Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230898

Research on two popular recommendation algorithms for anime

Zhefei Meng

School of Psychology, University of Nottingham, Nottingham, UK, NG7 2QL

xmxhuihui@163.com

Abstract. Anime is a popular eastern art form whose audience is mainly young people. In recent years, many people choose to watch anime on the websites. The recommendation system plays a very important part in improving the user experience and saving time. This paper focuses on two basic recommendation algorithms based on machine learning and deep learning methods, including the content-based and collaborative filtering method. The data used in this paper was downloaded from a public dataset on Kaggle. This paper shows how the two methods perform in the anime dataset and compares the results of the two methods. However, two methods have their own advantages in different conditions. The content-based method works when the user wants some related contents. The collaborative filtering method works better while considering more factors other than contents. These two methods can be combined under a new algorithm to form an even more reliable and reasonable recommendation system in the future studies.

Keywords: anime, recommendation system, content-based, collaborative filtering, deep learning.

1. Introduction

Searching engine and recommendation system are common among video websites to show the users their favorite list. Current ideas of building a recommendation system are mainly based on two basic approaches, content-based method and collaborative method. There are also cases that are using a hybrid method. Many video websites which have large user groups are using the hybrid recommendation system, which is supposed to be reliable than only one single algorithm.

In this paper, the content-based and collaborative filtering methods were applied to the anime recommendation database collected in 2020 by a Kaggle user named Hernan Valdivieso. The dataset includes anime information data and users' watching and rating histories. In the first part, some data analysis was conducted on the anime information dataset to show the data structure in 2-dimensional space. Then the content-based recommendation algorithm was applied to the anime information dataset. The recommended anime along with the score was shown in the table. In the last part, the CF method was applied to both anime and user data. The results of two methods show their own advantages in different conditions. There will be an even more sophisticated hybrid recommendation algorithm proposed in the future based on these two algorithms to make a better recommendation system.

© 2023 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/6/20230898

2. Methodology

2.1. Content-based method

Content-based method is a basic and common way to build a recommendation system. This method is mainly based on the description and profile of user preferences[1]. It is suitable to the situation that the information(name, genre, description) of an item is collected. Content-based recommendation system takes it as a classification task and learns a classifier to find out the similar item list for user. However, the recommended item is restricted to the items with similar contents, so that it is hard for users to find out the items from other genres.

Here is the process of how content-based method works. The full anime data were first embedded into vectors. Then choose a certain item to which the user wants to search the similar items. Compare all the vectors and generate the similarity to the selected item. Finally list the items whose similarity is above the threshold and sort them in order. A good recommendation system should not include too many items. So it is important to set the threshold properly in order to limit the recommendation list to a small size.

2.2. Vectorizing and dimensionality reduction visualization

Term frequency-inverse document frequency(TF-IDF) is a numerical statistic that intends to reflect the importance of a word to a paragraph of text or a collection of corpora[2]. It is often used as a weighting factor in searches for information retrieval and text mining[3]. The importance of each single word increases proportionally if the word occurs with a higher frequency. TF-IDF is widely used in many search engines and also in the field of natural language processing(NLP).

Term frequency(TF) is the relative frequency that the word t appears in the document d is defined by:

$$\mathrm{tf}(i,d) = \frac{f_{t,d}}{\prod_{t' \in d} f_{t',d}} \tag{1}$$

Inverse document frequency(IDF) measures how much information a word provides, which also reflects whether a word is common or rare in a document. The definition is given by:

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \tag{2}$$

The TF-IDF value is the multiplication of TF and IDF.

$$\mathrm{TF-IDF} = \mathrm{TF} \times \mathrm{IDF} \tag{3}$$

2.3. Clustering

In this paper TF-IDF is used to get the keyword from the full content of anime. The full content of each anime includes the name, synopsis, genres, type, studios and rating. Each paragraph of contents were embedded into a vector. The whole vectors forms a matrix, while each element represents the importance or impact of the corresponding word.

Next, the data were analyzed with principal component analysis(PCA) and k-means method to cluster the data into several groups on a 2-dimensional space[4]. PCA is a technique that preserves the maximum amount of information to show the data on a 2D or 3D space. It is accomplished by linearly transforming the data into a new coordinate system. The first two components were maintained to plot the data on a 2-dimensional plain in this paper. K-means is a method of vector quantization that divide the items into several clusters in which each item belongs to the cluster with the nearest mean. The standard algorithm, also called "naive k-means" proceeds by alternating two steps[5]:

Step 1: Assign each item to the cluster with the nearest mean.

