School of Computing and Information Systems
The University of Melbourne
COMP90049 Knowledge Technologies (Semester 2, 2017)
Workshop exercises: Week 10

1. For the following dataset:

ID	*Outl*	*Temp*	*Humi*	*Wind*	PLAY
TRAINING INSTANCES					
A	s	h	h	F	N
B	s	h	h	T	N
C	o	h	h	F	Y
D	r	m	h	F	Y
E	r	c	n	F	Y
F	r	c	n	T	N
TEST INSTANCES					
G	o	c	n	T	?
H	s	m	h	F	?

(a) Classify the test instances using the method of 0-R.

- 0-R classifies a test instance based on the majority class of the training data.
- In this case, the training data has 3 N instances and 3 Y instances: there is no majority class.
- Consequently, we believe that both classes are equally adequate at describing this data, and we would just choose one arbitarily. Let's say we choose the lexicographically smaller one: N. Both test instances would then be classified as N.

(b) Classify the test instances using the method of 1-R.

- 1-R chooses a single attribute on which to make the decision for classifying all of the test instances. The attribute that it chooses is the one which makes the fewest errors, when used to classify the training data.
- The class chosen for a given attribute value is the equivalent of applying the 0-R method, for the instances with that attribute value. Consequently, each error it makes on the training data corresponds to an instance **not** of the majority class.
- Let's look at it in action:
- First, *Outl*:
  - *Outl* takes 3 values: `s`, `o` and `r`.
  - There are 2 `s` instances, which both have the class N. If we apply 0-R here, we would choose N for both instances, and make 0 errors.
  - There is just 1 `o` instance, which has the class Y. 0-R would (trivially) choose Y, and make 0 errors.
  - There are 3 `r` instances, 2 Y and 1 N. If we apply 0-R here, we would choose Y for all 3 instances, and make 1 error (for the `r` instance with the class Y).
  - Altogether, this is $0 + 0 + 1 = 1$ error.
- *Temp*:
  - *Temp* takes 3 values: `h`, `m` and `c`.
  - There are 3 `h` instances, 2 Y and 1 N. If we apply 0-R here, we would choose Y for all 3 instances, and make 1 error.
  - There is just 1 `m` instance, which has the class Y. 0-R makes 0 errors.
  - There are 2 `c` instances, 1 Y and 1 N. If we apply 0-R here, we would make 1 error, regardless of whether we chose Y or N.

  - Altogether, this is $1 + 0 + 1 = 2$ errors.
- *Humi*:
  - *Humi* takes 2 values: `h` and `n`.
  - There are 4 `h` instances, 2 Y and 2 N. 0-R makes 2 errors.
  - There are 2 `n` instances, 1 Y and 1 N. 0-R makes 1 error.
  - Altogether, this is $2 + 1 = 3$ errors.
- *Wind*:
  - *Wind* takes 2 values: `F` and `T`.
  - There are 4 `F` instances, 3 Y and 1 N. 0-R makes 1 error.
  - There are 2 `T` instances, both N. 0-R makes 0 errors.
  - Altogether, this is $1 + 0 = 1$ error.
- So, we choose the attribute with the fewest errors; in this case, there is a tie between *Outl* and *Wind* (1 error in total). We would need to break the tie arbitrarily.
- Let's say we chose *Outl*, how would we classify the test instances? Well, we would read off the value of *Outl* for each test instance, and classify it using 0-R (for the training attributes with the same value of *Outl*), similar to the above. For the test instance `G`, *Outl* is `o`, and we would choose Y, based on the single training instance with `o`. For `H`, *Outl* is `s`, and we would classify it as `n`.
- If instead we had chosen *Wind*, `G` takes the value `T`, so we choose N, and `H` takes the value `F`, so we choose Y. Note that these are the opposite classifications as the ones we would have made based on *Outl*!
- I can't help but notice that we skipped one of the attributes: *ID*. What happens for that attribute:
  - *ID* takes 6 different values (in the training data): `A`, `B`, `C`, `D`, `E`, and `F`.
  - When *ID* is `A`, there is just a single instance, of class N: 0-R makes 0 errors.
  - When *ID* is `B`, there is just a single instance, of class N: 0-R makes 0 errors.
  - When *ID* is `C`, there is just a single instance, of class Y: 0-R makes 0 errors.
  - And so on. You can see that we obviously aren't going to make any errors with this attribute, because each attribute value only occurs with a single training instance.
- So, 1-R thinks that, actually *ID* is the best attribute — unfortunately, though, it's completely useless for classifying the test data, because we haven't seen those values (`G` and `H`) in the training data!

