جامعة الطائف
TAIF UNIVERSITY

DATA MINING

Assoc. Prof. Dr. Salha Alzahrani

College of Computers and Information Technology
Taif University
Saudi Arabia

s.zahrani@tu.edu.sa

Classification | Classification using Decision Trees

Recap of Previous Lecture

Content of This Lecture

Summary & Checklist

Recap of Lecture 5

- Trees
- Decision Rules and Decision Trees
- Decision Trees: The Golf Example
- How to construct a Decision Tree?
- Functions of Decision Trees
- Decision Trees: The degrees Dataset
- The TDIDT Algorithm
- The TDIDT Algorithm: Adequacy Condition

Recap of Previous Lecture

Content of This Lecture

Summary & Checklist

Content of Lecture 6

- Rule Post-pruning
- Exercise: Rule Post-pruning
- Inducing Modular Rules for Classification: The Prism Algorithm
- The Prism Algorithm: The lens24 Example
- Handout (5): Using the Prism Algorithm for Rules Induction from the lens24 dataset.
- Summary & Checklist.

Classification | Rule Post-pruning

- Generating classification rules via the intermediate form of a decision tree is a widely used technique.

- The Rule Post-pruning method begins by converting a decision tree to an equivalent set of rules and then examines the rules with the aim of **simplifying** them without any loss of **predictive accuracy**.

- Each branch of the tree corresponds to a classification rule and so the rules equivalent to the decision tree can be extracted from it branch by branch.

