Data Mining
(EECS 4412)

Decision Rule Learning

Parke Godfrey
EECS
Lassonde School of Engineering
York University

Thanks to

Professor Aijun An

for curation & use of these slides.

Outline

- What are decision rules?
- How to learn decision rules?
- Sequential covering algorithm
- Classification with rules

What Are Decision Rules?

- *If-then* rules that can be used for classification (also called *classification rules*)

  If $condition_1$ and $condition_2$ and ..., then *class*

- Types of if-then rules

  - Propositional rules: rules without variables
    - Relate an example's class to its attribute values

    - Example:
      - If (Outlook = "sunny") and (temperature < 30) then (PlayTennis = yes)
      - If (Outlook = "overcast") and (wind ≠ "strong"), then (PlayTennis = yes)
  - First-order rules: rules containing variables
    - Example:
      - If Parent(x, y), then Ancestor(x, y)
      - If Parent(x, z) ∧ Ancestor(z, y), then Ancestor(x, y)

Decision Rules vs Decision Trees

- Decision rules are easier to understand: most *human readable* representation
- Rules are more flexible than decision trees
  - No overlapping among branches in a decision tree
  - Branches in a decision tree share at least one attribute
- If-then rules can be used with expert systems

Learning Rules

- Two general ways to learn rules from data
  - Rules can be derived from other representations (e.g., decision trees)
  - Rules can be learned ***directly***. Here, we are concentrating on the direct method.
- Advantage of direct rule learning algorithms
  - more flexible rules can be learned
  - Some algorithms can learn sets of ***first-order rules*** which have much more representational power than the ***propositional*** rules that can be derived from decision trees.
- Decision rule learning algorithms
  - CN2, AQ family, HYDRA, PRISM, ELEM2, FOIL, etc.

