Data Mining & Knowledge Discovery

Classification - 03

Md. Biplob Hosen

Lecturer, IIT-JU

Email: biplob.hosen@juniv.edu

Types of Classifiers

Binary versus Multiclass:

- Binary classifiers assign each data instance to one of two possible labels, typically denoted as +1 and -1.
- The positive class usually refers to the category we are more interested in predicting correctly compared to the negative class (e.g., the category in email classification problems).
- If there are more than two possible labels available, then the technique is known as a multiclass classifier.
- As some classifiers were designed for binary classes only, they must be adapted to deal with multiclass problems.

Deterministic versus Probabilistic

- A deterministic classifier produces a discrete-valued label to each data instance it classifies whereas a probabilistic classifier assigns a continuous score between 0 and 1 to indicate how likely it is that an instance belongs to a particular class, where the probability scores for all the classes sum up to 1.
- Some examples of probabilistic classifiers include the naïve Bayes classifier, Bayesian networks, and logistic regression.
- Probabilistic classifiers provide additional information about the confidence in assigning an instance to a class than deterministic classifiers.
- A data instance is typically assigned to the class with the highest probability score.

Linear versus Nonlinear

- A linear classifier uses a linear separating hyperplane to discriminate instances from different classes whereas a nonlinear classifier enables the construction of more complex, nonlinear decision surfaces.
- Although the linearity assumption makes the model less flexible in terms of fitting complex data, linear classifiers are thus less susceptible to model overfitting compared to nonlinear classifiers.
- Furthermore, one can transform the original set of attributes x= (x1, x2, ⋯ ,xd), into a more complex feature set, e.g., Φ(x)= (x1, x2, x1x2, x12, x22, ⋯), before applying the linear classifier.
- Such feature transformation allows the linear classifier to fit data sets with nonlinear decision surfaces.

Global versus Local

- A global classifier fits a single model to the entire data set.
- Unless the model is highly nonlinear, this one-size-fits-all strategy may not be effective when the relationship between the attributes and the class labels varies over the input space.
- In contrast, a local classifier partitions the input space into smaller regions and fits a distinct model to training instances in each region.
- The k-nearest neighbor classifier is a classic example of local classifiers.
- While local classifiers are more flexible in terms of fitting complex decision boundaries, they are also more susceptible to the model overfitting problem, especially when the local regions contain few training examples.

Generative versus Discriminative

- Given a data instance , the primary objective of any classifier is to predict the class label, y, of the data instance.
- However, apart from predicting the class label, we may also be interested in describing the underlying mechanism that generates the instances belonging to every class label.
- For example, in the process of classifying spam email messages, it may be useful to understand the typical characteristics of email messages that are labeled as spam, e.g., specific usage of keywords in the subject or the body of the email.
- Classifiers that learn a generative model of every class in the process of predicting class labels are known as generative classifiers.
- Some examples of generative classifiers include the naïve Bayes classifier and Bayesian networks.
- In contrast, discriminative classifiers directly predict the class labels without explicitly describing the distribution of every class label.
- They solve a simpler problem than generative models since they do not have the onus of deriving insights about the generative mechanism of data instances.
- They are thus sometimes preferred over generative models, especially when it is not crucial to obtain information about the properties of every class.
- Some examples of discriminative classifiers include decision trees, rule-based classifier, nearest neighbor classifier, artificial neural networks, and support vector machines.

Rule-Based Classifier

- A rule-based classifier uses a collection of “if …then…” rules (also known as a rule set) to classify data instances.
- Following table shows an example of a rule set generated for the vertebrate classification problem.
- Each classification rule in the rule set can be expressed in the following way - r_i: (Condition$_i$) $\rightarrow y_i$
- The left-hand side of the rule is called the rule **antecedent or precondition**. It contains a conjunction of attribute test conditions:
  - Condition$_i$ = $(A_1$ op $v_1) \wedge (A_2$ op $v_2) \ldots (A_k$ op $v_k)$,
- where (A_j, v_j) is an attribute-value pair and op is a comparison operator chosen from the set {=, ≠, <, >, ≤, ≥}.
- The right-hand side of the rule is called the rule **consequent**, which contains the predicted class y_i.