Classification | Rule Post-pruning

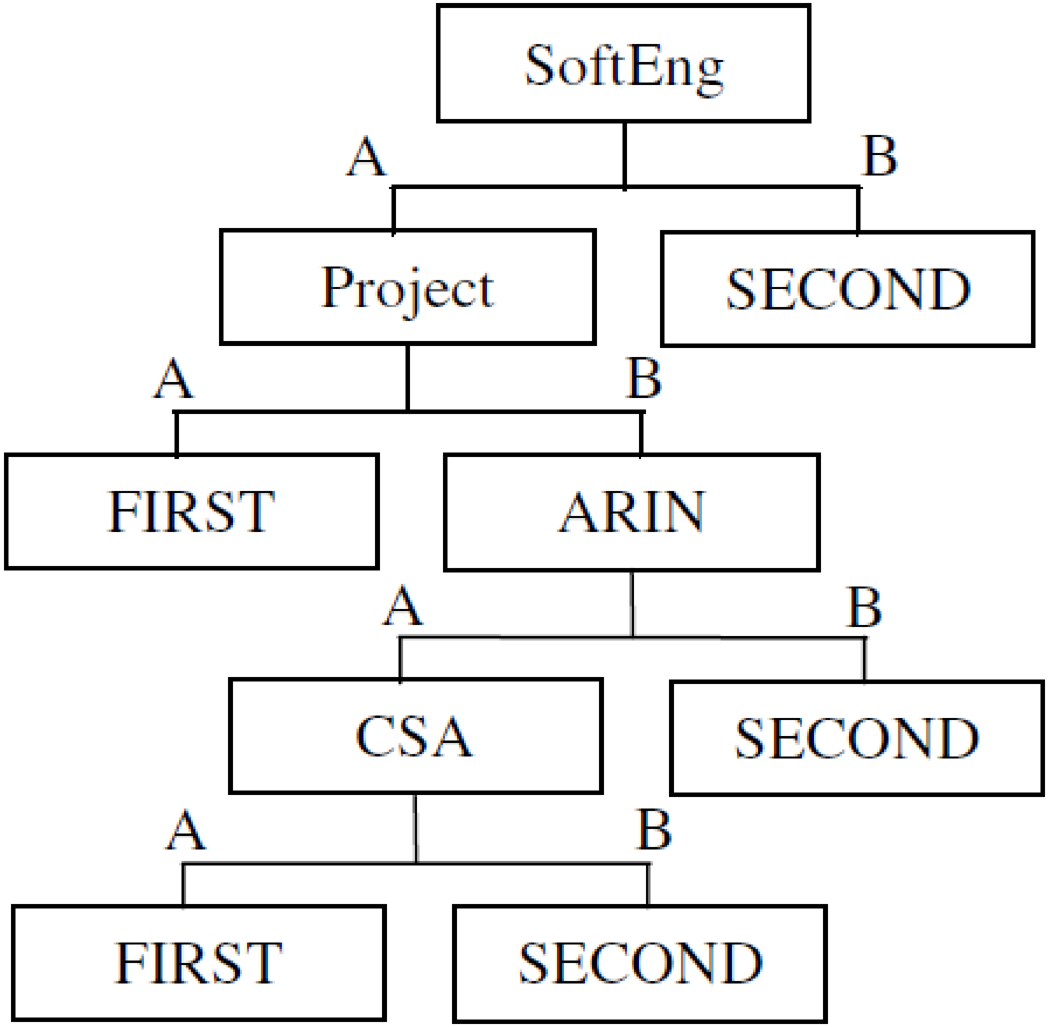

Figure 10.1 Decision Tree for the *degrees* Dataset

Classification | Rule Post-pruning

IF SoftEng = A AND Project = B AND
    ARIN = A AND CSA = A THEN Class = FIRST
IF SoftEng = A AND Project = A THEN Class = FIRST
IF SoftEng = A AND Project = B AND ARIN = A AND
    CSA = B THEN Class = SECOND
IF SoftEng = A AND Project = B AND ARIN = B THEN
    Class = SECOND
IF SoftEng = B THEN Class = SECOND

Classification | Rule Post-pruning

- We examine each of the rules to consider whether removing each of its terms **increases** or **reduces** its **predictive accuracy**.

- Thus, for the first rule

$$\boxed{\begin{array}{l}\text{IF SoftEng} = \text{A AND Project} = \text{B AND} \\ \quad \text{ARIN} = \text{A AND CSA} = \text{A THEN Class} = \text{FIRST}\end{array}}$$

- we consider the **four terms**
  - 'SoftEng = A', 'Project = B', 'ARIN= A' and 'CSA = A'.

- We need some way of estimating whether removing each of these terms singly would increase or decrease the accuracy of the resulting rule set.

Classification | Rule Post-pruning

Step 1: Remove the term that gives the **largest increase** in the predictive accuracy, say 'Project = B'.

IF SoftEng = A AND [redacted] ARIN = A AND CSA = A THEN Class = FIRST

OK

Step 2: Consider the removal of each of the other terms.

IF SoftEng = A AND [redacted] CSA = A THEN Class = FIRST

NOT OK

IF SoftEng = A AND [redacted] ARIN = A [redacted] THEN Class = FIRST

NOT OK

Step 3: The processing of a rule ends when removing any of the terms would reduce (or leave unchanged) the predictive accuracy.

Classification | Exercise: Rule Post-pruning

Exercise: Perform the rule post-pruning on the following classification rules:

IF SoftEng = A AND Project = B AND ARIN = A AND CSA = B THEN Class = SECOND

IF SoftEng = A AND Project = B AND ARIN = B THEN Class = SECOND

Classification | Inducing Modular Rules for Classification: The Prism Algorithm

- The aim is to **induce modular classification rules directly from the training set**. The algorithm assumes that all the attributes are **categorical**.

- Alternatively, the algorithm can be extended to deal with **continuous** attributes in much the same way as was described for TDIDT (define a split value, use $< and \ \geq$)

- The algorithm generates the rules concluding each of the possible classes in turn. Each rule is generated **term by term**, with each term of the form ‘attribute = value’.
- The attribute/value pair added at each step is chosen to maximise the **probability** of the target ‘outcome class’.

Classification | Inducing Modular Rules for Classification: The Prism Algorithm

For each classification (class = i) in turn and starting with the complete training set each time:

1. Calculate the probability that class = i for each attribute/value pair.
2. Select the pair with the largest probability and create a subset of the training set comprising all the instances with the selected attribute/value combination (for all classifications).
3. Repeat 1 and 2 for this subset until a subset is reached that contains only instances of class i. The induced rule is then the conjunction of all the attribute/value pairs selected.
4. Remove all instances covered by this rule from the training set.

