

---

# COMP9318 (2009 s1) Tutorial 3

Wei WANG

The University of New South Wales

[weiw@cse.unsw.edu.au](mailto:weiw@cse.unsw.edu.au)

---

## ① Classification

---

## Q1

Consider the following training dataset and the original decision tree induction algorithm (ID3).

*Risk* is the class label attribute. The *Height* values have been already discretized into disjoint ranges.

- ① Calculate the information gain if *Gender* is chosen as the test attribute.
- ② Calculate the information gain if *Height* is chosen as the test attribute.
- ③ Draw the final decision tree (without any pruning) for the training dataset.
- ④ Generate all the “IF-THEN” rules from the decision tree.

---

<i>Gender</i>	<i>Height</i>	<i>Risk</i>
F	(1.5, 1.6]	Low
M	(1.9, 2.0]	High
F	(1.8, 1.9]	Medium
F	(1.8, 1.9]	Medium
F	(1.6, 1.7]	Low
M	(1.8, 1.9]	Medium
F	(1.5, 1.6]	Low
M	(1.6, 1.7]	Low
M	(2.0, $\infty$ ]	High
M	(2.0, $\infty$ ]	High
F	(1.7, 1.8]	Medium
M	(1.9, 2.0]	Medium
F	(1.8, 1.9]	Medium
F	(1.7, 1.8]	Medium
F	(1.7, 1.8]	Medium

---

---

## SOLUTION TO Q1

① The original entropy is

$$I_{Risk} = I(Low, Medium, High) = I(4, 8, 3) = 1.4566.$$

Consider *Gender*.

<i>Gender</i>	entropy
<i>F</i>	$I(3, 6, 0)$
<i>M</i>	$I(1, 2, 3)$

The expected entropy is  $\frac{9}{15} \cdot I(3, 6, 0) + \frac{6}{15} \cdot I(1, 2, 3) = 1.1346$ . The information gain is  $1.4566 - 1.1346 = 0.3220$

② Consider *Height*.

---

<i>Height</i>	entropy
(1.5, 1.6]	$I(2, 0, 0)$
(1.6, 1.7]	$I(2, 0, 0)$
(1.7, 1.8]	$I(0, 3, 0)$
(1.8, 1.9]	$I(0, 4, 0)$
(1.9, 2.0]	$I(0, 1, 1)$
(2.0, $\infty$ ]	$I(0, 0, 2)$

The expected entropy is  $\frac{2}{15} \cdot I(2, 0, 0) + \frac{2}{15} \cdot I(2, 0, 0) + \frac{3}{15} \cdot I(0, 3, 0) + \frac{4}{15} \cdot I(0, 4, 0) + \frac{2}{15} \cdot I(0, 1, 1) + \frac{2}{15} \cdot I(0, 0, 2) = 0.1333$ . The information gain is  $1.4566 - 0.1333 = 1.3233$

③ ID3 decision tree:

→ According to the computation above, we should first choose *Height* to split

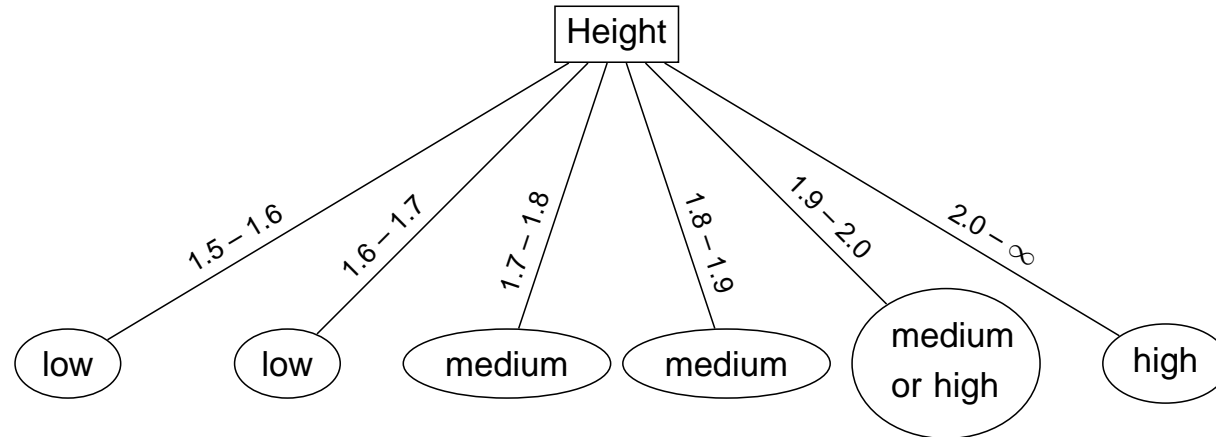
→ After split, the only problematic partition is the (1.9, 2.0] one.

