School of Computing and Information Systems
The University of Melbourne
COMP90049 Knowledge Technologies (Semester 1, 2017)
Workshop exercises: Week 11

1. 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, 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 overcome a **sampling bias** in the original dataset, which might produce an undesirable (deterministic) Decision Tree (because some class is over-represented, or there is a spurious correlation between some class and some 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 overcome sampling bias.

2. 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:

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

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

- 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:

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

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

- Here again, F_2 is the nearest neighbour, followed by F_1 and C_2. So, we choose FRUIT $(2 > 1)$.

(c) How might we incorporate the values that we have calculated into a **weighted** k-NN method?

- What if we were using “weighted” k-nearest neigbour? Well, each of the k best neighbours will be accorded a weighted vote, according to the cosine similarity score we calculated above. We then accumulate these weighted votes, and the class with the greater weighting is the one that we choose.
- In the case of T_1, the top–3 neighbours are F_2, F_1, and C_2. So, FRUIT has a total weight of 1.6379 (summing the scores of F_2 and F_1) and COMPUTER only 0.4324 (from C_2), so we choose FRUIT.
- T_2 has the same nearest neighbours, and we still choose FRUIT $(1.7956 > 0.2626)$.