DOI: 10.54254/2755-2721/6/20230898

$$S_i^t = \left\{x_p: \left\|x_p - m_i^{(t)}\right\|^2 \leq \left\|x_p - m_j^{(t)}\right\|^2 \forall j, 1 \leq j \leq k\right\} \quad (4)$$

Step 2: Update the mean for the items to each cluster.

$$m_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x_j \in S_i^{(t)}} x_j \quad (5)$$

mi(i) was first given an initial set of values. Then repeat the two steps until the algorithm converges and none of the assignments change. However, the result is not guaranteed to be optimal, which means that there could always be better way to cluster the data.

The elbow method is a heuristic used in determining clusters of data. The explained variation is plotted as a function of the number of clusters. The elbow of the curve is picked as the optimal value of the number of clusters. The elbow method shows that the optimal number of clusters is about 10, as shown in Figure 1. The total anime was classified into 10 categories each of which has a keyword list.

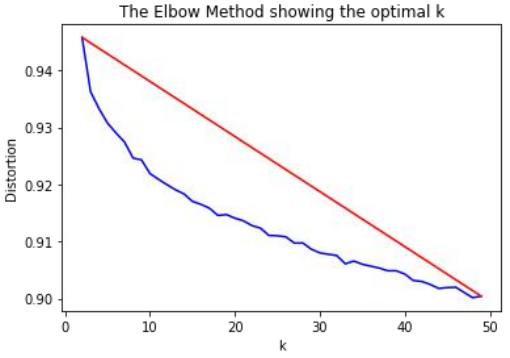
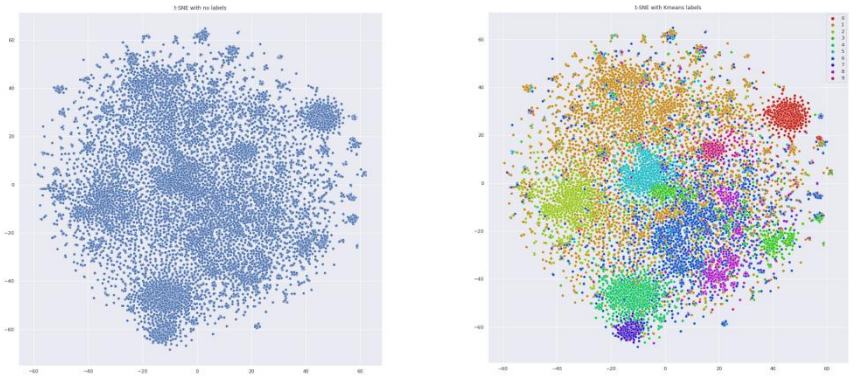


Figure 1. The optimal number of clusters for the anime data.

T-distributed stochastic neighbor embedding(t-SNE) is a method for showing a high dimensional dataset on a 2-dimension graph [6]. It is a nonlinear dimensional reduction method that embeds high dimensional data in a low dimensional space. Similar items are modelled by close points and others are modelled by distant points with high probability. The anime information data were shown in Figure 2 in clusters.



(a) (b)

Figure 2. (a) t-SNE with no labels (b) t-SNE with labels.

DOI: 10.54254/2755-2721/6/20230898

The next step is to find out what keywords are used to generate the 10 clusters. Latent Dirichlet allocation (LDA) is a generative statistical model that is adopted to find out the topics or keywords of a document[7]. The LDA model was combined with another numeric statistical model CountVectorizer. CountVectorizer is similar to TF-IDF method, which could also generates a frequency matrix of the whole data. Then the LDA method was applied to the frequency matrix to get a keyword list with highest frequencies. A word cloud is plotted in Figure 3 showing the top highest frequent words.

Figure 3. Keyword word cloud for the anime full information data.

2.4. Cosine similarity

Cosine similarity is a method to compare the similarity between two sequences or vectors[8]. In this paper, it is used to compare the vectors of each anime in the TF-IDF matrix, which is defined as the following equation:

$$S_C(A, B) := \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (6)$$

The parameters A and B are two vectors for comparison. The value of the cosine similarity is between -1 to 1. -1 means totally opposite, 1 means the same, and 0 indicates orthogonality or decorrelation. Based on the cosine similarity, the recommendation system shows the anime with the highest values.

