School of Computing and Information Systems
The University of Melbourne
COMP90049 Introduction to Machine Learning (Semester 2, 2020)

Sample Solutions: Week 12

1. For the following set of instances:

a_1	a_2	a_3	c
hot	windy	dry	Yes
mild	windy	rainy	No
hot	windy	rainy	Yes
cool	still	dry	Yes
cool	still	rainy	No
hot	still	dry	No
mild	still	dry	Yes

Construct all of the **1-itemsets** and calculate their confidences and supports. Discuss how you would continue mining for effective **Association Rules**.

The 1-itemsets are all of the values each of the attributes can take. Counting the "class" values (this is an unsupervised method), there are 9 such item sets: *{hot}, {mild}, {cool}, {windy}, {still}, {dry}, {rainy}, {Yes},* and *{No}*.

When we consider support and confidence, it's typically over an **association rule**, not an itemset. There are two (defective) association rules that we can generate for a given 1-itemset, say *{hot}*: one with the item in the antecedent and one with the item in the consequent:

$$\{hot\} \rightarrow \emptyset \quad (1)$$

$$\emptyset \rightarrow \{hot\} \quad (2)$$

Recall the formulae for confidence and support, for *N* instances in total:

$$Conf(A \rightarrow B) = \frac{n(A,B)}{n(A)}$$

$$Supp(A \rightarrow B) = \frac{n(A,B)}{N}$$

For rules of type (1) (with an empty consequent), the confidence will be 1 (because n(A, ∗) = n(A)). For rules of type (2) (with an empty antecedent), the confidence will be equal to the support (because n(∗) = N). Hence, the only interesting value we need to calculate at this point is support:

Item	*hot*	*mild*	*cool*	*windy*	*still*	*dry*	*rainy*	*YES*	*NO*
Support	$\frac{3}{7}$	$\frac{2}{7}$	$\frac{2}{7}$	$\frac{3}{7}$	$\frac{4}{7}$	$\frac{4}{7}$	$\frac{3}{7}$	$\frac{4}{7}$	$\frac{3}{7}$

If at this point, we are going to continue mining for effective association rules, we will want rules with good confidence and good support. In particular, we will set a pair of thresholds τ_c and τ_s, and we will attempt to extract every rule whose confidence and support are above both of these thresholds.

One way to attempt this is to just generate every rule and calculate both support and confidence, but in general, there will be far too many rules (as the number of rules is largely exponential in the number of attributes).

To restrict the number of rules under consideration, we observe that, from a given k-itemset, we can construct a number of k + 1 item sets by adding each one of the attribute values that aren't already in this rule — but that the support can only go **down** by adding another attribute value.

(Why?)

So, from the 1-itemsets above, if any of them fail to pass our support threshold, then we can exclude that itemset from being a subset of an interesting 2-itemset.

For example, if the support threshold is $\tau_s = 0.5$, the only interesting 1-itemsets are *{still}, {dry},* and *{Yes}*, and the only potentially interesting 2-itemsets are *{still,dry}, {still,YES},* and *{dry,YES}.*

We would then calculate the supports of those item sets (and continue to *{still, dry, YES}* if they continued to meet the support threshold of 0.5).

Once we can no longer construct further item sets that can meet the support threshold (because enough *k* item sets have supports below the threshold so as to block all of the *k + 1* item sets), we can generate all of the association rules from each itemset, calculate the confidences of these rules, and return with the ones that pass the confidence threshold τ_c.

(We can construct about 2^k different association rules for a *k*-itemset. If we have many item sets that meet the support threshold, there could still be too many rules to examine the confidences. In this case, we derive a similar procedure by observing that moving an attribute value from the antecedent to the consequent can only reduce the confidence of the rule. For a given k-itemset, we start with each of the k attribute values in the consequent one at a time, find the confidence, and proceed with rules that meet the confidence threshold.)

2. What does "correlation does not imply causation" mean? Why is it important to keep this adage in mind, when working in the field of Data Mining?

"Correlation does not imply causation" is an important adage which highlights the fact some events are reliably seen together, even though there isn't a causal relationship between the events.

