Data Analytics Week 4 Assignment

202011431 산업공학과 차승현

Network Visualization

건국대학교

1) Two node matrix(6x8)를 정의한다.

걸그룹 우주소녀의 멤버 수빈, 다영, 루다, 연정, 다원, 보나는 본인이 좋아하는 타이틀곡에 1점부터 5점까지 투표하였다. 이를 활용하여 멤버간 선호하는 컨셉을 분류하여 유닛 활동을 시작하려고 한다. 어느 멤버가 선호하는 무대와 곡의 컨셉이 비슷할지 알아보자.

	부탁해	HAPPY	비밀이야	LaLaLove	이루리	Butterfly	Unnatural	너에게닿기를
수빈	3	2	4	1	1	1	5	1
다영	2	5	3	2	5	2	5	4
루다	1	1	2	1	4	5	1	2
연정	1	1	3	4	2	4	3	1
다원	4	1	1	4	5	3	2	2
보나	1	4	2	3	1	1	1	2

2) Similarity 지표 중 1개를 선택하여 One mode matrix로 변환한다.

Similarity 지표 중, Cosine Similarity를 이용하여 One mode matrix로 변환하고자 한다.

$$similarilty(A,B) = \cos(\theta) = \frac{A \cdot B}{|A||B|}$$

$$= \frac{\sum_{i=-1}^{n} A_i B_i}{\sqrt{\sum_{i=-1}^{n} {A_i}^2}\sqrt{\sum_{i=-1}^{n} {B_i}^2}}$$

cosine similarity results

s(수빈,다영)	s(수빈,루다)	s(수빈,연정)	s(수빈,다원)	s(수빈,보나)
0.818881251	0.54108977	0.747854362	0.632598995	0.669185982
s(다영,루다)	s(다영,연정)	s(다영,다원)	s(다영,보나)	s(루다,연정)
0.752802491	0.763454916	0.791237847	0.838849184	0.818723302
s(루다,다원)	s(루다,보나)	s(연정,다원)	s(연정,보나)	s(다원,보나)
0.819329539	0.587130378	0.820445119	0.740356046	0.678882858

One mode matrix						
	수빈	다영	루다	연정	다원	보나
수빈	0	0. 818881251	0.54108977	0.747854362	0.632598995	0.669185982
다영	0.818881251	0	0.752802491	0.763454916	0.791237847	0.838849184
루다	0.54108977	0. 752802491	0	0.818723302	0.819329539	0.587130378
연정	0.747854362	0. 763454916	0.818723302	0	0.820445119	0.740356046
다원	0.632598995	0. 791237847	0.819329539	0.820445119	0	0.678882858
보나	0.669185982	0. 838849184	0.587130378	0.740356046	0.678882858	0

코사인 유사도의 범위는 0점 이상, 1점 이하이며 1에 가까울수록 유사도가 높음을 의미한다. (본인에 대한 코사인 유사도는 0으로 처리한다.)

3) Network visualization

Python을 사용하여 Network Visualization을 시행하였다.

사용한 모듈(library): pandas, network, matplotlib, warnings

```
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
%matplotlib inline
import warnings; warnings.simplefilter('ignore')
import matplotlib.font_manager as fm
from matplotlib import rc
font_name = fm.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
rc('font', family=font_name)
```

```
df = pd.read_csv(r'C:₩Users₩moon_₩OneDrive - konkuk.ac.kr₩바탕 화
df
```

	수빈	다영	루다	연정	다원	보나
수빈	0.000000	0.818881	0.541090	0.747854	0.632599	0.669186
다영	0.818881	0.000000	0.752802	0.763455	0.791238	0.838849
루다	0.541090	0.752802	0.000000	0.818723	0.819330	0.587130
연정	0.747854	0.763455	0.818723	0.000000	0.820445	0.740356
다원	0.632599	0.791238	0.819330	0.820445	0.000000	0.678883
보나	0.669186	0.838849	0.587130	0.740356	0.678883	0.000000

One mode matrix를 csv형태로 저장하여 data frame의 형태로 불러와 df라는 변수명에 저장하였다.

```
member = df.columns.tolist()
member
```

```
['수빈', '다영', '루다', '연정', '다원', '보나']
```

```
G_weighted = nx.Graph()

G_weighted.add_edge(member[0],  member[1], weight=df.iloc[0, 1])
G_weighted.add_edge(member[0],  member[2], weight=df.iloc[0, 2])
G_weighted.add_edge(member[0],  member[3], weight=df.iloc[0, 3])
G_weighted.add_edge(member[0],  member[4], weight=df.iloc[0, 4])
G_weighted.add_edge(member[0],  member[5], weight=df.iloc[0, 5])

G_weighted.add_edge(member[1],  member[2], weight=df.iloc[1, 2])
G_weighted.add_edge(member[1],  member[3], weight=df.iloc[1, 3])
G_weighted.add_edge(member[1],  member[4], weight=df.iloc[1, 4])
G_weighted.add_edge(member[1],  member[5], weight=df.iloc[1, 5])

G_weighted.add_edge(member[2],  member[3], weight=df.iloc[2, 3])
G_weighted.add_edge(member[2],  member[4], weight=df.iloc[2, 4])
G_weighted.add_edge(member[2],  member[5], weight=df.iloc[2, 5])

G_weighted.add_edge(member[3],  member[4], weight=df.iloc[3, 4])
G_weighted.add_edge(member[3],  member[5], weight=df.iloc[3, 5])

G_weighted.add_edge(member[4],  member[5], weight=df.iloc[4, 5])
```