Continue…

- A rule r covers a data instance x if the precondition of r matches the attributes of x. r is also said to be fired or triggered whenever it covers a given instance.
- For an illustration, consider the rule given in the following table and the following attributes for two vertebrates: hawk and grizzly bear.

r1: (Gives Birth=no) ∧ (Aerial Creature=yes)→Birds
r2: (Gives Birth=no)∧(Aquatic Creature=yes)→Fishes
r3:(Gives Birth=yes)∧(Body Temperature=warmblooded)→Mammals
r4:(Gives Birth=no)∧(Aerial Creature=no)→Reptiles
r5:(Aquatic Creature=semi)→Amphibians

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates
Hawk	Warm-blooded	Feather	No	No	Yes	Yes	No
Grizzly bear	Warm-blooded	Fur	Yes	No	No	Yes	Yes

Continue…

- r1 covers the first vertebrate because its precondition is satisfied by the hawk's attributes.
- The rule does not cover the second vertebrate because grizzly bears give birth to their young and cannot fly, thus violating the precondition of r1.
- The quality of a classification rule can be evaluated using measures such as coverage and accuracy.
- Given a data set D and a classification rule r: $A \rightarrow y$, the coverage of the rule is the fraction of instances in D that trigger the rule r.
- On the other hand, its accuracy or confidence factor is the fraction of instances triggered by r whose class labels are equal to y.
- The formal definitions of these measures are:
  - Coverage(r) =$|A| / |D|$
  - Accuracy (r) = $|A \cap y| / |A|$,

- where $|A|$ is the number of instances that satisfy the rule antecedent, $|A \cap y|$ is the number of instances that satisfy both the antecedent and consequent, and $|D|$ is the total number of instances.

Table . The vertebrate data set.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	Mammals
python	cold-blooded	scales	no	no	no	no	yes	Reptiles
salmon	cold-blooded	scales	no	yes	no	no	no	Fishes
whale	warm-blooded	hair	yes	yes	no	no	no	Mammals
frog	cold-blooded	none	no	semi	no	yes	yes	Amphibians
komodo dragon	cold-blooded	scales	no	no	no	yes	no	Reptiles
bat	warm-blooded	hair	yes	no	yes	yes	yes	Mammals
pigeon	warm-blooded	feathers	no	no	yes	yes	no	Birds
cat	warm-blooded	fur	yes	no	no	yes	no	Mammals
guppy	cold-blooded	scales	yes	yes	no	no	no	Fishes
alligator	cold-blooded	scales	no	semi	no	yes	no	Reptiles
penguin	warm-blooded	feathers	no	semi	no	yes	no	Birds
porcupine	warm-blooded	quills	yes	no	no	yes	yes	Mammals
eel	cold-blooded	scales	no	yes	no	no	no	Fishes
salamander	cold-blooded	none	no	semi	no	yes	yes	Amphibians

Rule-Based Classifier

- Consider the data set.
- The rule (Gives Birth=yes) ∧ (Body Temperature=warm-blooded)→Mammals - has a coverage of 33% since five of the fifteen instances support the rule antecedent.
- The rule accuracy is 100% because all five vertebrates covered by the rule are mammals.

How a Rule-Based Classifier Works

- A rule-based classifier classifies a test instance based on the rule triggered by the instance.
- To illustrate how a rule-based classifier works, consider the rule set shown in the rule table and the following vertebrates:

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates
lemur	warm-blooded	fur	yes	no	no	yes	yes
turtle	cold-blooded	scales	no	semi	no	yes	no
dogfish shark	cold-blooded	scales	yes	yes	no	no	no

Continue…

- The first vertebrate, which is a lemur, is warm-blooded and gives birth to its young. It triggers the rule r3, and thus, is classified as a mammal.
- The second vertebrate, which is a turtle, triggers the rules r4 and r5.
- Since the classes predicted by the rules are contradictory (reptiles versus amphibians), their conflicting classes must be resolved.
- None of the rules are applicable to a dogfish shark.
- In this case, we need to determine what class to assign to such a test instance.

Properties of a Rule Set

- The rule set generated by a rule-based classifier can be characterized by the following two properties.

Mutually Exclusive Rule Set:

- The rules in a rule set R are mutually exclusive if no two rules in R are triggered by the same instance.
- This property ensures that every instance is covered by at most one rule in R.

Exhaustive Rule Set:

- A rule set R has exhaustive coverage if there is a rule for each combination of attribute values. This property ensures that every instance is covered by at least one rule in R.
- Together, these two properties ensure that every instance is covered by exactly one rule.
- If the rule set is not exhaustive, then a default rule, r_d: ()→y_d, must be added to cover the remaining cases.
- A default rule has an empty antecedent and is triggered when all other rules have failed.
- y_d is known as the default class and is typically assigned to the majority class of training instances not covered by the existing rules.

Properties of a Rule Set

- If the rule set is not mutually exclusive, then an instance can be covered by more than one rule, some of which may predict conflicting classes.

