School of Computing and Information Systems
The University of Melbourne
COMP90049
Knowledge Technologies (Semester 1, 2018)
Workshop sample solutions: Week 11

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	?

Classify the test instances using a Decision Tree:

(a) Using the Information Gain as a splitting criterion

- For Information Gain, at each level of the decision tree, we're going to choose the attribute that has the largest difference between the entropy of the class distribution at the parent node, and the average entropy across its child nodes (weighted by the fraction of instances at each node);

$$IG(A|R) \quad = \quad H(R) - \sum_{i \in A} P(A=i)H(A=i)$$

- In this dataset, we have 6 instances total — 3 Y and 3 N. The entropy at the top level of our tree is $H(R) = -[\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}] = 1$.
- This is a very even distribution. We're going to hope that by branching the tree according to an attribute, that will cause the children to have an uneven distribution — which means that we will be able to select a class with more confidence — which means that the entropy will go down.
- For example, for the attribute *Outl*, we have three attribute values: `s`, `o`, `r`.
  - When *Outl*=`s`, there are 2 instances, which are both N. The entropy of this distribution is $H(O = \texttt{s}) = -[0\log 0 + 1\log 1] = 0$. Obviously, at this branch, we will choose N with a high degree of confidence.
  - When *Outl*=`o`, there is a single instance, of class Y. The entropy here is going to be 0 as well.
  - When *Outl*=`r`, there are 2 Y instances and 1 N instance. The entropy here is $H(O = \texttt{r}) = -[\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}] \approx 0.9183$.
- To find the average entropy (the "mean information"), we sum the calculated entropy at each child multiplied by the fraction of instances at that child: $MI(O) = \frac{2}{6}(0) + \frac{1}{6}(0) + \frac{3}{6}(0.9183) \approx 0.4592$.
- The overall information gain here is $IG(O) = H(R) - MI(O) = 1 - 0.4592 = 0.5408$.
- The table overleaf lists the Mean Information and Information Gain, for each of the 5 attributes.

	R	*Outl*			*Temp*			H		*Wind*		*ID*					
		s	o	r	h	m	c	h	n	T	F	A	B	C	D	E	F
Y	3	0	1	2	1	1	1	2	1	0	3	0	0	1	1	1	0
N	3	2	0	1	2	0	1	2	1	2	1	1	1	0	0	0	1
Total	6	2	1	3	3	1	2	4	2	2	4	1	1	1	1	1	1
$P(Y)$	$\frac{1}{2}$	0	1	$\frac{2}{3}$	$\frac{1}{3}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{3}{4}$	0	0	1	1	1	0
$P(N)$	$\frac{1}{2}$	1	0	$\frac{1}{3}$	$\frac{2}{3}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{4}$	1	1	0	0	0	1
H	1	0	0	0.9183	0.9183	0	1	1	1	0	0.8112	0	0	0	0	0	0
MI			0.4592			0.7924			1		0.5408				0		
IG			0.5408			0.2076			0		0.4592				1		
$GINI$	0.5	0	0	0.4444	0.4444	0	0.5	0.5	0.5	0	0.375	0	0	0	0	0	0
GS			0.2778			0.1111			0		0.25				0.5		

- At this point, *ID* has the best information gain, so hypothetically we would use that to split the root node. At that point, we would be done, because each child is purely of a single class — however, we would be left with a completely useless classifier! (Because the IDs of the test instances won't have been observed in the training data.)
- Instead, let's take the second best attribute: *Outl*.
- There are now three branches from our root node: for `s`, for `o`, and for `r`. The first two are pure, so we can't improve them any more. Let's examine the third branch (*Outl*=`r`):
  - Three instances (D, E, and F) have the attribute value `r`; we've already calculated the entropy here to be 0.9183.
  - If we split now according to *Temp*, we observe that there is a single instance for the value `m` (of class N, the entropy is clearly 0); there are two instances for the value `c`, one of class Y and one of class N (so the entropy here is 1). The mean information is $\frac{1}{3}(0) + \frac{2}{3}(1) \approx 0.6667$, and the information gain at this point is $0.9183 - 0.6667 \approx 0.2516$.
  - For *Humi*, we again have a single instance (with value `h`, class Y, $H = 0$), and two instances (of `n`) split between the two classes ($H = 1$). The mean information here will also be 0.6667, and the information gain 0.2516.
  - For *Wind*, there are two `F` instances, both of class Y ($H = 0$), and one `T` instance of class N ($H = 0$). Here, the mean information is 0 and the information gain is 0.9183.
  - *ID* would still look like a good attribute to choose, but we'll continue to ignore it.
  - All in all, we will choose to branch based on *Wind* for this child.
- All of the children of `r` are pure now, so our decision tree is complete. We can represent a decision tree over a 2-class problem like this as a pair of Boolean formulae, for example:
  - *Outl*=`o` $\cup$(*Outl*=`r` $\cap$ *Wind*=`F`) $\rightarrow$ Y (so we classify `G` as Y)
  - *Outl*=`s` $\cup$(*Outl*=`r` $\cap$ *Wind*=`T`) $\rightarrow$ N (so we classify `H` as N)