There are a number of reasons why this might be the case:

- The causal relationship exists, but not in the direction that we think; for example: "farmers usually have breakfast just before sunrise" leads us to wrongly conclude that "sunrise is caused by farmers having breakfast"
- The casual relationship exists, but with an unseen intermediary; for example: "I usually have breakfast just after sunrise" leads us to wrongly conclude that "sunrise causes me to have breakfast" (whereas I might actually have breakfast just before work, whose starting time is correlated with sunrise)
- There is an unseen factor which is the actual cause of both events; for example, "the month-over-month consumption of ice cream is correlated with observed incidences of sunburns" leads us to wrongly conclude either "eating ice cream causes sunburns" or "people eat ice cream when they have a sunburn", whereas the causal relationship is due to the exterior temperature ("people eat more ice cream and get more sunburns in summer")
- The statistical correlation is entirely coincidental; there were a number of examples of this in the lecture, for example "the number of people who die annually from drowning by falling into a swimming pool is correlated with the number of films in which Nicholas Cage appears" leads us to wrongly conclude that "Nicholas Cage appearing in films causes people to drown in swimming pools" (!) or "people drowning in swimming pools causes Nicholas Cage to appear in films" (!!)

In Data Mining, we are looking for patterns about the data, and we are hoping to find patterns which are:

(i). Valid: they are actually attested in the data; the data has support for this pattern

(ii). Non-trivial: the pattern isn't immediately self-evident from the data, like: "the given instances are composed of attributes"; or, for something at least somewhat interpreted: "average household water use is correlated with the number of persons residing in the

household"

(iii). Previously unknown: finding such a rule tells us something we didn't already know about the data; many non-trivial patterns are already well-understood by experts, and so data mining doesn't necessarily give any extra value in these circumstances (other than saving the experts a little bit of time)

The point is that, for a given pattern, it is difficult to assess whether the pattern is **useful**, by meeting the three guidelines above (as well as being actionable in whatever context the data is from). An algorithm can posit patterns that it believes to be valid but aren't actually useful for some reason.

If we consider a large number of possible patterns (as we do in, say, Association Rule Mining), we are likely to find some which are entirely statistical quirks, and do not actually imply any causal relationship — hence the pattern is not actually useful, even though it might appear to be so.

3. 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 have 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 incentivized to submit more ratings (so that they can receive better recommendations).

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

ID	Item A	Item B	Item C	Item D	Item E	Item F
User 1	5	6	7	4	3	?
User 2	4	?	3	?	5	4
User 3	?	2	4	1	1	?
User 4	7	4	3	7	?	4
User 5	1	?	3	2	2	7

(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-centered** 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:

$$\mu_1 = \frac{r_{1a} + r_{1b} + r_{1c} + r_{1d} + r_{1e}}{|U_1|} = \frac{5+6+7+4+3}{5} = 5$$

$$\mu_2 = \frac{4+3+5+4}{4} = 4$$

$$\mu_3 = \frac{2+4+1+1}{4} = 2$$

$$\mu_4 = \frac{7+4+3+7+4}{5} = 5$$

$$\mu_5 = \frac{1+3+2+2+7}{5} = 3$$

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:

$$P(1,4) = \frac{\sum_{i \in U_1 \cap U_4}(r_{1i} - \mu_1)(r_{4i} - \mu_4)}{\sqrt{\sum_{i \in U_1 \cap U_4}(r_{1i} - \mu_1)^2}\sqrt{\sum_{i \in U_1 \cap U_4}(r_{4i} - \mu_4)^2}}$$

$$= \frac{(r_{1a} - \mu_1)(r_{4a} - \mu_4) + (r_{1b} - \mu_1)(r_{4b} - \mu_4) + (r_{1c} - \mu_1)(r_{4c} - \mu_4) + (r_{1d} - \mu_1)(r_{4d} - \mu_4)}{\sqrt{(r_{1a} - \mu_1)^2 + (r_{1b} - \mu_1)^2 + (r_{1c} - \mu_1)^2 + (r_{1d} - \mu_1)^2}\sqrt{(r_{4a} - \mu_4)^2 + (r_{4b} - \mu_4)^2 + (r_{4c} - \mu_4)^2 + (r_{4d} - \mu_4)^2}}$$

