What to Watch Next? Evaluating Recommender System Techniques in Online Movie Streaming

Team 5: Sameen Haroon, Abhishek Saha, Sharon Xu, Zeyu Zhu

Research Question: Which collaborative filtering recommendation technique(s) provide effective movie recommendations with low processing cost?

Introduction

Since the introduction of online stream entertainment content by Netflix in 2007[1], the subscription video-on-demand (SVoD) industry has become increasingly competitive. Today, the average American subscribes to three video streaming services, and more established providers such as Netflix, Amazon and Hulu are facing growing competition from up-and-coming services such as HBO Now, or the anticipated Disney+[2].

In this context, video streaming entertainment providers are increasingly looking for ways to both attract subscribers and increase consumption on their platforms. Due to high costs associated with retaining popular content on such platforms[3], many providers are spending growing sums on producing their own content, although with varying degrees of success.[4] At the same time, providers need to find ways to push the right internal or external content to the right consumer. With multiple platforms and entertainment options at their fingertips, consumers can lose interest within 60-90 seconds of trying to choose a movie to watch, and rarely review more than 10-20 titles (maybe 3 in detail).[5] As such, offering relevant, customized content suggestions through recommender systems continues to enable platform success.

In an effort to effectively personalize recommender system content, industry leaders rely on a variety of algorithms to deliver content to users. For instance, both Netflix and Amazon Prime Video use hybrid techniques that produce recommendations based on consumer viewing patterns and broader content popularity on their platforms. These techniques have varying computation needs and result relevance. They benefit from continued optimization to drive meaningful impact on business metrics such as take rate (the fraction of

---

[1] Miguel Helft "Netflix to Deliver Movies to the PC" New York Times. https://www.nytimes.com/2007/01/16/technology/16netflix.html (accessed December 9th, 2018).
[2] Adam Levy "How More Streaming Competition Affects Netflix and Amazon" The Motley Fool. https://www.fool.com/investing/2018/10/04/how-more-streaming-competition-impacts-netflix-and.aspx (accessed December 9th, 2018).
[3] Edmund Lee "Netflix Will Keep 'Friends' Through Next Year in a $100 Million Agreement" New York Times. https://www.nytimes.com/2018/12/04/business/media/netflix-friends.html (accessed December 9th, 2018).
[4] John Archer 'Netflix Beats Amazon Prime And Others In Five-Way Video Streaming Shoot Out' Forbes.https://www.forbes.com/sites/johnarcher/2018/05/22/netflix-beats-amazon-prime-and-others-in-five-way-video-streaming-shoot-out (accessed December 9th, 2018).
[5] Gomez-Uribe, Carlos A. and Neil Hunt. "The Netflix Recommender System: Algorithms, Business Value, and Innovation." ACM Trans. Management Inf. Syst. 6 (2015): 13:1-13:19.

recommendations offered resulting in a play) as well as broader goals of reducing churn to maintain subscriber growth and drive lifetime value.[6]

To optimize recommender systems, companies must balance the need for recommendations that are as personalized and relevant to users as possible with the need for low cost solutions, i.e. those that can be delivered in real-time with limited processing needs. Consequently, this research focuses on quantifying the tradeoff between effective performance and cost efficiency of commonly used recommender system techniques.

Specifically, we have chosen to evaluate techniques by building recommender systems on movie ratings data from Netflix users. As an industry leader, Netflix has a rich collection of user preference data that can be used to assess these techniques, and has released such data in the past with the explicit goal of improving its recommender system through analysis by the broader data science community.[7] Additional details on the specific techniques being evaluated, all of which are in use by Netflix and/or other major streaming providers today, is provided in the next section.

Background on recommender systems and techniques evaluated

Recommendation systems function as a personalized information technology to identify the set of items relevant to a user,[8] using 'statistical and knowledge discovery techniques to make recommendations based on previously recorded data.[9] A number of different approaches for creating recommendation systems exist today, and primarily rely on one of content-based or collaborative filtering, or both (known as hybrid filtering).

Content-based filtering relies on consumer preferences on product attributes to make recommendations.[10] For instance, movie attributes such as genre, release year can be matched with a user's preference e.g. to recommend a romantic comedy from the 1990s to a user who shows the interest in it. In general, the effectiveness of content-based techniques varies with the richness of the content tagging system that an entertainment provider has, and given data limitations, is not evaluated in this study.

---

[6] Ibid.
[7] Bennett, James, and Stan Lanning. "The netflix prize." In *Proceedings of KDD cup and workshop*, vol. 2007, p. 35. 2007.
[8] Deshpande, Mukund, and George Karypis. "Item-based top-n recommendation algorithms." *ACM Transactions on Information Systems (TOIS)* 22, no. 1 (2004): 143-177.
[9] Sarwar B, Karypis G, Konstan J, Riedl J (2000). "Analysis of recommendation algorithms for e-commerce." In EC '00: Proceedings of the 2nd ACM conference on Electronic commerce, pp. 158-167. ACM. ISBN 1-58113-272-7.
[10] Ansari A, Essegaier S, Kohli R (2000). "Internet Recommendation Systems." Journal of Marketing Research, 37, 363–375.

Collaborative filtering techniques, on the other hand, look for patterns in user ratings to produce recommendations based on rating similarity across users, movies, etc. For instance, if two users watched and liked the same set of movies, additional movies that are only viewed and highly rated by one user can be recommended to the other.[11] It has the benefit of producing recommendations across content categories and can dig out customers' potential interests instead of being restricted to recommending content directly similar to what a user has already seen. Also, it doesn't depend on the availability of rich descriptive data or well-organized user profiles as compared to content-based techniques. Collaborative filtering is more widely used than content-based approaches, especially amongst entertainment providers, since content can't be easily and adequately described by metadata.[12] As such, this study focuses on evaluating the three main collaborative filtering techniques: user-based, item-based and singular value decomposition. These are explained in further detail below.