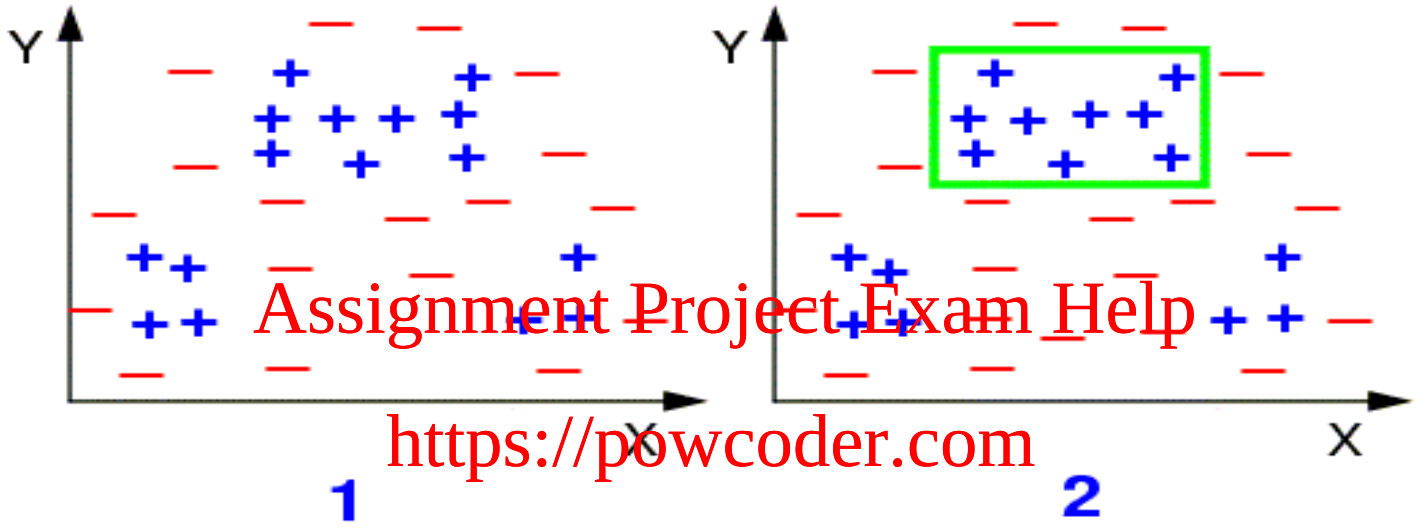
Learning Rules Directly

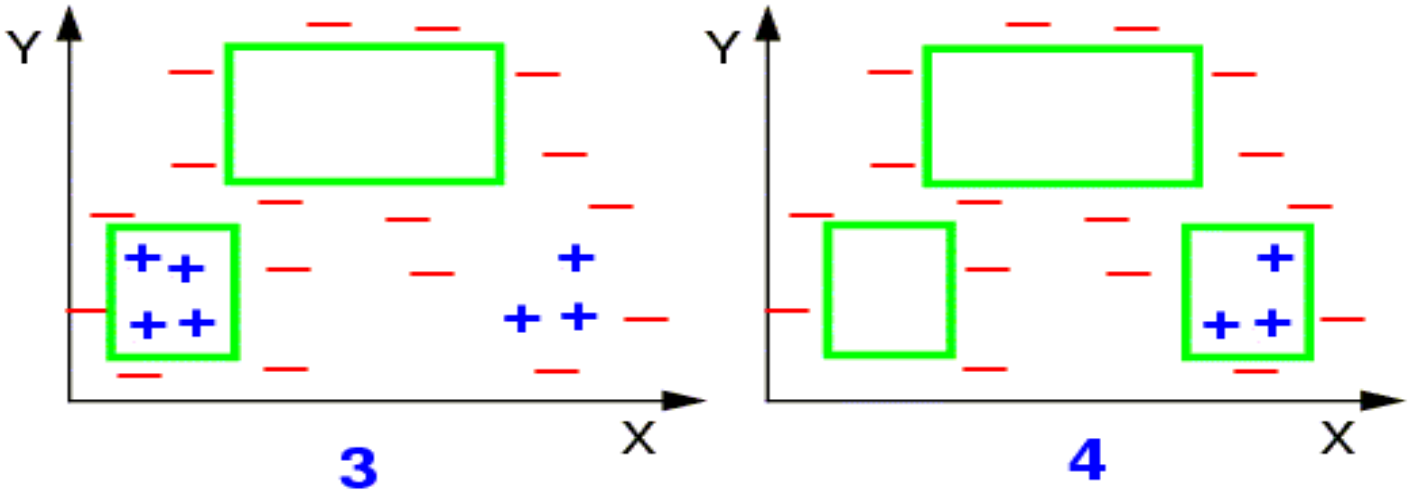
- Most of direct rule learning algorithms use *sequential covering* strategy
- A sequential covering algorithm learns a set of conjunctive rules in turn.
  - A conjunctive rule for class c_i:
    - $(A_1\ rel_1\ v_1) \wedge (A_2\ rel_2\ v_2) \wedge \ldots \wedge (A_k\ rel_k\ v_k) \rightarrow (Class = c_i)$, where $A_1, A_2, \ldots, A_k$ are attributes, $v_1, v_2, \ldots, v_k$ are attribute values and $rel_1, rel_2, \ldots, rel_k$ are relational operators ($=, \neq, \leq, >, \in$, etc.). Each attribute-value pair is called a conjunct.
  - Examples:
    - (Outlook = "sunny") $\wedge$ (temperature < 30) $\rightarrow$ (PlayTennis = yes)
    - (Outlook = "overcast") $\wedge$ (wind $\neq$ "strong") $\rightarrow$ (PlayTennis = yes)

Sequential Covering Algorithm

- To learn rules for class c_i from training set S,
  - Training set is separated into
    - positive examples: examples that belong to c_i
    - negative examples: examples that do not belong to c_i
  - Idea: greedily (sequentially) find rules that apply to (cover) positive examples in the training data
    - *Learn one rule*
    - Remove positive examples covered by this rule from the training data
    - Repeat

