Anime Recommendation System

Based on graph clustering

Jinil Chandarana
School of engineering and applied science
Ahmedabad University
Ahmedabad Gujarat India
jinil.c@ahduni.edu.in

Keyur Nagar
School of engineering and applied science
Ahmedabad University
Ahmedabad Gujarat India
keyur.n@ahduni.edu.in

Maulikkumar Bhalani
School of engineering and applied science
Ahmedabad University
Ahmedabad Gujarat India
maullikkumar.b@ahduni.edu.in

ABSTRACT

With the increase in the online entertainment platforms popularity, the popularity of animation art in Japan is gaining more and more acknowledgment. Every year, many new animes are released with their unique character characteristics, tales, plots, and fantastic visual animation. Therefore, a recommendation personalized for a particular user is needed. Many suggestions are based on collaborative or content-based filtering, taking the overall rating of anime. However, we might improve this recommendation based on the social network analysis. viewers might be influenced by friends or social media regarding their preferences, and these preferences are based on the similarity of their linking. Therefore, the type of anime, a social network analysis of the user sentiment/linking, and the anime categories can help recommend anime to viewers. In this report, we propose two clustering approach based on the Louvain algorithm to find the clusters of similar users linking similar anime for a recommendation using similarity matrices and bipartite graph. This avoids the suggestion of such anime which might have a good rating but a particular user might not like.

KEYWORDS

Recommendation, Anime, Louvain, Clustering, bipartite, modularity, Jaccard.

1 Introduction

With the worldwide rise in popularity, Japan has gained more than USD 18B in revenue with this entertainment art [1]. The anime industry is multiplied by an average of 2 trillion-yen market size from 2013 to 2018. In 2018, the overseas market exceeded 1 trillion yen or around 46.3% of the overall Japanese animation market size [2]. With the increase in liking anime and the growth of the variety of animes, the user might get overwhelmed searching for which anime is the best for him to watch.

A straightforward approach is to develop a recommendation system through which users can get the list of best preferences without going through every anime. The primary methods which are out there are based on two approaches. 1. Content-based and two collaborative filtering [3]. The CB filtering considers the similarity of the features for a recommendation of the product that is similar to the product a user has liked. On the other hand, the CF considers the user's linking based on the liking of the user having a similar taste in the product. As the recommendation of CF is based on the user's historical data, it has more efficiency. However, it does not deal with the fact that the user's taste can change over time, and he may like to try a different product. Also, introducing something new will be discarded for a recommendation. Therefore, a new hybrid approach is needed to solve these downfalls and take the best outcome from both systems [4].

This report shows the implementation of such hybrid approaches. The first approach is based on networking similar anime into various clusters depending on the common genres they share. By providing the liked anime and its rating by the user, this approach returns the list of most similar anime having the ratings around the anime liked by the user. The second approach considers the user rating on the anime and forms the cluster of the similar users and the similar anime selected by those users. Thus, a user can be in different groups connected to other anime clusters, taking into account the sentimental linking and the similarity of the anime. By these approaches, we aim to answer the following questions:

1. **Is the similarity of anime based on genres enough for a recommendation?**
2. **Will users get a new taste of anime when bored with one?**
3. **Is merely considering rating in the second approach appropriate?**
4. **Which system gives better results?**

In the proposed approaches, we perform network analysis based on the matrix provided to the Louvain algorithm, which forms the modules of nodes using the maximizing modularity approach. The detailed formulation and explanation of each are presented further in the report.

2 Dataset and finding

MyAnimeList is one of the most extensive ratings/ranking websites for animes. The dataset was obtained from Kaggle and is compiled from information gathered for each anime on MyAnimeList. The

dataset consists of two major sections: information about the anime and statistics specific to MyAnimeList, such as how many people are watching that particular anime, how many people have rated it, and the rating of each user on that anime, and so forth. The first part contains information about the title, score, genre, type of anime, episodes, popularity, etc. The second part includes the rating of any particular anime by any users. We have cleaned the data, and the dataset contains information regarding 17562 animes with all other information, but for the demonstration, we have only considered 76 anime and its rating [5].

This dataset contains 109 million rows, 17.562 different animes, and 325.772 different users. These are the columns of our dataset.

user_id: non-identifiable randomly generated user id.
anime_id: - MyAnimelist ID of the anime that this user has rated.
Rating: rating that this user has assigned.
Score: score between 1 to 10 given by the user. 0 if the user didn't assign a score
Watching_status: state ID from this anime in the anime list of this user.
Watched_episodes: numbers of episodes watched by the user.
Genres: comma-separated list of genres for this anime. (e.g., Action, Adventure, Comedy, Drama, Sci-Fi, Space)
Type: TV, movie, OVA, etc.
Episodes: number of chapters.
Rating: rating for any particular anime
Ranked: position based on the score [5].