User-based collaborative filtering (UBCF)

User-based collaborative filtering (UBCF) creates item recommendations based on consumption patterns of similar users. For a user, this approach finds his/her neighborhood i.e. users who have similar viewing and rating preferences. It then recommends a movie that has been seen by others in this neighborhood with a high average rating. Netflix's "Top picks for [UserName]" video reel is an example of this technique.[13] The UBCF algorithm takes an intuitive approach with results that can be quite relevant, but suffers from two important limitations related to data sparsity and scalability. The first occurs when users haven't rated enough items for a neighborhood of similar viewers to be found. This frequently happens in the 'cold start' case, i.e. for new users. The second limitation comes up when the number of users in a database becomes too large, and the resource requirements of UBCF algorithm can make it challenging to use in a real-time environment.[14]

Item-based collaborative filtering (IBCF)

Item-based collaborative filtering (IBCF) technique uses similarity between items to generate recommendations. Used in Netflix's "Because You Watched" video reel, this approach assumes that users will like a movie that is similar to what they've already watched and highly rated. Instead of using tags about a movie's content, however, it first finds users who have rated both movies and then calculates similarity

[11] This is an example of user-based collaborative filtering, a specific technique within collaborative filtering.
[12] Isinkaye, F. O., Y. O. Folajimi, and B. A. Ojokoh. "Recommendation systems: Principles, methods and evaluation." *Egyptian Informatics Journal* 16, no. 3 (2015): 261-273.
[13] Gomez-Uribe, et. al. "The Netflix Recommender System" (2015)
[14] Sarwar, B. et al. Item-based collaborative filtering recommendation algorithms (2001)

metrics for the movies. Once similar sets of items are found, a rating prediction can be made (e.g. using the weighted average).

When the ratio of users to items in a recommender system is large (as is often the case with successful entertainment platforms such as Netflix), the IBCF algorithm is better equipped to deal with large-scale data. It can be up to twice as fast as user-based recommender systems since it's typically model-based and only keeps a list of items that are similar.[15] However, this algorithm can potentially sacrifice recommendation quality.[16]

Singular Value Decomposition (SVD)

Singular Value decomposition (SVD) is less intuitive than user and item-based collaborative filtering approaches. In essence, it looks to identify hidden factors or traits of movies that drive user rating patterns. For a user, this approach measures the extent to which they like movies scored high on specific movie factors and makes recommendations based on high correspondence between item factors and user preference.

This technique is not formally identified on Netflix's platform, although it likely powers many of their recommendation video reels. It has also been used by data scientists who won awards for improving Netflix's recommender algorithm in the mid-2000s.[17] Key advantages of SVD include how it solves the problem of sparsity in UBCF and IBCF through its factorization approach, and enables much faster computations due to dimensionality reduction.[18] Additionally, this algorithm has been associated with more accurate results in empirical studies as well.[19]

Additional techniques

While the collaborative filtering techniques discussed above can be effective in making movie recommendations, they are of limited utility when little-to-no information is available about a user's preferences, or a given movie. This situation occurs most often as the 'cold start problem', when a system needs to provide recommendations to new users or have the ability to recommend new items[20]. In such cases, non-personalized approaches are used for recommendation, including the popularity technique and the random technique.

---

[15] Sarwar, B. et al. Item-based collaborative filtering recommendation algorithms (2001)
[16] Hahsler, Michael. *recommenderlab: A framework for developing and testing recommendation algorithms*. 2015.
[17] Kumar, B. (2016). "A Novel Latent Factor Model For Recommender System". Journal of Information Systems and Technology Management : JISTEM, 13(3), 497-514.
[18] Rajamanickam, S. (2009). Efficient algorithms for sparse singular value decomposition (Order No. 3400303). Available from ProQuest Dissertations & Theses A&I; ProQuest Dissertations & Theses Global. (304879163).
[19] Yehuda K., Robert B. & Chris V. (2009). Matrix factorization techniques for Recommendation Systems.
[20] Al Mamunur Rashid, Gerge Karypis, and John Riedl. "Learning preferences of new users in recommender system: An information approach." In *SIGKDD Workshop on Web Mining and Web Usage Analysis (WEBKDD)*. 2008.

The popularity technique, used in Netflix's 'Trending Now' and 'Popular on Netflix' video reels, recommends popular content for consumption, with popularity determined by widespread use. This technique is especially effective for new users, and can drive content consumption on platforms. However, it can also introduce prefix bias,[21] where popular items increasingly receive ratings while non-popular items get sidelined. The tendency to recommend mainstream content (with good user results) can produce an over-dependence on such content, which is generally associated with expensive licenses as discussed earlier, and can depreciate customer satisfaction for users with more niche tastes. As such, understanding how the main collaborative filtering techniques compare with the popular technique becomes especially relevant.

Another potential solution to the cold start problem is the random technique, which recommends items randomly to a user. This is less useful in practice, but serves as a good baseline strategy for evaluating recommender system performance.

Data overview

Before diving into recommender system buildout and evaluation, however, it is worth spending a little time getting acquainted with the data used for this study. Given Netflix's prominence in the industry, its use of multiple techniques for producing recommendations, and the practical benefit of the data being readily available (via Netflix's 2006 prize competition), this study focuses on evaluating recommender techniques based on Netflix user ratings of content. While this technically includes non-movies i.e. television shows, documentaries, etc., for the purposes of this report all such content is referred to as a movie.

The original data available from the Netflix prize competition spans 6 years from 2000-2006, with over 100M ratings of 17,770 movies by 480,189 users. Working with the full dataset is quite challenging, given the resources required for computation and object storage limits in R (for vectors, matrices, etc.). Instead, this analysis uses a smaller subset of the data that can still provide meaningful insights. Data for only a single year (2003) is examined, since this provides a large amount of data from one of the more recent years in the data but does not exceed R matrix and network operation limits. Examining data from a single year also reduces any additional noise from temporal trends.

[21] Nadimi-Shahraki, Mohammad-Hossein, and Mozhde Bahadorpour. "Cold-start problem in collaborative recommender systems: Efficient methods based on ask-to-rate technique." *Journal of computing and information technology* 22, no. 2 (2014): 105-113.

In addition, only data for active users and movies is examined (to reduce sparsity and address the cold start problem), so the data is subset to include movies with at least 100 ratings and users who have provided at least 50 ratings. Summary information for the final data used are shown in Table 1.

Overview of data used	
Total observations	8,467,727
Number of users who provided ratings	51,374
Number of movies rated	6,177
Average movie rating	3.4
Median number of movies rated	108.0

Table 1: Overview of data

Taking a closer look at the distribution of ratings by user (Figure 1), we can see that most users rate movies at 3 or higher. This is unsurprising, since users presumably only rate movies they see, which are the ones they are more likely to enjoy.

