Table of contents

1. Introduction

Nearly all of this material is derived from the text, Han, Kamber and Pei, Chapter 8.4, or the corresponding powerpoint slides made available by the publisher. Where a source other than the text or its slides was used for the material, attribution is given. Unless otherwise stated, images are copyright of the publisher, Elsevier.

Here we briefly look at learning methods that are designed to learn rules as expressive and readable class descriptions. These methods have been somewhat fringe but are increasingly important due to their capacity to explain their decisions (alternatively, to produce human-interpretable models).

2. Rule-Based Classification (Text: 8.4)

In this section, we look at rule-based classifiers, where the learned model is represented as a set of IF-THEN rules.

IF-THEN rules for classification

Rules are a good way of representing information or bits of knowledge. A rule-based classifier uses a set of IF-THEN rules for classification.

How to Represent the knowledge in the form of IF-THEN rules

Example

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

- "IF" part
  - Rule antecedent / precondition
  - Condition consists of one or more attribute tests that are logically ANDed
- "Then" part
  - Rule consequent
  - Contains a class prediction

Rule Evaluation

How to measure the quality of a single rule **R**? Use both *coverage* and *accuracy.* The key thing to note is that the accuracy of a rule is in proportion to its coverage, not to the size of the dataset as a whole.

- *n_covers* = number of tuples covered by **R,** i.e. that satisfy the antecedent of **R**
- *n_correct* = number of tuples correctly classified by **R** i.e that satisfy both the antecedent and the consequent
- *D* = training data set
- ***coverage***(**R**) = *n_covers* /|*D*|
  - the proportion of tuples that are covered by the rule
- ***accuracy***(**R**) = *n_correct* / *n_covers*
  - the proportion of covered tuples that are correctly labelled

Evaluation measures other than accuracy can be similarly adapted to using cover just as for accuracy here.

How to measure the quality of a set of rules?

Conventional accuracy (correct tuples as a proportion of all tuples in the dataset) is used to evaluate a rule based classifier comprising a set (or sequence) of rules. Other conventional classification quality measures can also be used this way for a set of rules that together form a classifier.

Conflict Resolution

If more than one rule is triggered, need conflict resolution

- Size ordering: assign the highest priority to the triggering rules that has the “toughest” requirement
  - i.e., with the most attribute tests
- Rule ordering: prioritise the rules beforehand by class-based or rule-based
  - Class-based ordering: prioritise classes beforehand. If a tuple is classified into multiple classes, choose a class by the class order.
  - Rule-based ordering: the rules are organised into one long priority list, according to some measure of rule quality

Default Rule

A default rule can be applied if there is no rule satisfied by a tuple.

3. Rule Extraction from a Decision Tree (Text: 8.4.2)

We have seen before how rules may be derived from a decision tree classifier.

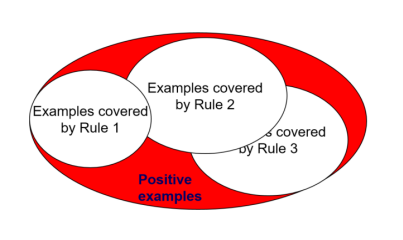
ACTION: Revisit this idea where we studied DecisionTrees

4. Rule Induction (Text 8.4.3)

IF-THEN rules can also be derived directly from the training data (i.e., without having to generate a decision tree first) using a sequential covering algorithm. Sometimes this is called *induction* or *abduction,* by reference to the language of logical inference, where rule languages have been extensively studied over a very long period.

Sequential covering algorithm

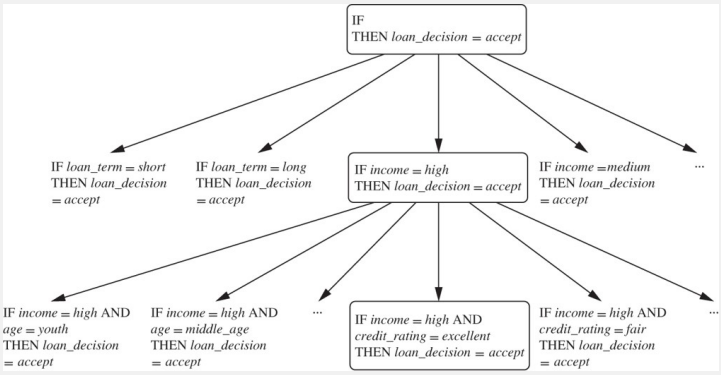
- Extracts rules directly from training data
- Typical sequential covering algorithms: FOIL, AQ, CN2, RIPPER
- Rules are learned sequentially, each for a given class C_i will cover many tuples of C_i but none (or few) of the tuples of other classes
- Steps:
  - Rules are learned one at a time
  - Each time a rule is learned, the tuples covered by the rules are removed from the training set
  - The process repeats on the remaining tuples until a termination condition
    - e.g., when no more training examples or when the quality of a rule returned is below a user-specified threshold



How each rule is learned

- Rules are grown in a general-to-specific manner.
- Choose a rule consequent (usually the positive class)
- Initialise an empty antecedent
- Repeat:
  - Add an attribute test to the rule antecedent
  - Until satisfied with this rule.

Example



Compared to rules derived from a Decision Tree

- Rules may be presented in a more expressive language, such as a relational language (e.g. can use relations like "adjacent" or "greater than" that relate attributes of covered tuples)

- Rules may be more compact and readable (do not branch from a common stem and tuples covered by an earlier rule are removed from training set)