Finding Similar Users with Recommender Systems

Florencia Valdés
Pontificia Universidad Católica de Chile
Santiago, Chile
ftvaldes1@uc.cl

Matías Ovalle
Pontificia Universidad Católica de Chile
Santiago, Chile
movalle4@uc.cl

ABSTRACT

Users of websites that focus on item collection may want to get to know other users with preferences similar to their own, especially if the website has a social component and users can add friends. We created a user recommender system in which similar users are found by the items they rate, based on methods such as user based collaborative filtering and deep learning.

KEYWORDS

recommender system, user recommendation, social recommendation

1 RELATED WORK

Many contributions in the user recommendation area are based around social graphs and user-user interactions. The current developments in the area also include user-item information, for example, in adding user-item graphs [4] or using matrix factorization [3] for user interests. Another method [2] uses the information of social relationships between users to enhance recommendations, featuring a self-attentive representation, learning the user-user implicit relationships. In other paper [1], researchers create a recommendation model in which users can be similar in two ways: they can have similar preferences or they can have similar "attractiveness" for the users that contact them. They use this information to predict, for a given user, other users that they may like to contact based on their similarity. The problem with all of these methods is that they heavily base their recommendations in the data of user-user interactions, which may not be available, as in this case.

One method [5] generated user distributions from the topics they applied to documents, and then used Kullback–Leibler divergence to compute the similarity between distributions. Although the results were favorable in the task of recommending similar users, it needs a dataset that can be converted to a probability distribution.

2 DATASET

2.1 Grouvee

Grouvee is a web page where users can catalog and rank videogames they have played. It also doubles as a social network, where users can share their activities, make status updates and comment on each other's posts. The users can also befriend each other, but the function is underutilized and most people see an unfiltered version of the front page, with activities of all users. This includes items the user may not be interested in. By providing a way of finding similar users, they can get the most out of the page by connecting to people and finding more content that interests them.

Grouvee's main dataset has almost 2 million entries, each containing a rating for a certain game made by a certain user. There's a secondary dataset that provides the genres of each game. The exact numbers are shown in table 1. The rating is a whole number from 1 to 5. The average rating is 3.7, the median rating is 4 and the average rating per game is 3.01, which suggests that users tend to rate more items that have a higher average rating than those with a lower rating. In average, each user made 67.27 ratings and each game was rated 57.53 times. The distribution can be seen in figures 1 and 2.

Table 1: Dataset summary

Ratings	1,960,658
Users	29,144
Games	34,075

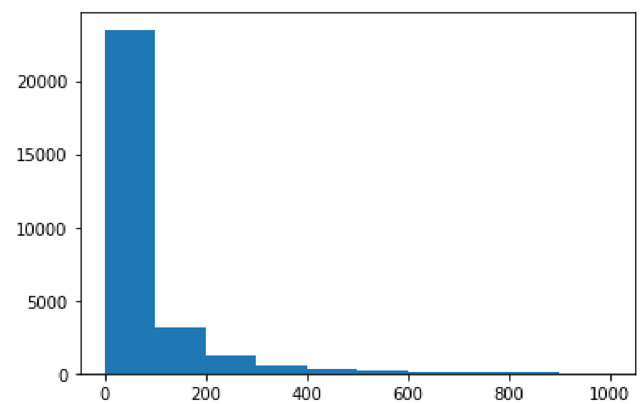

Figure 1: Number of users (y axis) vs. number of ratings per user (x axis). The graph only shows users who have made less than 1000 ratings.

One challenge that this dataset presents is that it is not possible to evaluate the user recommendations. The dataset doesn't provide friendship information between users and it is not possible to run online tests and ask users about the recommendations. To solve this, we used recommender systems that base their recommendation in finding similar users or in finding user representations and we evaluated them with traditional methods. We assume that better item recommendations in these types of recommender systems will correlate with better user representations or better user similarity functions, due to them being the core of the recommendation.