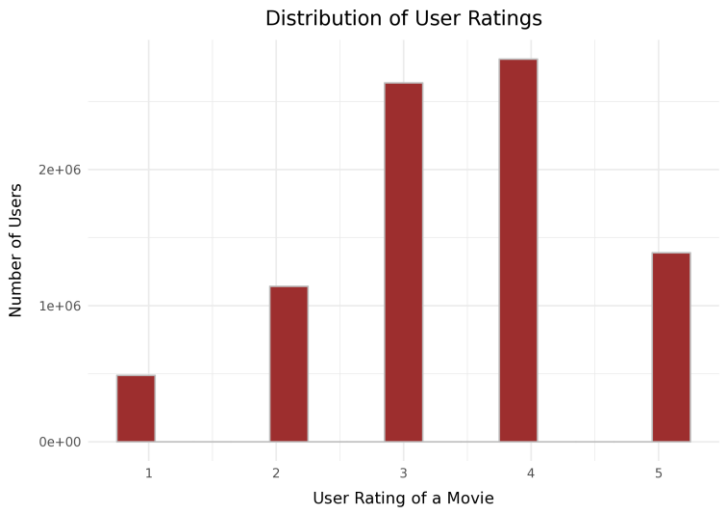

Figure 1: Distribution of user ratings

Looking at average rating by Netflix user in Figure 2, however, we do see variation in the rating tendencies of different users. To correct for this, in models for each recommender technique, ratings will be normalized by user to remove rating bias from individuals who consistently provide lower or higher ratings versus others. [22]

[22] Hahsler, Michael. *recommenderlab*. (2015).

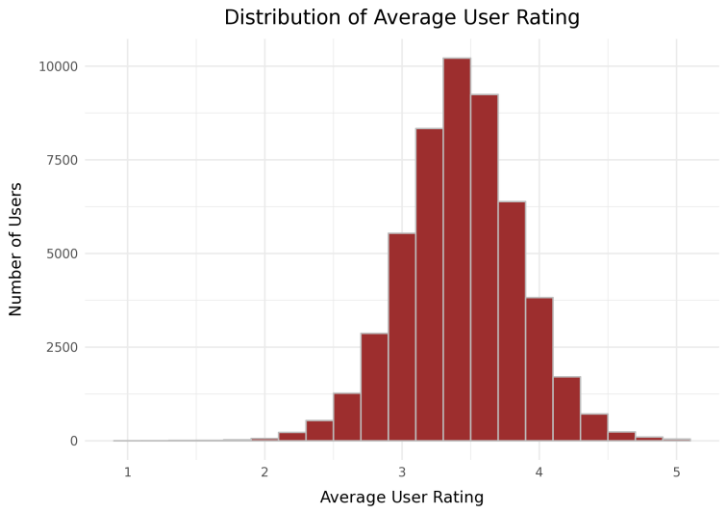

Figure 2: Distribution of average user ratings

Taking a closer look at the number of movies rated by each user (keeping in mind the minimum of 50 used to subset this data), a right-skewed distribution is observed (Figure 3). A number of users had very high movie ratings (such as 4000+) that would imply over 10 movies seen per day on average in 2003. This may be due to multiple individuals sharing an account, or due to the use of on-site surveys to get ratings of movies a user saw in the past.[23]

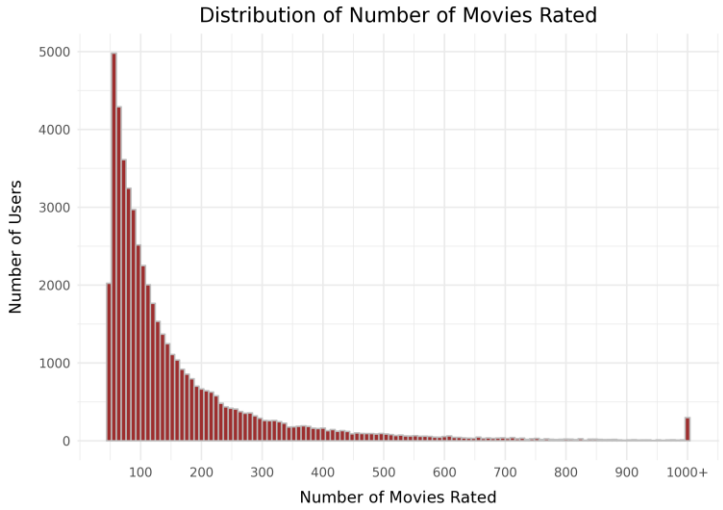

Figure 3: Distribution of Number of ratings per user

A list of the top 10 movies with the highest average rating and the highest number of total user ratings can be seen in Table 2 and 3 to help make the data more concrete. As communicated earlier, while the phrase ‘movie’ is used for each item, the actual content includes other material such as television shows. It is interesting to note how many of the top rated ‘movies’ for our 2003 data subset are seasons of different TV shows.

[23] While shared account use may have some implications for the effectiveness of these models, it is a limitation of the data set. This limitation is likely overcome in Netflix internal systems through the introduction of multiple profiles in accounts (feature released in 2013)

Top 10 'Movies': Highest average rating
Lord of the Rings: The Return of the King
City of God
Alias: Season 2
Raiders of the Lost Ark
CSI: Season 2
24: Season 2
CSI: Season 1
Family Guy: Vol. 2: Season 3
The Sopranos: Season 2
Alias: Season 1

Table 2: Top 10 best rated movies

Top 10 Movies: Highest number of user ratings
My Big Fat Greek Wedding
Catch Me If You Can
Two Weeks Notice
Sweet Home Alabama
Minority Report
Road to Perdition
Signs
Harry Potter and the Chamber of Secrets
Lord of the Rings: The Two Towers
The Bourne Identity

Table 3: Top 10 most rated movies

Network exploration

The collaborative filtering techniques being examined in this study rely on underlying relationships between users and movies as actors in different networks. For instance, we can examine relationships between movies that are seen together to gauge item similarity, or between users that tend to watch the same movies for user similarity. Prior to building formal recommender models, it is worthwhile to see what can be learned by examining networks that model such relationships. As such, this section further explores the three key networks involved in this data: user-movie, movie-movie, and user-user.

User-Movie Network

The user-movie network represents the basic relationship captured in all rows of the data, namely a user's decision to view a movie, which can be weighted by the rating value. An example of this two-node (bipartite) relationship in the network is visualized in Figure 4, based on random sampling of 200 users and all the movies they have rated. Random sampling is used due to processing constraints for a plot with all 57K+ nodes, but each movie watched by the sampled users is included to showcase the dense nature of the network, where many movies are seen/rated by a large number of users.