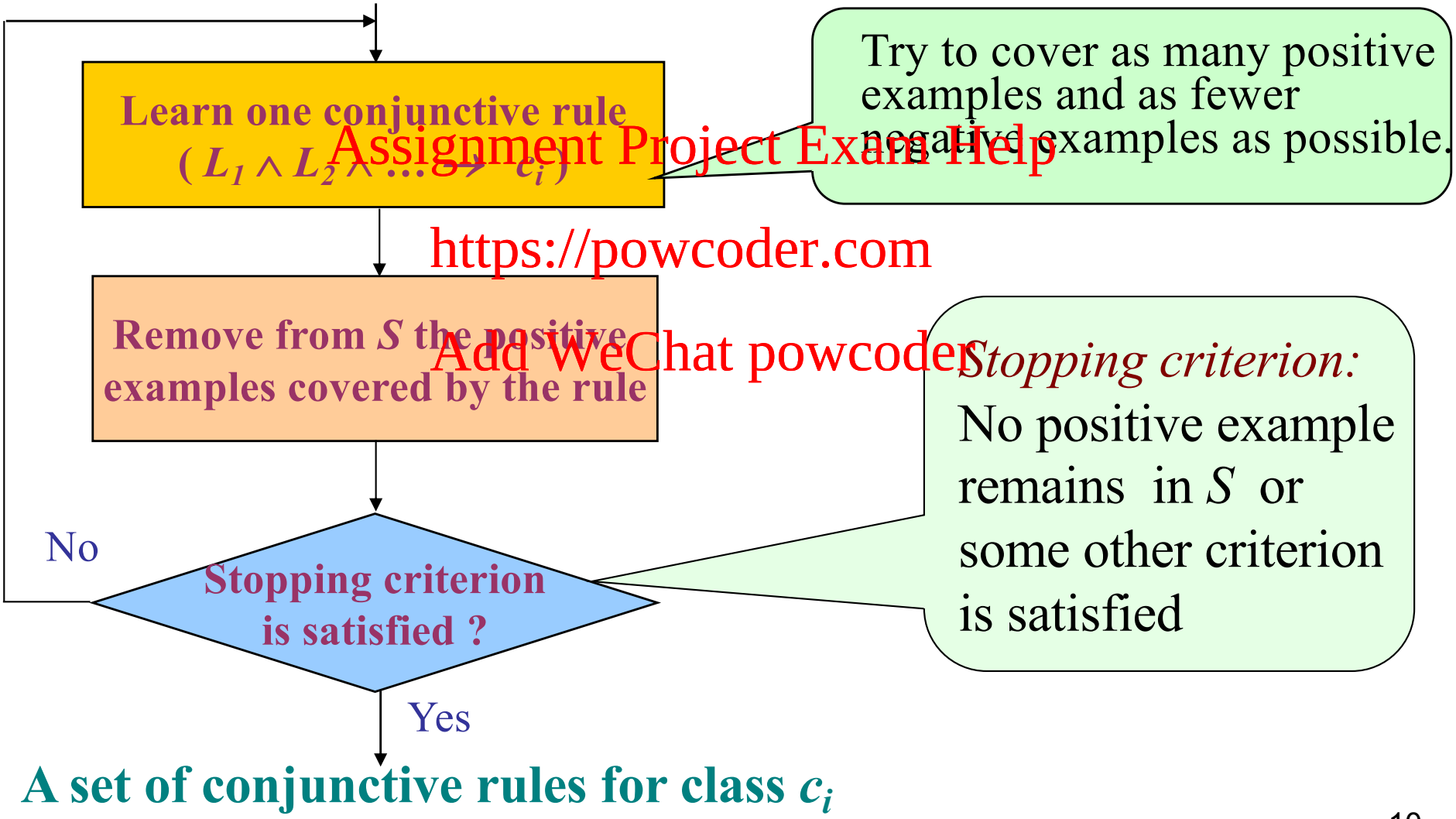
A Sequential Covering Example





Sequential Covering Algorithm