As a result of our analysis, we found that all animes in the dataset averaged 6.74 out of 10, just passing the passing grade. There are also 38,927 anime viewers per episode, close to 40,000. To find the types of animes in this dataset, we have analyzed the Kaggle dataset and found that there are four types of anime, TV, OVA, movie, and music. Moreover, we found that TV shows and OVAs account for more than 80% of all anime in the dataset, and their percentages are pretty similar. Among the movie types, 14% are movies, and music types fill the remaining 2% [5].

As with other arts, anime has many genres, and this dataset contains a total of 41 genres. Popular genres include comedy, action, adventure, romance, a slice of life, etc. Each year, the production of the above genres can be around 30-60, while we can only see less than ten productions for the less popular genres. Comedy has the most significant production per year out of these ten genres, while the supernatural has the most negligible production. And by looking at graphs, we found out that the average anime audience for comedy shows is half that of anime for schools. For anime producers, it is best to make school animes, romance animes, or supernatural animes.

3 Problem formulation

A. Similarity matrix:

TABLE 1

	e1	e2	e3	e4
e1	1	0.75	0	0
e2	0.75	1	0	0
e3	0	0	1	0.4
e4	0	0	0.4	1

The recommendation is based on the fact that a particular viewer who likes certain content is now willing to see new content similar to the previous one. Thus, it is logical to understand and find the similarity between the two elements. In statistics, the similarity between the two objects by quantifying similar traits between the elements using a real-valued function [6]. Over a large range of the similarity, the values can be 0 or negative with different objects. This similarity is represented in the form of a matrix in the given table illustration, table 1. The Aij represents the similarity between row i and column j. Aij = 0 if there is no similarity at all, Aij = 1 if i=j. The elements are represented in rows and columns, and the entries indicate the similarity between those elements e1, e2, e3, and e4. Analyzing the matrix, we can observe that the elements e1 and e2 are more similar than elements e3 and e4. It is worth noticing that the similarity matrix is the inverse of the distance matrix in some sense.
Clustering refers to grouping the data points in a data set into several groups such that the data points in a particular group have similar properties. A similarity matrix is used in several clustering algorithms, such as the Leiden algorithm, Louvain algorithm, spectral clustering, and so on, to cluster or form a community of similar elements.

B. Modularity;

While dealing with the graphs and graph theories, we aim to understand the representation of connections of the nodes and interpret a new helpful insight. Say, while coping with the graphs, we wish to determine whether the structure of a graph says something about itself. A complex graph will be challenging to understand. The modularity in this situation determines whether there exists a natural division of the nodes that forms a community or cluster which are nonoverlapping. With this, the structure of a graph can be simplified and can be understandable.
Modularity quantifies the strength of a graph to form divisions or modules it can naturally form. A graph network would have a sparse connection between vertices of different modules, and modules with similar vertices will have densely connected modules. With its wide variety of use, modularity maximization forms better results with accuracy. It is a fraction of the difference of the expected edges between the nodes falling into a group and the fraction of them within the group. From the above mathematical definition, the partition of the community for modularity can be defined as

$$P = \frac{1}{T_{..}} \sum_{ij} \left[T_{ij} - \frac{T_{i.} T_{.j}}{T_{..}} \right] \delta_{c_i c_j}$$

TiTj/T.. Is the expected fraction of node j and i are connected. Ti and Tj are inside and outside the strength of the nodes. Total strength is given by T.. is total. delta is Kronecker's delta. Tij is the adjacency matrix [7].

C. Jaccard similarity

Jaccard similarity is the measure used to find the similarity between two sets of data. We have used Jaccard similarity to find the coefficient for commonality between two animes with reference to Genres [8].

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Where J = Jaccard distance
A = set 1
B = set 2

D. Louvain Algorithm

The Louvain Approach is an unsupervised algorithm for detecting a network's community [8].

This approach uses the idea of modularity. This approach is divided into two phases:

1. Modularity optimization
2. Community aggregation

The algorithm is initialized with a weighted graph of N nodes (i.e. Adjacency matrix). In the first phase, different communities will be assigned to each node. At this stage, it checks the neighbor around a particular node and evaluates the modularity by eliminating certain nodes from the current community. The node will be placed at the other community if and only if the modularity gain is maximized. This process will continue running until no improvement is visible. If the local maxima of modularity is found, the first phase will terminate. In the second phase, the algorithm builds the network based on the community derived from the first phase. These operations will continue until the highest level of modularity is achieved.

