Assignment 4

Gengcong Yan - 1009903
CS-E4650 Methods of Data mining

November 28, 2021

1 Task 1

1.1 Task1.a

The graphs shown in Fig 1 and 2 represent different measures in questions, the bigger the nodes are, the higher the corresponding values are. Answers follow:

1. Node degree. Nodes $[1477, 1443, 1502, 1457]$ have most degrees.

2. Weighted degree. Nodes $[1437, 1563]$ have most weighted degrees.

3. Closeness centrality. Nodes $[1477, 1443, 1502, 1457]$ have the largest closeness centrality.

4. Betweenness centrality. Nodes $[1443, 1477]$ have the largest betweenness centrality.

5. Although the node with the most degrees has more nodes connected to them, but they are not ranked at the top because the weights between nodes are all small. when the weights are considered for calculation in weighted degree measure, Nodes $[1437, 1563]$ are the most important nodes because some of the edges have a significant weight of 100 or more.

6. In my opinion, the nodes with most degrees are the most critical for the information flow. They are $[1477, 1443, 1502, 1457]$. The connection of nodes to nodes does not mean that they are close and become good friends. But because they are able to have connections with the most nodes, information spreads through them the fastest.

1.2 Task1.b

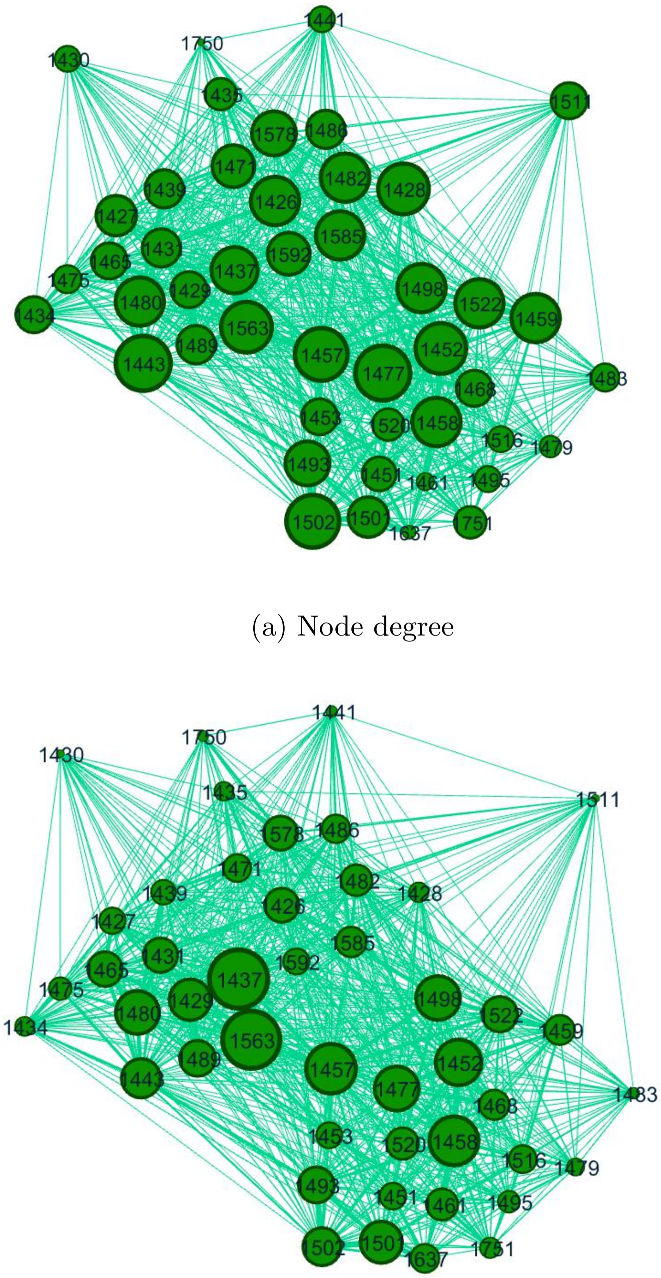
From the cluster figure in Fig 3, we can see there are 4 communities rendered in different colors, which represents different groups of students with different features. All students belong to 5A or 5B. The groups in Fig 3 shows students with same gender in the same class are more connected to others student. Moreover, you can even know which class they belongs to from the figure. **The green group and blue group represent males and females in class 5B, respectively. The purple group and orange group represent males and females in class 5A, respectively.**

1.3 Task1.c

1. I visualized all of graphs in the questions above via Gephi, the images are in the corresponding sections.

2. The gender of person 1637 in the class 5A are unkown. Through the observation in the communities graph above, I think 1637 are a girl in class 5A.

