COMP9318 Tutorial 2: Classification

Wei Wang @ UNSW

Q1 I

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.

Gender	*Height*	*Risk*
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, ∞]	High
M	(2.0, ∞]	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 I

1. The original entropy is $I_{Risk} = I(Low, Medium, High) = I(4,8,3) = 1.4566$. Consider *Gender*.

Gender	entropy
F	$I(3,6,0)$
M	$I(1,2,3)$

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 $1.4566 - 1.1346 = 0.3220$

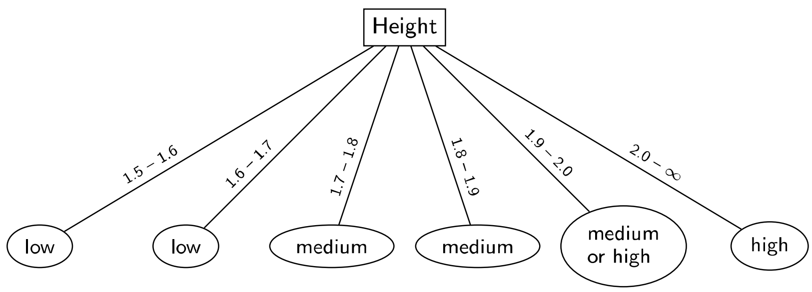
2. Consider *Height*.

Height	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 $1.4566 - 0.1333 = 1.3233$

Solution to Q1 II

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

Q2 I

Consider applying the SPRINT algorithm on the following training dataset

Age	*CarType*	*Risk*
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 I

- Attribute list of *Age* is:

Age	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:

CarType	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:

Age	class	Index
17	High	2
20	High	6
23	High	1

Solution to Q2 II

Attribute list of *CarType* is:

CarType	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:

Age	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

Solution to Q2 III

therefore, the best split should be $Age > 27.5$.
Consider the attribute list of *CarType*:

CarType	High	Low
f	2	1
s	2	0
t	0	1

Consider all the possible cuts:

CarType	High	Low
f	2	1
s, t	2	1

CarType	High	Low
s	2	0
f, t	2	2

CarType	High	Low
t	0	1
f, s	4	1

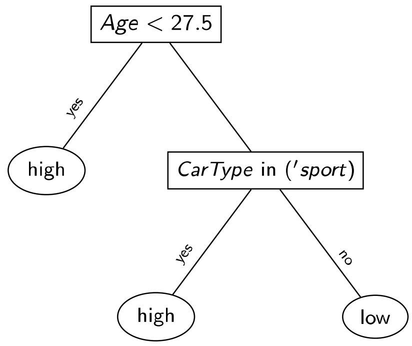
Each of them have gini index value as: 0.44, 0.33, 0.27, respectively.
Therefore, the best split is *CarType* in $('truck')$.
Obviously, splitting on *Age* is better. Therefore, we shall split by $Age > 27.5$.
The attribute lists for each of the child node have already been computed.
Since the tuples in the partition for $Age < 27.5$ are all "high", we only need to look at the partition for $Age \geq 27.5$.

Solution to Q2 IV

Age	class	Index
32	Low	5
43	High	3
68	Low	4

CarType	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 I

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 $\overline{\$}$ are those not containing any \$).

Class	\$	$\overline{\$}$
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 ($\bar{L}$)).

Q3 II

Class	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 I

1. The prior probabilities are:

$$P(\texttt{SPAM}) = \frac{100}{1000} = 0.10$$
$$P(\texttt{NOSPAM}) = \frac{1000 - 100}{1000} = 0.90$$

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

$P(\$ \mid \texttt{SPAM})$	$\frac{91}{100} = 0.91$
$P(\$ \mid \texttt{NOSPAM})$	$\frac{63}{900} = 0.07$

Solution to Q3 II

To classify the new object, we calculate the posterior probability for both classes as:

$$
\begin{aligned}
P(\texttt{SPAM} \mid X) &= \frac{1}{P(X)} \cdot P(X \mid \texttt{SPAM}) \cdot P(\texttt{SPAM}) \\
&= \frac{1}{P(X)} \cdot P(\$ \mid \texttt{SPAM}) \cdot P(\texttt{SPAM}) \\
&= \frac{1}{P(X)} \cdot 0.91 \cdot 0.10 = \frac{1}{P(X)} \cdot 0.091 \\
P(\texttt{NOSPAM} \mid X) &= \frac{1}{P(X)} \cdot P(X \mid \texttt{NOSPAM}) \cdot P(\texttt{NOSPAM}) \\
&= \frac{1}{P(X)} \cdot P(\$ \mid \texttt{NOSPAM}) \cdot P(\texttt{NOSPAM}) \\
&= \frac{1}{P(X)} \cdot 0.07 \cdot 0.90 = \frac{1}{P(X)} \cdot 0.063
\end{aligned}
$$

So the prediction will be SPAM.