For the following dataset:

apple	*ibm*	*lemon*	*sun*	CLASS
TRAINING INSTANCES				
4	0	1	1	FRUIT
5	0	5	2	FRUIT
2	5	0	0	COMPUTER
1	2	1	7	COMPUTER

2. Build a contingency table for each of the four attributes on the data collection above.

   - We want four contigency tables (see next page); one for each attribute. We will use a binary interpretation of this data (i.e. we will interpret the probabilities with respect to the instances).
   - Now, the *apple* contingency table will be built around the number of instances which contain (not equal to zero) *apple* for each of the classes. For example, *apple* is non-zero for 2 (both) of the *fruit* instances, and zero for 0 (neither), and so on.

   (a) According to "Pointwise Mutual Information", which attribute has the best correlation with the class COMPUTER?

apple	FRUIT	COMPUTER
Y	2	2
N	0	0

ibm	FRUIT	COMPUTER
Y	0	2
N	2	0

lemon	FRUIT	COMPUTER
Y	2	1
N	0	1

sun	FRUIT	COMPUTER
Y	2	1
N	0	1

Table 1: Contingency tables for the above dataset

- Pointwise Mutual Information is calculated according to the following formula (which you can compare with Mutual Information below):

$$PMI(A,C) = \log_2 \frac{P(A,C)}{P(A)P(C)}$$

- Here, we read these values as: A means $A = Y$ and C, $C = Y$.
- To assess the PMI of *apple* with COMPUTER, we first need to calculate each of the prior probabilities, and the joint probability (according to the instance–based model):

$$\begin{aligned} P(a) &= \frac{4}{4} \\ P(\textsc{c}) &= \frac{2}{4} \\ P(a,\textsc{c}) &= \frac{2}{4} \end{aligned}$$

- Now we can substitute:

$$\begin{aligned} PMI(a,\textsc{c}) &= \log_2 \frac{P(a,\textsc{c})}{P(a)P(\textsc{c})} \\ &= \log_2 \frac{\frac{2}{4}}{\frac{4}{4}\frac{2}{4}} \\ &= \log_2 \frac{0.5}{0.5} = \log_2(1) = 0 \end{aligned}$$

- The PMI of *apple* with respect to the class COMPUTER is 0, which means that there is no correlation, and (probably) no predictive capacity.
- What about the other attributes?

$$\begin{aligned} P(i) &= \frac{2}{4} \\ P(i,\textsc{c}) &= \frac{2}{4} \\ PMI(i,\textsc{c}) &= \log_2 \frac{P(i,\textsc{c})}{P(i)P(\textsc{c})} \\ &= \log_2 \frac{\frac{2}{4}}{\frac{2}{4}\frac{2}{4}} \\ &= \log_2 \frac{0.5}{0.25} = \log_2(2) = 1 \end{aligned}$$

$$\begin{aligned}
P(l) &= \frac{3}{4} \\
P(l,\textsc{c}) &= \frac{1}{4} \\
PMI(l,\textsc{c}) &= \log_2 \frac{P(l,\textsc{c})}{P(l)P(\textsc{c})} \\
&= \log_2 \frac{\frac{1}{4}}{\frac{3}{4}\frac{2}{4}} \\
&= \log_2 \frac{0.25}{0.375} = \log_2(\frac{2}{3}) \approx -0.58 \\
P(s) &= \frac{3}{4} \\
P(s,\textsc{c}) &= \frac{1}{4} \\
PMI(s,\textsc{c}) &= \log_2 \frac{P(s,\textsc{c})}{P(s)P(\textsc{c})} \\
&= \log_2 \frac{\frac{1}{4}}{\frac{3}{4}\frac{2}{4}} \\
&= \log_2 \frac{0.25}{0.375} = \log_2(\frac{2}{3}) \approx -0.58
\end{aligned}$$

- All in all, *ibm* has the greatest Pointwise Mutual Information with the class COMPUTER, as we might expect, because they are perfectly correlated.
- What if we had found the PMI of FRUIT instead? Well, in that case, we would have thought that *lemon* and *sun* were the best attributes (with a PMI of 0.42); we wouldn't have found *ibm* as a good attributem because it is reverse–correlated with FRUIT.

(b) Use the method of "Mutual Information" to rank the "goodness" of the four features in predicting this two–class problem, according to the following formula:

$$MI(A,C) = \sum_{i\in\{a,\bar{a}\}} \sum_{j\in\{c,\bar{c}\}} P(i,j) \log_2 \frac{P(i,j)}{P(i)P(j)}$$

- This is very similar to PMI, but we are going to combine the PMIs of the attribute occurring together with each class, as well as not occurring (having a frequency of 0) with each class, weighted by the number of times this has happened.
- All of this information can be trivially read off the contigency tables above. For example, for *apple*:

$$\begin{aligned}
P(a) &= \frac{4}{4} \\
P(\bar{a}) &= 0 \\
P(\textsc{f}) &= \frac{2}{4} \\
P(\textsc{c}) &= \frac{2}{4} \\
P(a,\textsc{f}) &= \frac{2}{4} \\
P(a,\textsc{c}) &= \frac{2}{4} \\
P(\bar{a},\textsc{f}) &= 0 \\
P(\bar{a},\textsc{c}) &= 0
\end{aligned}$$

