DATA MINING 2
Rule-based Classifiers

Riccardo Guidotti

a.a. 2022/2023

Slides edited from Tan, Steinbach, Kumar, Introduction to Data Mining
Rule-based Classifier

• Classify records by using a collection of “if...then...” rules

• Rule: \((Condition) \rightarrow y\)
  • where
    • *Condition* is a conjunction of tests on attributes
    • *y* is the class label
  • Examples of classification rules:
    • \((\text{Blood Type}=\text{Warm}) \land (\text{Lay Eggs}=\text{Yes}) \rightarrow \text{Birds}\)
    • \((\text{Taxable Income} < 50K) \land (\text{Refund}=\text{Yes}) \rightarrow \text{Evade}=\text{No}\)
Rule-based Classifier (Example)

R1: (Give Birth = no) \( \land \) (Can Fly = yes) \( \rightarrow \) Birds
R2: (Give Birth = no) \( \land \) (Live in Water = yes) \( \rightarrow \) Fishes
R3: (Give Birth = yes) \( \land \) (Blood Type = warm) \( \rightarrow \) Mammals
R4: (Give Birth = no) \( \land \) (Can Fly = no) \( \rightarrow \) Reptiles
R5: (Live in Water = sometimes) \( \rightarrow \) Amphibians

<table>
<thead>
<tr>
<th>Name</th>
<th>Blood Type</th>
<th>Give Birth</th>
<th>Can Fly</th>
<th>Live in Water</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
<td>warm</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>mammals</td>
</tr>
<tr>
<td>python</td>
<td>cold</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>reptiles</td>
</tr>
<tr>
<td>salmon</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>fish</td>
</tr>
<tr>
<td>whale</td>
<td>warm</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>mammals</td>
</tr>
<tr>
<td>frog</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>bat</td>
<td>warm</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>mammals</td>
</tr>
<tr>
<td>pigeon</td>
<td>warm</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>birds</td>
</tr>
<tr>
<td>cat</td>
<td>warm</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>mammals</td>
</tr>
<tr>
<td>leopard shark</td>
<td>cold</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>fish</td>
</tr>
<tr>
<td>turtle</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>reptiles</td>
</tr>
<tr>
<td>penguin</td>
<td>warm</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>birds</td>
</tr>
<tr>
<td>porcupine</td>
<td>warm</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>mammals</td>
</tr>
<tr>
<td>eel</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>fish</td>
</tr>
<tr>
<td>salamander</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>amphibians</td>
</tr>
<tr>
<td>gila monster</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>reptiles</td>
</tr>
<tr>
<td>platypus</td>
<td>warm</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>mammals</td>
</tr>
<tr>
<td>owl</td>
<td>warm</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>birds</td>
</tr>
<tr>
<td>dolphin</td>
<td>warm</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>mammals</td>
</tr>
<tr>
<td>eagle</td>
<td>warm</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>birds</td>
</tr>
</tbody>
</table>
Application of Rule-Based Classifier

• A rule \( r \) covers an instance \( x \) if the attributes of the instance satisfy the condition of the rule.

\[
\begin{align*}
R1: & \ (\text{Give Birth} = \text{no}) \land (\text{Can Fly} = \text{yes}) \rightarrow \text{Birds} \\
R2: & \ (\text{Give Birth} = \text{no}) \land (\text{Live in Water} = \text{yes}) \rightarrow \text{Fishes} \\
R3: & \ (\text{Give Birth} = \text{yes}) \land (\text{Blood Type} = \text{warm}) \rightarrow \text{Mammals} \\
R4: & \ (\text{Give Birth} = \text{no}) \land (\text{Can Fly} = \text{no}) \rightarrow \text{Reptiles} \\
R5: & \ (\text{Live in Water} = \text{sometimes}) \rightarrow \text{Amphibians}
\end{align*}
\]

<table>
<thead>
<tr>
<th>Name</th>
<th>Blood Type</th>
<th>Give Birth</th>
<th>Can Fly</th>
<th>Live in Water</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>hawk</td>
<td>warm</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>?</td>
</tr>
<tr>
<td>grizzly bear</td>
<td>warm</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>?</td>
</tr>
</tbody>
</table>