3. The likelyhood of the new feature for each class is:

$P(L \mid \texttt{SPAM})$	$\frac{40}{100} = 0.40$
$P(L \mid \texttt{NOSPAM})$	$\frac{400}{900} = 0.44$

Solution to Q3 III

(Note: we can easily obtain probabilities, e.g.,
$P(\bar{L} \mid \texttt{SPAM}) = 1 - P(\bar{L} \mid \texttt{SPAM}) = 0.60$)
To classify the new object, we calculate the posterior probability for both classes as:

$$
\begin{aligned}
P(\texttt{SPAM} \mid X) &= \frac{1}{P(X)} \cdot P(X \mid \texttt{SPAM}) \cdot P(\texttt{SPAM}) \\
&= \frac{1}{P(X)} \cdot P(\$, \bar{L} \mid \texttt{SPAM}) \cdot P(\texttt{SPAM}) \\
&= \frac{1}{P(X)} \cdot P(\$ \mid \texttt{SPAM}) \cdot P(\bar{L} \mid \texttt{SPAM}) \cdot P(\texttt{SPAM}) \\
&= \frac{1}{P(X)} \cdot 0.60 \cdot 0.91 \cdot 0.10 = \frac{1}{P(X)} \cdot 0.055 \\
P(\texttt{NOSPAM} \mid X) &= \frac{1}{P(X)} \cdot P(X \mid \texttt{NOSPAM}) \cdot P(\texttt{NOSPAM}) \\
&= \frac{1}{P(X)} \cdot P(\$, \bar{L} \mid \texttt{NOSPAM}) \cdot P(\texttt{NOSPAM}) \\
&= \frac{1}{P(X)} \cdot P(\$ \mid \texttt{NOSPAM}) \cdot P(\bar{L} \mid \texttt{NOSPAM}) \cdot P(\texttt{NOSPAM}) \\
&= \frac{1}{P(X)} \cdot 0.56 \cdot 0.07 \cdot 0.90 = \frac{1}{P(X)} \cdot 0.035
\end{aligned}
$$

Solution to Q3 IV

So the prediction will be `SPAM`.

Q4 I

Based on the data in the following table,

1. estimate a Bernoulli Naive Bayes classifer (using the add-one smoothing)
2. apply the classifier to the test document.
3. estimate a multinomial Naive Bayes classifier (using the add-one smoothing)
4. apply the classifier to the test document

You do not need to estimate parameters that you don't need for classifying the test document.

	docID	**words in document**	**class = `China`?**
training set	1	Taipei Taiwan	Yes
	2	Macao Taiwan Shanghai	Yes
	3	Japan Sapporo	No
	4	Sapporo Osaka Taiwan	No
test set	5	Taiwan Taiwan Taiwan Sapporo Bangkok	?

Solution to Q3 I

We use the following abbreviations to denote the words, i.e., TP = Taipei, TW = Taiwan, MC = Macao, SH = Shanghai, JP = Japan, SP = Sapporo, OS = Osaka. The size of the vocabulary is 7.

1. (**Bernoulli NB**) We take each word in the vocabulary as a feature/attribute, and hence can obtain the following "rational" training set.

docID	**TP**	**TW**	**MC**	**SH**	**JP**	**SP**	**OS**	**class**
1	1	1	0	0	0	0	0	Y
2	0	1	1	1	0	0	0	Y
3	0	0	0	0	1	1	0	N
4	0	1	0	0	0	1	1	N

The testing document is (ignoring the unknown token `Bangkok`):

docID	**TP**	**TW**	**MC**	**SH**	**JP**	**SP**	**OS**	**class**
5	0	1	0	0	0	1	0	?

Solution to Q3 II

By looking at the test data, we calculate the *necessary* probabilities for the 'Y' class as (note that there are 2 possible values for each variable)

$$P(Y) = \frac{2}{4}$$
$$P(TP = 0|Y) = \frac{1+1}{2+2}$$
$$P(TW = 1|Y) = \frac{2+1}{2+2}$$
$$P(MC = 0|Y) = \frac{1+1}{2+2}$$
$$P(SH = 0|Y) = \frac{1+1}{2+2}$$
$$P(JP = 0|Y) = \frac{2+1}{2+2}$$
$$P(SP = 1|Y) = \frac{0+1}{2+2}$$
$$P(OS = 0|Y) = \frac{2+1}{2+2}$$

Solution to Q3 III

Finally,

$$\begin{aligned} P(Y|X) &\propto P(Y) \cdot P(TP = 0|Y) \cdot P(TW = 1|Y) \cdot P(MC = 0|Y) \cdot P(SH = 0|Y) \\ &\quad \cdot P(JP = 0|Y) \cdot P(SP = 1|Y) \cdot P(OS = 0|Y) \\ &= \frac{1}{2}\frac{1}{2}\frac{3}{4}\frac{1}{2}\frac{1}{2}\frac{3}{4}\frac{1}{4}\frac{3}{4} = \frac{27}{4096} \approx 0.0066 \end{aligned}$$