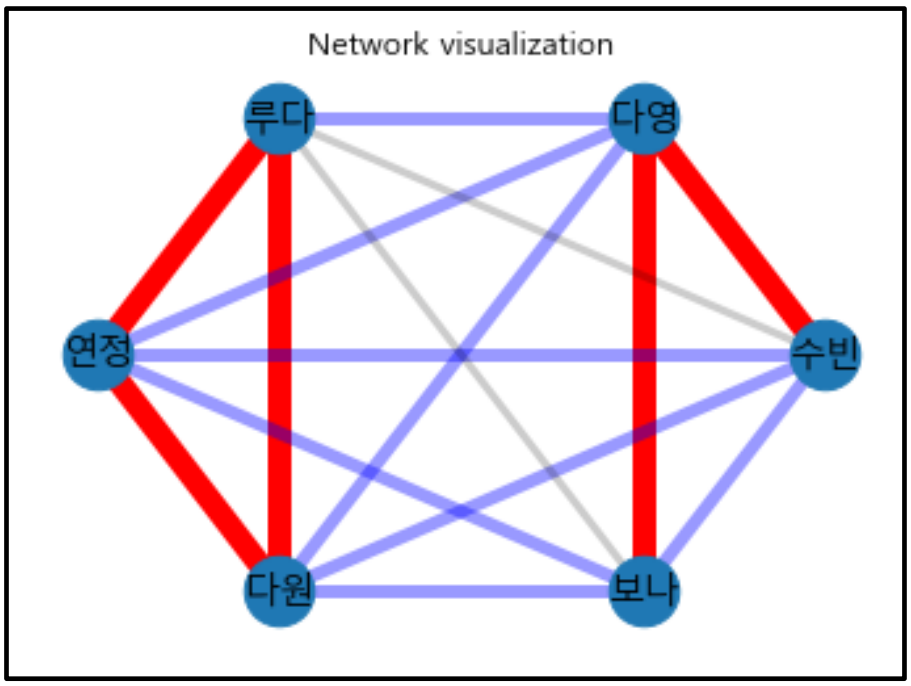
데이터프레임의 컬럼명(멤버의 이름)을 list의 형태로 저장한 후 networkx모듈을 사용하여 weight와 from 값, target값을 부여하였다.

```
e_1 = [(u, v) for (u, v, d) in G_weighted.edges(data=True) if d['weight'] >= 0.8]
e_2 = [(u, v) for (u, v, d) in G_weighted.edges(data=True) if ((d['weight'] < 0.8) & (d['weight'] >= 0.6))]
e_3 = [(u, v) for (u, v, d) in G_weighted.edges(data=True) if ((d['weight'] < 0.6) & (d['weight'] >= 0.4))]

pos = nx.circular_layout(G_weighted)

#nodes
nx.draw_networkx_nodes(G_weighted, pos, node_size=700)

#edges
nx.draw_networkx_edges(G_weighted, pos, edgelist=e_1, edge_color='r', width=9)
nx.draw_networkx_edges(G_weighted, pos, edgelist=e_2, alpha=0.4, edge_color='b', width=5)
nx.draw_networkx_edges(G_weighted, pos, edgelist=e_3, alpha=0.2, width=3)


#labels
nx.draw_networkx_labels(G_weighted, pos,font_family=font_name, font_size=14)

plt.title('Network visualization')
plt.axis('off')
plt.show();
```

이후 가중치가 0.8 이상인 수치를 e_1으로, 0.6 이상이며 0.8 미만인 수치를 e_2로, 0.4 이상이며 0.6 미만인 수치를 e_3 총 세가지의 범주로 분류하였다.

e_1에 해당하는 network는 붉은 색과 굵은 width로 표시하였으며, e_2에 해당하는 network는 푸른 색과 불투명도를 0.4로 설정한 후 중간 크기의 width, e_3에 해당하는 network는 검정 색과 불투명도를 0.2로 설정하고 얇은 width를 선택하여 세 분류의 network를 한 눈에 들어오기 쉽게 표시하였다.

(같은 색으로 하니 불투명도를 달리해도 시각적으로 유사하다고 생각이 들었기에 색상, 불투명도, width 모든 값에 변화를 주었다.)



Visualization result