The rule R1 covers a hawk => Bird
The rule R3 covers the grizzly bear => Mammal
Rule Coverage and Accuracy

- **Coverage of a rule:**
  - Fraction of records that satisfy the antecedent of a rule

- **Accuracy of a rule:**
  - Fraction of records that satisfy the antecedent that also satisfy the consequent of a rule

<table>
<thead>
<tr>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Single</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Married</td>
<td>100K</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>Single</td>
<td>70K</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Married</td>
<td>120K</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Divorced</td>
<td>95K</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>Married</td>
<td>60K</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>Divorced</td>
<td>220K</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>Single</td>
<td>85K</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Married</td>
<td>75K</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>Single</td>
<td>90K</td>
<td>Yes</td>
</tr>
</tbody>
</table>

(Status=Single) → No

Coverage = 40%, Accuracy = 50%
How does Rule-based Classifier Work?

R1: (Give Birth = no) \land (Can Fly = yes) \rightarrow Birds
R2: (Give Birth = no) \land (Live in Water = yes) \rightarrow Fishes
R3: (Give Birth = yes) \land (Blood Type = warm) \rightarrow Mammals
R4: (Give Birth = no) \land (Can Fly = no) \rightarrow Reptiles
R5: (Live in Water = sometimes) \rightarrow Amphibians

<table>
<thead>
<tr>
<th>Name</th>
<th>Blood Type</th>
<th>Give Birth</th>
<th>Can Fly</th>
<th>Live in Water</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>lemur</td>
<td>warm</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>?</td>
</tr>
<tr>
<td>turtle</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>sometimes</td>
<td>?</td>
</tr>
<tr>
<td>dogfish shark</td>
<td>cold</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>?</td>
</tr>
</tbody>
</table>

A lemur triggers rule R3, so it is classified as a mammal
A turtle triggers both R4 and R5
A dogfish shark triggers none of the rules
Characteristics of Rule Sets: Strategy 1

• **Mutually exclusive rules**
  • Classifier contains mutually exclusive rules if the rules are independent of each other
  • Every record is covered by at most one rule

• **Exhaustive rules**
  • Classifier has exhaustive coverage if it accounts for every possible combination of attribute values
  • Each record is covered by at least one rule
Characteristics of Rule Sets: Strategy 2

• **Rules are not mutually exclusive**
  - A record may trigger more than one rule
  - Solution?
    - Ordered rule set
    - Unordered rule set – use voting schemes

• **Rules are not exhaustive**
  - A record may not trigger any rules
  - Solution?
    - Use a default class
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

<table>
<thead>
<tr>
<th>Rule</th>
<th>Condition</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>(Give Birth = no) ∧ (Can Fly = yes)</td>
<td>Birds</td>
</tr>
<tr>
<td>R2</td>
<td>(Give Birth = no) ∧ (Live in Water = yes)</td>
<td>Fishes</td>
</tr>
<tr>
<td>R3</td>
<td>(Give Birth = yes) ∧ (Blood Type = warm)</td>
<td>Mammals</td>
</tr>
<tr>
<td>R4</td>
<td>(Give Birth = no) ∧ (Can Fly = no)</td>
<td>Reptiles</td>
</tr>
<tr>
<td>R5</td>
<td>(Live in Water = sometimes)</td>
<td>Amphibians</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Blood Type</th>
<th>Give Birth</th>
<th>Can Fly</th>
<th>Live in Water</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>turtle</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>sometimes</td>
<td>?</td>
</tr>
</tbody>
</table>
Rule Ordering Schemes

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

- **Class-based ordering**
  - Rules that belong to the same class appear together

### Rule-based Ordering

- (Refund=Yes) => No
- (Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) => No
- (Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) => Yes
- (Refund=No, Marital Status={Married}) => No

### Class-based Ordering

- (Refund=Yes) => No
- (Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) => No
- (Refund=No, Marital Status={Married}) => No
- (Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) => Yes
Building Classification Rules

• Direct Method:
  • Extract rules directly from data
  • Examples: RIPPER, CN2, Holte’s 1R

• Indirect Method:
  • Extract rules from other classification models (e.g. decision trees).
  • Examples: C4.5rules
Direct Method: 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
Example of Sequential Covering

(i) Original Data

(ii) Step 1
Example of Sequential Covering...

