Steam Explorer: A Games and Friends Recommendation System

Illinois Institute of Technology, CS579 Project Milestone
Jiajia Liu A20426786, **Jia Tan**, A20433208, **Yixin Peng** A20426267

1 Overview

Steam is an online game platform and the Steam community network is a large social network of players on this platform. Based on Steam, we want to perform online social network analysis with friends and games recommendation features. Steam has an average of 18.5 million concurrent players and 9,300 new video games released in 2018, which is a magnificent challenge for finding good partners and favorite games. Our application work with the visualizable community network and some filtered statistical data to makes steam users find friends and games easily. We collected historical and real-time data in steam, processed, trained and tested them to make the result more precise.

2 Related work

Recommendation system has been an hot topic not only to the game field. For example, recommendations for goods in Amazon, videos in Youtube and friends in Twitter are all evaluated carefully by many researchers [1, 2, 3]. There are three types of recommendation system: collaborative filtering, content-based filtering, hybrid. [4] Collaborative filtering aims at predicting the user interest for a given item based on a collection of user profiles. It can be further divided to user-based and item-based.[5] And content-based system recommend an item to a user based upon a description of the item and a profile of the user's interests[6].

For game recommendation system developing, Rafet [7] used an non-linear archetypal model and outperformed neighborhood model. And according to [8], collaborative filtering model beat both neighborhood and matrix factorization.

For friends recommendation, inspired by gamer behavior analysis in [9], players tend to befriend those who are similar in terms of popularity, playtime, money spent, and games owned. So these can be features to build models.

In this project, we are going to start with the most common used content-based methods in building games recommendation systems, and analyze the performance of link prediction model in Steam social network as a friends recommendation method.

3 Data

3.1 Historical Data

3.1.1 Database Summary

The Steam historical data are provided by BYU Internet Research Lab[1]. The data are in MySql format and collected between 2013 May - 2014 Nov. Table 1 and Figure 1 give an overview of the two tables that we are going to use as training data. A detailed description can be found in their paper [9].

Table Name	DB Size (MB)	Rows
Friends	44,981.06	392,732,150
Games_Daily	4,411.87	85,681,026

Table 1: BYU Database Summary

3.2 Real-time Data

Real-time data can be collected by Valve, a Steam official web API Provider[2]. This is mainly used for testing.

- **User-friend-pair**: Contains users' Steam ID, friends' Steam ID and date when became friends.
- **User-game-pair**: Contains users' Steam ID, games' ID and play time. Because of the change of Steam privacy policy, only 5,657 users in Games_Daily are now public.
- **Games features**: Contains description, price, genre and developer, etc. There are 76,398 games as at 2019 Oct.

3.3 Data Pre-processing

3.3.1 Subset selection

- **User-friend-pair**: 22,050 users are selected from 33,119,781 because of computational limit. The selection rule is keeping users in the intersection of Friends and Games_Daily, friends count between 10-100 and games count more than 10.

[1] https://steam.internet.byu.edu/
[2] https://developer.valvesoftware.com/wiki/Steam_Web_API

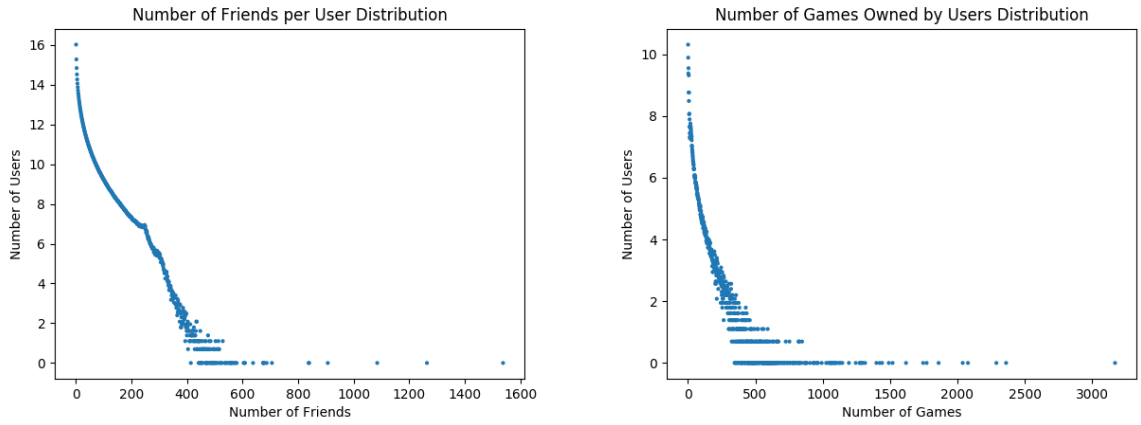

Figure 1: Friends and Games count distribution in 2014

- **User-game-pair**: For real-time data, 5,222 users who bought new games between 2014 to 2019 are selected. For historical data, as we can see in Figure 1, most of the users own less than 300 games. For speed consideration we choose 44383 users with games count between 20-200.
- **Games features**: Games not released or description language not English are dropped. Similar to [7], we remove games whose type is not 'game' or 'dlc'.

3.3.2 Text processing

- **Remove stop words**: Removed English stop words given by scikit-learn.
- **Stem vocabularies**: We used SnowballStemmer provided by nltk to transform English volabulary to it's stem.
- **Remove low/high frequency words**: Words with absolute frequency lower than min_fq or relative frequency higher than max_fq are removed. Best min_fq and max_fq will be selected in model validation and applied in later testing.