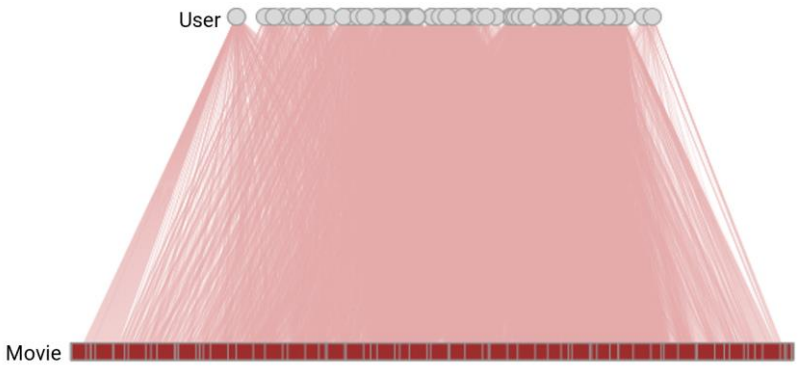

Figure 4: User-movie bipartite network (random sampled 200 users)

As we dig deeper, however, an important finding comes to light regarding how the number of ratings a movie has received (i.e. its degree centrality) is related to its average rating. A clear positive association exists between degree centrality and average rating. The logged value of degree centrality is used to enable linear modeling, since this transforms the right-skewed distribution of the number of ratings a movie receives to a relatively more normal distribution, as shown in Figures 5 and 6.

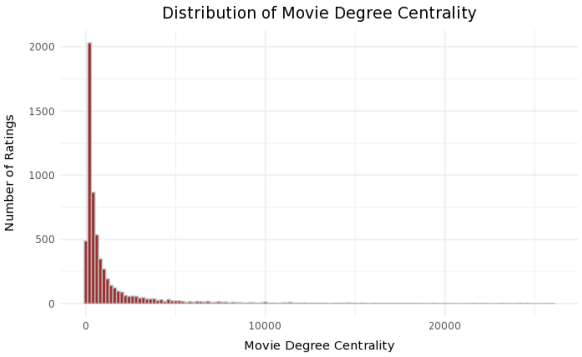

Figure 5: Distribution of Movie Degree Centrality

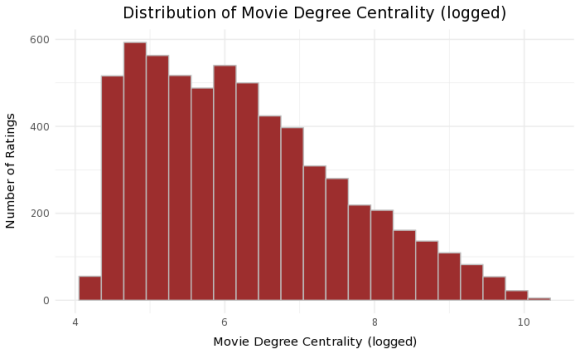

Figure 6: Distribution of Movie Degree Centrality after log transformation

The positive, statistically significant relationship between a movie's inbound degree and its average rating (Table 4) suggests that movies that receive more ratings also tend to be more well-liked. This makes sense intuitively, since movies that are popular among a large group of people (hence less likely to be niche) are more likely to be viewed and liked by others as well. This pattern could also be an example of the effective use of the 'popular' technique by Netflix, where movies that are seen by a large number of people are recommended to more and more people and can maintain high ratings due to their universal appeal.

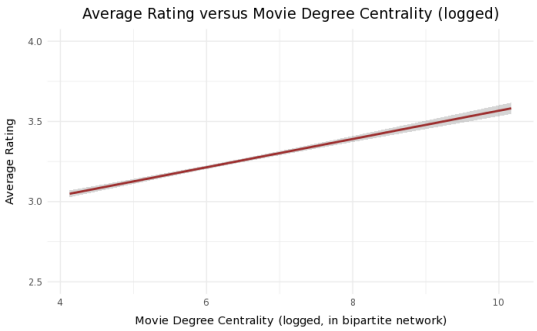

Figure 7: Average rating versus movies degree centrality (logged)

Formula: Average rating ~ log(Movie degree centrality)				
	Estimate	Std. Error	t value	Pr(> | t |)
(Intercept)	2.685234	0.027784	96.64	<2e-16 ***
log(Movie degree centrality)	0.088176	0.004357	20.24	<2e-16 ***
Residual standard error: 0.4489 on 6175 degrees of freedom				
Multiple R-squared: 0.06221	Adjusted R-squared: 0.06206			
F-statistic: 409.6 on 1 and 6175 DF	p-value: <2.2e-16			

Table 4: Linear regression of average rating on logged movie degree centrality

Looking at degree centrality for users in the user-movie network, the reverse pattern is observed (Figure 8). More specifically, users that rate a higher number of movies tend to give lower average ratings, with the negative relationship statistically significant (although of small magnitude, per Table 5). This may be because users who watch a lot of movies are 'binge-watching', or they run out of great content to see on the platform, and hence watch a wider range of movies that are less attuned to their tastes. Alternatively, the movies could generally be more sub-par, bringing their average rating down. It is also possible that the users' frequent movie viewing may raise the bar on their definition of good (because of over-exposure to movie cliches/plotlines), such that ratings are depressed down. For our modeling, this affirms the need to normalize the ratings by user, to ensure user rating bias doesn't influence recommendations.

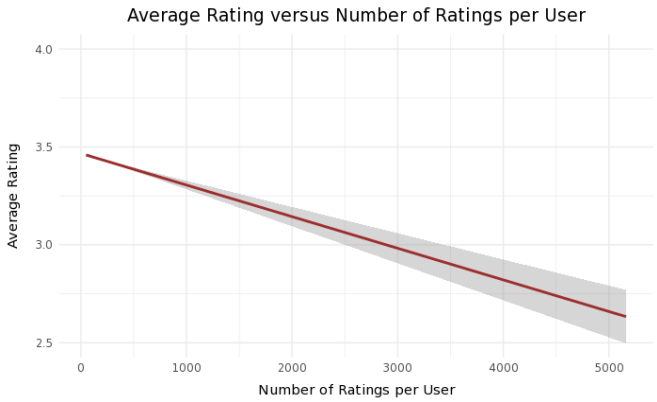

Figure 8: Average rating versus Number of ratings per user

Formula: Average rating ~ User degree centrality (i.e. number of ratings)				
	Estimate	Std. Error	t value	Pr(> | t |)
(Intercept)	3.467e+00	4.299e-03	806.47	<2e-16 ***
Number of ratings per user	-1.615e-04	1.409e-05	-11.46	<2e-16 ***
Residual standard error: 0.4845 on 28410 degrees of freedom				
Multiple R-squared: 0.004599		Adjusted R-squared: 0.004563		
F-statistic: 131.2 on 1 and 28410 DF		p-value: <2.2e-16		

Table 5: Linear regression of average rating on user degree centrality

Movie-Movie Network

In the movie-movie network, ties exists between movies that get watched together by a single user. This can provide insights into the value of item-based similarity techniques for recommender systems. Once again, random sampling is used to visualize the network in Figure 9, and a core-periphery pattern emerges. The core

here may be universally popular/well-liked movies that are seen often by people (and hence can be seen by the same person), while the periphery may represent movies that are less universal.[24]