2.5. Collaborative filtering

Collaborative filtering is a method of making predictions automatically with the information of user preferences and usage history from other users[9, 10]. This method is based on the assumption that the decision made in the past would continue into the future. In other words, people do not change their preferences when they make the next decision compared with the past. The recommendation system would also look into other similar users' searching history and generate a item list that probably favors the current user. CF method is widely used in the recommendation system of video websites. It is also used as a part of the hybrid recommendation system. Here is an example to show the process of CF method. Suppose user A has a similar taste to user B because they have a partially similar watching list of anime. Hence, the system could recommend some of the common items to user A from the watching list of user B. If the user group is a large one, then there could be multiple users who share the same interest with the user A. Hence the system could integrate this group of users' watching lists and recommend anime to the user A.

The collaborative filtering system has many types and forms, but the common workflow can be concluded by the following two steps:

Step 1: Find the similar users who share the same rating patterns with the active user.

Step 2: Calculate a recommendation list with the ratings from the first step.

DOI: 10.54254/2755-2721/6/20230898

3. Results

3.1. Validation of clustering

If the data is properly clustered, it should be possible to train a classifier to predict which cluster each anime belongs to. In this paper, a stochastic gradient descent(SGD) classifier was trained to check how well the data was clustered. The training set and test set were split by the ratios of 0.8 and 0.2. The accuracy is shown in Table 1. It could be accepted that the data was well clustered.

Table 1. SGD classifier accuracy.

	Training Set	Test Set
Accuracy Score	92.445%	89.639%
Precision	93.280%	91.966%
Recall	93.069%	90.176%
F1 Score	93.149%	91.006%
Mean Cross Validation Score	92.531%	

3.2. Result of the content based recommendation system

This paper takes the famous anime Naruto as an example to find the similar anime and list them in order of similarity. The results are shown in the Table 2 with the name and score of recommended anime. The anime shown in the table are mostly the movies and spinoff of the Naruto TV series.

Table 2. Name and score of similar anime to Naruto.

Name	Score
Naruto: Shippuuden	8.16
Boruto: Naruto The Movie - Naruto ga Hokage ni Natta Hi	7.40
Naruto: Shippuuden Movie 6 - Road to Ninja	7.67
Boruto: Naruto Next Generations	5.81
Naruto: Shippuuden - Shippuu! ”Konoha Gakuen” Den	7.15
Naruto: Shippuuden Movie 4 - The Lost Tower	7.42
Naruto: Honoo no Chuunin Shiken! Naruto vs. Konohamaru!!	7.16
Boruto: Naruto the Movie	7.50
Naruto: Dai Katsugeki!! Yuki Hime Shinobu Houjou Dattebayo! - Konoha no Sato no Dai Undoukai	6.87
Naruto: Shippuuden Movie 2 - Kizuna	7.29
Naruto: Takigakure no Shitou - Ore ga Eiyuu Dattebayo!	6.76
Naruto Soyokazeden Movie: Naruto to Mashin to Mitsu no Onegai Dattebayo!!	6.97
Naruto SD: Rock Lee no Seishun Full-Power Ninden	7.14
The Last: Naruto the Movie	7.76
Naruto: Shippuuden Movie 5 - Blood Prison	7.46
Naruto Movie 2: Dai Gekitotsu! Maboroshi no Chiteiiseki Dattebayo!	6.88

DOI: 10.54254/2755-2721/6/20230898

Table 2. (continued).

Naruto: Shippuuden Movie 1	7.29
Naruto: Shippuuden Movie 3 - Hi no Ishi wo Tsugu Mono	7.35
Naruto: Akaki Yotsuba no Clover wo Sagase	6.52

3.3. Anime recommendation system based on collaborative filtering

3.3.1. Results given a certain anime. A neural network was trained to get the embedding of anime and user[11]. The structure of the network is shown in Figure 4. The rating information for each user was used to generate the recommendation list. Before sending the ratings into the neural network, the values were scaled between 0 and 1 for higher speed and accuracy. Then the rating values and corresponding user and anime index were sent to the input layers for training. Figure 5 shows the loss curves of the training set and test sets.