Repeat 1–4 until all instances of class i have been removed

Figure 10.5 The Basic Prism Algorithm

Classification | The Prism Algorithm: The *lens24* Example

- **Show how to generate the classification rules from this dataset using the Prism Algorithm?**

age	specRx	astig	tears	class
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

Figure 10.6 The *lens24* Training Set

Classification | The Prism Algorithm: The *lens24* Example

First Rule

- The probability of **class = 1** occurring for each attribute/value pair over the whole training set (**24 instances**).
- The maximum probability is when ***astig* = 2** or ***tears* = 2**.
- Choose ***astig* = 2 arbitrarily**.

Attribute/value pair	Frequency for class = 1	Total frequency (out of 24 instances)	Probability
age = 1	2	8	0.25
age = 2	1	8	0.125
age = 3	1	8	0.125
specRx = 1	3	12	0.25
specRx = 2	1	12	0.083
astig = 1	0	12	0
astig = 2	4	12	0.33
tears = 1	0	12	0
tears = 2	4	12	0.33

Figure 10.7 First Rule: Probability of Attribute/value Pairs (Version 1)

- Incomplete rule induced so far:

IF astig = 2 THEN class = 1

Classification | Inducing Modular Rules for Classification

age	specRx	astig	tears	class
1	1	2	1	3
1	1	2	2	1
1	2	2	1	3
1	2	2	2	1
2	1	2	1	3
2	1	2	2	1
2	2	2	1	3
2	2	2	2	3
3	1	2	1	3
3	1	2	2	1
3	2	2	1	3
3	2	2	2	3

Figure 10.8 First Rule: Subset of Training Set Covered by Incomplete Rule (Version 1)

First Rule

- The probability of each attribute/value pair (**not involving attribute *astig***) occurring for this subset.
- The maximum probability is when ***tears* = 2**.
- Incomplete rule induced so far:

Attribute/value pair	Frequency for class = 1	Total frequency (out of 12 instances)	Probability
age = 1	2	4	0.5
age = 2	1	4	0.25
age = 3	1	4	0.25
specRx = 1	3	6	0.5
specRx = 2	1	6	0.17
tears = 1	0	6	0
tears = 2	4	6	0.67

Figure 10.9 First Rule: Probability of Attribute/value Pairs (Version 2)

IF astig = 2 and tears = 2 THEN class = 1

Classification | Inducing Modular Rules for Classification

age	specRx	astig	tears	class
1	1	2	2	1
1	2	2	2	1
2	1	2	2	1
2	2	2	2	3
3	1	2	2	1
3	2	2	2	3

Figure 10.10 First Rule: Subset of Training Set Covered by Incomplete Rule (Version 2)

First Rule

- The probability of each attribute/value pair (**not involving attributes *astig* or *tears***) occurring for this subset.
- The maximum probability is when ***age*** **= 1** or ***specRx*** **= 1**.
- Choose (arbitrarily) ***age*** **= 1**.
- Incomplete rule induced so far:

Attribute/value pair	Frequency for class = 1	Total frequency (out of 6 instances)	Probability
age = 1	2	2	1.0
age = 2	1	2	0.5
age = 3	1	2	0.5
specRx = 1	3	3	1.0
specRx = 2	1	3	0.33

Figure 10.11 First Rule: Probability of Attribute/value Pairs (Version 3)

IF astig = 2 and tears = 2 and age = 1 THEN class = 1

age	specRx	astig	tears	class
1	1	2	2	1
1	2	2	2	1

Figure 10.12 First Rule: Subset of Training Set Covered by Incomplete Rule (Version 3)

- **This subset contains only instances of class 1.**
- **The final induced rule is therefore:**

First Rule

IF astig = 2 and tears = 2 and age = 1 THEN class = 1

Classification | Inducing Modular Rules for Classification

- Removing the **two instances** covered by the first rule from the training set gives a new training set with 22 instances.

age	specRx	astig	tears	class
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

Figure 10.13 The *lens24* Training Set (Reduced)

