CS-F415: Data Mining

07 Rule-Based Classification

Dr. Kamlesh Tiwari
Associate Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Feb 16, 2024 M/W/F 4:00pm 6101 @ BITS-Pilani [Jan-May 2024]

http://ktiwari.in/dm

Classification

Classification maps data - into *predefined* labels.

Example: Lots of mails are there in my mail box. Can you tell me which ones are SPAM?

- Task of supervised learning
- Often based on some patterns or characteristics
- We can use the frequency of words
- Assumption is that some words appears more or less frequently in a SPAM mail

Rule-Based Classifier

Classify records by using a collection of

if...then...

rules

- **Rule:** (*condition*) $\rightarrow y$

where

- ⋆ **antecedent:** condition is a conjunction of tests on attributes
- ⋆ **consequent:** y is class label

Examples of classification rules:

- ⋆ $(bloodType = warm) \wedge (layEgs = Yes) \rightarrow Birds$
- ⋆ $(taxableIncome < 50L) \wedge (Refund = Yes) \rightarrow Evade = No$

Example

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: $(giveBirth = No) \wedge (canFly = Yes) \rightarrow Birds$
R2: $(giveBirth = No) \wedge (liveInWater = Yes) \rightarrow Fishes$
R3: $(giveBirth = Yes) \wedge (bloodType = warm) \rightarrow Mammals$
R4: $(giveBirth = No) \wedge (canFly = No) \rightarrow Reptiles$
R5: $(liveInWater = sometimes) \rightarrow Amphibians$

Application of Rule-Based Classifier

A rule R covers an instance X, if the attributes of X satisfy the condition of the rule

R1: $(giveBirth = No) \wedge (canFly = Yes) \rightarrow Birds$
R2: $(giveBirth = No) \wedge (liveInWater = Yes) \rightarrow Fishes$
R3: $(giveBirth = Yes) \wedge (bloodType = warm) \rightarrow Mammals$
R4: $(giveBirth = No) \wedge (canFly = No) \rightarrow Reptiles$
R5: $(liveInWater = sometimes) \rightarrow Amphibians$

- hawk (bloodType, giveBirth, canFly, liveInWater) = (warm, no, yes, no) $\rightarrow$?
  Is Bird by using R1
- grizzly bear (bloodType, giveBirth, canFly, liveInWater) = (warm, yes, no, no) $\rightarrow$?
  Is Mammal by using R3

Coverage and Accuracy: Decide quality of rules

Coverage

Fraction of records that satisfy the antecedent of a rule

$$n_{covers}/|D|$$

Accuracy

Fraction of records that satisfy the antecedent that also satisfy the consequent of a rule

$$n_{crrect}/n_{covers}$$

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Consider: $(Status = Single) \rightarrow No$

Coverage = 40%

Accuracy = 50%

How does Rule-based Classifier Work?

R1: $(giveBirth = No) \wedge (canFly = Yes) \rightarrow Birds$
R2: $(giveBirth = No) \wedge (liveInWater = Yes) \rightarrow Fishes$
R3: $(giveBirth = Yes) \wedge (bloodType = warm) \rightarrow Mammals$
R4: $(giveBirth = No) \wedge (canFly = No) \rightarrow Reptiles$
R5: $(liveInWater = sometimes) \rightarrow Amphibians$

- lemur (bloodType, giveBirth, canFly, liveInWater) = (warm, yes, no, no) →?
  Rule R3 applies
- turtle (bloodType, giveBirth, canFly, liveInWater) = (cold, no, no, sometimes) →?
  Rule R4 + R5 applies
- dogfish shark (bloodType, giveBirth, canFly, liveInWater) = (cold, yes, no, yes) →?
  No Rule applies

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
- Resolve Ties By
  - 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

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

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

?? Reptiles

Schemes for Rule Ordering

Rule-based ordering: use their quality

$(refund = yes) \rightarrow no$

$(refund = no, maritalStatus = \{single, divorces\}, taxableIncome < 80K) \rightarrow no$

$(refund = no, maritalStatus = \{single, divorces\}, taxableIncome > 80K) \rightarrow yes$

$(refund = no, maritalStatus = \{married\}) \rightarrow no$

Rule-based ordering: same class ones appear together

$(refund = yes) \rightarrow no$

$(refund = no, maritalStatus = \{single, divorces\}, taxableIncome < 80K) \rightarrow no$

$(refund = no, maritalStatus = \{married\}) \rightarrow no$

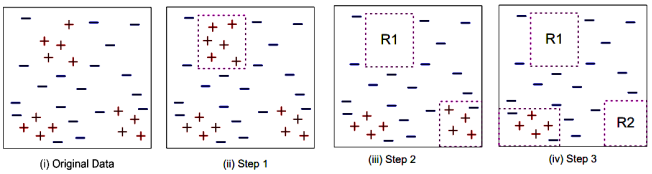
$(refund = no, maritalStatus = \{single, divorces\}, taxableIncome > 80K) \rightarrow yes$

How to Build Classification Rules?

- **Direct Method:**
  Extract rules directly from data
  Examples: Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Climbing and Pruning Rule Learne (CN2), Holte's 1R
- **Indirect Method:**
  Extract rules from other classification models (e.g. decision trees, neural networks, etc).
  Examples: C4.5 rules

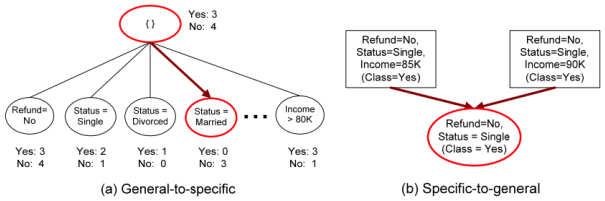
Sequential Covering (Direct Method)

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



Rule Growing

Two common strategies



Rule Evaluation

Foil's Information Gain

FOIL: First Order Inductive Learner - an early rule based learning algorithm

R0: $\{\} \rightarrow$ class (initial rule)

R1: $\{A\} \rightarrow$ class (rule after adding conjunct)

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

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

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 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 (contd..)

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 (contd..)

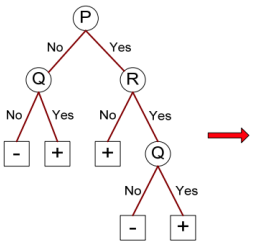
Building a Rule Set:

- 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

Optimize the rule set:

- For each rule r in the rule set R consider 2 alternative rules: (1) Replacement rule (r*): grow new rule from scratch, (2) 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
- Repeat rule generation and rule optimization for remaining +ve examples

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: C4.5 rules

Extract rules from an unpruned decision tree

- For each rule, $r : A \rightarrow y$
  1. consider an alternative rule $rz : A' \rightarrow y$ where A' is obtained by removing one of the conjuncts in A
  2. Compare the pessimistic error rate for r against all r's
  3. Prune if one of the alternative rules has lower pessimistic error rate
  4. Repeat until we can no longer improve generalization error

Thank You!

Thank you very much for your attention!

Queries ?