However, the only remaining attribute *Gender* cannot divide them.

---

As there is a draw, we can take any label.

→ The final tree is show in the figure below.



④ The rules are

- **IF**  $height \in (1.5, 1.6]$ , **THEN**  $Rish = \text{Low}$ .
- **IF**  $height \in (1.6, 1.7]$ , **THEN**  $Rish = \text{Low}$ .
- **IF**  $height \in (1.7, 1.8]$ , **THEN**  $Rish = \text{Medium}$ .
- **IF**  $height \in (1.8, 1.9]$ , **THEN**  $Rish = \text{Medium}$ .
- **IF**  $height \in (1.9, 2.0]$ , **THEN**  $Rish = \text{Medium (or High)}$ .
- **IF**  $height \in (2.0, \infty]$ , **THEN**  $Rish = \text{High}$ .

---

## Q2

Consider applying the SPRINT algorithm on the following training dataset

<i>Age</i>	<i>CarType</i>	<i>Risk</i>
23	family	High
17	sports	High
43	sports	High
68	family	Low
32	truck	Low
20	family	High

Answer the following questions:

- ① Write down the attribute lists for attribute *Age* and *CarType*, respectively.

- 
- ② Assume the first split criterion is  $Age < 27.5$ . Write down the attribute lists for the left child node (i.e., corresponding to the partition whose  $Age < 27.5$ ).
  - ③ Assume that the two attribute lists for the root node are stored in relational tables name  $AL\_Age$  and  $AL\_CarType$ , respectively. We can in fact generate the attribute lists for the child nodes using standard SQL statements. Write down the SQL statements which will generate the attribute lists for the left child node for the split criterion  $Age < 27.5$ .
  - ④ Write down the final decision tree constructed by the SPRINT algorithm.

---

## SOLUTION TO Q2

→ Attribute list of *Age* is:

<i>Age</i>	class	Index
17	High	2
20	High	6
23	High	1
32	Low	5
43	High	3
68	Low	4

Attribute list of *CarType* is:

---

<i>CarType</i>	class	Index
family	High	1
sports	High	2
sports	High	3
family	Low	4
truck	Low	5
family	High	6

→ Attribute list of *Age* is:

<i>Age</i>	class	Index
17	High	2
20	High	6
23	High	1

Attribute list of *CarType* is:

---

<i>CarType</i>	class	Index
family	High	1
sports	High	2
family	High	6

→ SQL for the attribute list of *Age*:

```
SELECT Age, Class, Index
FROM AL_Age
WHERE Age < 27.5
```

SQL for the attribute list of *CarType*:

```
SELECT C.CarType, C.Class, C.Index
FROM AL_Age A, AL_CarType C
WHERE A.Age < 27.5
      AND A.index = C.index
```

→ Consider the attribute list of *Age*: there are 5 possible “cut” positions, each of them have gini index value as:

---

<i>Age</i>	above	below	$gini_{split}$
17 – 20	(1, 0)	(3, 2)	0.40
20 – 23	(2, 0)	(2, 2)	0.33
23 – 32	(3, 0)	(1, 2)	0.22
32 – 43	(3, 1)	(1, 1)	0.42
43 – 68	(4, 1)	(0, 1)	0.27

therefore, the best split should be  $Age > 27.5$ .

Consider the attribute list of *CarType*:

<i>CarType</i>	High	Low
f	2	1
s	2	0
t	0	1

Consider all the possible cuts:

---

<i>CarType</i>	High	Low
f	2	1
s, t	2	1

<i>CarType</i>	High	Low
s	2	0
f, t	2	2

<i>CarType</i>	High	Low
t	0	1
f, s	4	1

Each of them have gini index value as: 0.44, 0.33, 0.27, respectively. Therefore, the best split is *CarType* in ('truck').

Obviously, splitting on *Age* is better. Therefore, we shall split by  $Age > 27.5$ .

The attribute lists for each of the child node have already been computed. Since the tuples in the partition for  $Age < 27.5$  are all "high", we only need to look at the partition for  $Age \geq 27.5$ .