(b) Using the Gini Index as a splitting criterion

- For the Gini Index, at each level of the decision tree, we're going to choose the attribute that has the largest difference between the Gini Index of the class distribution at the parent node, and the averaged Ginis across its child nodes (weighted by the fraction of instances at each node); this is sometimes called *GINI-split*:

$$GS(A|R) \quad = \quad GINI(R) - \sum_{i \in A} P(A = i) GINI(A = i)$$

- Observe that this is the same formula as for Information Gain above.
- How do we calculate GINI for this dataset?

$$GINI(X) \quad = \quad 1 - [p(Y)^2 + p(N)^2]$$

- You might like to compare this with the formula for entropy to see why these values are closely correlated.

- Anyway, since the steps of the method are so similar to Information Gain, we've simply recorded the GINI values and GINI-split values in the table above. You can double-check that the same tree is produced as for Information Gain.

2. What is **bagging**, in the context of **Decision Trees**?

   - Bagging and Random Forests are both variants of Decision Trees.
   - In Bagging, we build a number of Decision Trees by re-sampling the data:
     - For each tree, we randomly select (with repetition) N instances out of the possible N instances, so that we have the same sized data as the deterministic decision tree, but each one is based around a different data set
     - We then build the tree as usual.
     - We classify the test instance by **voting** — each tree gets a vote (the class it would predict for the test instance), and the class with the plurality wins.

   (a) What is a **Random Forest**?

   - In Random Forests, we follow the same strategy as Bagging, but:
     - When we build a tree, for each node in the tree, we randomly select some subset of the possible attributes. (Typically, roughly $\log k$ for k attributes in total.)
     - This is different to building a deterministic tree, where we always consider every possible attribute available (unless we already used it further up the tree).

   (b) What advantages does a Random Forest have, with comparison to a (deterministic) Decision Tree model, or a bag of Decision Trees?

   - This seemingly small change gives Random Forests a number of very important benefits:
     - As in Bagging, by using many trees, we can deal with **outlier instances** in the original dataset, which might produce an undesirable (deterministic) Decision Tree (because we perceive a spurious correlation between some class and some rare attribute)
     - By using many trees, we can overcome the problem of **irrelevant attributes** — if some attribute has a spurious correlation with the class, it will appear near the top of the (deterministic) Decision Tree, but a given attribute will only be available occasionally ($\frac{\log k}{k}$), and many of the trees will find (hopefully) better attributes near the top of the tree.
     - By using a small proportion of the attribute set, we can build many trees in a reasonable amount of time; Bagging, on the other hand, often takes too long to generate enough trees to be worthwhile.

3. For the following dataset:

apple	*ibm*	*lemon*	*sun*	CLASS	Label	Length
TRAINING INSTANCES						
4	0	1	1	FRUIT	F_1	$\sqrt{18}$
5	0	5	2	FRUIT	F_2	$\sqrt{54}$
2	5	0	0	COMPUTER	C_1	$\sqrt{29}$
1	2	1	7	COMPUTER	C_2	$\sqrt{55}$
TEST INSTANCES						
2	0	3	1	?	T_1	$\sqrt{14}$
1	0	1	0	?	T_2	$\sqrt{2}$

   (a) Using the **Euclidean distance** measure, classify the test instances using the 1-NN method.

