SMM638 Final Course Project Description

Simone Santoni

2024-11-25

Synopsis This notebook illustrates the final course project for the SMM638 course. The project is based on the analysis of a dataset regarding the friendship network and genre preferences of Deezer users. The first section of the notebook provides an overview of the project, the second section describes the data, the third section outlines the problem, and the fourth section describes the submission package.

1 Overview

Like other streaming platforms, Deezer contains a wealth of digital traces, which can be used to analyze user behavior, and, therefore, to create or refine products and improve business model execution (e.g., by adopting a recommendation system that help a platform business better engage with audiences).

Network analysis methods and tools play a key role when it comes to analyzing digital-traces like the one we have in the Deezer dataset. Particularly, network analysis offers an effective framework within which to appreciate the similarity between entities — being users or the genres they may favorite — and, possibly, cluster these entities into homogenous groups — e.g., users that share similar music genres or genres that are liked by the same users. Let us consider a two-mode (or bipartite) network X, where N users are connected to K genres via the 'like' relationship:

$$X = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nk} \end{bmatrix}$$

where a_{ij} is the 'like' relationship between user i and genre j. The matrix X can be used to create a user-user network Y (N x N) and a genre-genre network Z (K x K):

$$Y = X \cdot X^T$$

$$Z = X^T \cdot X$$

The user-user network Y is a one-mode, non-directed, weighted graph where nodes are users and edges are mutual likes, i.e., the counts of music genres that users i and j share. The genre-genre network Z is a one-mode, non-directed, weighted graph where nodes are genres and edges are mutual likers, i.e., the counts of users that like both genres i and j. Consider the following example of 'like' network, including five users and three music genres:

$$X = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

The user-user network Y ($X \cdot X^T$) is:

$$Y = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 2 \\ 1 & 0 & 1 & 2 & 2 \end{bmatrix}$$

whereas the genre-genre network Z ($X^T \cdot X$) is:

$$Z = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 3 \\ 0 & 3 & 4 \end{bmatrix}$$

Both Y and Z can be further analyzed using network analysis tools — e.g., block-modeling — or conventional statistical tools — e.g., cluster analysis – to identify homogenous groups of entities (users and genres for Y and Z, respectively).

2 Data

The data for the final course project is stored in the `data/deezer_clean_data` directory of GitHub repository of SMM638. The data, which were gathered for a network science project,[1] are also available in the website of Stanford Network Analysis Project.

Below are some key aspects about the data:

- The data were scraped from Deezer in November 2017
- `**_edges.csv` represent friendships networks of users from 3 European countries, that is, Croatia, Hungary, and Romania. Nodes represent the users and edges are the mutual friendships[2]
- `**_genres.json` contain the genre preferences of users — each key is a user identifier, the genres loved are given as lists. Genre notations are consistent across users. In each dataset users could like 84 distinct genres. Liked genre lists were compiled based on the liked song lists

[1] Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. 2018. GEMSEC: Graph Embedding with Self Clustering. arXiv preprint arXiv:1802.03997.

2.1 Friendship networks

For illustrative purposes, let us inspect the friendship network for the case of Croatia. First, we load `Pandas` and `NetworkX`, then we load the data:

```
# load modules
import pandas as pd
import networkx as nx
# load data
fr = pd.read_csv('../data/deezer_clean_data/HR_edges.csv')
# data preview
fr.head()
```

	node_1	node_2
0	0	4076
1	0	29861
2	0	53717
3	0	23820
4	0	39945

The data preview shows that the friendship network for Croatia is a list of edges, where each edge is a pair of user identifiers. The data can be used to create a network object using `NetworkX`:

```
fr_g = nx.from_pandas_edgelist(fr, source='node_1', target='node_2')
fr_g?
```

Using code introspection, it is possible to see that the network object `fr_g` is a NetworkX object of type `Graph` and that it has 54,573 nodes and 498,202 edges. To familiarize with the data, we test if `fr_g` is connected:

```
nx.is_connected(fr_g)
```

```
True
```

