Department of Computing and Information Systems
The University of Melbourne
COMP90049
Knowledge Technologies (Semester 2, 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	?

(a) Classify the test instances using a Decision Tree:

i. 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 daughter 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 daughters 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 daughter multiplied by the fraction of instances at that daughter: $MI(O) = \frac{2}{6}(0) + \frac{1}{6}(0) + \frac{3}{6}(0.9183) \approx 0.4592$.
- The overall information gain here is $IG(O) = H(R) - MI(O) = 1 - 0.4592 = 0.5408$.
- The table below lists the Mean Information and Information Gain, for each of the 5 attributes.
- 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 daughter 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.)

	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					
SI		1.459			1.459			0.9183		0.9183		2.585					
GR		0.3707			0.1423			0		0.5001		0.3868					
$GINI$	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					

- 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 daughter.
- All of the daughters of r are pure now, so our decision tree is complete:
  - $Outl$=o $\cup$($Outl$=r $\cap$ $Wind$=F) → Y (so we classify G as Y)
  - $Outl$=s $\cup$($Outl$=r $\cap$ $Wind$=T) → N (so we classify H as N)

ii. Using the Gain Ratio as a splitting criterion

- Gain ratio is similar, except that we're going to weight down (or up!) by the "split information" — the entropy of the distribution of instances across the daughters of a given attribute.
- For example, we found that, for the root node, *Outl* has an information gain of 0.5408. There are 2 (out of 6) instances at the s daughter, 1 at the o daughter, and 3 at the r daughter.
- The split information for *Outl* is $SI(O) = -[\frac{2}{6}\log_2\frac{2}{6} + \frac{1}{6}\log_2\frac{1}{6} + \frac{3}{6}\log_2\frac{3}{6}] \approx 1.459$. The Gain ratio is consequently $GR(O) = \frac{IG(O)}{SI(O)} \approx \frac{0.5408}{1.459} \approx 0.3707$.
- The values for split information and gain ratio for each attribute at the root node are shown in the table above. The best attribute (with the greatest gain ratio) at the top level this time is *Wind*.
- *Wind* has two branches: T is pure, so we focus on improving F (which has 3 Y instances (C, D, E), and 1 N instance (A)). The entropy of this daughter is 0.8112.

  - For *Outl*, we have a single instance at `s` (class N, $H = 0$), a single instance at `o` (class `y`, $H = 0$), and 2 Y instances at `r` ($H = 0$). The mean information here is clearly 0; the information gain is 0.8112. The split information is $(SI(O \mid (W = F)) = -[\frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{2}\log_2\frac{1}{2}] = 1.5$, so the gain ratio is $GR(O \mid (W = F)) = \frac{0.8112}{1.5} \approx 0.5408$.
  - For *Temp*, we have two `h` instances (one Y and one N, so $H = 1$), a single `m` instance (Y, $H = 0$), and a single `c` instance (Y, $H = 0$). The mean information is $\frac{2}{4}(1) + \frac{1}{4}(0) + \frac{1}{4}(0) = 0.5$, so the information gain is $0.8112 - 0.5 = 0.3112$. The distribution of instances here is the same as *Outl*, so the split information is also 1.5, and the gain ratio is $GR(T \mid (W = F)) = \frac{0.3112}{1.5} \approx 0.2075$.
  - For *Humi*, we have 3 `h` instances (2 Y and 1 N, $H = 0.9183$), and 1 `n` instance (Y, $H = 0$): the mean information is $\frac{3}{4}(0.9183) + \frac{1}{4}(0) = 0.6887$ and the information gain is $0.8112 - 0.6887 = 0.1225$. The split information is $SI(H \mid (W = F)) = -[\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}] \approx 0.8112$, so the gain ratio is $GR(H \mid (W = F)) = \frac{0.1225}{0.8112} \approx 0.1387$.
  - For *ID*, the mean information is obviously still 0, so the information gain is 0.8112. The split information at this point is $\frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4} = 2$, so the gain ratio is approximately 0.4056.
- Of our four choices at this point, *Outl* has the best gain ratio. The resulting daughters are all pure, so the decision tree is finished:
  - *Wind*=`F` $\cap$(*Outl*=`o` $\cup$ *Outl*=`r`) → Y
  - *Wind*=`T` $\cup$(*Wind*=`F` $\cap$ *Outl*=`s`) → N (so we classify `G` and `H` as N)
- Note that this decision tree is superficially similar to the tree above, but gives different classifications because of the order in which the attributes are considered.
- Note also that we didn't need to explicitly ignore the *ID* attribute for Gain Ratio (as we needed to do for Information Gain) — the split information pushed down its "goodness" to the point where we didn't want to use it anyway!

iii. 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 daughter 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 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 to 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. Review the concepts of **Recommendation Systems**:

(a) What is Content-based Recommendation?