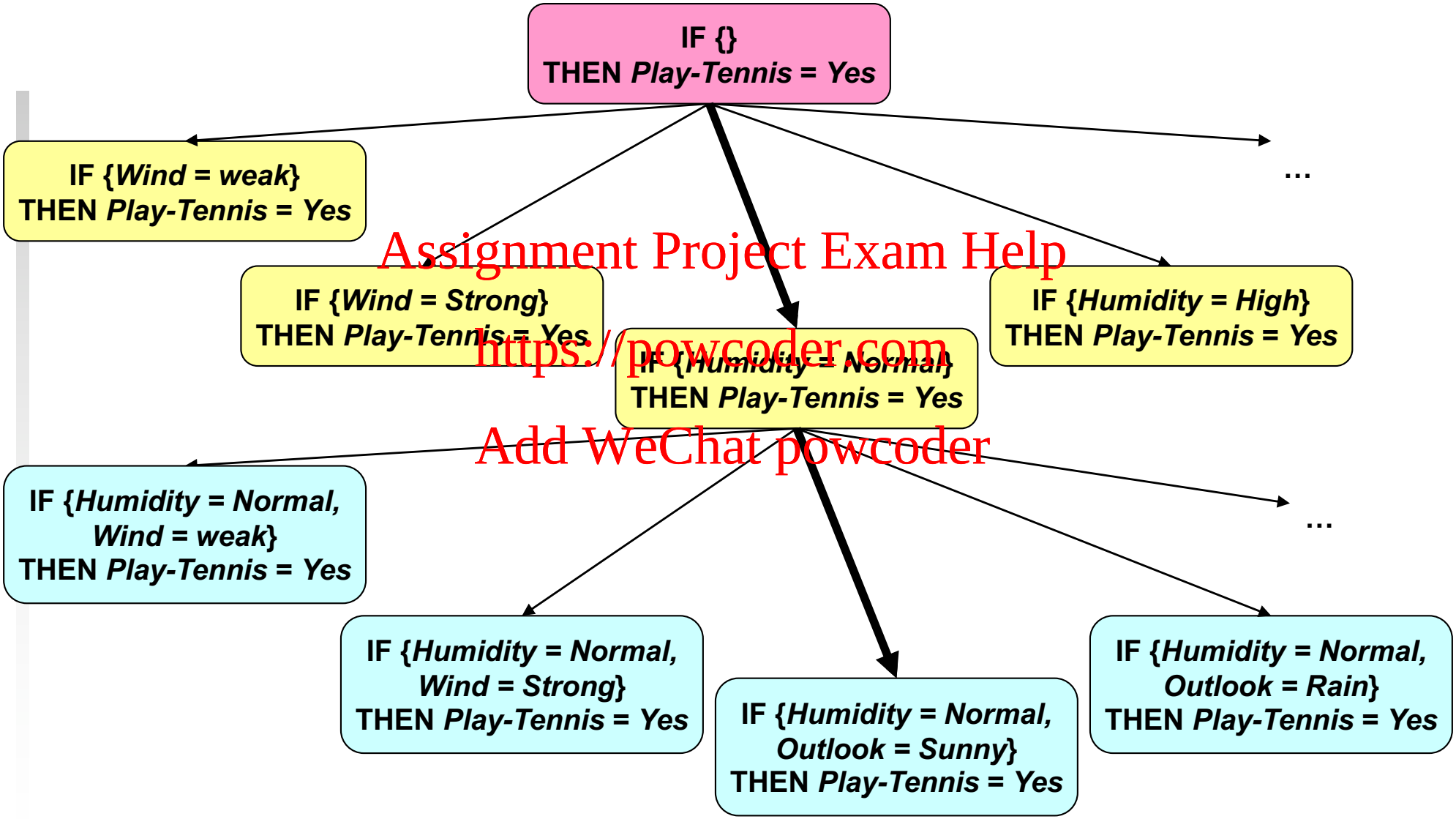
- Learn a set of rules for class c_i from training set S:



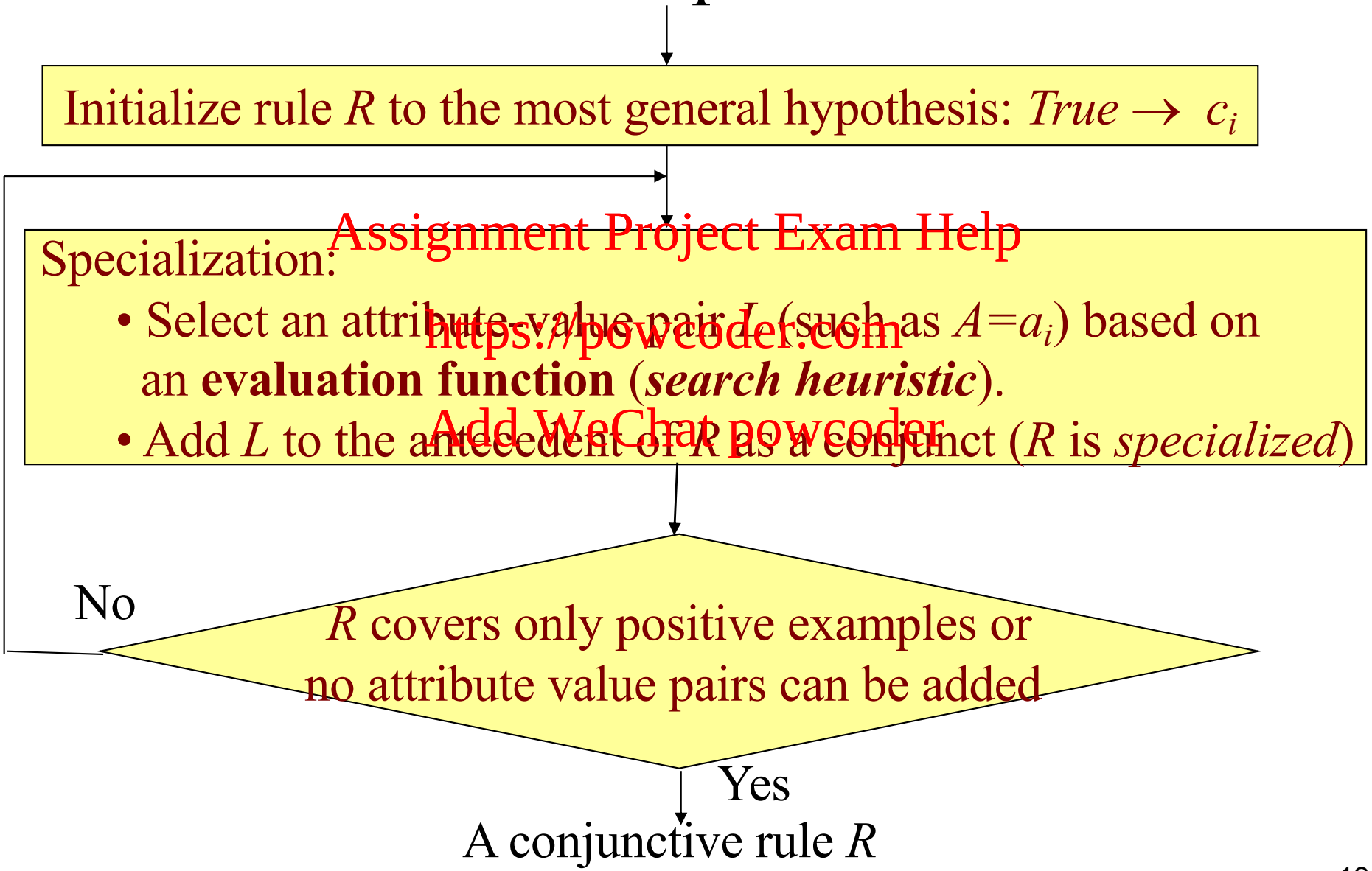
How to Learn One Conjunctive Rule

- General-to-Specific Search
  - Start with empty condition (the most general condition), add conjuncts (attribute-value pairs)
- Specific-to-General Search
  - Start with the most specific condition (such as an example), drop conjuncts (attribute-value pairs)
- Bi-directional search

General to Specific Search



Learning One Conjunctive Rule with General-to-Specific Search



Evaluation Functions for Selecting an Attribute-value Pair

- Training accuracy of the rule (AQ15):
  - Accuracy of the rule on the training examples:

$$p/t$$

where t is the total number of training examples that the new rule covers, and p is the number of these that are positive (i.e., belong to the class in question)

- Information gain (CN2, PRISM): $\log_2 \frac{p}{t} - \log_2 \frac{P}{T}$

where p and t are the same as above, and P and T are the corresponding numbers of examples that satisfied the rule before the new attribute-value pair was added.

  - It is equivalent to using p/t

Evaluation Functions for Selecting an Attribute-value Pair (*Cont'd*)

- *Training accuracy* itself is not a reliable indicator of the predictive accuracy on future test examples
  - Rules that have a high training accuracy may be too specific and overfit the data.
  - *Coverage* of the rule should be considered in the evaluation function.
- Information gain with coverage (used in FOIL):

$$p\left[\log_2 \frac{p}{t} - \log_2 \frac{P}{T}\right]$$

Contact lens data

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
middle-aged	myope	no	reduced	none
middle-aged	myope	no	normal	soft
middle-aged	myope	yes	reduced	none
middle-aged	myope	yes	normal	hard
middle-aged	hypermetrope	no	reduced	none
middle-aged	hypermetrope	no	normal	soft
middle-aged	hypermetrope	yes	reduced	none
middle-aged	hypermetrope	yes	normal	none
old	myope	no	reduced	none
old	myope	no	normal	none
old	myope	yes	reduced	none
old	myope	yes	normal	hard
old	hypermetrope	no	reduced	none
old	hypermetrope	no	normal	soft
old	hypermetrope	yes	reduced	none
old	hypermetrope	yes	normal	none

Learn the first rule from the data for class "hard"

- To begin, we seek a rule

  If ? then recommendation = hard

- For ?, we have 9 choices:

Assume *training accuracy* is used as the attribute selection criterion

Attribute-value pair	p/t
age = young	2/8
age = middle-aged	1/8
age = old	1/8
spectacle pres. = myope	3/12
spectacle pres. = hypermetrope	1/12
astigmatism = no	0/12
astigmatism = yes	4/12
tear prod. rate = reduced	0/12
tear prod. rate = normal	4/12

- The highest fraction is 4/12.
- Arbitrarily choose one or choose the first one:

If astigmatism = yes then recommendation = hard

Learn the first rule from the data for class "hard" (*Cont'd*)

- The rule

If astigmatism = yes then recommendation = hard

is not very accurate

- It covers 12 examples
- Needs to refine it:

If astigmatism = yes and ?
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
middle-aged	myope	yes	reduced	none
middle-aged	myope	yes	normal	hard
middle-aged	hypermetrope	yes	reduced	none
middle-aged	hypermetrope	yes	normal	none
old	myope	yes	reduced	none
old	myope	yes	normal	hard
old	hypermetrope	yes	reduced	none
old	hypermetrope	yes	normal	none

Learn the first rule from the data for class "hard" (*Cont'd*)

- For ? in:

If astigmatism = yes and ?
then recommendation = hard

- Consider 7 choices:

Attribute-value pair	p/t
age = young	2/4
age = middle-aged	1/4
age = old	1/4
spectacle pres. = myope	3/6
spectacle pres. = hypermetrope	1/6
tear prod. rate = reduced	0/6
tear prod. rate = normal	4/6

- The last one is the winner
- The rule is refined as

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

Learn the first rule from the data for class "hard" (*Cont'd*)

- The rule

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

is still not very accurate

- Refine it further:

