be the class in majority or the majority class of the instances that were not covered by any rule.
  - The default rule is evaluated at the end, if and only if no other rule covers ***X***.
  - The condition in the default rule is empty.
  - In this way, the rule fires when no other rule is satisfied.

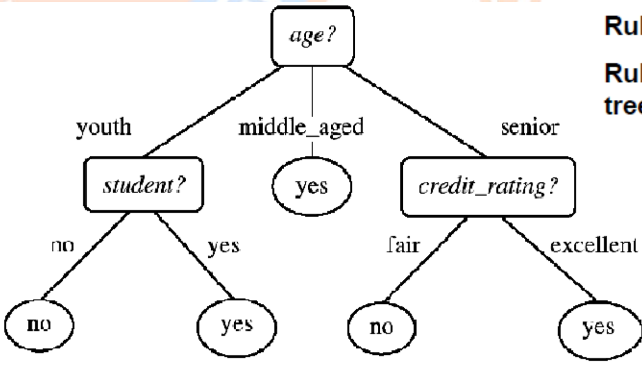
Building Classification Rules

- **Indirect Method**
  - Extract rules from other classification models
  - e.g. Decision trees, Neural Networks
- **Direct Method**
  - Extract rules directly from data
  - e.g.: Holte's One Rule (1R), Sequential Covering Algorithms (PRISM, RIPPER, CN2, FOIL)

Indirect Method
(Rule Extraction from Decision Tree) – [1]

- Decision trees can become large and difficult to interpret.
  - Rules are easier to understand than large trees
  - One rule is created for each path from the root to a leaf
  - Each attribute-value pair along a path forms a precondition: the leaf holds the class prediction
  - The order of the rules does not matter
- **Rules extracted from Decision Tree are:**
  - **Mutually exclusive**: no two rules will be satisfied for the same instance
  - **Exhaustive**: there is one rule for each possible attribute-value combination

Indirect Method
(Rule Extraction from Decision Tree) – [2]



Rules are mutually exclusive and exhaustive

Rule set contains as much information as the tree

R1: IF *age* = *youth* AND *student* = *no* THEN *buys_computer* = *no*
R2: IF *age* = *youth* AND *student* = *yes* THEN *buys_computer* = *yes*
R3: IF *age* = *middle_aged* THEN *buys_computer* = *yes*
R4: IF *age* = *senior* AND *credit_rating* = *excellent* THEN *buys_computer* = *yes*
R5: IF *age* = *senior* AND *credit_rating* = *fair* THEN *buys_computer* = *no*

Direct Method
(Holte's One Rule Method)

- An easy way to find very simple classification rule
- 1R: rules that test one particular attribute
- Basic version
  - One branch for each value
  - Each branch assigns most frequent class
  - Error rate: proportion of instances that don't belong to the majority class of their corresponding branch
  - Choose attribute with lowest error rate (*assumes nominal attributes*)
- "Missing" is treated as a separate attribute value

Direct Method
(Pseudo-code for One Rule)

For each attribute,

For each value of the attribute, make a rule as follows:

count how often each class appears

find the most frequent class

make the rule assign that class to this attribute-value

Calculate the error rate of the rules

Choose the rules with the smallest error rate

Illustrating One Rule Method
(Example: The Weather Dataset)

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Illustrating One Rule Method
(Example: Evaluating Weather Attributes)

	Attribute	Rules	Errors	Total errors
1	outlook	sunny → no	2/5	4/14
		overcast → yes	0/4	
		rainy → yes	2/5	
2	temperature	hot → no*	2/4	5/14
		mild → yes	2/6	
		cool → yes	1/4	
3	humidity	high → no	3/7	4/14
		normal → yes	1/7	
4	windy	false → yes	2/8	5/14
		true → no*	3/6	

Illustrating One Rule Method
(Example: Attribute with Smallest Error)

	Attribute	Rules	Errors	Total errors
1	outlook	sunny → no	2/5	4/14
		overcast → yes	0/4	
		rainy → yes	2/5	

one rule for this example

IF overcast THEN Play

ELSE IF sunny THEN Don't Play

ELSE IF rain THEN Play

Direct Method
(Description of Sequential Covering)

- The rules are learned sequentially (one at a time)
- Each rule for a given class will ideally cover many of the instances of that class (and hopefully none of the instances of other classes).
- Each time a rule is learned, the instances covered by the rule are removed, and the process repeats on the remaining instances.