Second Rule

- The table of frequencies is now as given in for attribute/value pairs corresponding to ***class* = 1**.
- The maximum probability is achieved by ***astig* = 2** and ***tears* = 2**.
- Choose ***astig* = 2** arbitrarily.
- Incomplete rule induced so far:

Attribute/value pair	Frequency for class = 1	Total frequency (out of 22 instances)	Probability
age = 1	0	6	0
age = 2	1	8	0.125
age = 3	1	8	0.125
specRx = 1	2	11	0.18
specRx = 2	0	11	0
astig = 1	0	12	0
astig = 2	2	10	0.2
tears = 1	0	12	0
tears = 2	2	10	0.2

Figure 10.14 Second Rule: Probability of Attribute/value Pairs (Version 1)

IF astig=2 THEN class = 1

Classification | Inducing Modular Rules for Classification

age	specRx	astig	tears	class
1	1	2	1	3
1	2	2	1	3
2	1	2	1	3
2	1	2	2	1
2	2	2	1	3
2	2	2	2	3
3	1	2	1	3
3	1	2	2	1
3	2	2	1	3
3	2	2	2	3

Figure 10.15 Second Rule: Subset of Training Set Covered by Incomplete Rule (Version 1)

Second Rule

- The maximum probability is for ***tears*= 2**
- Incomplete rule induced so far:

Attribute/value pair	Frequency for class = 1	Total frequency (out of 10 instances)	Probability
age = 1	0	2	0
age = 2	1	4	0.25
age = 3	1	4	0.25
specRx = 1	0	5	0
specRx = 2	2	5	0.4
tears = 1	0	6	0
tears = 2	2	4	0.5

Figure 10.16 Second Rule: Probability of Attribute/value Pairs (Version 2)

IF astig = 2 and tears = 2 then class = 1

Classification | Inducing Modular Rules for Classification

age	specRx	astig	tears	class
2	1	2	2	1
2	2	2	2	3
3	1	2	2	1
3	2	2	2	3

Figure 10.17 Second Rule: Subset of Training Set Covered by Incomplete Rule (Version 2)

Second Rule

- The maximum probability is achieved by **specRx = 1**.
- Incomplete rule induced so far:

Attribute/value pair	Frequency for class = 1	Total Frequency (out of 4 instances)	Probability
age = 1	0	0	–
age = 2	1	2	0.5
age = 3	1	2	0.5
specRx = 1	2	2	1.0
specRx = 2	0	2	0

Figure 10.18 Second Rule: Probability of Attribute/value Pairs (Version 3)

IF astig = 2 and tears = 2 and specRx = 1 THEN class = 1

Classification | Inducing Modular Rules for Classification

age	specRx	astig	tears	class
2	1	2	2	1
3	1	2	2	1

Figure 10.19 Second Rule: Subset of Training Set Covered by Incomplete Rule (Version 3)

- This subset contains only instances of class 1. So the final induced rule is:

Second Rule

IF astig = 2 and tears = 2 and specRx = 1 THEN class = 1

- Removing the two instances covered by this rule from the current version of the training set (which has 22 instances) gives a training set of **20 instances** from which all instances of class 1 have now been removed. **So the Prism algorithm terminates (for class= 1).**

- The final pair of rules induced by Prism for class 1 are:
  - IF astig = 2 and tears = 2 and age = 1 THEN class = 1
  - IF astig = 2 and tears = 2 and specRx = 1 THEN class = 1

- The algorithm will now go on to generate rules for the remaining classifications.
- It produces 3 rules for class 2 and 4 for class 3.

Classification | Handout (5): Using the Prism Algorithm for Rules Induction from the *lens24* dataset

Summary & Checklist

- ☑ Rule Post-pruning
- ☑ Exercise: Rule Post-pruning
- ☑ Inducing Modular Rules for Classification: The Prism Algorithm
- ☑ The Prism Algorithm: The lens24 Example
- ☑ Handout (5): Using the Prism Algorithm for Rules Induction from the lens24 dataset.

Reminder | Next Lecture !

Next Lecture…

Association Rule Mining (Part I)

- *Be ready!*
- *Download & print the lecture notes before your class.*

Thank You !

✉ s.zahrani@tu.edu.sa