# CS 301 – Intro to Data Mining

## Algorithms for Constructing Rules from Coverings (continued)

- **PRISM** looked for a covering each time it needed to put together a part of a rule for one particular value of a decision attribute (i.e., class)

Ex: If ? then ***Recommended Lenses*** = hard

| | Correctness |
|---|---|
| ***Age*** = young | 2/8 |
| ***…*** | |
| ***Astigmatism*** = yes | 4/12 |
| ***…*** | |
| ***Tear Production Rate*** = normal | 4/12 |

Best is 4/12, so could pick either ***Astigmatism*** = yes or ***Tear Production Rate*** = normal

If ***Astigmatism*** = yes then ***Recommended Lenses*** = hard

Only correct for 4 of the 12 instances it covers (hardly "perfect"!), so try to add more conditions to the rule

If ***Astigmatism*** = yes and ? then ***Recommended Lenses*** = hard

| | Correctness |
|---|---|
| ***Astigmatism*** = yes and ***Age*** = young | 2/4 |
| ***Astigmatism*** = yes and ***Age*** = pre-presbyopic | 1/4 |
| ***Astigmatism*** = yes and ***Age*** = presbyopic | 1/4 |
| ***Astigmatism*** = yes and ***Spectacle Prescription*** = myope | 3/6 |
| ***…*** | |

Instead you can find **one** covering from which you can generate the rules for **all** values of the decision attribute

- **Finding all coverings for a decision attribute**

You might decide you like one covering better than another ☺

Let S be the set of all non-decision attributes

For every non-empty subset P of S compute P*

where P* is a partitioning of the instances of the dataset on some attributes such that if instances $x_i$ and $x_j$ are in the same block in P* it means that they are <u>indistinguishable</u> in their values for the attributes in P

<u>Ex</u>: Decision attribute is ***f***

| | a | b | c | d | f | color1 | color2 |
|---|---|---|---|---|---|---|---|
| **x1** | 0 | L | 0 | L | 0 | B | W |
| **x2** | 0 | R | 1 | L | 1 | B | W |
| **x3** | 0 | L | 0 | L | 0 | W | B |
| **x4** | 0 | R | 1 | L | 1 | W | B |
| **x5** | 1 | R | 0 | L | 2 | B | W |
| **x6** | 1 | R | 0 | L | 2 | W | B |
| **x7** | 2 | S | 2 | H | 3 | B | W |
| **x8** | 2 | S | 2 | H | 3 | W | B |

{a}* = $\{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}\}$

{b}* = $\{\{x_1, x_3\}, \{x_2, x_4, x_5, x_6\}, \{x_7, x_8\}\}$

{c}* = $\{\{x_1, x_3, x_5, x_6\}, \{x_2, x_4\}, \{x_7, x_8\}\}$

{a, c}* = {a, b}* = {b, c}* = {a, b, c, d}* = $\{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}\}$

{a, b, c, d, color1, color2}* = $\{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8\}\}$

…

Later we're also going to need the partition for the decision attribute

{f}* = **$\{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}\}$**

**Attribute dependency inequality**:

Let $P_1$, $P_2$ be partitions

Then $\mathbf{P_1 \le P_2}$ if for each block B of $P_1$ there exists a block B' of $P_2$ such that B is a proper subset of B'

Ex: $\{\{x_1, x_3\}, \{x_2, x_4, x_5, x_6\}, \{x_7, x_8\}\} \le \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}\}$ ***yes/no?***

$\{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}\} \le \{\{x_1, x_3\}, \{x_2, x_4, x_5, x_6\}, \{x_7, x_8\}\}$ ***yes/no?***

**Covering**:

Let S be the set of all non-decision attributes and let R be the set of decision attributes

P is a **covering** of R if:

(1) P is a subset of S,

(2) $P^* \le R^*$, and

(3) P is minimal.

Ex: Which of the partitions computed earlier qualifies as a **covering** for R = {f}?

$\{f\}^* = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}\}$ S = {a, b, c, d, color1, color2}

$\{a\}^* = \{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}\}$ ***yes/no?***

$\{b\}^* = \{\{x_1, x_3\}, \{x_2, x_4, x_5, x_6\}, \{x_7, x_8\}\}$ ***yes/no?***

$\{c\}^* = \{\{x_1, x_3, x_5, x_6\}, \{x_2, x_4\}, \{x_7, x_8\}\}$ ***yes/no?***

$\{a, c\}^* = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}\}$ ***yes/no?***

$\{a, b\}^* = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}\}$ ***yes/no?***

$\{b, c\}^* = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}\}$ ***yes/no?***

$\{a, b, c, d\}^* = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}, \{x_7, x_8\}\}$ ***yes/no?***

$\{a, b, c, d, color1, color2\}^* = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8\}\}$ ***yes/no?***

- **Finding (i.e., "inducing") rules from coverings - RICO**

Pick the covering you prefer from the ones you found

initialize E to the set of all instances;
initialize rule set to empty;
**while** E contains instances do
  create rule that uses attribute values of first instance in E for each attribute
    in covering;
  add rule to rule set;
  remove instances in E covered by this rule;
**end-while**

Ex: If we had picked covering {a, c} what rule set would we get?

| | a | b | c | d | f | color1 | color2 |
|---|---|---|---|---|---|---|---|
| **x1** | 0 | L | 0 | L | 0 | B | W |
| **x2** | 0 | R | 1 | L | 1 | B | W |
| **x3** | 0 | L | 0 | L | 0 | W | B |
| **x4** | 0 | R | 1 | L | 1 | W | B |
| **x5** | 1 | R | 0 | L | 2 | B | W |
| **x6** | 1 | R | 0 | L | 2 | W | B |
| **x7** | 2 | S | 2 | H | 3 | B | W |
| **x8** | 2 | S | 2 | H | 3 | W | B |