Direct Method
(Outline of Sequential Covering)

1. Start from an empty rule
2. Grow a rule using the Learn-One-Rule function
3. Remove training records covered by the rule
4. Repeat Step (2) and (3) until stopping criterion is met

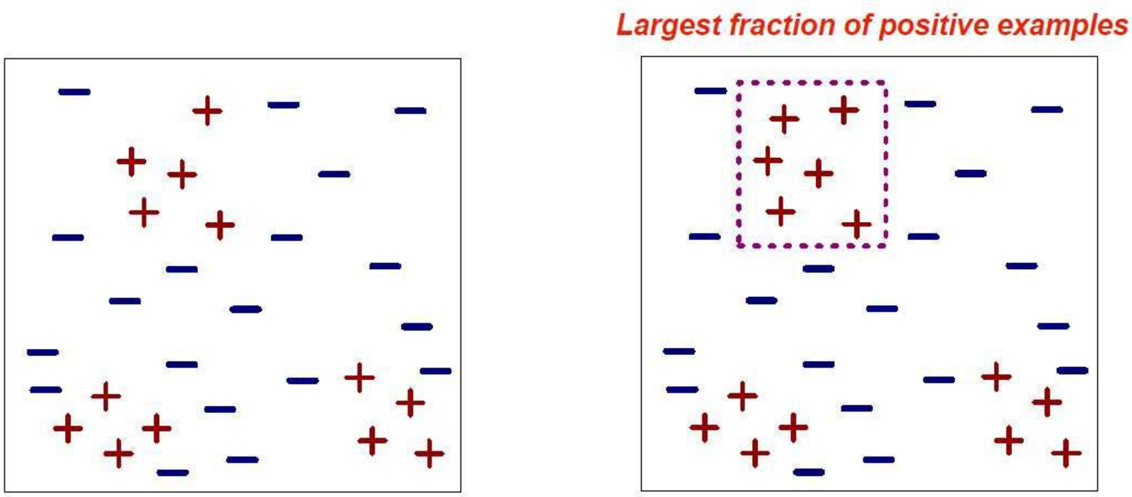
Direct Method
(Pseudo-code of Sequential Covering)

(E: training examples, A: set of attributes)

1. Let R = { } be the initial rule set
2. While stopping criteria is not met
   1. r := Learn-One-Rule (E, A);
   2. Remove instances from E that are covered by r ;
   3. Add r to rule set: R = R + {r};

- Ex. Stopping criteria = "E is empty"

Direct Method
(Illustration of Sequential Covering) – [1]

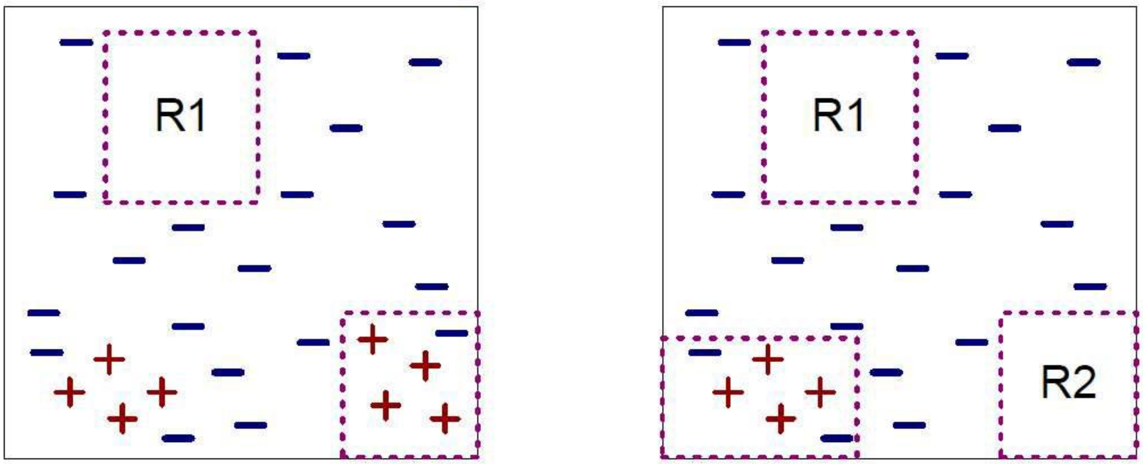


(i) Original Data

(ii) Step 1

Direct Method
(Illustration of Sequential Covering) – [2]