- Substituting in (and taking $0 \log x \equiv 0$), we find:

$$\begin{aligned}
MI(a) &= P(a,f)\log_2 \frac{P(a,f)}{P(a)P(f)} + P(\bar{a},f)\log_2 \frac{P(\bar{a},f)}{P(\bar{a})P(f)} + \\
&\quad P(a,c)\log_2 \frac{P(a,c)}{P(a)P(c)} + P(\bar{a},c)\log_2 \frac{P(\bar{a},c)}{P(\bar{a})P(c)} \\
&= \frac{2}{4}\log_2 \frac{\frac{2}{4}}{\frac{4}{4}\frac{2}{4}} + 0\log_2 \frac{0}{(0)\frac{2}{4}} + \frac{2}{4}\log_2 \frac{\frac{2}{4}}{\frac{4}{4}\frac{2}{4}} + 0\log_2 \frac{0}{(0)\frac{2}{4}} \\
&= \frac{2}{4}(0) + 0 + \frac{2}{4}(0) + 0 = 0
\end{aligned}$$

- The value of 0 here means that, regardless of whether *apple* is present or absent from the instance, it has no predictive power for either of these classes.
- What about the other attributes? *ibm*:

$$\begin{aligned}
P(i) &= \frac{2}{4} \\
P(\bar{i}) &= \frac{2}{4} \\
P(i,\textsc{f}) &= 0 \\
P(i,\textsc{c}) &= \frac{2}{4} \\
P(\bar{i},\textsc{f}) &= \frac{2}{4} \\
P(\bar{i},\textsc{c}) &= 0
\end{aligned}$$

$$\begin{aligned}
MI(i) &= P(i,f)\log_2 \frac{P(i,f)}{P(i)P(f)} + P(\bar{i},f)\log_2 \frac{P(\bar{i},f)}{P(\bar{i})P(f)} + \\
&\quad P(i,c)\log_2 \frac{P(i,c)}{P(i)P(c)} + P(\bar{i},c)\log_2 \frac{P(\bar{i},c)}{P(\bar{i})P(c)} \\
&= 0\log_2 \frac{0}{\frac{2}{4}\frac{2}{4}} + \frac{2}{4}\log_2 \frac{\frac{2}{4}}{\frac{2}{4}\frac{2}{4}} + \frac{2}{4}\log_2 \frac{\frac{2}{4}}{\frac{2}{4}\frac{2}{4}} + 0\log_2 \frac{0}{\frac{2}{4}\frac{2}{4}} \\
&= 0 + \frac{2}{4}(1) + \frac{2}{4}(1) + 0 = 1
\end{aligned}$$

- *lemon*:

$$\begin{aligned}
P(l) &= \frac{3}{4} \\
P(\bar{l}) &= \frac{1}{4} \\
P(l,\textsc{f}) &= \frac{2}{4} \\
P(l,\textsc{c}) &= \frac{1}{4} \\
P(\bar{l},\textsc{f}) &= 0 \\
P(\bar{l},\textsc{c}) &= \frac{1}{4}
\end{aligned}$$

$$
\begin{aligned}
MI(l) &= P(l,f)\log_2\frac{P(l,f)}{P(l)P(f)} + P(\bar{l},f)\log_2\frac{P(\bar{l},f)}{P(\bar{l})P(f)} + \\
&\quad P(l,c)\log_2\frac{P(l,c)}{P(l)P(c)} + P(\bar{l},c)\log_2\frac{P(\bar{l},c)}{P(\bar{l})P(c)} \\
&= \frac{2}{4}\log_2\frac{\frac{2}{4}}{\frac{3}{4}\frac{2}{4}} + 0\log_2\frac{0}{\frac{1}{4}\frac{2}{4}} + \frac{1}{4}\log_2\frac{\frac{1}{4}}{\frac{3}{4}\frac{2}{4}} + \frac{1}{4}\log_2\frac{\frac{1}{4}}{\frac{1}{4}\frac{2}{4}} \\
&\approx \frac{2}{4}(0.42) + 0 + \frac{1}{4}(-0.58) + \frac{1}{4}(1) \approx 0.32
\end{aligned}
$$

- *sun* is exactly the same as *lemon* (check the contingency tables above), so I won't repeat the calculations here.
- All in all, *ibm* appears to be the best attribute for this dataset, as we might expect. *lemon* and *sun* are both marginally useful, and *apple* is completely useless (so maybe we should get rid of it! `:)`).