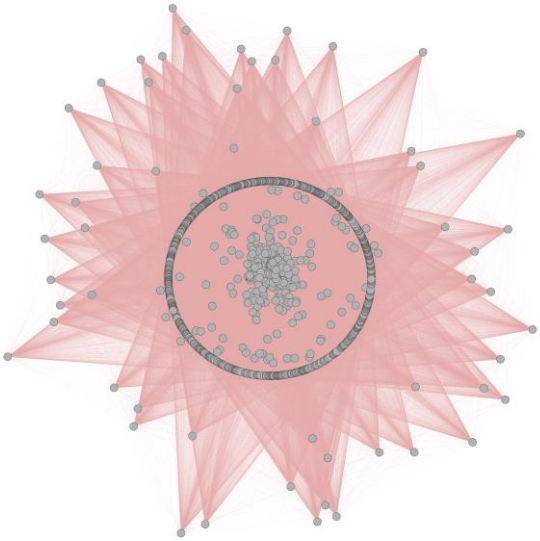

Figure 9: Movie-Movie network, based on 1000 random nodes sampled from network of 6177

Once again, the relationship between network centrality and average movie rating is examined. In this case, looking at movie-movie degree centrality was found to be less meaningful. This could be because most movies in the data had similar values for degree, likely due to the minimum thresholds on movie and user ratings applied to our data and the scale of the data. Two alternative measures of movie centrality in this network, namely closeness and eigen centrality, are examined instead.

In this network, movies with lower closeness are those with fewer direct links and longer indirect links to other movies, i.e. they are rarely seen with other movies. Here, higher closeness centrality implies a movie that either gets seen often with other movies, or is likely a few clicks away from being watched by a user. Examining the relationship between movie closeness and average rating, an interesting pattern emerges. Movies that tend to be less close to others, possibly representing more niche tastes (e.g. cult movies of a certain type), actually tend to have higher ratings than movies that have high closeness (Figure 10).

Conversely, movies with high-closeness i.e. those that are either watched often with other movies or may be watched next with higher probability, are actually associated with lower average ratings (although it is worth noting the floor is an average rating of 3). This may be because such movies represent a safe but unexciting movie choice for a lot of users, and hence receive middling reviews. While other explanations are possible,

[24] It is worth noting that if a minimum threshold of 100 ratings per movie isn't used, this network pattern may appear different.

ultimately, this result suggests that an item-based technique which only recommends movies based on rating similarity may not be the best approach. The technique could lead to over-recommending similar high-closeness movies that may not be well received, and may rarely find reason to suggest more 'isolated' movies with lower closeness.

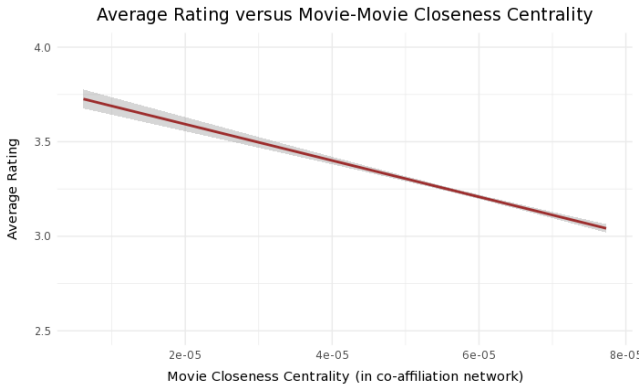

Figure 10: Average rating versus movies-movie closeness centrality

Formula: Average rating ~ Movie closeness centrality				
	Estimate	Std. Error	t value	Pr(> | t |)
(Intercept)	3.785e+00	2.851e-02	132.75	<2e-16 ***
Movie closeness centrality	-9.606e+03	-9.606e+03	-19.67	<2e-16 ***
Residual standard error: 0.4497 on 6175 degrees of freedom				
Multiple R-squared: 0.05896		Adjusted R-squared: 0.05881		
F-statistic: 386.9 on 1 and 6175 DF		p-value: <2.2e-16		

Table 6: Regression of average rating on movie-movie closeness centrality

The results on movie-movie eigen centrality provide an added dimension to our analysis. While movies that get watched with many other movies have depressed average ratings, if a movie is watched with other movies that are also frequently viewed with other movies, the trend reverses (Figure 13). Here, being connected to multiple 'popular' movies i.e. high eigen centrality, is associated with higher ratings. This is possibly because a movie being seen together with lots of other popular movies likely means it is popular itself, and hence is well-reviewed by most users. This relationship, modeled using logged movie-movie eigen centrality to facilitate the use of linear regression(Figures 11-12), is statistically significant, as shown in Table 7.