Rule creation and removed examples



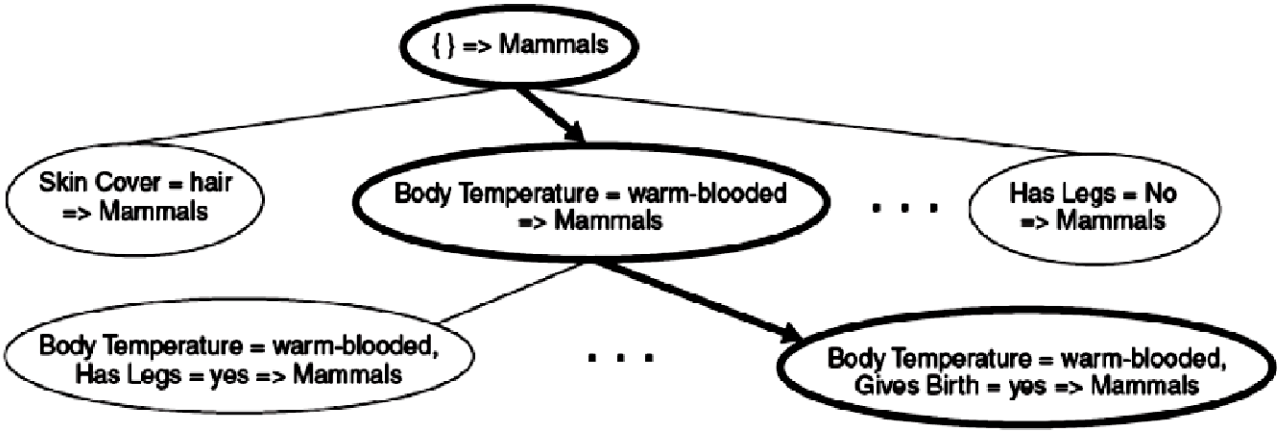
(iii) Step 2 (iv) Step 3

Direct Method
(Rule Growing in Sequential Covering) – [1]

- **Two Common Strategies:**
  - – General to Specific
  - – Specific to General

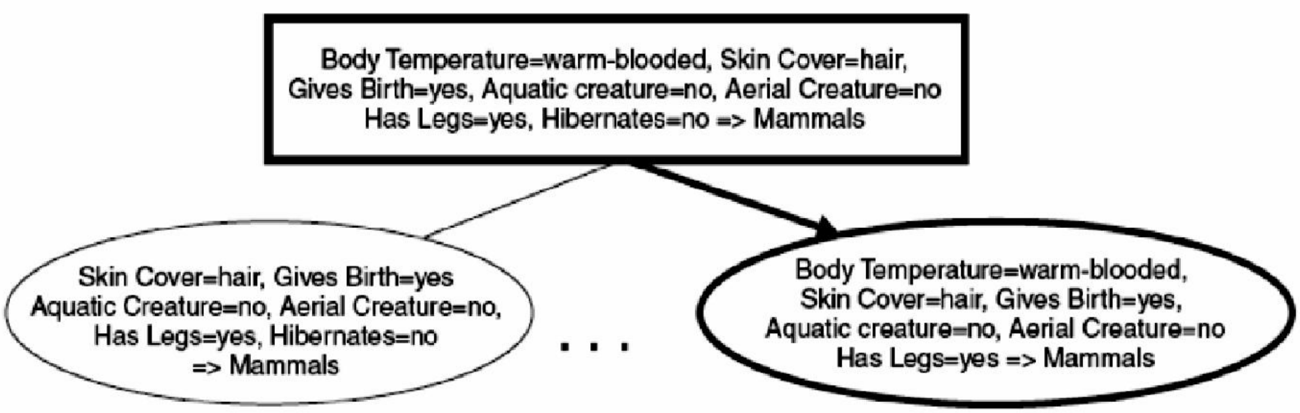
Direct Method
(Rule Growing in Sequential Covering) – [2]

- **General to Specific**



Direct Method
(Rule Growing in Sequential Covering) – [3]

- **Specific to General**



Direct Method: Sequential Covering
(General to Specific Rule Growing)

- Typically, rules are grown in a **general-to-specific manner**
- We start with an empty rule and then gradually keep appending attribute tests to it.
- We append by adding the attribute test as a logical conjunct to the existing condition of the rule antecedent.

Direct Method: Sequential Covering
(Example: General to Specific Rule Growing) – [1]

- Suppose our training set, *D*, consists of loan application data.
- Attributes regarding each applicant include their:
  - age
  - income
  - education level
  - residence
  - credit rating
  - the term of the loan.
- The classifying attribute is *loan_decision*, which indicates whether a loan is accepted (considered **safe**) or rejected (considered **risky**).