(iii) Step 2

(iv) Step 3
Rule Growing

- Two common strategies

(a) General-to-specific

(b) Specific-to-general
Rule Evaluation

• FOIL’s Information Gain
  
  • R0: {} => class (initial rule)
  • R1: {A} => class (rule after adding conjunct)

  \[
  \text{Gain}(R_0, R_1) = p_1 \times \left[ \log_2 \left( \frac{p_1}{p_1 + n_1} \right) - \log_2 \left( \frac{p_0}{p_0 + n_0} \right) \right]
  \]

  • \(p_0\): number of positive instances covered by R0
  • \(n_0\): number of negative instances covered by R0
  • \(p_1\): number of positive instances covered by R1
  • \(n_1\): number of negative instances covered by R1

FOIL: First Order Inductive Learner – an early rule-based learning algorithm
Minimum Description Length (MDL)

- \[ \text{Cost(Model,Data)} = \text{Cost(Data|Model)} + \alpha \times \text{Cost(Model)} \]
  - Cost is the number of bits needed for encoding.
  - Search for the least costly model.
- \[ \text{Cost(Data|Model)} \] encodes the misclassification errors.
- \[ \text{Cost(Model)} \] uses node encoding (number of children) plus splitting condition encoding.
Pessimistic Error Estimate

• **Pessimistic Error Estimate** of a rule set $T$ with $k$ rules:

$$err_{gen}(T) = err(T) + \Omega \times \frac{k}{N_{train}}$$

• $err(T)$: error rate on all training records
• $\Omega$: trade-off hyper-parameter relative cost of adding a rule
• $k$: number of rules
• $N_{train}$: total number of training records
Direct Method: RIPPER

• For 2-class problem, choose one of the classes as positive class, and the other as negative class
  • Learn rules for positive class
  • Negative class will be the default class

• For multi-class problem
  • Order the classes according to increasing class prevalence (fraction of instances that belong to a particular class)
  • Learn the rule set for smallest class first, treat the rest as negative class
  • Repeat with next smallest class as positive class
Direct Method: RIPPER

• Growing a rule:
  • Start from empty rule
  • Add conjuncts as long as they improve FOIL’s information gain
  • Stop when rule no longer covers negative examples
  • Prune the rule immediately using incremental reduced error pruning
  • Measure for pruning: \( v = (p-n)/(p+n) \)
    • \( p \): number of positive examples covered by the rule in the validation set
    • \( n \): number of negative examples covered by the rule in the validation set
  • Pruning method: delete any final sequence of conditions that maximizes \( v \)
Direct Method: RIPPER

• Building a Rule Set:
  • Use sequential covering algorithm
    • Finds the best rule that covers the current set of positive examples
    • Eliminate both positive and negative examples covered by the rule
  • Each time a rule is added to the rule set, compute the new description length
    • Stop adding new rules when the new description length is $d$ bits longer than the smallest description length obtained so far
Direct Method: RIPPER

• Optimize the rule set:
  • For each rule $r$ in the rule set $R$
    • Consider 2 alternative rules:
      • Replacement rule ($r*$): grow new rule from scratch
      • Revised rule ($r'$): add conjuncts to extend the rule $r$
    • Compare the rule set for $r$ against the rule set for $r*$ and $r'$
    • Choose rule set that minimizes MDL principle
Indirect Methods

Rule Set

r1: (P=No, Q=No) ==> -
r2: (P=No, Q=Yes) ==> +
r3: (P=Yes, R=No) ==> +
r4: (P=Yes, R=Yes, Q=No) ==> -
r5: (P=Yes, R=Yes, Q=Yes) ==> +
Indirect Method: C4.5rules