**If *a* = 0 and *c* = 0 then *f* = 0**

**If *a* = 0 and *c* = 1 then *f* = 1**

**If *a* = 1 and *c* = 0 then *f* = 2**

**If *a* = 2 and *c* = 2 then *f* = 3**

**Could any conditions be dropped in any of those rules?**

**If *a* = 0 and *c* = 0 then *f* = 0**

**If *a* = 0 and *c* = 1 then *f* = 1**

**If *a* = 1 ~~and *c* = 0~~ then *f* = 2**

**If ~~*a* = 2~~ and *c* = 2 then *f* = 3 OR If *a* = 2 ~~and *c* = 2~~ then *f* = 3**

- **Essential attribute:** attribute that appears in <u>every</u> covering; *what is the significance of this???*

- **Some advantages to this algorithm:** (1) will always find at least one covering and (2) will always generate rules that are 100% correct

<u>Ex</u>: Decision attribute is ***Play***

| | **Outlook** | **Temperature** | **Humidity** | **Windy** | **Play** |
|---|---|---|---|---|---|
| **x1** | sunny | hot | high | FALSE | no |
| **x2** | sunny | hot | high | TRUE | no |
| **x3** | overcast | hot | high | FALSE | yes |
| **x4** | rainy | mild | high | FALSE | yes |
| **x5** | rainy | cool | normal | FALSE | yes |
| **x6** | rainy | cool | normal | TRUE | no |
| **x7** | overcast | cool | normal | TRUE | yes |
| **x8** | sunny | mild | high | FALSE | no |
| **x9** | sunny | cool | normal | FALSE | yes |
| **x10** | rainy | mild | normal | FALSE | yes |
| **x11** | sunny | mild | normal | TRUE | yes |
| **x12** | overcast | mild | high | TRUE | yes |
| **x13** | overcast | hot | normal | FALSE | yes |
| **x14** | rainy | mild | high | TRUE | no |

Coverings are {***Outlook***, ***Temperature***, ***Windy***} and {***Outlook***, ***Humidity***, ***Windy***}

Rule set for covering {***Outlook***, ***Temperature***, ***Windy***}:

[[[sunny, hot, TRUE, no], 1], [[sunny, hot, FALSE, no], 1], [[sunny, mild, TRUE, yes], 1], [[sunny, mild, FALSE, no], 1], [[sunny, cool, FALSE, yes], 1], [[overcast, hot, FALSE, yes], 2], [[overcast, mild, TRUE, yes], 1], [[overcast, cool, TRUE, yes], 1], [[rainy, mild, TRUE, no], 1], [[rainy, mild, FALSE, yes], 2], [[rainy, cool, TRUE, no], 1], [[rainy, cool, FALSE, yes], 1]]

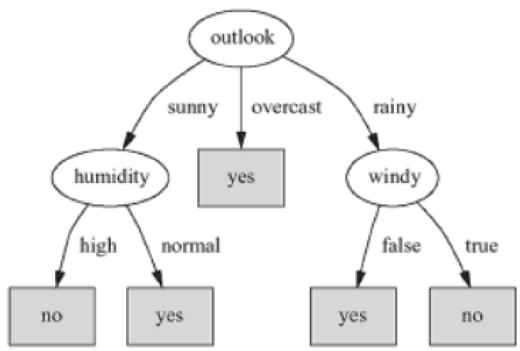
Here ***Outlook*** and ***Windy*** would be essential attributes

Without dropping conditions, we get 12 rules as compared to:

6 rules produced by **PRISM**

```
Prism rules
----------
If outlook = overcast then yes
If humidity = normal
   and windy = FALSE then yes
If temperature = mild
   and humidity = normal then yes
If outlook = rainy
   and windy = FALSE then yes
If outlook = sunny
   and humidity = high then no
If outlook = rainy
   and windy = TRUE then no
```

At most 2 decisions required in result produced by **ID3/J48**



Ex: Decision attribute is ***Recommended Lenses***

**Table 1.1** Contact Lens Data

| Age | Spectacle Prescription | Astigmatism | Tear Production Rate | Recommended Lenses |
|---|---|---|---|---|
| young | myope | no | reduced | none |
| young | myope | no | normal | soft |
| young | myope | yes | reduced | none |
| young | myope | yes | normal | hard |
| young | hypermetrope | no | reduced | none |
| young | hypermetrope | no | normal | soft |
| young | hypermetrope | yes | reduced | none |
| young | hypermetrope | yes | normal | hard |
| pre-presbyopic | myope | no | reduced | none |
| pre-presbyopic | myope | no | normal | soft |
| pre-presbyopic | myope | yes | reduced | none |
| pre-presbyopic | myope | yes | normal | hard |
| pre-presbyopic | hypermetrope | no | reduced | none |
| pre-presbyopic | hypermetrope | no | normal | soft |
| pre-presbyopic | hypermetrope | yes | reduced | none |
| pre-presbyopic | hypermetrope | yes | normal | none |
| presbyopic | myope | no | reduced | none |
| presbyopic | myope | no | normal | none |
| presbyopic | myope | yes | reduced | none |
| presbyopic | myope | yes | normal | hard |
| presbyopic | hypermetrope | no | reduced | none |
| presbyopic | hypermetrope | no | normal | soft |
| presbyopic | hypermetrope | yes | reduced | none |
| presbyopic | hypermetrope | yes | normal | none |

**Only covering we would find for this dataset is the entire set of attributes, with a rule for each instance**

**A downside to using this particular approach ☹**