Direct Method: Sequential Covering
(Example: General to Specific Rule Growing) – [2]

- To learn a rule for the class "accept," we start off with the most general rule possible, that is, the condition of the rule precondition is empty.
  - The rule is:

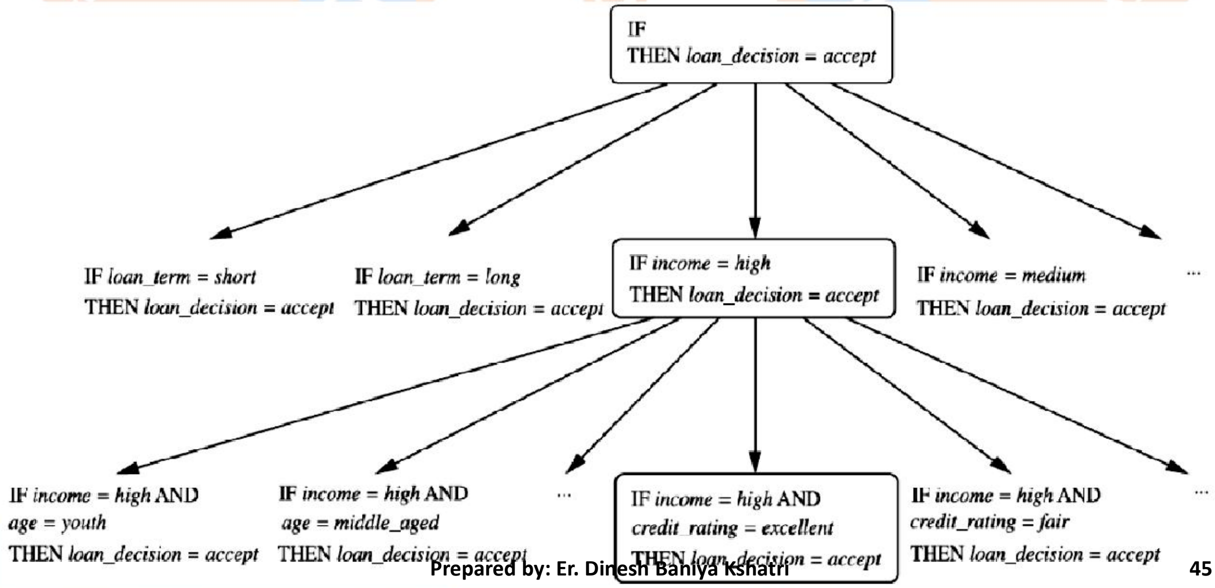
$$\text{IF} \quad \text{THEN } loan\_decision = accept.$$

- We then consider each possible attribute test that may be added to the rule.

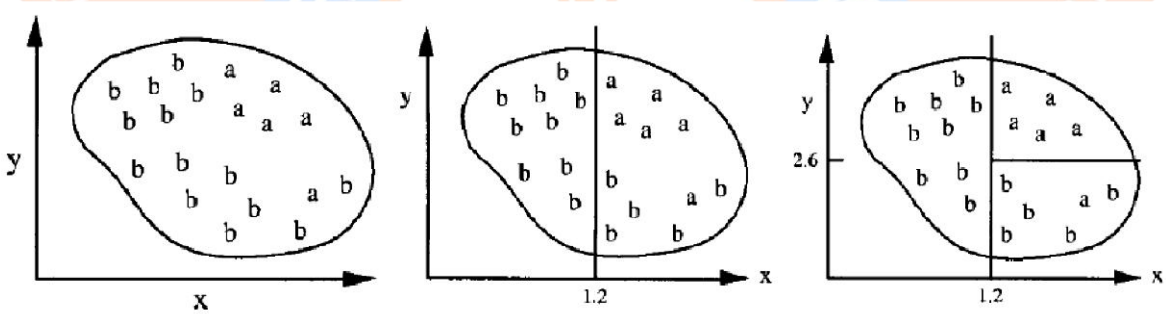
Direct Method: Sequential Covering
(Example: General to Specific Rule Growing) – [3]

- Each time it is faced with adding a new attribute test to the current rule, it picks the one that **most improves the rule quality**, based on the training samples.
- The process repeats, where at each step, we continue to greedily grow rules until the resulting rule meets an acceptable quality level.