- As you might expect, this describes methods of making recommendation based on the content of items, usually by comparing them to items that the users has seen or enjoyed.
- This is generally in terms of the metadata of the various items, but might take into account the actual textual content under certain circumstances.

(b) What is Collaborative Filtering?

- Collaborative Filtering is a strategy for recommender systems where the recommendations for one user are based on *different* users' preferences.
- To be effective, this typically requires having a large number of users who have submitted ratings of various items — consequently, it is a problem which is difficult to **start** to solve, but once the system becomes useful, then users are incentivised to submit more ratings (so that they can receive better recommendations).

3. Consider the following rating table between five users and six items:

ID	Item A	Item B	Item C	Item D	Item E	Item F	Mean	$P(4)$
User 1	5	6	7	4	3	?	5	-0.793
User 2	4	?	3	?	5	4	4	0.667
User 3	?	2	4	1	1	?	2	-0.894
User 4	7	4	3	7	?	4	5	N/A
User 5	1	?	3	2	2	7	3	-0.605
Mean	4.25	4	4	3.5	2.75	5		
$AC(e)$	0.408	-0.894	-0.882	0.943	N/A	-0.707		
$P(e)$	0.259	0.71	-0.076	0.990	N/A	-0.707		

(a) Predict the value of the unknown rating for User 4 using User-based Collaborative Filtering. (i.e. Find the Pearson correlation between users, and adjust User 4's mean score).

- In User-based Collaborative Filtering, we begin by constructing a model of similarity between our target user (in this case, User 4) and other users.
- Here, we are going to use the Pearson correlation statistic to determine that similarity:

$$P(X,Y) = \frac{\sum_{i \in X \cap Y}(r_{xi} - \mu_x)(r_{yi} - \mu_y)}{\sqrt{\sum_{i \in X \cap Y}(r_{xi} - \mu_x)^2}\sqrt{\sum_{i \in X \cap Y}(r_{yi} - \mu_y)^2}}$$

- If you compare this equation to the Cosine Similarity, you will see that this is very similar to finding the cosine of the angle between the vectors as defined by the users' ratings, except that we are subtracting away the average, to get a **mean-centred** value.
- Observe that we only compare the values which **both** users have rated — this is important for comparison, but it means that we might get unreliable results if the two users only share a small number of ratings.
- Anyway, we begin by calculating each user's average rating (μ), of items that they have actually rated:

$$\begin{aligned}
\mu_1 &= \frac{r_{1a} + r_{1b} + r_{1c} + r_{1d} + r_{1e}}{|U_1|} \\
&= \frac{5+6+7+4+3}{5} = 5 \\
\mu_2 &= \frac{4+3+5+4}{4} = 4 \\
\mu_3 &= \frac{2+4+1+1}{4} = 2 \\
\mu_4 &= \frac{7+4+3+7+4}{5} = 5 \\
\mu_5 &= \frac{1+3+2+2+7}{5} = 3
\end{aligned}$$

- Now, we need to find the similarity — according to the Pearson correlation coefficient — for each user with respect to our target user (4). For User 1, both users have rated Items

A, B, C, and D, so these are the basis for our calculation:

$$\begin{aligned}
P(1,4) &= \frac{\sum_{i \in U_1 \cap U_4} (r_{1i} - \mu_1)(r_{4i} - \mu_4)}{\sqrt{\sum_{i \in U_1 \cap U_4} (r_{1i} - \mu_1)^2} \sqrt{\sum_{i \in U_1 \cap U_4} (r_{4i} - \mu_4)^2}} \\
&= \frac{(r_{1a} - \mu_1)(r_{4a} - \mu_4) + (r_{1b} - \mu_1)(r_{4b} - \mu_4) + (r_{1c} - \mu_1)(r_{4c} - \mu_4) + (r_{1d} - \mu_1)(r_{4}}{\sqrt{(r_{1a} - \mu_1)^2 + (r_{1b} - \mu_1)^2 + (r_{1c} - \mu_1)^2 + (r_{1d} - \mu_1)^2} \sqrt{(r_{4a} - \mu_1)^2 + (r_{4b} - \mu_1)^2 + (r_{4c} - \mu}} \\
&= \frac{(5-5)(7-5) + (6-5)(4-5) + (7-5)(3-5) + (4-5)(7-5)}{\sqrt{(5-5)^2 + (6-5)^2 + (7-5)^2 + (4-5)^2} \sqrt{(7-5)^2 + (4-5)^2 + (3-5)^2 + (7-5)^2}} \\
&= \frac{(0)(2) + (1)(-1) + (2)(-2) + (-1)(2)}{\sqrt{0^2 + 1^2 + 2^2 + (-1)^2} \sqrt{2^2 + (-1)^2 + (-2)^2 + (2)^2}} \\
&= \frac{-7}{\sqrt{6}\sqrt{13}} \approx -0.793
\end{aligned}$$