Then, we consider the degree distribution of the network:

```
# import further modules
import numpy as np
from matplotlib import pyplot as plt
from collections import Counter
```

[2]The researchers who collected the data say that they have "*... reindexed the nodes in order to achieve a certain level of anonymity.*"

```
# compute node degree
dd = Counter(dict(fr_g.degree()).values())
# plot the degree distribution
fig = plt.figure(figsize=(4, 3))
ax = fig.add_subplot(111)
ax.scatter(dd.keys(), dd.values(), color="limegreen", alpha=0.15)
ax.set_yscale("log")
ax.set_xscale("log")
ax.set_xlabel("Log(Degree)")
ax.set_ylabel("Log(Counts of nodes)")
ax.grid(True, ls="--")
plt.show()
```

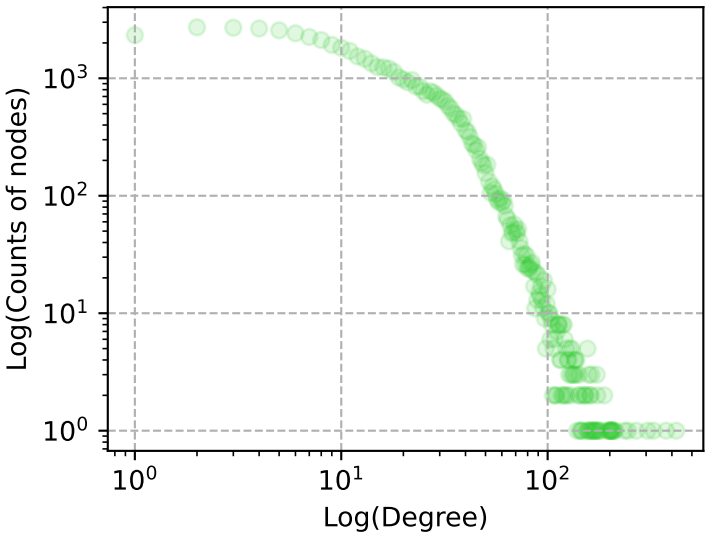

Figure 1: Degree distribution of the friendship network for Croatia

It is self-explanatory that the degree distribution of the friendship network for Croatia is right-skewed, which is a common feature of social networks. We can try to getter a better understanding of the network — including the presence and locatio of 'hub' users — by visualizing it. Since the network is large, we may benefit from using the visualization capabilities of `graph-tool`, a Python API wrapping around C++ code, a more efficient alternative to pure Python `NetworkX`:

```
# import further module
from graph_tool.all import *
# iterate over the Pandas DataFrame to create the graph and edges to it
edges = [(str(u), str(v)) for u, v in fr[['node_1', 'node_2']].values]
fer_gt = Graph(edges, hashed=True, directed=False)
# plot the network
#       graph_tool.draw.graph_draw(fer_gt,       output_size=(500,       500),
output="fer_gt.png")
# load image
```

```
from IPython.display import Image
Image(filename='fer_gt.png')
```