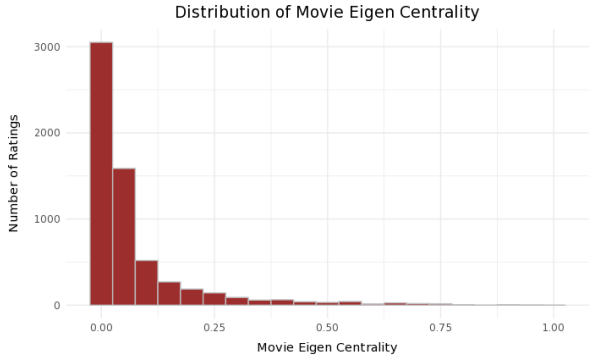

Figure 11: Distribution of Movie Eigen Centrality

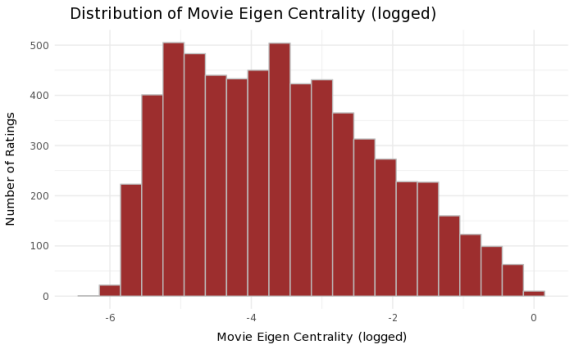

Figure 12: Distribution of Movie Eigen Centrality after log transformation

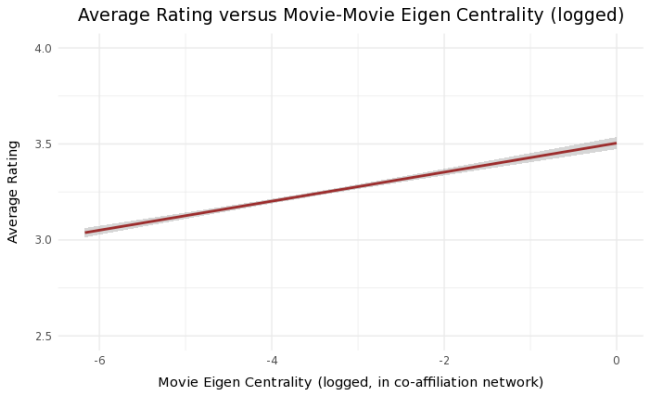

Figure 13: Average Rating versus movie eigen centrality

Formula: Average rating ~log(Movie eigen centrality)				
	Estimate	Std. Error	t value	Pr(> | t |)
(Intercept)	3.504601	0.016026	218.68	<2e-16 ***
log(Movie eigen centrality)	0.075645	0.004206	17.99	<2e-16 ***
Residual standard error: 0.4519 on 6175 degrees of freedom				
Multiple R-squared: 0.04978		Adjusted R-squared: 0.04963		
F-statistic: 323.5 on 1 and 6175 DF		p-value: <2.2e-16		

Table 7: Regression of average rating on movie-movie eigen centrality

User-User Network

The third network examined creates ties between users that have watched the same movie, and can be used to draw insights for collaborative filtering techniques that rely on user-based similarity (i.e. UBCF). Unfortunately, given the large number of users in the overall dataset (over 50K), the data had to be heavily subsetted to look at a co-affiliation network between users. The decision was made to only examine this network for highly active users on the Netflix platform, represented as users who rated 200 or more movies in

a single year (i.e. the year of our data, 2003). Three outliers with very large number of movie ratings (i.e.. 4000+) were excluded from this network as well.[25]

A randomly sampled subgraph of the user-user network is shown in Figure 14, which indicates a core-periphery network, suggesting that some users tend to be more unique than others in their movie preferences.

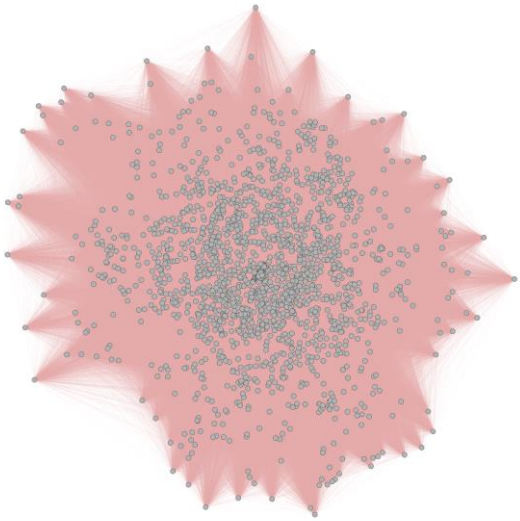

Figure 14: User-User network, visual of 2000 nodes randomly sampled out of 11, 869

Like the movie-movie network, given the high number of users with comparable degree, the association between network centrality and average rating is examined via closeness and eigen centrality measures instead. Here, individuals with low user closeness are those that are far removed from other users i.e. tend to have more niche tastes, while those with higher closeness tend to have more movies in common with other users and share similar tastes with a broader network. (Figure 15 and Table 8)

Keeping in mind that this data is already a subset for users who rated 200+ movies, we see that users with high closeness tend to provide higher ratings on average. This could indicate that users with wider or more mainstream interests tend to provide higher movie ratings. Alternatively, it could indicate that Netflix's recommender algorithm was finding meaningful patterns in user preferences for users who had a large network of similar users to learn from, but not so much for users who had a sparser network. This latter implication may mean that user-based techniques, while helpful, could have limited value for users with a smaller number of ratings if their taste profile is more unique. A technique like SVD which relies on finding underlying patterns could work well in such a situation.

[25] These users were removed from this subset network where they served as outliers, whereas they follow the natural distribution of the data in the original set. The exclusion primarily improved results for eigencentrality, which was being unnecessarily influenced by connections to these users otherwise.

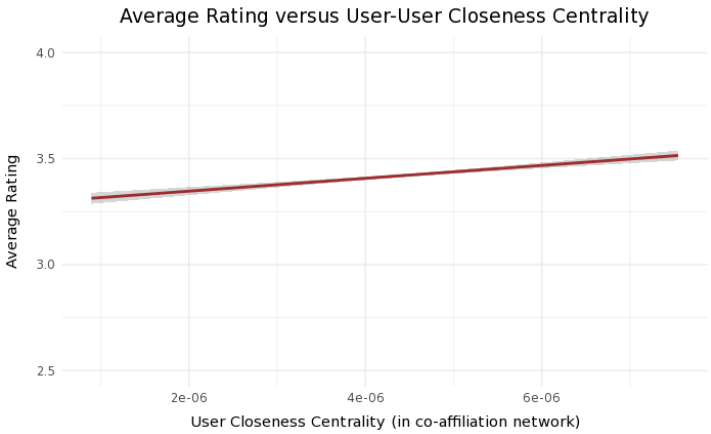

Figure 15: Average rating versus user closeness centrality

Formula: Average rating ~ User closeness centrality				
	Estimate	Std. Error	t value	Pr(> | t |)
(Intercept)	3.281e+00	1.664e-02	197.203	<2e-16 ***
log(User closeness centrality)	3.144e+04	3.650e+03	8.613	<2e-16 ***
Residual standard error: 0.4321 on 11867 degrees of freedom				
Multiple R-squared: 0.006213		Adjusted R-squared: 0.006129		
F-statistic: 74.19 on 1 and 11867 DF		p-value: <2.2e-16		

Table 8: Regression of average rating on user-user closeness

Similar to closeness, looking at user-user eigen centrality, a statistically significant positive relationship is observed (Table 9). Log-transformed eigen centrality is used to showcase the relationship given the distribution this measure takes (as in the movie-movie network). These results imply that users who tend to have a lot of movies in common with other users, where these other users have a lot of shared viewing patterns themselves, are associated with higher ratings. This reinforces the implications from the analysis of the user-user closeness centrality where users with lower eigen centrality, possibly those with unique tastes, could be better recommended by using techniques such as SVD.