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

age	Spectacle prescription	astigmatism	Tear production rate	Recommended lenses
young	myope	yes	normal	hard
young	hypermetrope	yes	normal	hard
middle -aged	myope	yes	normal	hard
middle -aged	hypermetrope	yes	normal	none
old	myope	yes	normal	hard
old	hypermetrope	yes	normal	none

Learn the first rule from the data for class "hard" (*Cont'd*)

- For ? in:

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

- Consider 5 choices

Attribute-value pair	p/t
age = young	2/2
age = middle-aged	1/2
age = old	1/2
spectacle pres. = myope	3/3
spectacle pres. = hypermetrope	1/3

- Choose

  spectacle pres. = myope

  because it has the highest ratio and its coverage is better than *age=young*

- The rule is refined to:

If astigmatism = yes and tear production rate = normal and spectacle pres. = myope then recommendation = hard

Contact lens data

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
middle-aged	myope	no	reduced	none
middle-aged	myope	no	normal	soft
middle-aged	myope	yes	reduced	none
middle-aged	myope	yes	normal	hard
middle-aged	hypermetrope	no	reduced	none
middle-aged	hypermetrope	no	normal	soft
middle-aged	hypermetrope	yes	reduced	none
middle-aged	hypermetrope	yes	normal	none
old	myope	no	reduced	none
old	myope	no	normal	none
old	myope	yes	reduced	none
old	myope	yes	normal	hard
old	hypermetrope	no	reduced	none
old	hypermetrope	no	normal	soft
old	hypermetrope	yes	reduced	none
old	hypermetrope	yes	normal	none

covered

Not Covered Positive example

Learn the second rule from the data for class "hard"

- The rule

If astigmatism = yes and tear production rate = normal and spectacle pres. = myope then recommendation = hard

is accurate, but covers only 2 positive examples

- Need to generate other rules to cover the rest of positive examples
  - Remove the 3 covered examples from the training data

  - Look for another rule using the same process
    - Start with a rule of the form:

      If ? then recommendation = hard
    - Using the same process, find the following rule:

      If age = young and astigmatism = yes and tear production rate = normal then recommendation = hard
    - This rule covers 2 of the original examples , one of which has been covered by the first rule. (It is ok) (It covers the first two positive examples)
    - Remove the example covered by the second rule from the training data. There is no positive example left. The process for learning rules for the "Hard" class stops.

Contact lens data

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
middle-aged	myope	no	reduced	none
middle-aged	myope	no	normal	soft
middle-aged	myope	yes	reduced	none
middle-aged	myope	yes	normal	hard
middle-aged	hypermetrope	no	reduced	none
middle-aged	hypermetrope	no	normal	soft
middle-aged	hypermetrope	yes	reduced	none
middle-aged	hypermetrope	yes	normal	none
old	myope	no	reduced	none
old	myope	no	normal	none
old	myope	yes	reduced	none
old	myope	yes	normal	hard
old	hypermetrope	no	reduced	none
old	hypermetrope	no	normal	soft
old	hypermetrope	yes	reduced	none
old	hypermetrope	yes	normal	none

Covered By the 2nd rule

Covered by the 1st rule

Overfitting Problem

- The learn-one-rule algorithm on the previous slides tries to learn a rule as accurate as possible
- Such a rule may
  - fit into noise
  - very complicated (containing many conjuncts)
  - exhibit low predictive accuracy on unseen examples
- Solution: approximate, but simpler rules
  - Pre-pruning
  - Post-pruning

Prepruning

- Stop the refinement of rules although they are still not accurate
- Most commonly used stopping criteria:
  - Minimum Purity Criterion
    - Requires a certain percentage of the examples covered by the rules is positive (i.e., a minimum training accuracy)
  - Difference Testing
    - Differences between the distribution of positive and negative examples covered by a rule and the overall distribution of positive and negative examples, or
    - Difference between distribution of instances covered by a rule and that of its direct predecessor
    - Only admits new attribute-value pair when the difference is above a user-set threshold (*cutoff*)
- Not as successful as post-pruning

Postpruning

- General idea
  - Learn a “pure” rule first
  - Remove some attribute value pairs from the rule to make it more general
  - Test the affect of the removal on a validation/pruning set on the quality of the rule
- Method 1
  - Separate the training data into growing set and validation set
  - Learn rules from the growing set
  - Test the accuracy of the pruned rule on a validation set

Postpruning (Cont'd)