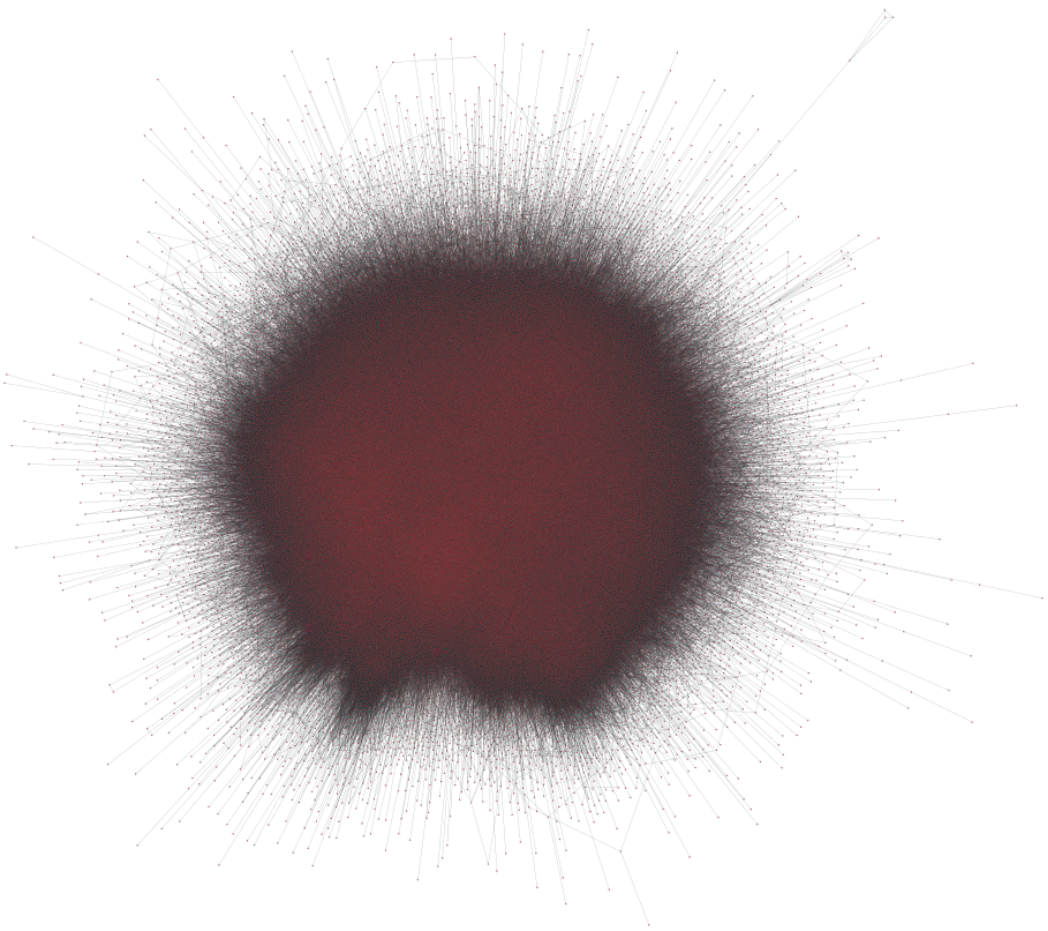

Figure 2: Friendship network for Croatia (N=54,573)

It is worth noticing the friendship network presents a periphery of users with low degree and, plausibly, a core of users with high degree. However, the figure does not provide a clear picture of the core of the network, which deserves further investigation.

2.2 Music genre preferences

Building on the previous sub-section, we consider the preferences of users as per `HR_genres.json` files. These files are JSON files, which can be loaded using the `json` module:

```
import json
with open('../data/deezer_clean_data/HR_genres.json', 'r') as f:
```

```
    pr_json = json.load(f)
pr_json["11542"]
```

```
['Indie Rock',
 'Indie Pop/Folk',
 'International Pop',
 'Rap/Hip Hop',
 'Pop',
 'Rock',
 'Indie Pop',
 'Alternative']
```

At this stage, we have a dictionary where each key is a user identifier and the corresponding value is a list of genres that the user likes. For example, above is the list of music genres that user `11542` likes. We can convert the dictionary into a `Pandas` DataFrame drawing upon Pandas' `json_normalize` function:

```
pr = pd.json_normalize(pr_json).T
pr.rename({0: 'genres'}, axis=1, inplace=True)
pr.head()
```

	genres
13357	[Pop]
11542	[Indie Rock, Indie Pop/Folk, International Pop…
11543	[Dance, Pop, Rock]
11540	[International Pop, Jazz, Pop]
11541	[Rap/Hip Hop]

The data preview shows that the DataFrame `pr` has a single column, `genres`, which contains lists of genres that users like. To make the data more amenable to analysis, we can explode the lists of genres into separate rows drawing upon Pandas' `explode` function:

```
pr = pr.explode('genres')
pr.reset_index(inplace=True)
pr.rename({'index': 'user_id'}, axis=1, inplace=True)
pr.head()
```

	user_id	genres
0	13357	Pop
1	11542	Indie Rock

	user_id	genres
2	11542	Indie Pop/Folk
3	11542	International Pop
4	11542	Rap/Hip Hop