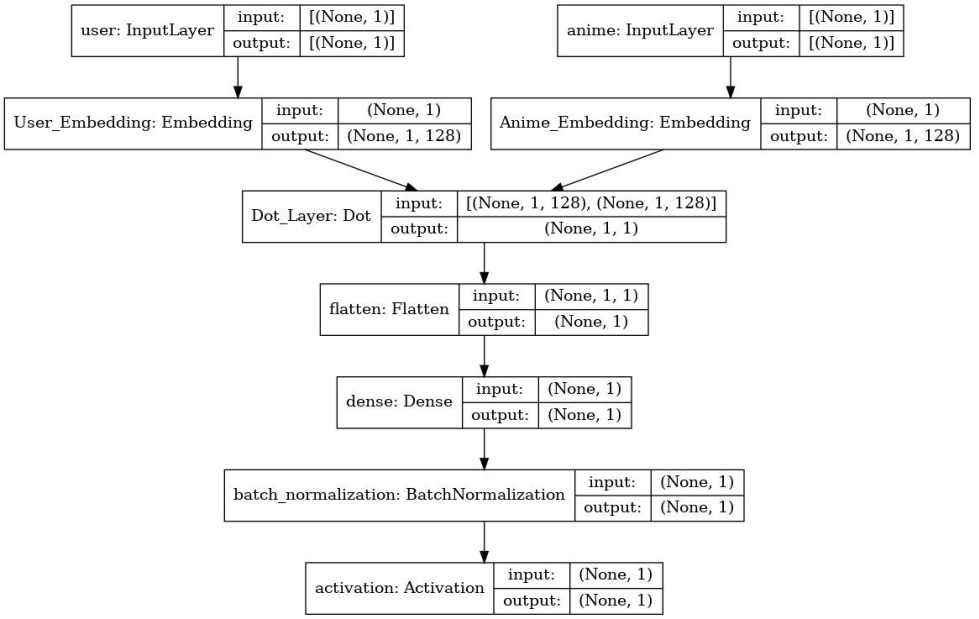


Figure 4. Loss curves during the training process.

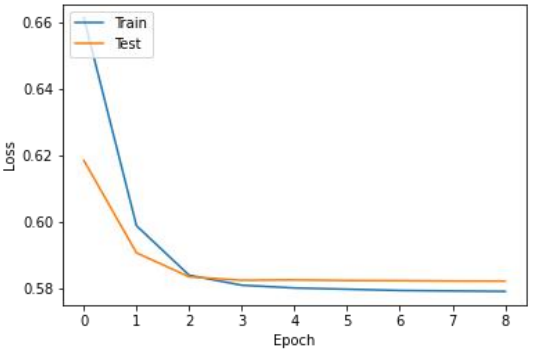


Figure 5. Loss curves during the training process.

DOI: 10.54254/2755-2721/6/20230898

This paper still takes the same anime Naruto as an example. The result is shown in Table 3. By referring to the genre, we could find that these anime do not seem to correlate to each other like the result of a content-based method. It is hard to say which method provides a more reasonable recommendation items or which method is better than the other, because people make decisions based on some other factors, such as recommendations from their friends, medias, etc. Doing a survey on how people feel about the recommendation items is one of the methods to know how the recommendation system works. But it is not studied in this paper. Also, the results of training the neural network can be different each time. Since the embedding vectors are different each time, it is possible that recommendation list could be different or even empty.

Table 3. Similar anime to Naruto with CF method.

Name	similarity	genre	Score
Loups=Garous	0.700322	Sci-Fi, Mystery, Thriller	6.3
Macross F Music Clip Shuu: Nyankuri	0.697030	Music	7.32
Devilman	0.691844	Action, Horror, Demons, Supernatural	6.45
Survivor	0.690542	Music, Supernatural	5.06
Fate/stay night:Heaven's Feel - III. Spring Song	0.681387	Action, Supernatural, Magic, Fantasy	8.79
Funny Faces in High School	0.674207	Comedy, Parody, Romance, School, Shounen	7.27
The Thousand Noble Musketeers	0.673864	Action, Military	4.93
Hello Kitty no Shiawase no Tulip	0.672215	Kids	NaN
Time Bokan Series: Gyakuten Ippatsuman	0.670703	Action, Comedy, Mecha, Sci-Fi	5.88
Xiao Hua Xian 2nd Season	0.667391	Kids, Magic	NaN

3.3.2. Results given a certain user. The dataset includes both the anime information and user preferences on these anime. Hence, we could also give each user a recommendation list based on their taste and ratings. The first step is to find out the similar users in order to create a candidate list. Suppose a random user is selected, and the similar users are shown in the Table 4. The selected user has preferences in the following genres, which are plotted in the word cloud in the Figure 6. The recommendation list based on the CF method is shown in the Table 5, which is sorted by the time of each item appearing in the total list.

Table 4. Similar users to user #168350.

similar users	similarity
255338	0.340584
183910	0.340268
139898	0.335110
63804	0.333554
108452	0.329461

DOI: 10.54254/2755-2721/6/20230898

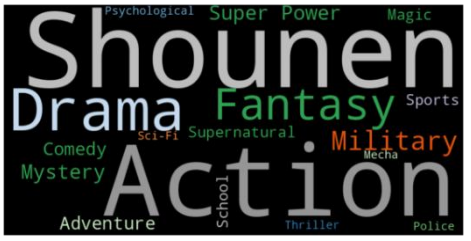


Figure 6. Favorite genres of the user #168350.