4 Proposed Methods

A. Louvain clustering (Adjacency Matrix)

If a user likes a particular anime, eventually, they want the genres connected to that Anime. If we somehow introduce the new Anime with the same set of genres, the possibility of liking the Anime will be high. So, anime recommendations rely on a set of genres with previously liked Anime. This section includes Genre-based similarities among Animes. Using this approach, We have classified different Anime based on their genres. We determined the Anime-Anime similarity based on Genres. Then we applied Louvain clustering for the classification of Animes. Suppose we have Anime {A1, A2, A3, ……., An}, for each Anime Ai, there are some genres {G1, G2, G3…., G_k} associated with them.

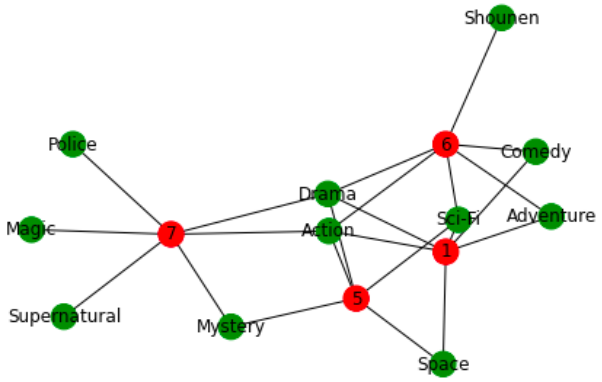

Figure 1: Scuffles bipartite graph of anime and genres.

The graph given above is the undirected graph. There are two types of nodes in the graph: red and green nodes. The red nodes represent the anime, and the green nodes represent the Genres associated with anime. Looking at the above graph, we can infer that the same genre is connected with multiple anime, and each anime is connected with various genres. The similarity between the two Anime A_i and A_j can be founded by

$$\text{Jaccard Similarity} = \frac{(\text{set of genres of Ai}) \cap (\text{set of genres of Aj})}{(\text{set of genres of Ai}) \cup (\text{set of genres of Aj})}$$

The n x n adjacency matrix where rows and columns represent Anime can be found. Each entry Aij in the adjacency matrix represents the Jaccard similarity between A_i and A_j. We can form the complete graph of n vertices and n (n-1)2 edges using this adjacency matrix. Where vertices denote Anime and weights of edges denote Jaccard similarity between two vertices.

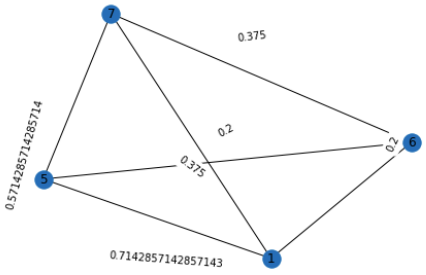

Figure 2: Anime graph with Jaccard as weight

As shown in the above graph, there are four anime with id 1, 5, 6, and 7. Anime 1-5, 1-7, and 1-6 have Jaccard similarity values of 0.7142, 0.2, and 0.2, respectively. We may infer from this graph that if a person likes anime 1, there's a good chance he'll enjoy anime five as well, due to the intense closeness of genres.

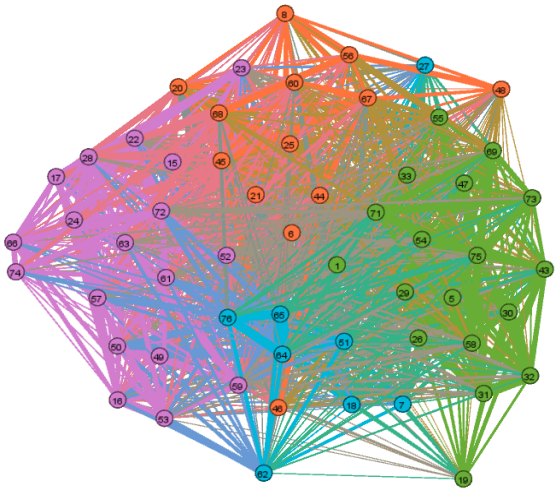

Figure 3: Anime similarity graph from using Louvain (done in Gephi for better visualization)

B. Louvain Coclustering for bipartite graph

The clustering based on the genres of the anime is not enough as a pair of anime might have a very similar set of genres, but the quality of the animes might be different. Furthermore, an anime has various features, such as its contents, direction, plot, and many more, making it better. This leads us to consider the aspect of the viewer's liking. A viewer might like particular anime, but another user may not pick that anime. Also, both the users might like a specific anime regardless of the first situation. Therefore, a hybrid recommendation system is needed which considers the similarity of the anime and the similarity of the user. A co-clustering approach of a bipartite graph using a bi-adjacency matrix embedded in the Louvain algorithm takes care of this.

bi-adjacency matrix.