   - For this part, we are interested in the (Euclidean) distances between the instances — this is more sensitive to the length of the instance (vector) than the cosine similarity, which may or may not be appropriate, depending on the data set.

- Recall the Euclidean distance between two points A and B:

$$\text{dist}(A, B) = \sqrt{\sum_k (a_k - b_k)^2}$$

- For T_1, we find the Euclidean distances to the four training instances:

$$\begin{aligned}
\text{dist}(F_1, T_1) &= \sqrt{\sum_k (F_{1,k} - T_{1,k})^2} \\
&= \sqrt{(4-2)^2 + (0-0)^2 + (1-3)^2 + (1-1)^2} \\
&= \sqrt{8} \approx 2.828 \\
\text{dist}(F_2, T_1) &= \sqrt{(5-2)^2 + (0-0)^2 + (5-3)^2 + (2-1)^2} \\
&= \sqrt{14} \approx 3.742 \\
\text{dist}(C_1, T_1) &= \sqrt{(2-2)^2 + (5-0)^2 + (0-3)^2 + (0-1)^2} \\
&= \sqrt{35} \approx 5.916 \\
\text{dist}(C_2, T_1) &= \sqrt{(1-2)^2 + (2-0)^2 + (1-3)^2 + (7-1)^2} \\
&= \sqrt{45} \approx 6.708
\end{aligned}$$

- With this distance metric, close neighbours are ones with low scores. If we use the 1-nearest neighbour method, we observe that the closest instance is F_1. This is a FRUIT instance, so we choose FRUIT for this test instance.
- For the second test instance:

$$\begin{aligned}
\text{dist}(F_1, T_2) &= \sqrt{\sum_k (F_{1,k} - T_{2,k})^2} \\
&= \sqrt{(4-1)^2 + (0-0)^2 + (1-1)^2 + (1-0)^2} \\
&= \sqrt{10} \approx 3.162 \\
\text{dist}(F_2, T_2) &= \sqrt{(5-1)^2 + (0-0)^2 + (5-1)^2 + (2-0)^2} \\
&= \sqrt{36} = 6.000 \\
\text{dist}(C_1, T_2) &= \sqrt{(2-1)^2 + (5-0)^2 + (0-1)^2 + (0-0)^2} \\
&= \sqrt{27} \approx 5.196 \\
\text{dist}(C_2, T_2) &= \sqrt{(1-1)^2 + (2-0)^2 + (1-1)^2 + (7-0)^2} \\
&= \sqrt{53} \approx 7.280
\end{aligned}$$

- Once more, the best instances is F_1, so we choose FRUIT.

(b) It is also possible to use a similarity measure for k-NN, rather than a distance measure: using the **Cosine similarity**, classify the test instances using the 3-NN method.

- We're using the cosine measure of similarity, interpretting the instances as vectors in the feature space, and we'll find the angles between the vectors to find the nearest neighbours among the training instances to each of the test instances.
- Recall that the cosine measure between two vectors A and B is calculated as:

$$\cos(A, B) = \frac{A \cdot B}{\mid A \mid \cdot \mid B \mid}$$

- Let's start by pre-calculating the lengths of the vectors (they're shown in the table above). For example:

$$\begin{aligned}
\mid F_1 \mid &= \sqrt{4^2 + 0^2 + 1^2 + 1^2} \\
&= \sqrt{18} \approx 4.24
\end{aligned}$$

- To find the nearest neighbours for T_1, we'll calculate the cosine measure for each of the four training instances:

$$\begin{aligned}
\cos(F_1, T_1) &= \frac{F_1 \cdot T_1}{|\ F_1\ | \cdot |\ T_1\ |} \\
&= \frac{\langle 4,0,1,1\rangle \cdot \langle 2,0,3,1\rangle}{|\ \langle 4,0,1,1\rangle\ | \cdot |\ \langle 2,0,3,1\rangle\ |} \\
&= \frac{4 \cdot 2 + 0 \cdot 0 + 1 \cdot 3 + 1 \cdot 1}{\sqrt{18} \cdot \sqrt{14}} \\
&= \frac{12}{\sqrt{18} \cdot \sqrt{14}} \approx 0.7559 \\
\cos(F_2, T_1) &= \frac{F_2 \cdot T_1}{|\ F_2\ | \cdot |\ T_1\ |} \\
&= \frac{\langle 5,0,5,2\rangle \cdot \langle 2,0,3,1\rangle}{|\ \langle 5,0,5,2\rangle\ | \cdot |\ \langle 2,0,3,1\rangle\ |} \\
&= \frac{27}{\sqrt{54} \cdot \sqrt{14}} \approx 0.9820
\end{aligned}$$