Direct Method: Sequential Covering
(Example: General to Specific Rule Growing) – [4]



Direct Method: Sequential Covering
(Example: General to Specific Rule Growing) – [5]



- Possible rule set for class "a":

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

Rule Evaluation Metrics

- Accuracy $= \frac{f_+}{n}$
- Laplace $= \frac{f_+ + 1}{n + k}$
- M-estimate $= \frac{f_+ + kp_+}{n + k}$
- FOIL's information gain

$$= p_1\left(\log_2 \frac{p_1}{p_1 + n_1} - \log_2 \frac{p_0}{p_0 + n_0}\right)$$

n : Number of instances covered by rule

f_+ : Number of positive instances covered by rule

k : Number of classes

p_+ : Prior probability for positive class

p_1 : Number of positive instances covered by new rule

n_1 : Number of negative instances covered by new rule

FOIL's Information Gain

$$FOIL\_Gain = pos' \times \left(\log_2 \frac{pos'}{pos' + neg'} - \log_2 \frac{pos}{pos + neg}\right)$$

- where
  - *pos* (*neg*) be the number of positive (negative) instances covered by *R*
  - *pos'* (*neg'*) be the number of positive (negative) instances covered by *R'*
- It favors rules that have **high accuracy** and cover **many positive instances**

Example: Rule Evaluation

- Consider a training set that contains 60 positive examples and 100 negative examples.
  - Rule r1 covers 50 positive examples and 5 negative examples
  - Rule r2 covers 2 positive examples and no negative examples

Accuracy of r1=50/55=90.9%, accuracy of r2=2/2=100%

Laplace measure for r1=(50+1)/(55+2)=89.47%, r2=(2+1)/(2+2)=75%

Foil's information gain for r1=63.87, r2=2.83

- **r1 :**

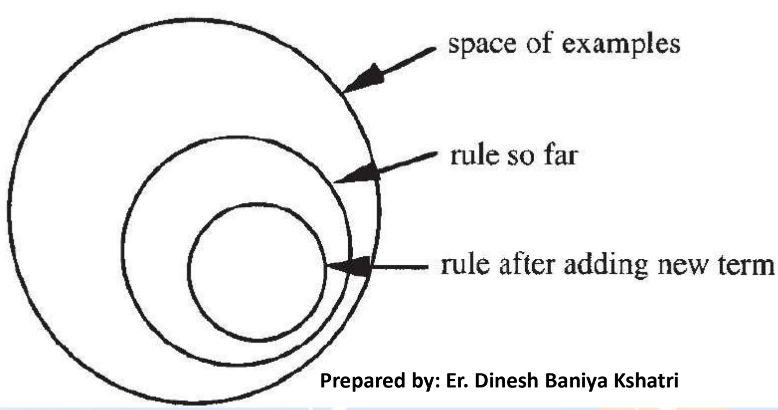
50 * (log2 (50 /(50 + 5)) - log2(60/(60+100))) = 63.87

- **r2 :**

2 * (log2 (2/(2+0)) - log2(60/(60+100))) = 2.83

Direct Method
(Overview of PRISM Method)

- **PRISM method** generates a rule by adding tests that maximize rule's accuracy
- Each new test reduces rule's coverage:



Direct Method
(Test Selection for PRISM Method)

- Goal: maximize accuracy
  - t total number of instances covered by rule
  - p positive examples of the class covered by rule
  - $t - p$ number of errors made by rule
  - Select test that maximizes the ratio p/t
- We are finished when $p / t = 1$ or the set of instances can't be split any further

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended 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 Lens Dataset

Direct Method
(PRISM Example: Contact Lens Dataset) – [1]

- To begin, we seek a rule:

```
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

Direct Method
(PRISM Example: Contact Lens Dataset) – [2]

- Rule with best test added and covered instances:

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

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

Direct Method
(PRISM Example: Contact Lens Dataset) – [3]

- 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

Direct Method
(PRISM Example: Contact Lens Dataset) – [4]

- 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

Direct Method
(PRISM Example: Contact Lens Dataset) – [5]

- 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

Direct Method
(PRISM Example: Contact Lens Dataset) – [6]

- 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

CN2 Algorithm

- **Originally developed by Clark & Niblett (CN), 1989**
  - Start from an empty conjunct: {}
  - Add conjuncts that minimizes the entropy measure: {A}, {A,B}, …
  - Determine the rule consequent by taking majority class of instances covered by the rule

RIPPER Algorithm

- **RIPPER = Repeated Incremental Pruning to Produce Error Reduction**
  - Start from an empty rule: {} => class
  - Add conjuncts that maximizes FOIL's information gain measure:
    - R0: {} => class (initial rule)
    - R1: {A} => class (rule after adding conjunct)
    - Gain(R0, R1) = t [log (p1/(p1+n1)) – log (p0/(p0 + n0))]
    - where t: number of positive instances covered by both R0 and R1
      p0: number of positive instances covered by R0
      n0: number of negative instances covered by R0
      p1: number of positive instances covered by R1
      n1: number of negative instances covered by R1