The bipartite graph with a set of vertices represents the viewers and another set of vertices represents the animes. This bipartite graph is represented using a similarity matrix called a bi-adjacent matrix where the 2 types of vertices are in the form of rows and columns. Thus, all the entries represent the connection of all the row nodes to the column nodes. Mind, as the number of viewers (row entries), will never be equal to the number of anime (column entries), the bi-adjacency matrix will be an mxn rectangular matrix. For graph G with a set of N and M type nodes, table 2 represents an NxM bi-adjacency matrix. Bij is the number of edges between row i and column j.

TABLE 2

	e1	e2	e3	e4
e1	0	1	0	0
e2	1	0	0	0
e3	0	0	0	1
e4	0	0	1	0

However, as we are interested in finding the similarity between the users as well as the anime, we take the entire Bij as the rating given by the viewer i to the anime j and then form a co-cluster of the matrix using MATLAB library with brings the shuffled similar rows as well as columns near each other. The reason for doing this is that by forming a bi-coclustering of the entries of the matrix, the set of rows giving a similar rating to the anime will be clustered together and thus we will have a cluster of viewers with the same liking and clusters of anime which those cluster of viewers given high rating.

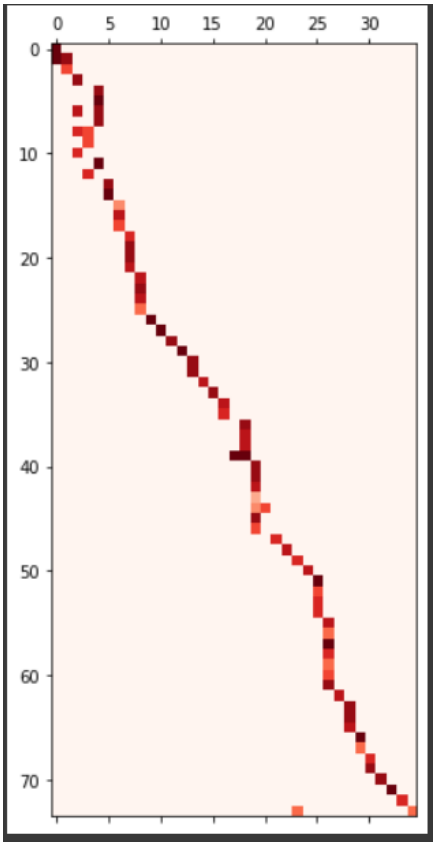

Figure 4: bi-adjacency matrix of user and anime sample dataset

This matrix on embedding in Louvain algorithms, which will find the modularity of these co-clusters, will give a bipartite graph with similar viewers cluster on one side and a cluster of anime liked by those users on the opposite side [9].

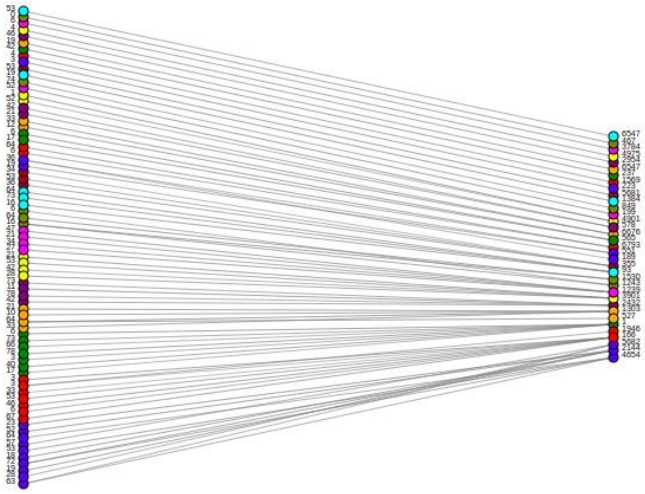

Figure 5**:** Bipartite graph of anime and user

5. Results and Conclusion

we evaluated both the approaches based on the graphical analysis and the recommendation results obtained. Looking at the Louvain algorithm using an adjacency matrix (fig 3), we get a good clustering of the animes based on their genres. A few examples of the anime in the rightmost the green cluster are shown in table 3.