$$\begin{aligned}
\cos(C_1, T_1) &= \frac{C_1 \cdot T_1}{|\ C_1\ | \cdot |\ T_1\ |} \\
&= \frac{\langle 2,5,0,0\rangle \cdot \langle 2,0,3,1\rangle}{|\ \langle 2,5,0,0\rangle\ | \cdot |\ \langle 2,0,3,1\rangle\ |} \\
&= \frac{4}{\sqrt{29} \cdot \sqrt{14}} \approx 0.1985 \\
\cos(C_2, T_1) &= \frac{C_2 \cdot T_1}{|\ C_2\ | \cdot |\ T_1\ |} \\
&= \frac{\langle 1,2,1,7\rangle \cdot \langle 2,0,3,1\rangle}{|\ \langle 1,2,1,7\rangle\ | \cdot |\ \langle 2,0,3,1\rangle\ |} \\
&= \frac{12}{\sqrt{55} \cdot \sqrt{14}} \approx 0.4324
\end{aligned}$$

- At this point, we consider the four values that we calculated. We're looking for the 3-best nearest neighbours: for the cosine measure, these are the instance with the greatest values. For T_1, this is F_2 with a score of 0.9820, and F_1 and C_2.
- Two of the 3-best neighbours are of class FRUIT, whereas only one is of class COMPUTER: we apply a voting procedure, and here FRUIT (with 2) out-votes COMPUTER (with 1), so we classify T_1 as FRUIT.
- T_2 is similar:

$$\begin{aligned}
\cos(F_1, T_2) &= \frac{F_1 \cdot T_2}{|\ F_1\ | \cdot |\ T_2\ |} \\
&= \frac{\langle 4,0,1,1\rangle \cdot \langle 1,0,1,0\rangle}{|\ \langle 4,0,1,1\rangle\ | \cdot |\ \langle 1,0,1,0\rangle\ |} \\
&= \frac{4 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 + 1 \cdot 0}{\sqrt{18} \cdot \sqrt{2}} \\
&= \frac{5}{\sqrt{18} \cdot \sqrt{2}} \approx 0.8333
\end{aligned}$$

$$
\begin{aligned}
\cos(F_2, T_2) &= \frac{F_2 \cdot T_2}{\mid F_2 \mid \cdot \mid T_2 \mid} \\
&= \frac{\langle 5,0,5,2\rangle \cdot \langle 1,0,1,0\rangle}{\mid \langle 5,0,5,2\rangle \mid \cdot \mid \langle 1,0,1,0\rangle \mid} \\
&= \frac{10}{\sqrt{54} \cdot \sqrt{2}} \approx 0.9623 \\
\cos(C_1, T_2) &= \frac{C_1 \cdot T_2}{\mid C_1 \mid \cdot \mid T_2 \mid} \\
&= \frac{\langle 2,5,0,0\rangle \cdot \langle 1,0,1,0\rangle}{\mid \langle 2,5,0,0\rangle \mid \cdot \mid \langle 1,0,1,0\rangle \mid} \\
&= \frac{2}{\sqrt{29} \cdot \sqrt{2}} \approx 0.2626 \\
\cos(C_2, T_2) &= \frac{C_2 \cdot T_2}{\mid C_2 \mid \cdot \mid T_2 \mid} \\
&= \frac{\langle 1,2,1,7\rangle \cdot \langle 1,0,1,0\rangle}{\mid \langle 1,2,1,7\rangle \mid \cdot \mid \langle 1,0,1,0\rangle \mid} \\
&= \frac{2}{\sqrt{55} \cdot \sqrt{2}} \approx 0.1907
\end{aligned}
$$

- Here again, F_2 is the nearest neighbour, followed by F_1 and C_1. So, we choose FRUIT ($2 > 1$).