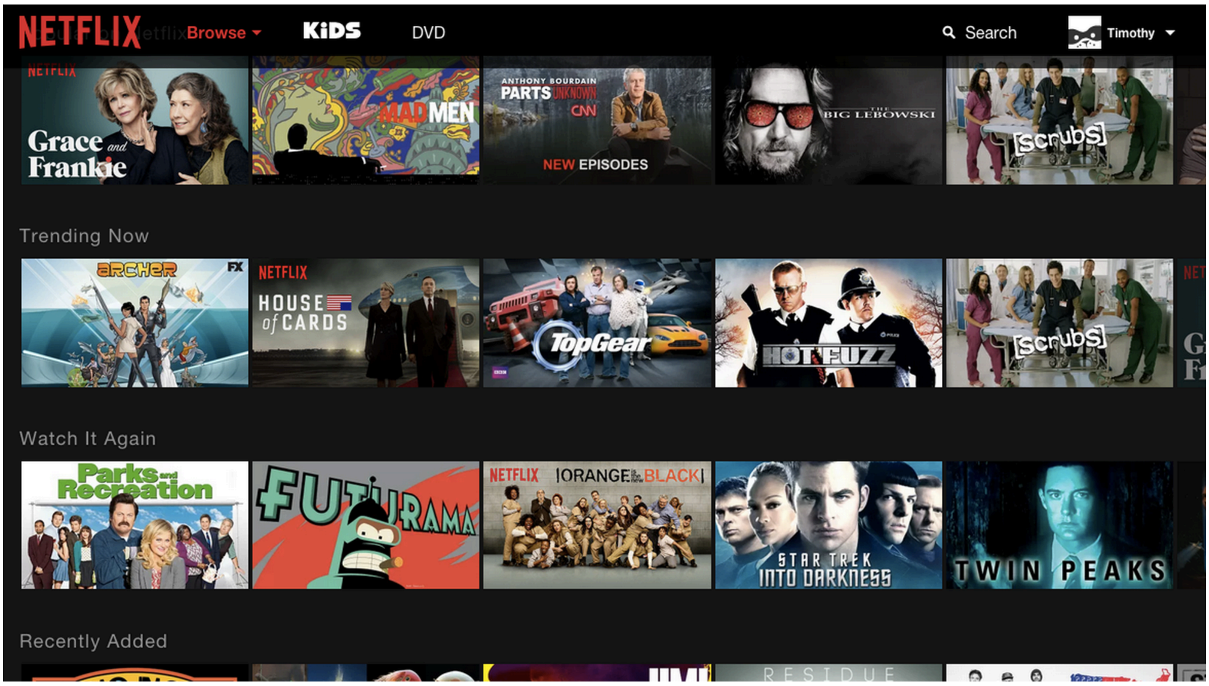
# Recommender System

*A user based collaborative filtering approach*





Ang Li

Ethan Xu

Ge Huang

Zhizheng Shi

# Introduction

While recommended systems have been implemented by major sites including Google and IMDB, there are always room for development in terms of personalisation. In this report, we aim to generalise a model which predicts the movie preference of a person, based on preferences of the similar users' group. We also aim to test the accuracy of our system using simple similarity function, which is also known as the cosine similarity.

# Related work

Netflix implements 5 major recommended systems, Personalised Video Ranker(PVR), Top-N Video Ranker, Trending Now and Continued Watching, Because you Watched. The first two algorithms are personalised while the latter three are unpersonalised. Meanwhile, a hybrid of the algorithms are often used, e.g. (unpersonalised) popularity is often blended into the results of personalised video ranking. As a results, even within the same genre, the recommended movies for different users in different regions at different times, is often different. Video-Video similarity :The sims algorithm is an unpersonalised algorithm that computes a ranked list of videos—the similars—for every video in our catalog. The key indicator is effective catalogue size, which measures the number of videos required to account for a typical hour streamed. It returns 1 if the most popular video accounted for most hours streamed, and N if all videos in the catalogue drove the same amount of streaming.

There are also work done for the research paper recommender system. The algorithm retrieves all the target paper's references and citations. For each of the references, it extracts all other papers from the web (google scholar to be precise) that also cited any of those target paper's references. In addition, for each of the target paper's citations, it extracts all other papers from the web that referenced any of those target paper's citations (in other words, all the references to the target paper's citations) and we refer to these extracted papers as the target papers nearest neighbours. For each of the neighbouring papers, we qualify candidate papers that are co-cited with the target paper and which has been referenced by at least any of the target papers references. We then measure the degree of similitude between these qualified candidate papers and the target paper by measuring their collaborative similarity. We then recommend the top-N most comparable papers to the researcher. In this recommendation system, the Jaccard similarity coefficient J is used.

# Implementation

Birds of feather, flock together, we follow the idea that similar users have similar tastes in movies. This means that for a particular unwatched movie, a user is more likely to watch it, provided the his/her companions (similar users).

We first split the data into training and testing sets. To do that, we take the 10% of each user's rating on the movies from the overall data as the test sets (in our case it is 14 movies per user), while keeping the remaining 90% as training sets. The test sets are used for evaluating the accuracy of our model, when comparing it with the predicted rating results.