3. We filter the graph based on degree and keep those with most. From the updated graph in Fig 4, I guess the remaining person in the graph may have taken on certain administrative duties in the class. They interacted more with the students in their

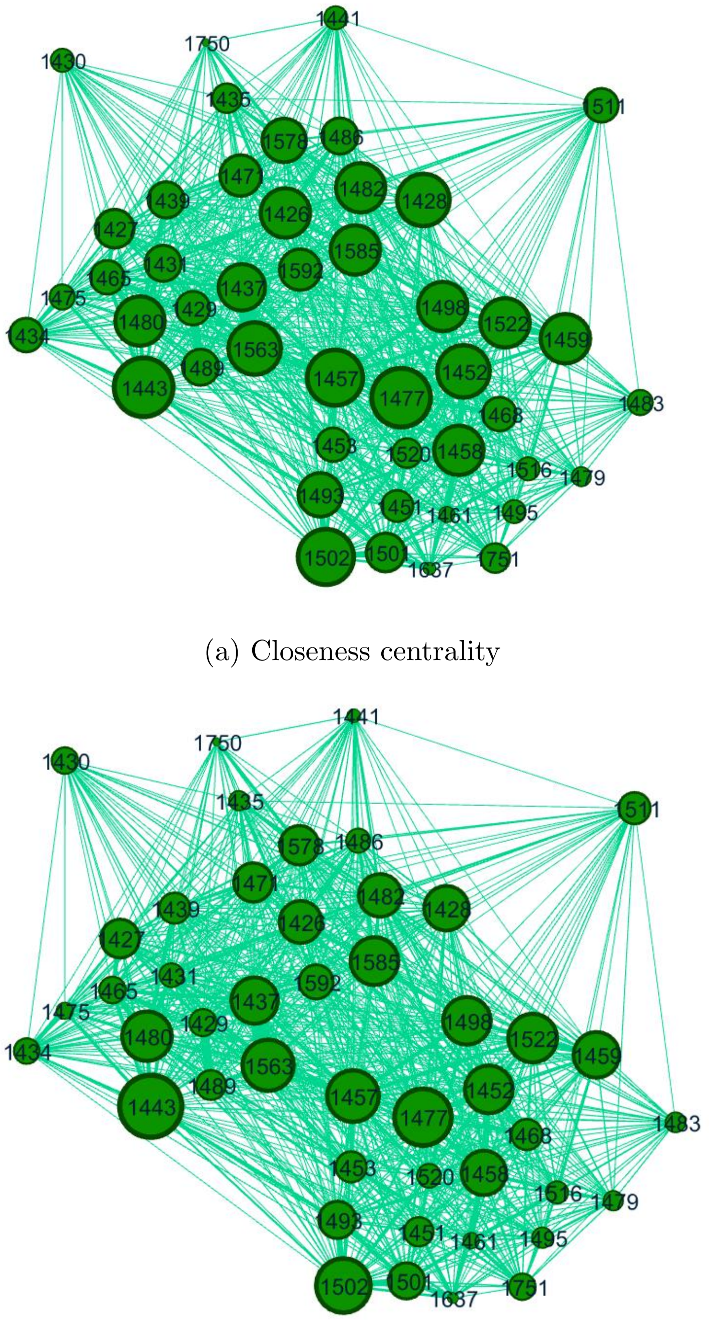


(a) Node degree

(b) Weighted Degree

Figure 1: Graph of classmates relationship

own class leading them to have more degrees. And they may also often communicate with each other about class work, interact more with other class administrators.