• Extract rules from an unpruned decision tree
• For each rule, $r: A \rightarrow y$,
  • consider an alternative rule $r': A' \rightarrow y$ where $A'$ is obtained by removing one of the conjuncts in $A$
  • Compare the pessimistic error rate for $r$ against all $r'$s
  • Prune if one of the alternative rules has lower pessimistic error rate
  • Remove duplicate rules
  • Repeat until we can no longer improve generalization error
Indirect Method: C4.5rules

• Instead of ordering the rules, order subsets of rules (class ordering)
  • Each subset is a collection of rules with the same rule consequent (class)
  • Compute description length of each subset
<table>
<thead>
<tr>
<th>Name</th>
<th>Give Birth</th>
<th>Lay Eggs</th>
<th>Can Fly</th>
<th>Live in Water</th>
<th>Have Legs</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>mammals</td>
</tr>
<tr>
<td>python</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>reptiles</td>
</tr>
<tr>
<td>salmon</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>fishes</td>
</tr>
<tr>
<td>whale</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>mammals</td>
</tr>
<tr>
<td>frog</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>sometimes</td>
<td>yes</td>
<td>amphibians</td>
</tr>
<tr>
<td>komodo</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>reptiles</td>
</tr>
<tr>
<td>bat</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>mammals</td>
</tr>
<tr>
<td>pigeon</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>birds</td>
</tr>
<tr>
<td>cat</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>mammals</td>
</tr>
<tr>
<td>leopard shark</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>fishes</td>
</tr>
<tr>
<td>turtle</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>sometimes</td>
<td>yes</td>
<td>reptiles</td>
</tr>
<tr>
<td>penguin</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>sometimes</td>
<td>yes</td>
<td>birds</td>
</tr>
<tr>
<td>porcupine</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>mammals</td>
</tr>
<tr>
<td>eel</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>fishes</td>
</tr>
<tr>
<td>salamander</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>sometimes</td>
<td>yes</td>
<td>amphibians</td>
</tr>
<tr>
<td>gila monster</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>reptiles</td>
</tr>
<tr>
<td>platypus</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>mammals</td>
</tr>
<tr>
<td>owl</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>birds</td>
</tr>
<tr>
<td>dolphin</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>mammals</td>
</tr>
<tr>
<td>eagle</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>birds</td>
</tr>
</tbody>
</table>
C4.5 versus C4.5rules versus RIPPER

**C4.5rules:**
- (Give Birth=No, Can Fly=Yes) → Birds
- (Give Birth=No, Live in Water=Yes) → Fishes
- (Give Birth=Yes) → Mammals
- (Give Birth=No, Can Fly=No, Live in Water=No) → Reptiles
- ( ) → Amphibians

**RIPPER:**
- (Live in Water=Yes) → Fishes
- (Have Legs=No) → Reptiles
- (Give Birth=No, Can Fly=No, Live In Water=No) → Reptiles
- (Can Fly=Yes, Give Birth=No) → Birds
- ( ) → Mammals
C4.5 versus C4.5rules versus RIPPER

C4.5 and C4.5rules:

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
<th>Amphibians</th>
<th>Fishes</th>
<th>Reptiles</th>
<th>Birds</th>
<th>Mammals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amphibians</td>
<td>Amphibians</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fishes</td>
<td>Fishes</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Reptiles</td>
<td>Reptiles</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Birds</td>
<td>Birds</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Mammals</td>
<td>Mammals</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

RIPPER:

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
<th>Amphibians</th>
<th>Fishes</th>
<th>Reptiles</th>
<th>Birds</th>
<th>Mammals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amphibians</td>
<td>Amphibians</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Fishes</td>
<td>Fishes</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Reptiles</td>
<td>Reptiles</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Birds</td>
<td>Birds</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Mammals</td>
<td>Mammals</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>
Advantages of Rule-Based Classifiers

• Has characteristics quite similar to decision trees
  • As highly expressive as decision trees
  • Easy to interpret
  • Performance comparable to decision trees
  • Can handle redundant attributes

• Better suited for handling imbalanced classes

• Harder to handle missing values in the test set
References

• Rule-Based Classifiers. Chapter 5.1. Introduction to Data Mining.