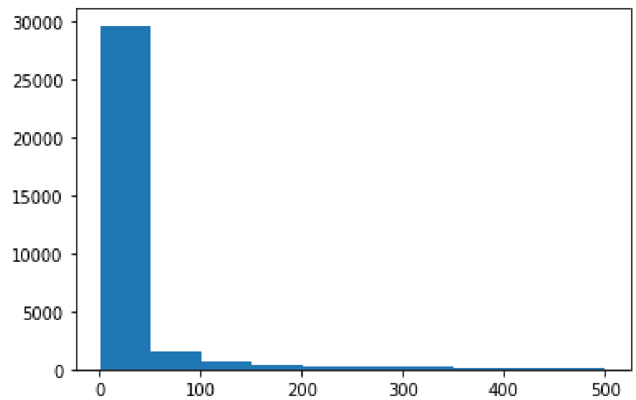

Figure 2: Number of games (y axis) vs. number of ratings per game (x axis). The graph only shows games with less than 1000 ratings.

2.2 MyAnimeList

My anime list is a web page where users can make a list of anime (japanese cartoons) they have watched and rank them. The site provides users with a list system to organize and score animes and mangas. It also works as a social network. Users can add friends, write blog posts and create clubs of people with similar interests.

MyAnimeList's dataset has 600,000 entries, with the same format of Grouvee's. The rating is a whole number from 1 to 10. The average rating is 7.4 and the median rating is 8. In average, users rate 136 anime and an anime is rated 60 times.

3 METHODOLOGY

We experimented with different recommender methods that based their recommendations in user similarity. This allowed us to evaluate the user recommendation indirectly by evaluating the item recommendation, which is necessary due to not being able to directly measure the effectiveness of users recommendations.

The first method used was User Based Collaborative Filtering. We did our own implementation of User KNN. We first computed the distance matrix between users, based on the user rows from the user item matrix. Then, we calculated the K nearest users according to this distance matrix. Finally, we made the recommendations by ranking the highest rated games played by the closest users.

A second method used was implemented with deep learning. We made a neural network that received the user row and the item column from the user-item matrix as separate inputs, and then learned a representation from them with smaller dimensions. The model then added both representations in order to predict a rating from 1 to 5. The network architecture can be seen in figure 3 and it has 65 million parameters.

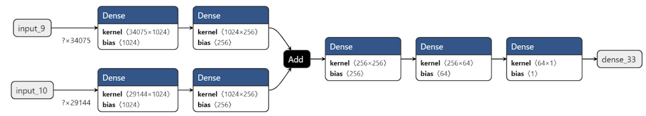

Figure 3: Neural Network's Architecture

4 PARAMETER ANALYSIS

There are different ways to represent the rating in the user-item matrix, and therefore the score to predict:

Binary representation: 1 if the user had rated the item and a 0 otherwise. The idea behind this was to find people that interacted with the same items and therefore had more to talk about, regardless of their opinion on the items.

Triple representation: -1 if the user did not like the item (rating < 3), 1 if the user did like the item (rating ≥ 3) and 0 otherwise. This representation helps with the problem of people rating more positive or negatively, and also helps with recommending people with similar taste in a broader sense (liking or disliking, rather than an exact number).

No representation: use the rating without any modification (1 to 5 rating). 0 if the user did not rate the item. We made the most tests with this method because it keeps all the information about the users interests and hence it will probably recommend the most similar users.

In user based collaborative filtering, the recommendations are made from the k most similar users according to a similarity function. These are the distance functions that we experimented with:

Cosine distance: $cosine(X, Y) = \frac{X \cdot Y}{\sqrt{X \cdot X}\sqrt{Y \cdot Y}}$

Euclidean: $euclidean(X, Y) = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2}$

City Block: $city\_block(X, Y) = \sum_{i=1}^{n} |X_i - Y_i|$

Jaccard: $jaccard(X, Y) = \frac{X \cap Y}{X \cup Y}$

We also tried different values for k in uKNN, with k being the number of closest users we chose to get the recommendations.