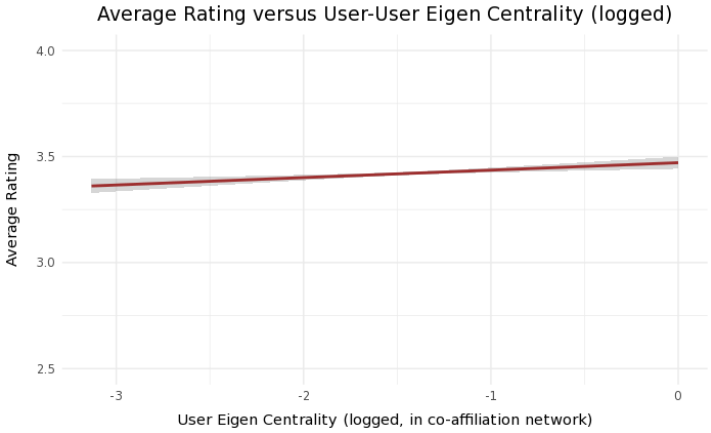

Figure 16: Average rating versus user eigen centrality

Formula: Average rating ~ log(User eigen centrality)				
	Estimate	Std. Error	t value	Pr(> | t |)
(Intercept)	3.471050	0.014603	237.691	<2e-16 ***
log(User eigen centrality)	0.035005	0.009662	3.623	2.92e-4 ***
Residual standard error: 0.4325 on 11864 degrees of freedom				
Multiple R-squared: 0.001105		Adjusted R-squared: 0.001021		
F-statistic: 13.13 on 1 and 11864 DF		p-value: 0.0002925		

Table 9: Regression of average rating on user-user eigen centrality

Recommender System Evaluation Approach

While the findings from network exploration provide interesting insights into the possible usefulness of different recommender systems for our data set, more formal model buildout and evaluation for each technique is still needed.

In order to ensure separation between model training and evaluation, a 75%-25% split of users into training and test data is implemented prior to model creation. Relying on a built-in evaluation function in the recommenderlab package in R, for each user in the test set, 20 ratings were used to train or produce predictions for that user, and these predictions were then evaluated on the remaining ratings in the test data.

For model evaluation, both effective performance and cost efficiency are assessed, in line with our research goal articulated earlier. To measure predictive performance, our approach examines model results based on classification and prediction accuracy.

Classification Accuracy

Classification accuracy lets us assess the frequency with which the recommender techniques evaluated make correct or incorrect decisions about whether a movie will be liked by a customer. In most cases, users wish to distinguish between good and bad movies, not between good movies. Per Henlocker et. al, "I don't necessarily need the best movie, only a movie I will enjoy". In our modeling, a rating of 4 out of 5 stars is used as a cutoff to establish a movie as being well-received, with ratings of 3 or below being regarded as undesirable.[26]

For evaluation, the top N list of recommended items for a user is compared to the test list of rated items for that user. Items recommended are seen as successful or relevant if they occur in the test list of rated items and have a high rating (i.e. 4 or more). Performance can be quantified visually using a receiver operating characteristics (ROC) curve, or through established metrics such as precision and recall.

[26] Herlocker, Jonathan L., Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. "Evaluating collaborative filtering recommender systems." *ACM Transactions on Information Systems (TOIS)* 22, no. 1 (2004): 5-53.

The ROC curve looks to measure the extent to which the recommender system can differentiate between relevance and noise.[27] It plots the probability of a movie being identified correctly (encoded as true positive rate TPR) in a top N list versus the probability of a false alarm (encoded as a false positive rate FPR). A larger amount of area under this curve indicates better performance.[28]

Precision gives the number of correctly recommended items as a fraction of the number of total recommended items. Recall, on the other hand, gives the number of correctly recommended items as a fraction of the total number of useful recommendations (commonly denoted by N in the top N classification). It is worth noting that these metrics are inversely related, such that high precision implies low recall and vice versa.[29] Given challenges associated with measuring recall precisely and the biases it can sometimes introduce, our evaluation will focus more on precision, especially since we do not need a list of every potentially relevant movie, just a subset e.g. top 10 that a user will find useful.[30]

Moreover, having many recommendations that are accurate out of the total set recommended (i.e. high precision) can be a useful measure since for most users, the cost of users being recommended an undesirable movie (false positive) or missing a single desirable movie (false negative) are both low, while the benefit of watching a desirable movie from the recommendations is large.[31]

Predictive accuracy

Predictive accuracy measures how closely a system's predicted rating for a movie mirrors the user's actual rating. Our evaluation will focus on looking at Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) since these are in the original unit of ratings, although Mean Squared Error (MSE) is also included for reference. While the exact rating of a movie is of less interest than its overall ranking to a user, the results can still provide a helpful evaluation of the algorithms' directional accuracy. In fact, MAE has well-established use in testing the significance of a difference between the predictive performance of two systems.[32] MSE and RMSE can also be helpful because they provide evidence of extremely inaccurate rating predictions, since they punish large errors more.

Cost efficiency

[27] Ibid (2004).
[28] Hahsler, Michael. *recommenderlab*. (2015).
[29] Ibid, (2015)
[30] Herlocker, et. al. "Evaluating collaborative filtering recommender systems". (2004).
[31] Ibid, (2004)
[32] Ibid, (2004)

In addition to examining effective performance, the model and prediction runtime for each algorithm is used to quantify the resource/cost efficiency of each technique. Model runtimes will be higher for model-based techniques such as IBCF that first find all matching movie sets and then evaluate a specific prediction when requested. Conversely, prediction run-times will be higher for memory-based techniques such as UBCF that find similar user neighborhoods for each user in real-time when the prediction request is made. Algorithms with high total runtime (model + prediction) will be seen as less efficient than those that run faster.

Model Results

Top-N classification performance results for all five algorithms are shown in Figure 17 and Figure 18. From these, we can conclude that in our 2003 movie rating network, the SVD algorithm performs the best in predicting top movies for users. It provides the ROC curve with the largest area below it (Figure 17), although overall true and false positive rates across algorithms are relatively low. The SVD model is also associated with the highest recall rates for all the top N recommendations (N from 1 to 50), followed by popularity and user-based models (Figure 18). For precision rates, SVD and popularity perform almost the same for N greater than 10, with 20% (or 2 out of 10) movies being correctly recommended in a top 10 classification.

In general, the popular model performed well on classification, with results slightly behind SVD. This suggests that recommending universally watched movies continues to be an effective strategy for customized content suggestions, although there are still strategic considerations to keep in mind as discussed earlier. These results are also consistent with earlier network findings on correlations between high movie viewership and average rating.

The user-based collaborative filtering model had middling results on classification, well behind SVD and popular but ahead of the item-based algorithm. This may be because of the present of users with more unique tastes, as discovered in the user-user network analysis.

The item-based collaborative filtering model, however, performs poorly across classification measures, with performance close to that of the random algorithm. This may be due to high variation in tastes across users, where looking to infer item similarity based on user ratings may not be producing accurate results. This is in keeping with earlier analysis of the movie-movie network, which suggested that movies with higher closeness were associated with lower ratings.