- The significance of this value is that the user's scores are **negatively** correlated: if one user likes an item more than their average, this is a moderately good indication that the other user will like that item **less** than their average.
- And we proceed with the other users. For User 2 and User 4, they have both rated Items A, C, and F:

$$\begin{aligned}
P(2,4) &= \frac{(4-4)(7-5) + (3-4)(3-5) + (4-4)(4-5)}{\sqrt{(4-4)^2 + (3-4)^2 + (4-4)^2} \sqrt{(7-5)^2 + (3-5)^2 + (4-5)^2}} \\
&= \frac{(0)(2) + (-1)(-2) + (0)(-1)}{\sqrt{0^2 + (-1)^2 + 0^2} \sqrt{2^2 + (-2)^2 + (-1)^2}} \\
&= \frac{2}{\sqrt{1}\sqrt{9}} \approx 0.667
\end{aligned}$$

- This user is positively correlated with our target user, meaning that this user's ratings (with respect to their average rating) are a moderately good predictor of user's ratings. (Albeit based only a single non-average rating, of Item D!)
- And so on for the other users. We have summarised the various similarities in the table above ($P(4)$).
- Now, after we have found all of the similarities, we will predict the missing rating (for Item E) for User 4. To do this, we would typically only base our calculations on a small proportion of the total set of users: for example, the users having the most positive scores (whose judgements correlate the best with our target user), or the users with the largest absolute-valued scores (because some users, like User 3 in this case, seem to be a good predictor that our target user won't actually like the same sorts of items). In this case, we will just use all 4 users for completeness.
- To predict the rating of Item E for User 4, we are going to estimate it with respect to User 4's average rating:

$$\hat{r}_{uj} = \mu_u + \frac{\sum_v P(u,v) \cdot (r_{vj} - \mu_v)}{\sum_v |P(u,v)|}$$

- However, we are only going to consider users who have actually rated this item: in this case, all four users have rated Item E, but if some hadn't, then they would be excluded from the following calculation:

$$\begin{aligned}
\hat{r}_{4e} &= \mu_4 + \frac{P(1,4)(r_{1e} - \mu_1) + P(2,4)(r_{2e} - \mu_2) + P(3,4)(r_{3e} - \mu_3) + P(5,4)(r_{5e} - \mu_5)}{|P(1,4)| + |P(2,4)| + |P(3,4)| + |P(5,4)|} \\
&\approx 5 + \frac{(-0.793)(3-5) + (0.667)(5-4) + (-0.894)(1-2) + (-0.605)(2-3)}{|(-0.793)| + (0.667) + |(-0.894)| + |(-0.605)|} \\
&= 5 + \frac{3.752}{2.959} \approx 6.268
\end{aligned}$$

- So, our prediction of the rating of Item E for User 4 is about 6.3, which would be a somewhat above-average item for User 4. (Effectively, what we are observing is that this item is above-average for the one user (2) who is positively correlated, and below-average for the negatively correlated users.)

(b) Predict the value of the unknown rating for User 4 using Item-based Collaborative Filtering. (i.e. Find the correlation between items (using "Adjusted Cosine Similarity"), and take a weighted average of User 4's scores).

