DATA MINING - 1DL105, 1DL111

Fall 2007

An introductory class in data mining

http://user.it.uu.se/~udbl/dm-ht2007/
alt. http://www.it.uu.se/edu/course/homepage/infoutv/ht07

Kjell Orsborn
Uppsala Database Laboratory
Department of Information Technology, Uppsala University,
Uppsala, Sweden
Data Mining
Classification: Alternative Techniques

(Tan, Steinbach, Kumar ch. 5)

Kjell Orsborn

Department of Information Technology
Uppsala University, Uppsala, Sweden
Rule-based classifier

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

• Rule: \((Condition) \rightarrow y\)
  - where
    • \(Condition\) is a conjunction of attributes
    • \(y\) is the class label
  - \(LHS\): rule antecedent or condition
  - \(RHS\): rule consequent
  - 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

<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>no</td>
<td>no</td>
<td>fishes</td>
</tr>
<tr>
<td>salmon</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>mammals</td>
</tr>
<tr>
<td>whale</td>
<td>warm</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>amphibians</td>
</tr>
<tr>
<td>frog</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>sometimes</td>
<td>reptiles</td>
</tr>
<tr>
<td>komodo</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>mammals</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>yes</td>
<td>fishes</td>
</tr>
<tr>
<td>turtle</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>sometimes</td>
<td>reptiles</td>
</tr>
<tr>
<td>penguin</td>
<td>warm</td>
<td>no</td>
<td>no</td>
<td>sometimes</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>yes</td>
<td>fishes</td>
</tr>
<tr>
<td>salamander</td>
<td>cold</td>
<td>no</td>
<td>no</td>
<td>sometimes</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>

**R1:** (Give Birth = no) ∧ (Can Fly = yes) → Birds  
**R2:** (Give Birth = no) ∧ (Live in Water = yes) → Fishes  
**R3:** (Give Birth = yes) ∧ (Blood Type = warm) → Mammals  
**R4:** (Give Birth = no) ∧ (Can Fly = no) → Reptiles  
**R5:** (Live in Water = sometimes) → Amphibians
Application of Rule-based classifier

- A rule $r$ covers an instance $x$ if the attributes of the instance satisfy the condition of the rule

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>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 $\Rightarrow$ Bird
The rule R3 covers the grizzly bear $\Rightarrow$ 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 both the antecedent and 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) ∧ (Can Fly = yes) → Birds
R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes
R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals
R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles
R5: (Live in Water = sometimes) → 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-based classifier

• 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
From Decision trees to Rules

Classification Rules

- (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

Rules are mutually exclusive and exhaustive

Rule set contains as much information as the tree
Rules can be simplified

Initial Rule: \((\text{Refund}=\text{No}) \land (\text{Status}=\text{Married})\) \rightarrow \text{No}

Simplified Rule: \((\text{Status}=\text{Married})\) \rightarrow \text{No}
Effect of rule simplification

• Rules are no longer mutually exclusive
  – A record may trigger more than one rule
  – Solution?
    • Ordered rule set
    • Unordered rule set – use voting schemes

• Rules are no longer 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

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>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}) ==> Yes
Building classification rules

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

• Indirect Method:
  • Extract rules from other classification models (e.g. decision trees, neural networks, etc).
  • e.g.: C4.5rules
Direct method: Sequential covering

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

• To learn one rule we use one of the strategies below:
  • Top-down:
    – Start with maximally general rule
    – Add literals one by one
  • Bottom-up:
    – Start with maximally specific rule
    – Remove literals one by one
  • Combination of top-down and bottom-up:
    – Candidate-elimination algorithm
Example of sequential covering

(i) Original Data

(ii) Step 1
Example of sequential covering...

(iii) Step 2

(iv) Step 3
Aspects of sequential covering

- Rule Growing
- Instance Elimination
- Rule Evaluation
- Stopping Criterion
- Rule Pruning
Rule growing

Two common strategies

(a) General-to-specific

(b) Specific-to-general
Rule growing examples

• **CN2 Algorithm:**
  - 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:**
  - 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)
    - \text{Gain}(R0, R1) = t \left[ \log \left( \frac{p1}{p1+n1} \right) - \log \left( \frac{p0}{p0 + n0} \right) \right]
    - 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 \( R1 \)
Instance elimination

- Why do we need to eliminate instances?
  - Otherwise, the next rule is identical to previous rule
- Why do we remove positive instances?
  - Ensure that the next rule is different
- Why do we remove negative instances?
  - Prevent underestimating accuracy of rule
  - Compare rules R2 and R3 in the diagram
Rule evaluation

• Heuristics for Learning One Rule - When is a rule “good”?
  – High accuracy;
  – Less important: high coverage.
• Metrics:
  – Accuracy, (relative frequency): $n_c/n$
  – Laplace: $(n_c + 1)/(n+k)$
  – M-estimate of accuracy: $(n_c + kp)/(n+k)$,
    • where $n_c$ is the number of correctly classified instances, and
    • $n$ is the number of instances covered by the rule, and
    • $p$ is the prior probability of the class predicted by the rule, and
    • $k$ is the number of classes or the weight of $p$.
  – Entropy
• The Laplace, M-estimate and Entropy metrics take rule coverage into account
Stopping criterion and rule pruning

• Stopping criterion
  – Compute the gain
  – If gain is not significant, discard the new rule

• Rule Pruning
  – Similar to post-pruning of decision trees
  – Reduced Error Pruning:
    • Remove one of the conjuncts in the rule
    • Compare error rate on validation set before and after pruning
    • If error improves, prune the conjunct
Summary of direct method

- Grow a single rule
- Remove instances from rule
- Prune the rule (if necessary)
- Add rule to Current Rule Set
- Repeat
Indirect methods

Rule Set

r1: \((P=\text{No}, Q=\text{No}) \implies -\)
r2: \((P=\text{No}, Q=\text{Yes}) \implies +\)
r3: \((P=\text{Yes}, R=\text{No}) \implies +\)
r4: \((P=\text{Yes}, R=\text{Yes}, Q=\text{No}) \implies -\)
r5: \((P=\text{Yes}, R=\text{Yes}, Q=\text{Yes}) \implies +\)
Advantages of rule-based classifiers

• As highly expressive as decision trees
• Easy to interpret
• Easy to generate
• Can classify new instances rapidly
• Performance comparable to decision trees