Workshop Week 11 - COMP20008 2020

1. Consider the following data set for a binary class problem:

Feature A	Feature B	Class Label
T	F	+
T	T	+
T	T	+
T	F	-
T	T	+
F	F	-
F	F	-
F	F	-
T	T	-
F	F	-

We wish to select the feature that best predicts the class label using the χ^2 method.

- Write down the observed and expected contingency tables for feature A
- Calculate the $\chi^2(A, Class)$ value.
- Using the table below, conclude whether feature A is independent of the class label for $p = 0.05$.

df	P = 0.05	P = 0.01	P = 0.001
1	3.84	6.64	10.83
2	5.99	9.21	13.82
3	7.82	11.35	16.27
4	9.49	13.28	18.47
5	11.07	15.09	20.52
6	12.59	16.81	22.46

- Repeat the process for feature B and decide which feature could be best used for predicting the class label.

Observed table:

A	A=T	A=F	Total
Class=+	4	0	4
Class=-	2	4	6
Total	6	4	10

Expected table:

A	A=T	A=F	Total
Class=+	2.4	1.6	4
Class=-	3.6	2.4	6
Total	6	4	10

$\chi^2(A, Class) = \frac{(4-2.4)^2}{2.4} + \frac{(0-1.6)^2}{1.6} + \frac{(2-3.6)^2}{3.6} + \frac{(4-2.4)^2}{2.4} = 4.44$

Degrees of freedom $= (2-1) \times (2-1) = 1$
Lookup value in table (3.84). Since our calculated χ^2 value is greater than the critical value in the table, conclude A is not independent of Class for $p = 0.05$

For feature B:
Observed table:

B	B=T	B=F	Total
Class=+	3	1	4
Class=-	1	5	6
Total	4	6	10

Expected table:

B	B=T	B=F	Total
Class=+	1.6	2.4	4
Class=-	2.4	3.6	6
Total	4	6	10

$\chi^2(B, Class) = \frac{(3-1.6)^2}{1.6} + \frac{(1-2.4)^2}{2.4} + \frac{(1-2.4)^2}{3.6} + \frac{(5-3.6)^2}{3.6} = 3.40$

Degrees of freedom $= (2-1) \times (2-1) = 1$
Lookup value in table (3.84). Since our calculated χ^2 value is less than the critical value in the table, conclude B is independent of Class for $p = 0.05$

Feature A best predicts class label.

2. Open Jupyter Notebook and implement a coded solution to Question 1. Ensure that your code gives the same answer as your calculation in Question 1.

3. Consider the following dataset:

User	Iron Man	Superman	Batman	Spiderman	Ant-Man	Wonder Woman
Anne	3		3	3.5	2.5	3
Bob	4	3.5	2.5	4	3	3
Chris	3	3	3			4
Dave		3.5	2	4	2.5	
Eve		3		3	2	5
Frank	2.5	4		5	3.5	5
Gary		4.5	3	4	2	

- Use the Item-based recommender systems approach discussed in lectures to predict

Frank's rating for Batman. First comput averages for each movie:

$$\{IronMan : 3.13, Superman : 3.58, Batman : 2.7,$$
$$Spiderman : 3.92, Antman : 2.58, WonderWoman : 4\}$$

Use these as imputed values for all missing entries.
Then calculate the similarity scores with Batman for each movie using $sim(i_i, i_j) = \frac{1}{1+d(i_i,i_j)}$ where $d(i_i, i_j) = \sqrt{\sum_{k=1}^{m}(r_{ki} - r_{kj})^2}$:

$$\{IronMan : 0.34, Superman : 0.27,$$
$$Spiderman : 0.21, Antman : 0.36, WonderWoman : 0.20\}$$

Finally, use the weighted average of the three most similar values to predict the final result: $(0.36 \times 3.5 + 0.34 \times 2.5 + 0.27 \times 4)/(0.36 + 0.34 + 0.27) = 3.29$

- Use the User-based recommender systems approach to predict Frank's rating for Batman First comput averages for each person:

  $$\{Anne : 3, Bob : 3.33, Chris : 3.25, Dave : 3, Eve : 3.25, Frank : 4, Gary : 3.38\}$$

  Use these as imputed values for all missing entries.
  Then calculate the similarity scores with Frank for each person using $sim(i_i, i_j) = \frac{1}{1+d(i_i,i_j)}$ where $d(i_i, i_j) = \sqrt{\sum_{k=1}^{m}(r_{ki} - r_{kj})^2}$:

  $$\{Anne : 0.245, Bob : 0.240, Chris : 0.284, Dave : 0.236, Eve : 0.257, Gary : 0.262\}$$

  Finally, use the weighted average of the three most similar values to predict the final result: $(0.284 \times 3 + 0.262 \times 3 + 0.245 \times 3)/(0.284 + 0.262 + 0.245) = 3$
- Identify the advantages and disadvantages of each approach **The item based approach works by considering the ratings of similar movies, while the user based approach works by considering the opinions of similar users. Item based measures tend to perform better in many practical cases. Generally, users are likely to be added more frequently than movies meaning the offline computation needs to be updated more often for a user-based approach. It's also very difficult to make recommendations to new users with a user-based approach**