Solution to Q3 IV

We calculate the *necessary* probabilities for the 'N' class as

$$\begin{aligned} P(N) &= \frac{2}{4} \\ P(TP = 0|N) &= \frac{2+1}{2+2} \\ P(TW = 1|N) &= \frac{1+1}{2+2} \\ P(MC = 0|N) &= \frac{2+1}{2+2} \\ P(SH = 0|N) &= \frac{2+1}{2+2} \\ P(JP = 0|N) &= \frac{1+1}{2+2} \\ P(SP = 1|N) &= \frac{2+1}{2+2} \\ P(OS = 0|N) &= \frac{1+1}{2+2} \end{aligned}$$

Solution to Q3 V

Finally,

$$\begin{aligned}P(N|X) \propto\ & P(N) \cdot P(TP=0|N) \cdot P(TW=1|N) \cdot P(MC=0|N) \cdot P(SH=0|N) \\ & \cdot P(JP=0|N) \cdot P(SP=1|N) \cdot P(OS=0|N) \\ =\ & \frac{1}{2}\frac{3}{4}\frac{1}{2}\frac{3}{4}\frac{3}{4}\frac{1}{2}\frac{3}{4}\frac{1}{2} = \frac{81}{4096} \approx 0.020\end{aligned}$$

Therefore, doc 5 should belong to the 'No' class.

2. (**Multinomial NB**) We form the mega-documents for each class as:

Doc	**class**
TP TW MC TW SH	Y
JP SP SP OS TW	N

The testing document is (ignoring the out-of-vocabulary (OOV) words `Bangkok`):

Doc	**class**
TW TW TW SP	?

Solution to Q3 VI

By looking at the test data, we calculate the *necessary* probabilities for the 'Y' class as (note that there are 7 possible values for the variable w_i)

$$\begin{aligned}P(Y) &= \frac{2}{4} \\ P(w_i = TW|Y) &= \frac{2+1}{5+7} \\ P(w_i = SP|Y) &= \frac{0+1}{5+7}\end{aligned}$$

Finally,

$$\begin{aligned}P(Y|X) \propto & P(Y) \cdot P(w_i = TW|Y) \cdot P(w_i = TW|Y) \\ & \cdot P(w_i = TW|Y) \cdot P(w_i = SP|Y) \\ & = \frac{1}{2}\frac{1}{4}\frac{1}{4}\frac{1}{4}\frac{1}{12} = \frac{1}{1536} \approx 0.000651\end{aligned}$$

Solution to Q3 VII

We calculate the *necessary* probabilities for the 'Y' class as

$$\begin{aligned} P(N) &= \frac{2}{4} \\ P(w_i = TW|N) &= \frac{1+1}{5+7} \\ P(w_i = SP|N) &= \frac{2+1}{5+7} \end{aligned}$$

Finally,

$$\begin{aligned} P(N|X) \propto P(N) &\cdot P(w_i = TW|N) \cdot P(w_i = TW|N) \\ &\cdot P(w_i = TW|N) \cdot P(w_i = SP|N) \\ &= \frac{1}{2}\frac{1}{6}\frac{1}{6}\frac{1}{6}\frac{1}{4} = \frac{1}{1728} \approx 0.000579 \end{aligned}$$

Therefore, doc 5 should belong to the '**Yes**' class.

Q5 I

Consider a binary classification problem.

1. First, we randomly obtained 47 training examples among which we have 22 negative instances (denoted as "-"), and 25 positive instances (denoted as "+").
   What is your estimate of the probability that a novel test instance belongs to the positive class?
2. We then identify a feature x, and rearrange the 47 training examples based on their x values. The result is shown in the table below.

Q5 II

x	y	count
1	-	6
1	+	2
2	-	5
2	+	2
3	-	7
3	+	6
4	-	3
4	+	7
5	-	1
5	+	8

Table: Training Data

For each of the group of training examples with the same x value, compute its probability p_i and $logit(p) := \log \frac{p}{1-p}$.

3. What is your estimate of the probability that a novel test instance belongs to the positive class if its x value is 1?
4. We can run a linear regression on the $(x, logit)$ pairs from each group. Will this be the same as what Logistic Regression does?

Solution to Q5 I

1. $\Pr(+) = \frac{25}{47}$.
2. See table below.

x	$cnt(y=0)$	$cnt(y=1)$	p	$logit(p)$
1	6	2	0.250000	-1.098612
2	5	2	0.285714	-0.916291
3	7	6	0.461538	-0.154151
4	3	7	0.700000	0.847298
5	1	8	0.888889	2.079442

3. $\Pr(+|x=1) = \frac{2}{8}$.
4. Not the same. The main reason is that Logistic regression will maximize the likelihood of the data, and this is in generally different from minimizing the SSE as in Linear Regression.