(a) Closeness centrality

(b) Betweenness centrality

Figure 2: Graph of classmates relationship

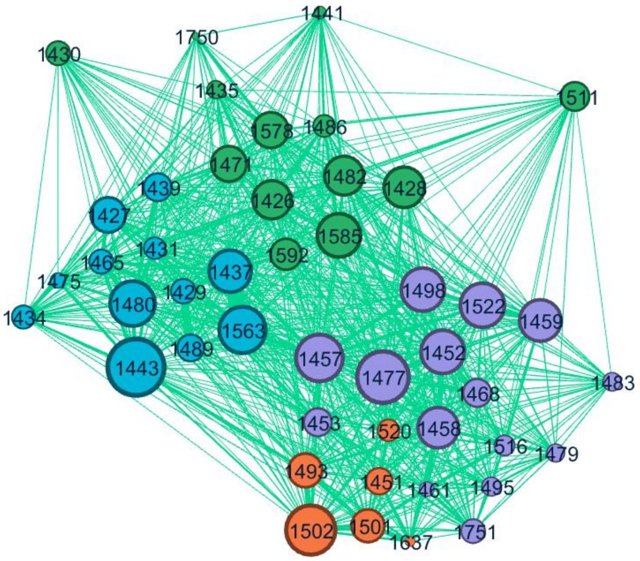


Figure 3: Communities in Graph

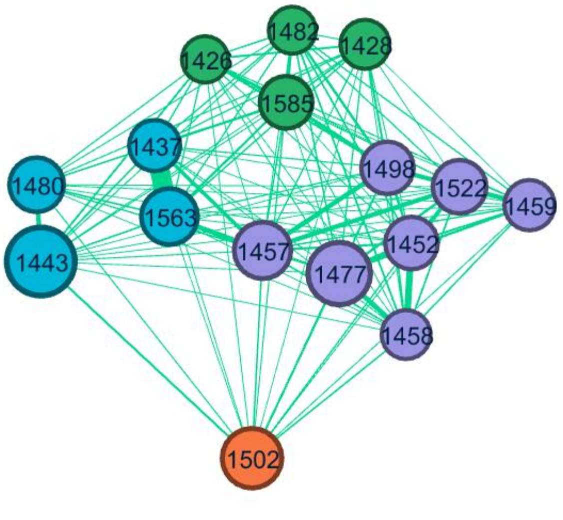


Figure 4: Filtered graph
(Degree between (39,43))

2 Task 2

2.1 Task2.a

According to formula in lecture 9, we can compute nearest neighbours of class M molecules using Udist and Mdist. M contains 3 molecules,so we average MCG distances on all molecules. The results are below:

Udist:

$$\begin{aligned} Udist(M,G_4) &= \frac{1}{3}(Udist(G_1,G_4)+Udist(G_2,G_4)+Udist(G_3,G_4)) \\ &= \frac{1}{3}[(1-\frac{|MCG(G_1,G_4)|}{|G_1|+|G_4|-MCG(G_1,G_4)})+Udist(G_2,G_4)+Udist(G_2,G_4)] \\ &= \frac{1}{3}[(1-\frac{9}{13+15-9})+0.71+0.56] \\ &= 0.60 \end{aligned} \tag{1}$$

$$Udist(M,G_5) = 0.80 \tag{2}$$

$$Udist(M,G_6) = 0.37 \tag{3}$$

Mdist:

$$\begin{aligned} Mdist(M,G_4) &= \frac{1}{3}(Mdist(G_1,G_4)+Mdist(G_2,G_4)+Mdist(G_3,G_4)) \\ &= \frac{1}{3}[(1-\frac{|MCG(G_1,G_4)|}{max\{|G_1|,|G_4|\}})+Mdist(G_2,G_4)+Mdist(G_3,G_4)] \\ &= \frac{1}{3}[(1-\frac{9}{15})+0.60+0.47] \\ &= 0.51 \end{aligned} \tag{4}$$

$$Udist(M,G_5) = 0.69 \tag{5}$$

$$Udist(M,G_6) = 0.35 \tag{6}$$

In both Udist and Mdist measures, The Ranking of nearest neighbours of M are G_6, G_4, G_5.

2.2 Task2.b

The maximum common subgraph G for all class M molecules shows in Fig 5. Then we can compute the confidence of graph class rules $G \to M$ and $\neg G \to \neg M$. Rule condition G means that subgraph G occurs in the graph and $\neg G$ means that it doesn't. we compute confidence as follows:

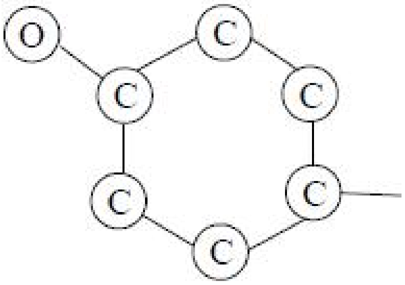


Figure 5: Maximum common subgraph G for all class M

$$Conf(G \to M) = \frac{Supp(G \cup M)}{Supp(G)} = \frac{3}{4} \tag{7}$$

$$Conf(\neg G \to \neg M) = \frac{Supp(\neg G \cup \neg M)}{Supp(\neg G)} = \frac{2}{2} = 1 \tag{8}$$

According to the current confidences, we can say if there are G in a molecule, this molecules most likely belongs to class M (monoamines). If there are no G in a molecule, this molecules won't belongs to class M (monoamines).