- We proceed the same way as before, however, this time we are interested in the similarity between the various items and our target item (E).
- The Pearson correlation coefficient, in this case between two items M and N, would be with respect to the users who have rated both items:

$$P(M,N) = \frac{\sum_{i \in M \cap N} (r_{im} - \mu_m)(r_{in} - \mu_n)}{\sqrt{\sum_{i \in M \cap N} (r_{im} - \mu_m)^2} \sqrt{\sum_{i \in M \cap N} (r_{in} - \mu_n)^2}}$$

- However, we are instead asked to use the "Adjusted Cosine", which actually looks very similar. Rather than centring the mean with respect to the item's average, we centre with respect to the user's average:

$$AC(M,N) = \frac{\sum_{i \in M \cap N} (r_{im} - \mu_i)(r_{in} - \mu_i)}{\sqrt{\sum_{i \in M \cap N} (r_{im} - \mu_i)^2} \sqrt{\sum_{i \in M \cap N} (r_{in} - \mu_i)^2}}$$

- (If you would like to compare with Pearson here, the item-item coefficients ($P(e)$) are given in the table above.)
- So, we calculate the similarity between each item and our target item. For Items A and E, both are rated by Users 1, 2, and 5:

$$\begin{aligned} AC(a,e) &= \frac{(r_{1a} - \mu_1)(r_{1e} - \mu_1) + (r_{2a} - \mu_2)(r_{2e} - \mu_2) + (r_{5a} - \mu_5)(r_{5e} - \mu_5)}{\sqrt{(r_{1a} - \mu_1)^2 + (r_{2a} - \mu_2)^2 + (r_{5a} - \mu_5)^2} \sqrt{(r_{1e} - \mu_1)^2 + (r_{2e} - \mu_2)^2 + (r_{5e} - \mu_5)^2}} \\ &= \frac{(5-5)(3-5) + (4-4)(5-4) + (1-3)(2-3)}{\sqrt{(5-5)^2 + (4-4)^2 + (1-3)^2} \sqrt{(3-5)^2 + (5-4)^2 + (2-3)^2}} \\ &= \frac{(0)(-2) + (0)(1) + (-2)(-1)}{\sqrt{0^2 + 0^2 + (-2)^2} \sqrt{(-2)^2 + 1^2 + (-1)^2}} \\ &= \frac{2}{\sqrt{4}\sqrt{6}} \approx 0.408 \end{aligned}$$

- This item's ratings are somewhat correlated with our target item's ratings (after taking the average ratings into account — in general, Item A was pretty average, but one user had both A and E below average).
- For Item B, we only have Users 1 and 3 who have rated both:

$$\begin{aligned} AC(b,e) &= \frac{(6-5)(3-5) + (2-2)(1-2)}{\sqrt{(6-5)^2 + (2-2)^2} \sqrt{(3-5)^2 + (1-2)^2}} \\ &= \frac{(1)(-2) + (0)(-1)}{\sqrt{1^2 + 0^2} \sqrt{(-2)^2 + (-1)^2}} \\ &= \frac{-2}{\sqrt{1}\sqrt{5}} \approx -0.894 \end{aligned}$$

- This item is quite negatively correlated (which is different to Pearson here). And so we proceed: we have summarised the item similarities in the table above ($AC(e)$).
- As with the User-based filtering, we might like to only base our rating estimate on a small proportion of the total set of items: in this case, Item D looks particularly similar to Item E, whereas Items B and C look quite negatively correlated. In practice, we don't have this luxury, because there is no guarantee that our user has rated the more predictive

items; consequently, we are usually reduced to using whichever ratings the user has given us. From a numerical point of view, however, negative coefficients will ruin our average (this is the downside of this simpler rating formula, as compared to the mean-adjusted one from question (a)), so we typically exclude items that are negatively correlated with the target item from our average[1].

- To predict the rating of Item E for User 4, we are going to estimate it by taking a weighted average of User 4's other ratings:

$$\hat{r}_{uj} \quad = \quad \frac{\sum_i AC(i,j) \cdot r_{ui}}{\sum_v |P(i,j)|}$$

- In this case, User 4 has rated all of the other items, but we ignore Items B, C and F (which are negatively correlated):

$$\begin{aligned} \hat{r}_{4e} &= \frac{P(a,e)r_{4a} + P(d,e)r_{4d}}{|P(a,e)| + |P(d,e)|} \\ &\approx \frac{(0.408)(7) + (0.943)(7)}{(0.408) + (0.943)} \\ &= \frac{9.457}{1.351} = 7 \end{aligned}$$

- This time, we end up with a well above-average predicted rating. It's notable that we expect that User 4 will enjoy Item E much more than any other user (because the have a high average to begin with, and rated Items A and D so highly!).

[1] You might be wondering what happens if the user has *only* rated items that are negatively correlated with our target item. In that case, it makes sense that this item is unlikely to relevant to our user anyway!