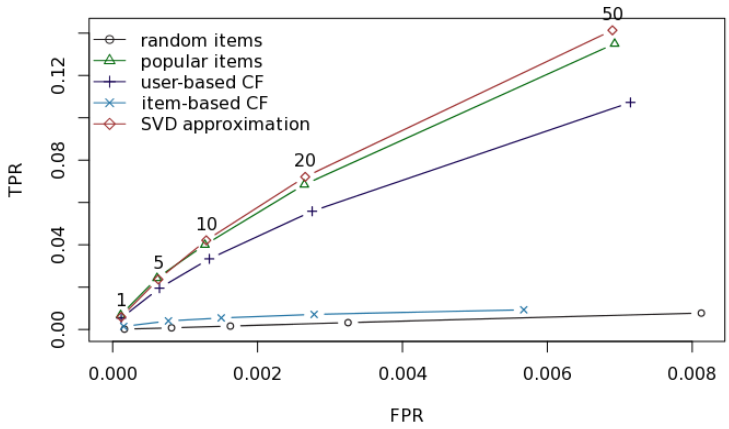

Figure 17: ROC Curves for 5 algorithms in predicting Top-N lists

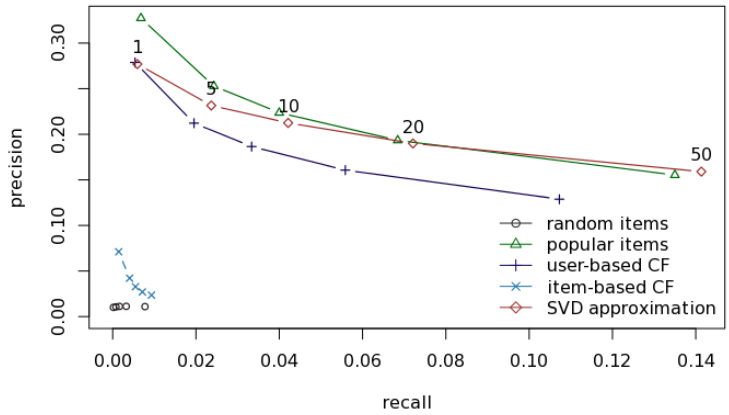

Figure 18: Recall-Precision curves for 5 algorithms in predicting Top-N lists

Looking at predictive accuracy, model performance is consistent with observations on classification accuracy (Figure 19). Namely, SVD and popular provide the lowest levels of RMSE and MAE, while the random and item-based algorithms have poor accuracy. It is worth noting that the margin of error for SVD, popularity and UBCF is below 1, indicating that a recommendation for a movie with a rating of 4 out of 5 can likely only result in a low of 3.

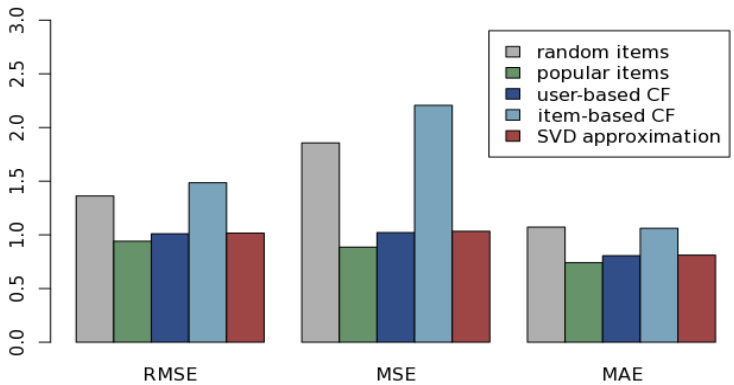

Figure 19: Predictive accuracy metrics for 5 algorithms in predicting ratings

Looking at algorithm efficiency, random and SVD appear as the least resource-intensive based on total runtime. Popular takes a non-trivial amount of runtime for top-N classification although it is quite fast for

rating prediction. More notably, UBCF is extremely resource-intensive in terms of runtime, taking over 178x the amount of time SVD took. IBCF also proves inefficient, although it is not as slow to run as UBCF.

	Top-N Classification Prediction			**Rating Prediction**		
Algorithm	Model Runtime (sec)	Prediction Runtime (sec)	Total runtime (sec)	Model Runtime (sec)	Prediction Runtime (sec)	Total Runtime (sec)
Random	0.069	124.403	124.472	0.067	54.937	55.004
Popular	1.211	4112.694	4113.905	1.013	31.599	32.612
UBCF	1.228	36012.91	36014.138	0.914	34170.9	34171.81
IBCF	12666.36	16.449	12682.809	12895.81	12.755	12908.57
SVD	74.248	127.818	**202.066**	73.395	56.64	**130.035**

Table 10: Computation times for all 5 algorithms

Overall, given SVD's effectiveness in producing relevant recommendations and its low-cost runtime, it emerges as the best technique across the dimensions of our evaluation. The use of popularity-based recommendations also proves effective, although with slightly slower performance for generating top-N suggestions. The additional benefit of SVD over the popular technique is that it doesn't constrain a platform to recommend mainstream content that is often associated with high license costs. Moreover, in a broader dataset that captures users with fewer ratings and more niche tastes, SVD has the added advantage of picking up on their preference patterns to produce more meaningful recommendations.

While user-based approaches make sense intuitively, our findings suggest middling performance with extremely high computation costs, indicating cautionary use of UBCF. Direct item-based approaches provided poor performance as well, suggesting that collaborative filtering may not be appropriate for item-based recommendations. It is possible that content-based recommendations for similar items may produce better performance.

Overall, in a practical context, our results indicate that even a simplistic recommender model can produce recommendations that correctly capture 2 out of a user's top 10 preferred movies to watch next.

Conclusion and Limitations

These results suggest that an effective recommender system for online movie streaming should leverage collaborative filtering based on the Singular Value Decomposition (SVD) algorithm. This technique produces the best results on Netflix's user data, well ahead of performance on other collaborative filtering techniques.

However, our research has some important limitations that are worth noting. Firstly, our study relies on a limited subset of data from a single year with additional restrictions placed on the minimum number of movie and user ratings. The applicability of our findings to broader datasets will need to be validated. Secondly, this research relied on publicly available data that was released in 2003, before Netflix introduced its online streaming capabilities or user-specific profiles within an account. Ideally, this study should be replicated on more recent data. Thirdly, our study is limited to a select number of collaborative filtering techniques, but other approaches such as content-based and hybrid techniques are worth evaluating as well. For future research, studies which offset some or all of these limitations could add substantial value.