$$= \frac{(5-5)(7-5) + (6-5)(4-5) + (7-5)(3-5) + (4-5)(7-5)}{\sqrt{(5-5)^2 + (6-5)^2 + (7-5)^2 + (4-5)^2}\sqrt{(7-5)^2 + (4-5)^2 + (3-5)^2 + (7-5)^2}}$$

$$= \frac{(0)(2) + (1)(-1) + (2)(-2) + (-1)(2)}{\sqrt{0^2 + 1^2 + 2^2 + (-1)^2}\sqrt{2^2 + (-1)^2 + (-2)^2 + 2^2}}$$

$$= \frac{-7}{\sqrt{6}\sqrt{13}} \approx -0.793$$

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:

$$P(2,4) = \frac{(4-4)(7-5) + (3-4)(3-5) + (4-4)(4-5)}{\sqrt{(4-4)^2 + (3-4)^2 + (4-4)^2}\sqrt{(7-5)^2 + (4-5)^2 + (3-5)^2}}$$

$$= \frac{(0)(2) + (-1)(-2) + (0)(-1)}{\sqrt{0^2 + (-1)^2 + 0^2}\sqrt{2^2 + (-1)^2 + (-2)^2}}$$

$$= \frac{2}{\sqrt{1}\sqrt{9}} \approx 0.667$$

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 summarized the various similarities in the following table:

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

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:

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

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:

$$\hat{r}_{uj} = \mu_u + \frac{P(1,4).(r_{1e} - \mu_1) + P(2,4).(r_{2e} - \mu_2) + P(3,4).(r_{3e} - \mu_3) + P(5,4).(r_{5e} - \mu_5)}{|P(1,4)| + |P(2,4)| + |P(3,4)| + |P(5,4)|}$$

$$\approx 5 + \frac{(0.793)(3-5) + (0.667)(5-4) + (-0.894)(1-2) + (-0.605)(2-3)}{|(-0.793)| + |0.667| + |(-0.894)| + |(-0.605)|}$$

$$= 5 + \frac{3.752}{2.959} \approx 6.268$$

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 centering the mean with respect to the item's average, we center 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 below.)

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		

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:

$$AC(a,e) = \frac{(r_{1a}-\mu_1)(r_{1e}-\mu_1)+(r_{2a}-\mu_2)(r_{2e}-\mu_2)+(r_{5a}-\mu_5)(r_{5e}-\mu_5)}{\sqrt{(r_{1a}-\mu_1)^2+(r_{2a}-\mu_2)^2+(r_{5a}-\mu_5)^2}\sqrt{(r_{1e}-\mu_1)^2+(r_{2e}-\mu_2)^2+(r_{5e}-\mu_5)^2}}$$

$$= \frac{(5-5)(3-5)+(4-4)(5-4)+(1-3)(2-3)}{\sqrt{(5-5)^2+(4-4)^2+(1-3)^2}\sqrt{(3-5)^2+(5-4)^2+(2-3)^2}}$$

$$= \frac{(0)(-2)+(0)(1)+(-2)(-1)}{\sqrt{0^2+0^2+(-2)^2}\sqrt{(-2)^2+1^2+(-1)^2}} = \frac{2}{\sqrt{4}\sqrt{6}} \approx 0.408$$

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:

$$AC(b,e) = \frac{(6-5)(3-5)+(2-2)(1-2)}{\sqrt{(6-5)^2+(2-2)^2}\sqrt{(3-5)^2+(1-2)^2}}$$

$$= \frac{(1)(-2)+(0)(-1)}{\sqrt{1^2+0^2}\sqrt{(-2)^2+(-1)^2}} = \frac{-2}{\sqrt{1}\sqrt{5}} \approx -0.894$$

This item is quite negatively correlated (which is different to Pearson here). And so, we proceed: we have summarized 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:

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

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

[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!

$$\hat{r}_{4e} = \frac{AC(a,e).r_{4a} + AC(d,e).r_{4d}}{|AC(a,e)| + |AC(d,e)|}$$
$$\approx \frac{(0.408)(7) + (0.943)(7)}{(0.408) + (0.943)} = \frac{9.457}{1.351} = 7$$

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 they have a high average to begin with, and rated Items A and D so highly!).