TABLE 3

Anime id	Name	genres
43	Akira	Action, Military, Sci-Fi, Adventure, Horror, Supernatural, Shounen
75	Dead Aggressor	Action, Mecha, Military, Sci-Fi, Drama
30	Neon Genesis	Action, Sci-Fi, Dementia, Psychological, Drama, Mecha

The cluster is clearly of action animes. There a lot of similar genres among the three; action, sci-fi, drama and there are a few different genres such as mecha or psychological. The advantage of having the same genre is the user will like the anime 75 and 30 if he liked 43 because it has the same theme. The advantage of having the different genres is to introduce them to new themes rather than sticking bored to one. A user, for instance, sees anime 43 and likes it, the system recommends anime 75. Now, there is a new genre "mecha" in anime 75 thus, a user is introduced to a new genre. Similarly, if he liked 75 i.e containing the "mecha" genre. The system recommends anime 30. 30 contains "mecha" as well as the new genre "psychological". Thus, going on, the user will travel through cluster as there are intermediate anime like anime 1, user will also get a taste of an entirely new collection of anime from a new cluster if he likes. Thus, the genre-based recommendation solves the problem of recommending products of only one type.

As for bipartite clustering using the Louvain algorithm, we get a good cluster of the user of similar taste on one side and a cluster of corresponding anime on the other. A small view of the cluster in fig 5 is provided in table 4.

TABLE 4

User id	Anime id	Rating	genres
17	1	8	Action, Adventure, Comedy, Drama, Sci-Fi, Space
78	1	9	Action, Adventure, Comedy, Drama, Sci-Fi, Space
40	3901	9	Action, Comedy, Historical, Mystery, Sci-Fi

The users 17, 78, and 40 of the blue clusters are connected to anime 1, and 3901 of the blue cluster. By looking at users 17 and 78, we can say that they have liked anime 1 as they have a similar rating. also, looking at the genres of the anime in the same cluster we can say all the users like action-related genres. Thus, by the above two, we can say the user clustering has users with similar likings. it is worth noticing that in spite of providing the rating as weight in the bi-adjacency matrix, there is the clustering of anime having similar genres list. This can be because of the users with similar likings like the anime with similar genres. We can say that this anime is indirectly clustered with the same genres based on user liking. Thus, merely rating is enough for recommendation using this approach because, if user 17 liked anime 1, it has the same taste as user 40 also, anime 3901 is liked by user 40 and have similar genres as the one user 17 liked. Thus, user 17 will definitely like anime 40.

looking at the modularity of both the models, we can say that Louvain with coclustering using bi-adjacency matrix forms a better clustering than Louvain clustering using adjacency matrix as the modularity of approach 1 is less than approach 2. This must be because of the zero entries in the matrix. There will be more 0 entries in the adjacency matrix as being a square matrix it has to fill the con-connected entries with 0. But this is not the case with the bi-adjacency matrix.

TABLE 5

Approach	**modularity**
Louvain using adjacency matrix	0.78364
Louvain coclustering for bipartite graph	0.93819

Finally, we can conclude that approach 2 is the best of the two as merely providing a rating can show the best recommendation on anime.

REFERENCES

[1] Bugle, B. (2021). 5 Reasons Anime Popularity Is Booming. Geek girl authority

[2] Hiromichi Masuda, Tadashi Sudo, Kazuo Rikukawa, Yuji Mori, Naofumi Ito, Yasuo Kameyama, and Megumi Onouchi. Anime industry report 2019, 2019.

[3] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, 42(8):30–37, 2009.

[4] Robin Burke. Hybrid Web Recommender Systems, pages 377–408. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[5] VALDIVIESO, HERNAN. “Anime Recommendation Database 2020.” Anime Recommendation Database 2020 | Kaggle, Accessed 3 May 2022.

[6] Vert, Jean-Philippe; Tsuda, Koji; Schölkopf, Bernhard (2004). "A primer on kernel methods".

[7] Kharrazi, A. (2008). Modularity - an overview | ScienceDirect Topics.

[8] Mishra, A. (2019, November 21). Demystifying Louvain’s Algorithm and Its implementation in GPU | by Abhishek Mishra | Walmart Global Tech Blog | Medium. Medium

[9] Thomas Bonald, Q. L. (2020). Louvain clustering. Retrieved from scikit-network.

.