For illustrative purposes, we can consider the distribution of genres liked by users in the dataset:

```
genres = Counter(pr.groupby('genres').size())
fig = plt.figure(figsize=(6, 3))
ax = fig.add_subplot(111)
ax.hist(genres.keys(), color="magenta", alpha=0.5)
ax.set_xticklabels(["{:,}".format(int(x)) for x in ax.get_xticks()])
ax.set_xlabel("Degree -- number of likers")
ax.set_ylabel("Counts of music genres")
ax.grid(True, ls="--")
plt.show()
```

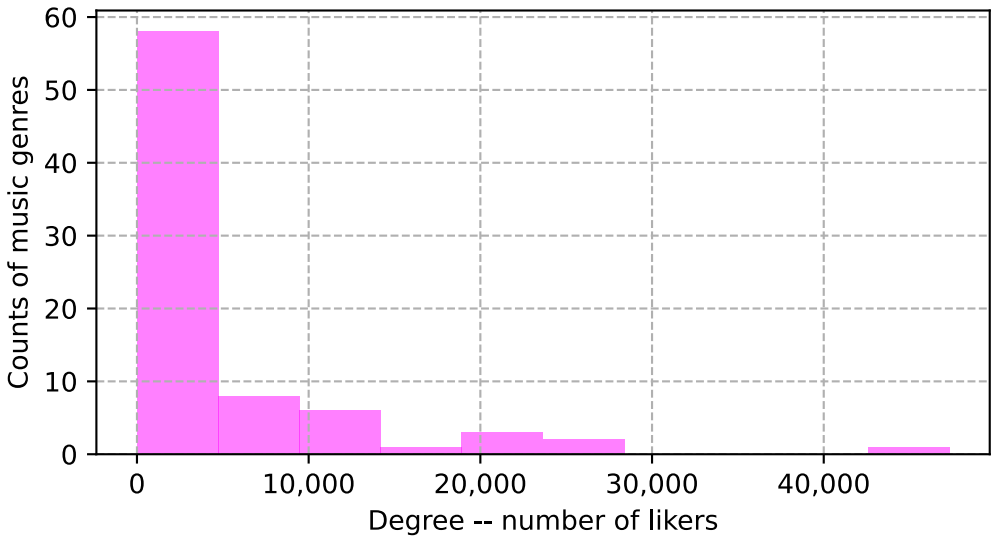

Figure 3: Degree distribution for music genres in the 'Croatia' dataset

The degree distribution of music genres liked by users is right-skewed, meaning there are few genres that are liked by many users and many genres that are liked by few users.

3 Problem description

Your employer is a consultancy firm that has been hired by 'Sonic,' a major music label, to better understand the categories that compose the music market. The artists and repertoire roles at Sonic have struggled to make sense of the association between the 'music genre' tags (e.g., 'International Pop') in Deezer and similar services. Some think that these tags are sometimes redundant. In other circumstances, they are unclear or meaningless. Therefore, it is hard for Sonic to see clear targets

in the market, and, consequently, correctly position new albums and musicians against consumer preferences. Sonic wants a map of the categories that form the music market. This map must be based on some digital traces – i.e., behavioral data – , easy to interpret, and well-grounded in demonstrable data patterns.

To help Sonic address its business problem, you have been asked to analyze the 'Croatia' dataset, briefly described in the previous section. You are expected to use network analytic methods and tools to:

1. Assess the similarity between Deezer music genres
2. Identify homogenous groups of Deezer music genres
3. Highlight how social ties among users influence the similarity between music genres

4 Submission package

The submission package consists of:

- A report that includes:
  - The description of the workflow you have followed to address the problem (300 words MAX)
  - The results of your analysis, comprising text (300 words MAX) and exhibits (five among figures and tables MAX)
  - The interpretation of the results in light of the business problem (300 words MAX)
- The computer code that allows me to fully reproduce your charts (being, R, Python, Julia, Rust, C++, Java, etc.). The code should be well-commented and easy to read. Non-reproducible exhibits will not be graded.