Ordered Rule Set:

- The rules in an ordered rule set R are ranked in decreasing order of their priority.
- An ordered rule set is also known as a decision list.
- The rank of a rule can be defined in many ways, e.g., based on its accuracy or total description length. When a test instance is presented, it will be classified by the highest-ranked rule that covers the instance.
- This avoids the problem of having conflicting classes predicted by multiple classification rules if the rule set is not mutually exclusive.
- An alternative way to handle a non-mutually exclusive rule set without ordering the rules is to consider the consequent of each rule triggered by a test instance as a vote for a particular class.
- The votes are then tallied to determine the class label of the test instance. The instance is usually assigned to the class that receives the highest number of votes.

Rule Extraction

- A rule-based classifier can be constructed using (1) direct methods, which extract classification rules directly from data, and (2) indirect methods, which extract classification rules from more complex classification models, such as decision trees and neural networks.

Direct Methods for Rule Extraction:

- The direct method for rule-based classification in data mining contains algorithms that extract rules directly from the dataset.

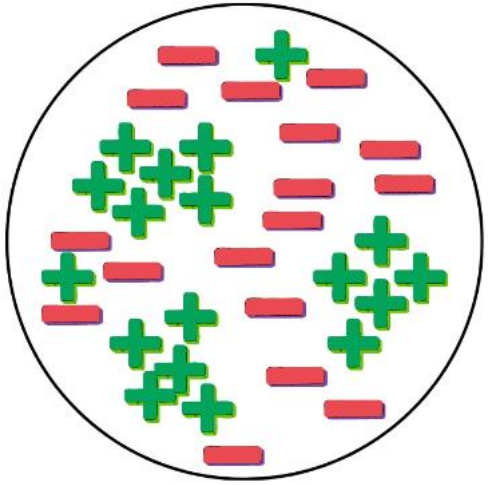
Sequentia Covering Algorithm:

- The sequential covering algorithm is the most widely used algorithm in rule-based classification in data mining. This algorithm defines rules to cover the maximum data records of one class and no data records of other classes.
- In these algorithms, rules are grown sequentially, one at a time. After a rule is grown, all data records covered by that rule are eliminated, and then this process keeps on repeating for the rest. This process will stop when we reach the stopping criteria, i.e., if the accuracy of the rule is below the required level, then we reject that rule and stop there.
- So we can understand the flow of the sequential covering algorithm through the following steps.

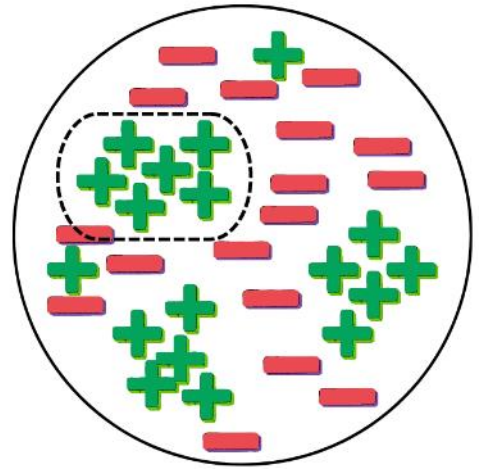
Sequentia Covering Algorithm

```
Algorithm: Sequential Covering

Input:
D, a data set class-labeled tuples,
Att_vals, the set of all attributes and their possible values.

Output:  A Set of IF-THEN rules.
Method:
Rule_set={ }; // initial set of rules learned is empty

for each class c do

   repeat
      Rule = Learn_One_Rule(D, Att_valls, c);
      remove tuples covered by Rule form D;
   until termination condition;

   Rule_set=Rule_set+Rule; // add a new rule to rule-set
end for
return Rule_Set;
```