4 Method

4.1 Friends Recommendation

We use Adamic/Adar approach to recommend friends. Steam as an online game network would follow the online link prediction. We select 22,050 users as the initial users, whose friends' number are bigger than 10 and less than 100. And their friends' friends would be our target user to recommand,which are about 9,000,000. If we use Jaccard coefficient approach, we need to know the number of friends of these 9,000,000 target friends too, which are too big data preprocess. So we choose Adamic/Adar approach to deal with the friend prediction.

Huge users data need to be processed. For each initial user, we can find their friends and their friends' friend. By using the friends' pair data, we can process these data by dictionary structure. Initial user's friends would be the common friends between the initial user and the target user. By calculate these common friends' friend number, we get the score table according to Adamic/Adar. Sorted all scores and get the target user's ranking. If we get same scores by Adamic/Adar, we will give them the same rank too, which will help to increase the accuracy too.

For evaluation, we use MMR to calculate prediction accuracy by comparing the actual new friends with the previous ranking result.

4.2 Games Recommendation

To start with, we developed the content-based game recommendation. The users-game-pair are randomly split into five folds for cross validation. The inputs for training models are:

- Users-games-matrix $\mathbf{U} \in \mathbb{R}^{n \times m}$: n is the number of users and m is the number of games. For every i-th user and j-th game pair in training folder, the according position in $\mathbf{U}_{ij}$ user-games-matrix is 1, and 0 if there is not such pair.
- Games-features-matrix $\mathbf{V} \in \mathbb{R}^{m \times k}$: m is the number of games and k is the number of features. Every row is a feature vector for this game. The game features are listed in Section 3.2. Among these features, games description's tf-idf matrix is combined with other features and scaled by sklearn.preprocessing.scale method.

Then, the users-features matrix $\mathbf{S} \in \mathbb{R}^{n \times k}$ are calcu-

lated by averaging user owned games feature vectors:

$$S_{ij} = \frac{\sum_{t=1}^{k} U_{it} V_{tj}}{\sum_{t=1}^{k} U_{it}}$$

By calculating the cosine-similarity between users' and games' feature vector, we can rank the those games which is not owned yet by a user and give recommendations accordingly.

5 Evaluation and Results

Similar to [7], a user based off-line evaluation is used to evaluate the hit recall based on the ranking assigned to a random game selected from test set combining with another randomly selected 100 games. After ranking these games and derive a top N list, if the test game is in this recommendation list, we add a count for hit. An ideal recall hit, 100%, will be achieved if all of the test games in the test set are returned, or ranked, in the recommended top-L list, and oppositely, 0% recall will be achieved if non of the games in the test set are returned in any of the recommendation steps.

The games recommendation recall rate reached 92.26% for following parameters: min_df=3, max_df =0.6, and features = [description].

6 Conclusion

References

[1] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7:76–80, 01 2003.

[2] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. The youtube video recommendation system. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 293–296, New York, NY, USA, 2010. ACM.

[3] Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. Collaborative personalized tweet recommendation. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 661–670, New York, NY, USA, 2012. ACM.

[4] Justin Basilico and Thomas Hofmann. Unifying collaborative and content-based filtering. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 9–, New York, NY, USA, 2004. ACM.

[5] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 501–508, New York, NY, USA, 2006. ACM.

[6] Michael J. Pazzani and Daniel Billsus. *Content-Based Recommendation Systems*, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[7] Rafet Sifa, Christian Bauckhage, and Anders Drachen. Archetypal game recommender systems. *CEUR Workshop Proceedings*, 1226:45–56, 01 2014.

[8] Barry Plunkett, Brandon Lin, Stephanie Shi, and Chris Painter. The steam engine: A recommendation system for steam users. Master's thesis, University of Pennsylvaniau, 2018.

[9] Mark O'Neill, Elham Vaziripour, Justin Wu, and Daniel Zappala. Condensing steam: Distilling the diversity of gamer behavior. In *Condensing Steam: Distilling the Diversity of Gamer Behavior*, pages 81–95, 11 2016.

Tasks	Jiajia Liu	Jia Tan	Yixin Peng
Data collection	Crawled real-time games data using API.	Crawled real-time User-Friends-Pair using API.	Crawled real-time User-Games-Pair using API.
Data pre-processing	Pre-processed games description text.		Select subset. Stem Vocabularies. Perform statistics analysis.
Model training	Add more features such like tags and price for content-based game recommendation.	Trained friends link prediction model.	Trained content-based game recommendation model.
Model evaluation		Evaluated friends link prediction model.	Used recall rate to evaluated content-based game recommendation.

Table 2: Group Member Contributions

Periods	Tasks
Sep 25 - Oct 02	Choose topic and propose
Oct 02 - Oct 21	Find raw data, read research paper, discuss model
Oct 21 - Oct 25	Data pre-processing: games, users.Plot stats
Oct 25 - Oct 28	Base-line: friends link prediction, content-based games recommendation
Oct 28 - Oct 30	milestone.pdf
Oct 30 - Nov 09	Explore and compare: feature selection, other models
Nov 09 - Nov 13	visualization, app
Nov 13 - Nov 20	report.pdf presentation.pdf

Table 3: Timeline