Q6

Consider two-dimensional vectors $\mathbf{A} = \left(\begin{smallmatrix} 2 \\ 3 \end{smallmatrix}\right)$ and $\mathbf{B} = \left(\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right)$ and $\mathbf{C} = \mathbf{A} + \mathbf{B}$.

- Represent the vectors in the non-orthogonal bases $\mathcal{B} = \left(\begin{smallmatrix} 1 & 2 \\ 0 & -2 \end{smallmatrix}\right)$.
- Let $\mathbf{Z}_p$ be a vector $\mathbf{Z}$ represented in the polar coordinate: (ρ, θ). What if we still do $\mathbf{Z}_p = \mathbf{A}_p + \mathbf{B}_p$ in the old "linear" way? Will $\mathbf{Z}_p$ be the same as $\mathbf{C}_p$?
- Can you construct a matrix $\mathbf{M}$ such that its impact on vectors represented in polar coordinates exhibit "linearality"? i.e., $\mathbf{M}(\mathbf{x} + \mathbf{y}) = \mathbf{M}\mathbf{x} + \mathbf{M}\mathbf{y}$?

Solution to Q6

- $\mathbf{C} = \left(\begin{smallmatrix} 1 \\ 3 \end{smallmatrix}\right)$

$$\begin{aligned} \left(\begin{smallmatrix} 2 \\ 3 \end{smallmatrix}\right) &= \mathcal{B}\mathbf{A}' \Rightarrow \mathbf{A}' = \left(\begin{smallmatrix} 5 \\ -1.5 \end{smallmatrix}\right) \\ \left(\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right) &= \mathcal{B}\mathbf{B}' \Rightarrow \mathbf{B}' = \left(\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right) \\ \left(\begin{smallmatrix} \\ 0 \end{smallmatrix}\right) &= \mathcal{B}\mathbf{C}' \Rightarrow \mathbf{C}' = \left(\begin{smallmatrix} 4 \\ -1.5 \end{smallmatrix}\right) \end{aligned}$$

  Obviously, we still have $\mathbf{C}' = \mathbf{A}' + \mathbf{B}'$.
- (Obivously) No.
- Let $\mathbf{M} = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$. Then $\mathbf{M}\left(\begin{smallmatrix} r \\ \rho \end{smallmatrix}\right) = \left(\begin{smallmatrix} ar+b\rho \\ cr+d\rho \end{smallmatrix}\right)$. To have the special "linearality" (for arbitrary r and ρ), we have to set $cr + d\rho = 0$, which means $c = d = 0$, i.e., $\mathbf{M} = \left(\begin{smallmatrix} a & b \\ 0 & 0 \end{smallmatrix}\right)$.

Q7

Consider a set of d-dimensional points arranged in a *data matrix* $\mathbf{X}_{n\times d} = \begin{pmatrix} \mathbf{o}_1 \\ \mathbf{o}_2 \\ \vdots \\ \mathbf{o}_n \end{pmatrix}$. Now we consider a linear projection $\mathbf{A}_{d\times m}$ of all the points to a m-dimensional space ($m < d$). Specifically, each $\mathbf{o}$ is mapped to a new vector $\pi(o_i) = \mathbf{A}^\top \mathbf{o}_i$.

- Computer $r := \frac{\|\pi(\mathbf{o}_i)\|^2}{\|\mathbf{o}_i\|^2}$. Can you guess what will be the maximum and minimum values of r?

Solution to Q7

- Since

$$\begin{aligned} \|\pi(\mathbf{o})\|^2 &= \pi(\mathbf{o})^\top \pi(\mathbf{o}) = (\mathbf{A}^\top \mathbf{o})^\top \mathbf{A}^\top \mathbf{o} \\ &= \mathbf{o}^\top \mathbf{A}\mathbf{A}^\top \mathbf{o} = \mathbf{o}^\top \left(\mathbf{A}\mathbf{A}^\top\right) \mathbf{o} \end{aligned}$$

Therefore,

$$r = \frac{\|\pi(\mathbf{o})\|^2}{\|\mathbf{o}\|^2} = \frac{\mathbf{o}^\top \left(\mathbf{A}\mathbf{A}^\top\right) \mathbf{o}}{\mathbf{o}^\top \mathbf{o}}$$

Comment: The above is the Rayleigh Quotient (c.f., its Wikipedia page) where $\mathbf{M} = \mathbf{A}\mathbf{A}^\top$. The maximum and minimum values of r are determined by the maximum and minimum eigenvalues of $\mathbf{M}$, respectively. This property is also used in the technical proof of the spectral clustering too (not required).