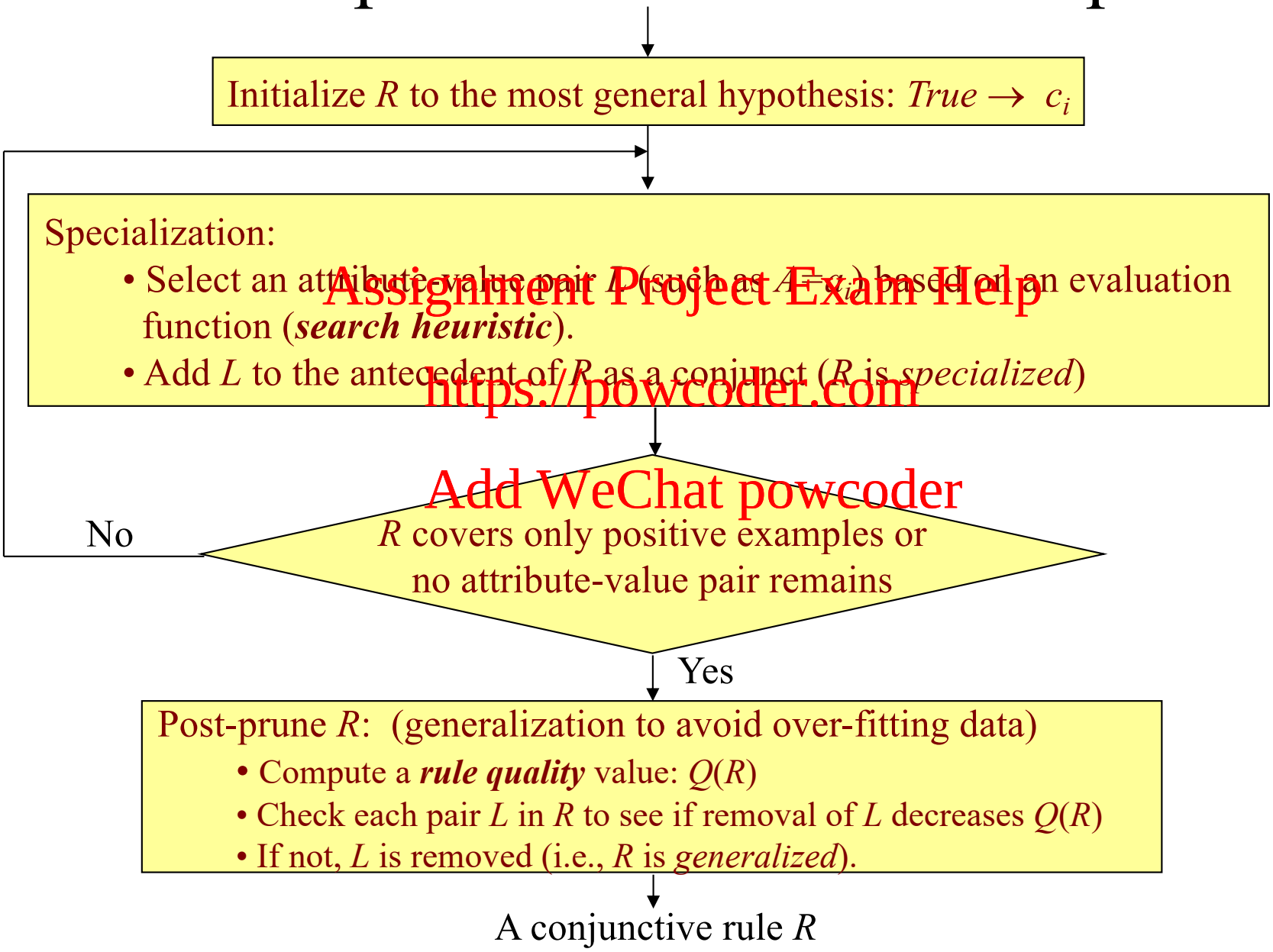
- Method 2
  - Use a rule quality measure, test the quality of the pruned rule on the same training set from which the rule was learned.
  - For example, ELEM2 uses rule quality formula

$$Q(R) = \log \frac{P(R \mid C)(1 - P(R \mid \overline{C}))}{P(R \mid \overline{C})(1 - P(R \mid C))}$$

where P(R|C) is the probability of an example covered by rule R given that the example belongs to C.