2.3 Task2.c

We would like to find most significant statistical associations between subgraphs and attributes in a database of compounds. Then we use the GraphApriori algorithm to find the frequent subgraphs in the database. Then we choose subgraphs that don't overlap too much, aiming to find more diverse subgraphs as features. We create new features based on remaining filtered subgraphs. So these compounds are represented by us in the form of different features as vectors. A compound may have more than one subgraph, i.e. more than one feature. Then We are able to use the previous text-based classification method to compute clustering on the data represented by frequent subgraphs features.

But it is a challenge to select frequent subgraphs with diversity. Only the richer and more different subgraphs are selected, the better results can be obtained in the subsequent feature expression. And because this is a graph-based method, it can only be implemented in small-sized graph databases, which consumes too much computing power and time when the number of compounds is too large or the structure is too complex.

3 Task3

The table below show the movie ratings by different users. Special value 0 means a missing rating.

	$m1$	$m2$	$m3$	$m4$	$m5$	$m6$
u1	3	1	2	2	0	2
u2	4	2	3	3	4	2
u3	4	1	3	3	2	5
u4	0	3	4	4	5	0
u5	2	5	5	0	3	3
u6	1	4	0	5	0	0

Table 1: Movie ratings (scale 1–5) by 6 users ($u1$–$u6$) on 6 movies ($m1$–$m6$).

3.1 Task3.a

After computation, the mean rating of each user are [**2 , 3 , 3 , 4 , 3.6 , 3.33**].

3.2 Task3.b

According to Pearson correlation in Aggarwal Equation 18.2, we can compute pairwise similarities between users shown in following table.

	$u1$	$u2$	$u3$	$u4$	$u5$	$u6$
u1	1.00	0.85	0.71	1.00	-0.82	-0.72
u2	0.85	1.00	0.00	1.00	-0.56	-0.72
u3	0.72	0.00	1.00	0.43	-0.59	-0.58
u4	1.00	1.00	0.43	1.00	-0.87	1.00
u5	-0.82	-0.56	-0.59	-0.87	1.00	1.00
u6	-0.72	-0.72	-0.58	1.00	1.00	1.00

Table 2: Pearson correlations between 6 users

3.3 Task3.c

```
def predict(user,movie,NN,avg,sim):
    movies=np.array(records.iloc[:,movie])

    m1,m2=movies[NN[0]]-avg[NN[0]],movies[NN[1]] -avg[NN[1]]
    s1,s2=sim.iloc[user,NN[0]],sim.iloc[user,NN[1]]

    rating=(s1*m1+s2*m2)/(s1+s2)+avg[user]
    return round(rating,2), rating>avg[user]
```

Based on the average ratings and similarities obtained above, now we can predict the missing ratings using two nearest neighbours ($K = 2$) with similarity r > 0.5. The predicted rating are in following table. T means the predicted rating is bigger than user's average rating, we will recommend this movie to current user.

	$m1$	$m2$	$m3$	$m4$	$m5$	$m6$
u1	3	1	2	2	**3.0(T)**	2
u2	4	2	3	3	4	2
u3	4	1	3	3	2	5
u4	**5.0(T)**	3	4	4	5	**3.5(F)**
u5	2	5	5	**0**	3	3
u6	1	4	**4.03(T)**	5	**3.53(T)**	**0**

Table 3: Updated Movie ratings

We need to know not all missing ratings can be predicted in current conditions. we can't predict the ratings on m4 by u5, m6 by u6. Because when we are finding their nearest neighbours, we fail to find enough 2 NNs with similarity $r > 0.5$, and the NNs already rated the movie we are predicting.

3.4 Task3.d

Now, we consider the item-based way of predicting the missing ratings of movies with adjusted cosine similarity. The formula shows below,

$$Cosine(\bar{U}, \bar{V}) = \frac{\sum_{i=1}^{s} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{s} u_i^2} \cdot \sqrt{\sum_{i=1}^{s} v_i^2}} \tag{9}$$

$U = (u_1, ..., u_s)$ and $V = (v_1, ..., v_s)$ of a pair of items' normalized ratings.

The adjusted cosine similarity of m3 and m4 are 1,because the movies ratings are same if not consider which users. So it means, with cosine similarity the model will give very similar predictions for different movies for different users, because they have too many similar watched records. This makes the differences between different users not reflected, making the whole system more and more homogeneous.

An alternative item-based solution is increasing some randomness when facing too much similarity. When the recommended movies become more and more similar, we can recommend some random movies in addition to the candidate movies. Through users' feedback on random movies, i.e. whether they choose to watch or not or how much they rate, we add diversity to our recommendation system and the recommended results are more personalized.