4. Recommender systems are challenged by the "cold start" problem - how to make recommendations to new users, about whom little is known, and how to make recommendations about new items. Suggest three strategies that might be used to address this.
   For new items:

   - **Find similar items in current dataset (based on description, author, title, category, etc), take the mean (or other summarization) of these neighbours as the initialization of the new item**
   - **Pay someone to rate the new item**

   For new users:

- **Use similar users' data to initilize the new user (possibly, based on gender, age, region, etc)**
- **Ask user questions to obtain more data in the first place, e.g. providing a collection of items and asking user to chose the ones they like**
- **Use the most popular items as the initial suggestion system wide (overall, no enough data to provide good recommendation)**
- **Use content-based recomendation at first, after getting enough data, then move to collaborative filtering method such as matrix factorization**

5. Recommender systems are sometimes criticised for over-recommending popular items to users and under-recommending rarer items. Why do you think this happens? How might it be addressed? **A Cycle:**
   popular items been recommended → users are more likely to give high rates to these item→ the popularity of these items increase
   Recommender system(RS) has no understanding of the items themselves, over and under recommending naturally occur as the system is performing its job

   Approaches to address this:

   - **Adding an extra level to manipulate the results from RS, to ensure the diversity (cover a broad range of items)**
   - **Giving weight to the timeliness of a item, to prevent users get the same recommendation all the time**

Exam 2016 - Questions 1d 3a-c

Consider the following dataset D which describes 3 people:

Age	Weight
20	40
30	50
25	25

a) (2 marks) Using a Euclidean distance function, construct the dissimilarity matrix for D. Show all working. (You may leave any square root terms unsimplified).

b) (2 marks) A dissimilarity matrix can be visualised as a heat map. Explain why it can be useful to reorder a dissimilarity matrix, using a technique such as the VAT algorithm. What information can this provide and why?

c) (2 marks) Would there be a benefit of applying principal components analysis to D to assist in visualisation? Explain.

Instance ID	1	2	3
1	0	$\sqrt{200}$	$\sqrt{250}$
2	$\sqrt{(200)}$	0	$\sqrt{650}$
3	$\sqrt{(250)}$	$\sqrt{650}$	0

Visualising as a reordered heat map using algorithm such as VAT – identify the cluster structure of the data. i.e. How many clusters there are and which objects are in each cluster. Might also help identify anomalies – which objects are not similar to other objects.

Little apparent benefit in applying PCA – the dataset is only 2 dimensions and already easy to visualize.

Exam 2016 - Question 3

d) Bob and Alice are having a debate about imputation methods for missing values. Bob argues that imputation using the mean of a feature works best. Alice argues that imputation using matrix factorization is better. To decide who is correct, they select a dataset with 10 features $F_1, \ldots, F_{10}$ and 100 instances $x_1, \ldots, x_{100}$. For each instance, they randomly make one of its feature values to be missing (yielding 100 missing values overall). They then separately apply each of the two imputation methods and assess which performs better in recovering the missing values.

i) (2 marks) What metric would be appropriate here for assessing performance of an imputation method? Provide a formula for your metric and justify it.

ii) (3 marks) Bob and Alice are joined by a third student, Barbara. Barbara says "You are both wrong. If an instance has any missing values, it is better to discard it entirely rather than trying to perform imputations." Describe three scenarios which support Barbara's reasoning.

Use a metric that finds aggregate deviation from the true answers. E.g. something like Mean squared error= $\frac{1}{100} * \sum_{i=1}^{i=100} (true\_value(x_i) - imputed\_value(x_i))^2$ **where i is an index that ranges over the 100 missing values.**

Could also use the mean absolute error as well (average of the absolute values of the deviations)

ii) Reasons it might be better to discard

- **we already have a large dataset, that contains sufficient information even when examples are discarded.**

- **if imputation method is likely to be computationally expensive (e.g. matrix factorization), then might choose discard instance if efficiency is important**
- **if we believe imputation is likely to cause problems or contaminate later analysis (due to its unreliability)**
- **scenarios where each instance is either complete (has nothing missing), or has mostly missing values.**

Exam 2018 - Question 10

Consider the following steps of the k-NN algorithm to classify a single test instance

```
1. Compute distance of the test instance to each of the instances
in the training set and store these distances
2. Sort the calculated distances
3. Store the K nearest points
4. Calculate the proportions of each class
5. Assign the class with the highest proportion
```

Step 1 may be very slow if the training set is large.
Suggest three possible strategies that might be used to speed up this step and describe a disadvantage of each.

Need three points, 1 mark each (0.5 mark for the strategy and 0.5 for its disadvantage)

Some possibilities:

- **could sample instances to reduce size of data (Disadvantage=lose information, less accurate)**
- **could apply PCA to reduce number of features (disadvantage=lose information)**
- **use a faster computer (disadvantage=expensive)**
- **use an index structure (disadvantage=complicated to implement, or might not work well for the distribution of data in the dataset)**