The cosine similarity between two vectors is defined as

$$cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|}$$

Hence if we have a matrix of M, each row representing the movie rating vector of each user. The cosine similarity matrix can be computed as

$$S = \frac{diagonal(M \cdot M^T)}{|diagonal(M \cdot X^M)|}$$

When we pick a particular user, we order the remaining users in a decreasing order of similarity, and choose the top k of them, so we have a vector $\vec{s}$ of size k representing the similarities between this user and other users.

Then for each movie, we compute the dot product of $\vec{s}$ and the particular column of the rating matrix for that movie $\vec{r}$, divide the result by the sum of vector $\vec{s}$.

We therefore obtain the prediction matrix for the user and movies

$$P_{i,j} = \frac{\vec{r} \cdot \vec{s}}{\sum s_i}$$

Finally we calculate mean square error between the prediction matrix and the actual matrix as follows

$$MSE = \sqrt{\sum_{i,j} (P_{i,j} - M_{i,j})^2}$$

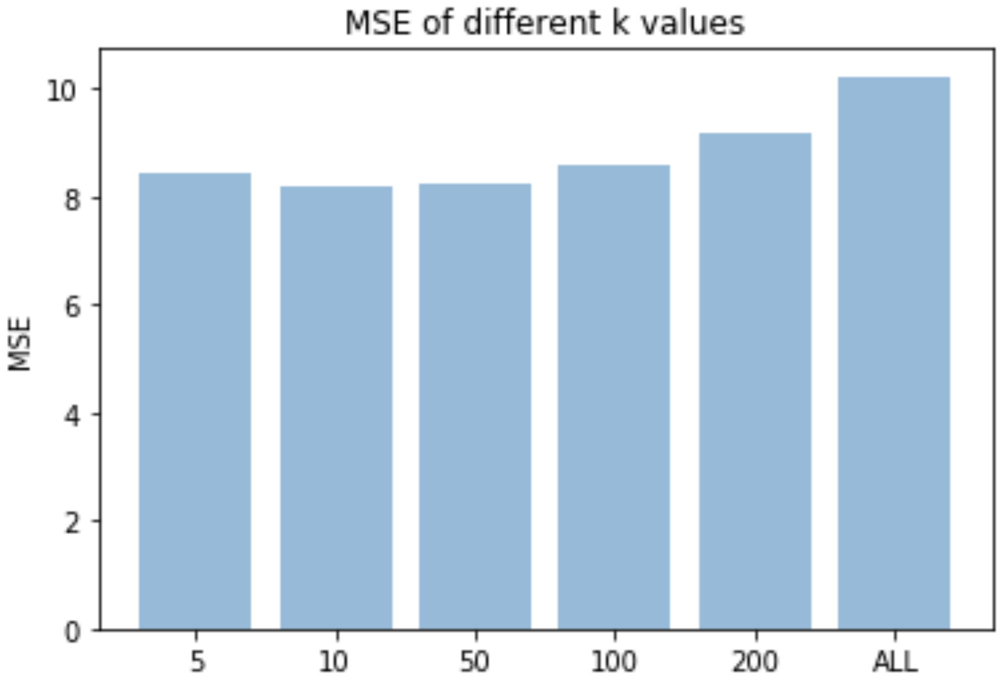
We perform the analysis for varying values of k from 5 via 10, 50, 100, 200 to size of the entire dataset, and plot the graph accordingly.

# Results

The MSE for different values of k is as follows

| 5 | 10 | 50 | 100 | 200 | All |
|---|---|---|---|---|---|
| 8.44 | 8.17 | 8.23 | 8.5 | 9.16 | 10.22 |

And the plot is




Also, by entering userID, we obtain the top 50 users with similar movie tastes with the user, (below is a sample output for user 1)

| UserID | MoveID | Title | Genre |
|---|---|---|---|
| 3 | 4 | Waiting to Exhale (1995) | Comedy\|Drama\|Romance |
| 6 | 7 | Sabrina (1995) | Comedy\|Romance |
| 18 | 19 | Ace Ventura: When Nature Calls (1995) | Comedy |
| 22 | 23 | Assassins (1995) | Action\|Crime\|Thriller |
| 33 | 35 | Carrington (1995) | Drama\|Romance |
| | …… | | |
| 603 | 714 | Dead Man (1995) | Drama\|Mystery\|Western |

And the 5 recommended movies are

- *Star Wars: Episode IV - A New Hope (1977)*
- *Fargo (1996)*

- *Godfather, The (1972)*
- *Star Wars: Episode V - The Empire Strikes Back (1980)*
- *Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)*

# Conclusion

When k < 50, the mean square error is decreasing as the value of k grows; on the other hand, MSE increases with the growth of k values as k > 50, i.e. MSE is minimum when k = 50, which is slightly less than 10% of the entire user population.

One possible explanation for the increasing MSE with larger values of k is that the model could be over-fitting. This is quite intuitive, just as it is impossible to accommodate everyone's needs. Having a model which takes into account of everyone's movie tastes, cannot accurately predict the personalised results.

# Reference

CARLOS A. GOMEZ-URIBE and NEIL HUNT, Netflix, Inc. *"The Netflix Recommender System: Algorithms, Business Value, and Innovation"*

Khalid Haruna, Maizatul Akmar Ismail, Damiasih Damiasih, Joko Sutopo, Tutut Herawan, "A collaborative approach for research paper recommender system"