In the deep learning mode we experimented with the network's architecture. We added and discarded layers and we also modified the size of them.

5 RESULTS

We did 6 experiments with the different item matrices. To measure the performance, we used mean absolute error as the evaluation metric. The results can be viewed on figure 2.

For each type of user item matrix we experimented with different numbers of k, with k being the number of nearest neighbors. As the number of neighbors increases, the mean average error decreases

Table 2: Results of User Based Collaborative Filtering

Experiment	Type of user-item Matrix	Distance Function	Number of Nearest Neighbors	MAE
1	Binary representation	Cosine	5	0.8076
2	Binary representation	Euclidean	5	0.9437
3	Binary representation	City Block	5	0.9438
4	No representation	Cosine	5	0.8861
5	Triple representation	Cosine	5	0.8767
6	No representation	Jaccard * Cosine	5	0.98

for the fourth experiment and increases for the fifth and sixth experiments. The results can be viewed on figure 3.

The results from MyAnimeList's database can be seen in figure 4. Given that the ratings range from 1 to 10, we can see that it performs similarly to the previous dataset.

For the deep learning model, the mean average error was 1.12 in the validation set.

6 CONCLUSIONS

We proposed a model for recommending similar users. This is particularly useful in websites that focus on item collections, like Grouvee for games or MyAnimeList for anime and also provide social features that haven't been utilized to the fullest extent. The current state of the art mostly deals with user-user interaction datasets, something that sometimes is not available. We evaluated traditional and current recommender systems methods such as User Based Collaborative Filtering and Deep Learning. We also evaluated their performance at recommending users by their item recommendation, with the assumption that a model that bases its item recommendations in user similarities will therefore perform better in the user and item recommendation tasks at the same time. We achieved a MAE of 0.8076 in our bests trials, and got the opportunity to connect thousands of users by recommending their next best friend.

In our future work we hope to try different techniques to create better recommendations, such as deeper neural networks. We would also like to use additional information, such as the game genre, to help the models make better recommendations.

REFERENCES

[1] Xiongcai Cai, Michael Bain, Alfred Krzywicki, Wayne Wobcke, Yang Sok Kim, Paul Compton, and Ashesh Mahidadia. 2010. Collaborative filtering for people to people recommendation in social networks. In *Australasian Joint Conference on Artificial Intelligence*. Springer, 476–485.

[2] Hongxu Chen, Hongzhi Yin, Tong Chen, Weiqing Wang, Xue Li, and Xia Hu. 2020. Social Boosted Recommendation with Folded Bipartite Network Embedding. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[3] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference*. 417–426.

[4] Ke Xu, Xushen Zheng, Yi Cai, Huaqing Min, Zhen Gao, Benjin Zhu, Haoran Xie, and Tak-Lam Wong. 2018. Improving user recommendation by extracting social topics and interest topics of users in uni-directional social networks. *Knowledge-Based Systems* 140 (2018), 120–133.

[5] Tom Chao Zhou, Hao Ma, Michael R Lyu, and Irwin King. 2010. UserRec: A User Recommendation Framework in Social Tagging Systems.. In *AAAI*. Citeseer.

Table 3: MAE for different number of neighbors (K)

Experiment	K = 1	K = 5	K = 10	K = 25	K = 50	K = 100	K = 500
4	0.9510	0.8861	0.8557	0.8215	0.8004	0.7833	0.7589
5	0.8711	0.8767	0.8798	0.8823	0.8876	0.8919	0.9049
6	0.9820	0.9834	0.9909	1.052	1.212	1.569	2.404

Table 4: MAE for different number of neighbors for MyAnimeList dataset

	K = 1	K = 5	K = 10	K = 25	K = 50	K = 100	K =500
MAE	1.4124	1.4161	1.3836	1.3218	1.2767	1.2422	1.1912