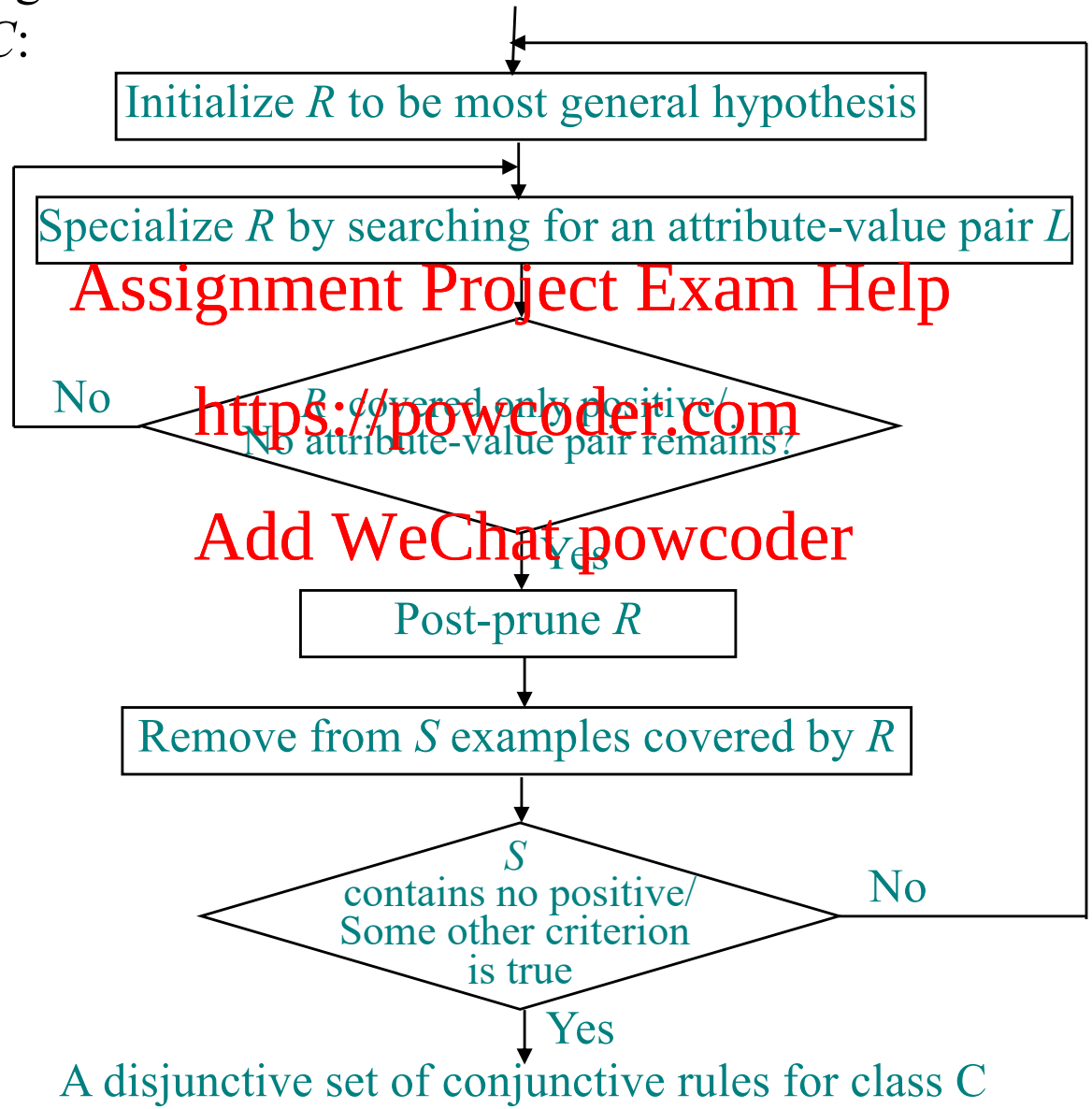
  - Procedure of postpruning with rule quality measure:
    - Compute a ***rule quality*** value: $Q(R)$
    - Check each attribute-value pair L in R in the reversed order of their generation to see if removal of L decreases $Q(R)$
    - If not, L is removed (i.e., R is *generalized*) and repeat the process.

Learning One Conjunctive Rule with General-to-Specific Search and Postpruning



Learn a set of conjunctive rules for a class

- Given a training set S and a class C:



Classification of New Example with Sets of Rules

- Three situations are possible when matching a new example with a set of rules:
  - Single-match
    - Only one rule is matched with the example. Classify the example into the class indicated by the rule.

Classification of New Example with Sets of Rules

- Multiple-match
  - Multiple rules are matched with the example
  - Two situations:
    - The matched rules indicate the same class. Classify the example into that class.
    - The matched rules indicate different classes.
      - Method 1: Rank the rules according to a criterion, use the first matched rule to classify the example
      - Method 2: Compute a decision score for each of the involved classes:

$$DS(C) = \sum_i Q(r_i)$$

where $Q(r_i)$ is a quality measure of rule r_i belonging to C and matched with the example.

choose the one with the highest decision score.

Classification of New Example with Sets of Rules *(Cont'd)*

- No-match
  - Method 1:
    - Use a default rule (the majority class) to classify the example
  - Method 2:
    - Partial matching is performed.
    - Calculate a matching score for each partially matched rule and a decision score for each involved class:

$$PMS(r_i) = \frac{\text{Number of matched attribute\_value\_pairs}}{\text{Number of attribute\_value\_pairs in } r_i} \times Q(r_i)$$

    - For each class involved, compute a decision score by summing up the partial matching scores for each partially matched rule within that class: $DS(C) = \sum_i PMS(r_i)$
    - Choose the class with the highest decision score.