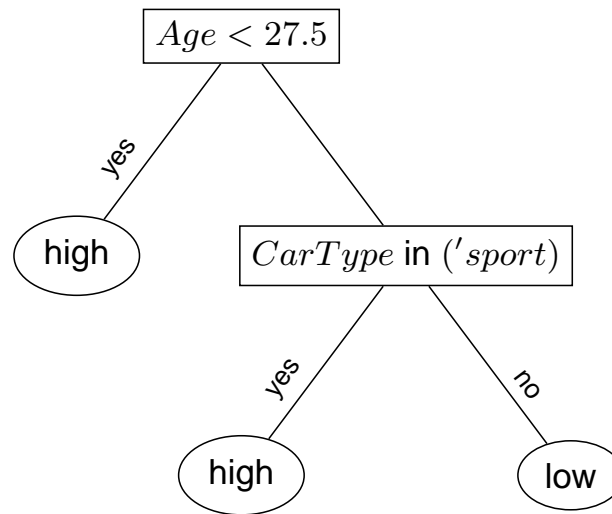
---

<i>Age</i>	class	Index
32	Low	5
43	High	3
68	Low	4

<i>CarType</i>	class	Index
sports	High	3
family	Low	4
truck	Low	5

It is obvious that *CarType* in ('*sports*') can immediately cut this partition into two “pure” partitions and thus will have 0 as the gini index value. So we can skip a lot of calculations.

The final tree is:



---

## Q3

Consider a (simplified) email classification example. Assume the training dataset contains 1000 emails in total, 100 of which are spams.

- ① Calculate the class prior probability distribution. How would you classify a new incoming email?
- ② A friend of you suggests that whether the email contains a \$ char is a good feature to detect spam emails. You look into the training dataset and obtain the following statistics (\$ means emails containing a \$ and  $\bar{\$}$  are those not containing any \$).

<b>Class</b>	\$	$\bar{\$}$
SPAM	91	9
NOSPAM	63	837

Describe the (naive) Bayes Classifier you can build on this new piece of “evidence”. How would this classifier predict the class label for a

---

new incoming email that contains a \$ character?

- ③ Another friend of you suggest looking into the feature of whether the email's length is longer than a fixed threshold (e.g., 500 bytes). You obtain the following results (this feature denoted as  $L$  ( $\bar{L}$ )).

<b>Class</b>	$L$	$\bar{L}$
SPAM	40	60
NOSPAM	400	500

How would a naive Bayes classifier predict the class label for a new incoming email that contains a \$ character and is shorter than the threshold?

---

## SOLUTION TO Q3

① The prior probabilities are:

$$P(\text{SPAM}) = \frac{100}{1000} = 0.10$$

$$P(\text{NOSPAM}) = \frac{1000 - 100}{1000} = 0.90$$

② In order to build a (naive) bayes classifier, we need to calculate (and store) the likelihood of the feature for each class.

---

$$P(\$ | \text{SPAM}) = \frac{91}{100} = 0.91$$

$$P(\$ | \text{NOSPAM}) = \frac{63}{900} = 0.07$$

---

To classify the new object, we calculate the posterior probability for

---

both classes as:

$$\begin{aligned} P(\text{SPAM} | X) &= \frac{1}{P(X)} \cdot P(X | \text{SPAM}) \cdot P(\text{SPAM}) \\ &= \frac{1}{P(X)} \cdot 0.91 \cdot 0.10 = \frac{1}{P(X)} \cdot 0.091 \end{aligned}$$

$$\begin{aligned} P(\text{NOSPAM} | X) &= \frac{1}{P(X)} \cdot P(X | \text{NOSPAM}) \cdot P(\text{NOSPAM}) \\ &= \frac{1}{P(X)} \cdot 0.07 \cdot 0.90 = \frac{1}{P(X)} \cdot 0.063 \end{aligned}$$

So the prediction will be SPAM.

③ The likelihood of the new feature for each class is:

---

$$\begin{aligned} P(L | \text{SPAM}) &= \frac{40}{100} = 0.40 \\ P(L | \text{NOSPAM}) &= \frac{400}{900} = 0.44 \end{aligned}$$

---

To classify the new object, we calculate the posterior probability for

---

both classes as:

$$\begin{aligned}P(\text{SPAM} | X) &= \frac{1}{P(X)} \cdot P(X | \text{SPAM}) \cdot P(\text{SPAM}) \\ &= \frac{1}{P(X)} \cdot 0.60 \cdot 0.91 \cdot 0.10 = \frac{1}{P(X)} \cdot 0.546\end{aligned}$$

$$\begin{aligned}P(\text{NOSPAM} | X) &= \frac{1}{P(X)} \cdot P(X | \text{NOSPAM}) \cdot P(\text{NOSPAM}) \\ &= \frac{1}{P(X)} \cdot 0.56 \cdot 0.07 \cdot 0.90 = \frac{1}{P(X)} \cdot 0.028\end{aligned}$$

So the prediction will be SPAM.