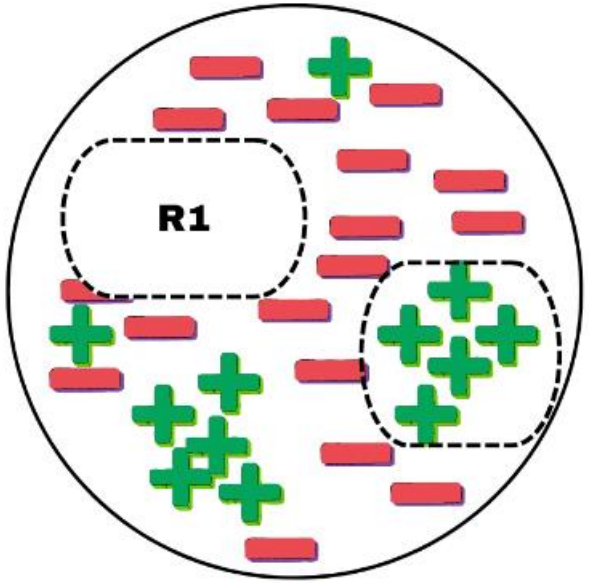
Sequentia Covering Algorithm



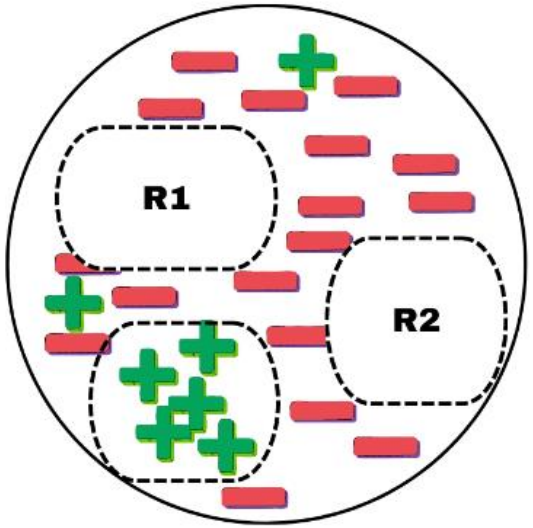
This is the original state



Rule Gowing(Rule 1)



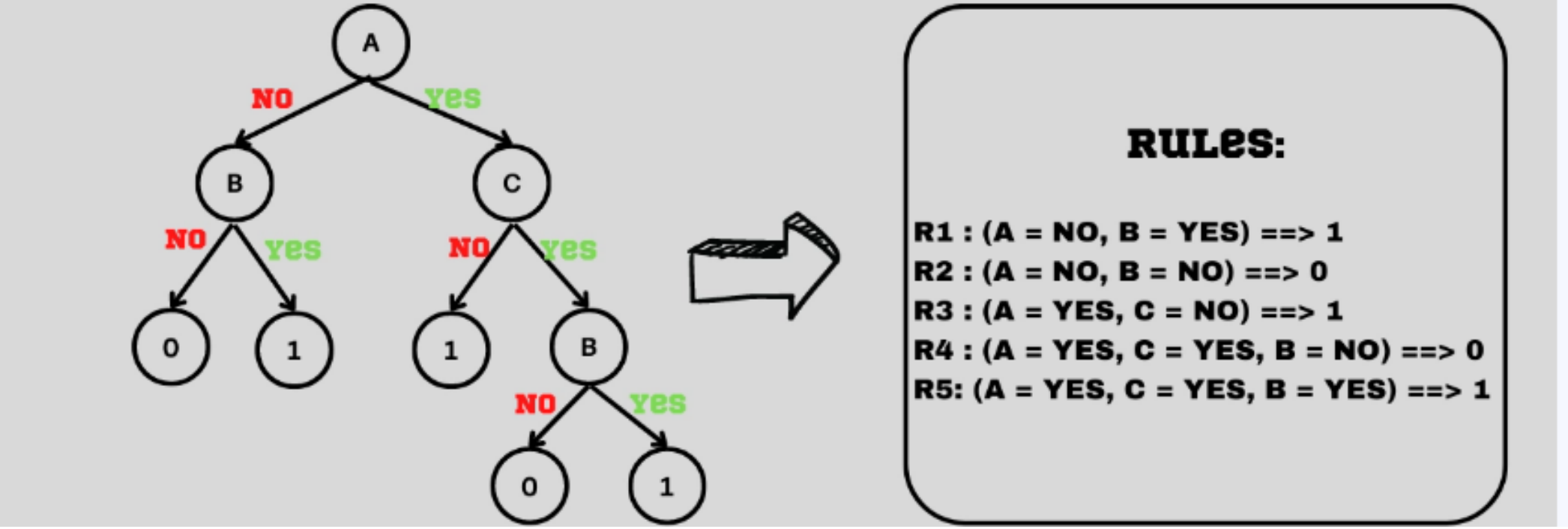
Instance Elimination(Removing Rule1 and growing Rule2)



Continuing this process until meeting the stopping criteria

Indirect Method

- In the indirect method of rule-based classification in data mining, we extract rules from different other classification models. Some prominent classifications from which we extract the rules are decision trees and neural networks.
- For example: Converting the decision tree to the rule set.



Exercise

1. Solution: https://csucidatamining.weebly.com/assign-5.html **(5.10.1)**
2. C4.5rules is an implementation of an indirect method for generating rules from a decision tree. RIPPER is an implementation of a direct method for generating rules directly from data.
   - Discuss the strengths and weaknesses of both methods.
3. Consider a training set that contains 100 positive examples and 400 negative examples. For each of the following candidate rules,
   R1: A→+(covers 4 positive and 1 negative examples),
   R2:B→+(covers 30 positive and 10 negative examples),
   R3:C→+(covers 100 positive and 90 negative examples),
   - Determine which is the best and worst candidate rule according to rule accuracy.

Thank You!