Table 5. Recommendation list for user #168350.

Frequency	Name	Genre	Score
5	Fullmetal Alchemist:Brotherhood	Action, Military, Adventure, Comedy, Drama, Magic, Fantasy, Shounen	9.19
5	Hunter x Hunter	Action, Adventure, Fantasy, Shounen, Super Power	9.1
4	Death Note	Mystery, Police, Psychological, Supernatural, Thriller, Shounen	8.63
4	Attack on Titan Season 3 Part 2	Action, Drama, Fantasy, Military, Mystery, Shounen, Super Power	9.1
4	Durarara!!	Action, Mystery, Supernatural	8.18
4	Haikyu!!	Comedy, Sports, Drama, School, Shounen	8.53
4	Magi:The Kingdom of Magic	Action, Adventure, Magic, Fantasy, Shounen	8.28
4	Attack on Titan Season 3	Action, Military, Mystery, Super Power, Drama, Fantasy, Shounen	8.59
4	Code Geass:Lelouch of the Rebellion R2	Action, Military, Sci-Fi, Super Power, Drama, Mecha	8.91
4	Haikyu!! 2nd Season	Comedy, Sports, Drama, School, Shounen	8.73

4. Conclusion

This paper basically studies two basic algorithms in the recommendation system, the content-based method and the CF method. The results of recommended anime are shown in Table 2 and Table 3. Although the result of the CF method does not seem as highly correlated as the content-based method, it is closer to the human choice. In reality, people do not always make consistent choice. There are some cases that people would like to turn to a new option. Content is not the only factor that users consider in the selection process. Many other factors and contingencies could also affect users' choice. Hence, it is a more reasonable way to conjecture users' preferences from other users' interests. In recent years, more video websites choose to use a hybrid recommendation system, which could utilize both advantages of the two basic algorithms.

It is always important to improve the user experience of the recommendation system. Showing what users want to see is a big task for the designers. The recommendation list should not be very long, otherwise it could take a very long time before users could find their favorite anime. Besides, the order

DOI: 10.54254/2755-2721/6/20230898

in which the anime are shown could also help users find their favorite anime more efficiently. This means the recommendation system should also help their users arrange a proper order to watch the anime. Though it could be hard for a machine to find out a human's taste perfectly, the algorithm for the recommendation system still needs to be improved and updated for better user experience in the future.

References

[1] Aggarwal, C., 2018. Recommender System: The Textbook. SPRINGER.
[2] Rajaraman, A.; Ullman, J.D. (2011). ”Data Mining” (PDF). Mining of Massive Datasets. pp. 1–17.
[3] Lin, W.-C., Tsai, C.-F. and Chen, H. (2022) “Factors affecting text mining based stock prediction: Text feature representations, Machine Learning Models, and news platforms,” Applied Soft Computing, 130, p. 109673.
[4] Galluccio, L. et al. (2012) “Graph based K-means clustering,” Signal Processing, 92(9), pp. 1970–1984.
[5] MacKay, David (2003). ”Chapter 20. An Example Inference Task: Clustering” (PDF). Information Theory, Inference and Learning Algorithms. Cambridge University Press. pp. 284–292.
[6] van der Maaten, Laurens & Hinton, Geoffrey. (2008). Viualizing data using t-SNE. Journal of Machine Learning Research. 9. 2579-2605.
[7] Blei, David M.; Ng, Andrew Y.; Jordan, Michael I (January 2003). Lafferty, John (ed.). ”Latent Dirichlet Allocation”. Journal of Machine Learning Research. 3 (4–5): pp. 993–1022.
[8] Tan, P., Steinbach, M., Karpatne, A. and Kumar, V., 2005. Introduction to data mining. p.500.
[9] John S. Breese; David Heckerman & Carl Kadie (1998). Empirical analysis of predictive algorithms for collabo- rative filtering. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence.
[10] Afoudi, Y., Lazaar, M. and Al Achhab, M. (2021) “Hybrid recommendation system combined content-based filtering and collaborative prediction using Artificial Neural Network,” Simulation Modelling Practice and Theory, 113, p. 102375.
[11] Zhang, Y., Liu, Z. and Sang, C. (2021) “Unifying paragraph embeddings and neural collaborative filtering for hybrid recommendation,” Applied Soft Computing, 106, p. 107345.