A Hybrid Approach to Recommender Systems

CONTENT ENHANCED COLLABORATIVE FILTERING

JONATHAN OHLSSON

JESPER SANDSTRÖM
A Hybrid Approach to Recommender Systems

Content enhanced collaborative filtering

Jonathan Ohlsson and Jesper Sandström

Degree Project in Computer Science DD143X
Supervised by
Michael Schliephake
Examiner
Örjan Ekeberg

May 10, 2016
Abstract

Recommender systems help shape the way the internet is used by leading users directly to the content which will interest them most. Traditionally, collaborative recommender systems based purely on user ratings have been proven to be effective. This report focuses specifically on film recommender systems. It investigates how the film content parameters Actor, Director and Genre can be used to further enhance the accuracy of predictions made by a purely collaborative approach, specifically with regards to the set of films chosen when performing the prediction calculations. The initial results showed that relying solely on content in this selection led to poorer predictions due to a lack of ratings. However, the investigation finds that using a hybrid approach between the two selection techniques with a bias for content solved this problem as well as increasing the overall prediction accuracy by over 11%.
Sammanfattning

Rekommendationssystem hjälper till att forma hur internet används genom att dirigera användare direkt till innehåll som är relevant för dom. Traditionellt sett har kollaborativa rekommendationsystem visat sig vara effektiva. Denna rapport fokuserar på rekommendationsystem för filmer. Den undersöker hur meta-informationen Skädespelare, Regissör samt Genre kan användas för att ytterligare förbättra prediktionen hos ett kollaborativt system, specifikt under urvalsprocessen av filmer att använda under prediktskalkylen. Initiala resultat visade att endast förliata sig på dessa meta-parametrar resulterar i dåliga prediktioner, på grund av en avsaknad av betyg att använda i kalkylen. Dock visar undersökningen att en hybrid mellan selektionsmetoderna där en partiskhet mot meta parametrarna användes ökade prediktionsnoggrannheten med över 11%.
Contents

1 Introduction ................................................. 4
   1.1 Recommender Systems ..................................... 4
   1.2 Purpose .................................................. 4
   1.3 Limitations ............................................... 5
   1.4 Report Overview ......................................... 5

2 Background .................................................. 5
   2.1 Collaborative Recommender Systems ....................... 5
   2.2 Content-Based Recommender Systems ...................... 6
      2.2.1 Advantages of Content-Based Recommender Systems .... 7
      2.2.2 Disadvantages of Content-Based Recommender Systems ... 7
   2.3 Item and User Similarity Methods ......................... 8
   2.4 Prediction Computation .................................... 10

3 Method ....................................................... 11
   3.1 Purely Collaborative Approach .......................... 11
   3.2 Content-Based Enhancement .............................. 11
   3.3 Improvements to the Content Enhanced Algorithm ........ 13
   3.4 Dataset .................................................. 15
   3.5 Content Fetching System ................................ 15

4 Results ....................................................... 16
   4.1 Purely Collaborative ..................................... 16
   4.2 Preliminary Content Enhanced Approach ................... 16
   4.3 Content Enhanced Improvements .......................... 17
   4.4 Summary of Results ...................................... 20

5 Discussion .................................................... 21
   5.1 Initial Content-Based Approach .......................... 21
   5.2 Improvements the Content-Based Approach ............... 21
   5.3 Effects of Individual Content Parameters ............... 21
   5.4 Strengths of Content-Enhanced Filtering .................. 22
   5.5 Limitations of Content-Enhanced Filtering ............... 22
   5.6 Evaluation of This Study ................................ 22
   5.7 Conclusion ............................................... 23
   5.8 Future Works ............................................. 23
1 Introduction

1.1 Recommender Systems

In the past decades the rapid increase in the amount of content on the Internet has called for the implementation of systems to automatically help users find the things that interest them most. A tool used for this purpose is known as a recommender system. These systems often rely on information gathered from analyzing the items users have purchased, have spent the most time viewing, or have explicitly rated. This report will explore various approaches to recommender systems for films.

Companies and consumers alike stand to benefit from implementing recommender systems. These systems help to enhance the user experience which can lead to an increase in user retention [1]. Furthermore, companies can use these systems to gather information about their customers’ preferences, which can help them track trends and tailor their goals. For example, a web service such as Netflix can greatly improve its user experience by guiding users directly to the content which will be of interest to them. By doing this they can avoid unnecessarily taxing the patience of their users. A web service such as Amazon can help users navigate their catalogue more efficiently and thus improve the probability of a user making a purchase. These benefits in efficiency are amplified for users with slower connections or limited bandwidth, since they do not have to navigate through uninteresting content [1]. In the case of films, a recommendation system can save users both time and money as the cost of a bad recommendation is the amount of time the user spends watching the film and the monetary cost of the film [1].

Recommender systems present two main challenges. These systems need to be implemented for thousands (possibly millions) of users with millions of data points. This requires the algorithms to be scalable. However, in order to actually fulfill their purpose these systems need to produce high quality recommendations. There is therefore a necessary tradeoff between the efficiency and efficacy of such systems. Many solutions have been proposed to solve this dilemma such as systems based on correlation between different users and the items they like, and solutions based on the analyzing the items themselves.

This investigation will focus on two common approaches to creating recommender systems: collaborative and content-based filtering. Collaborative filtering is grounded in the idea that similar users will like similar items. In this approach all calculations are based on ratings given by users on items [2]. The second, content-based filtering, is based on analysing item meta-data in order to generate predictions [3].

1.2 Purpose

Recommender systems purely based off of the collaborative approach have been shown to be very effective [2]. However, there is always motivation for improvement. The purpose of this report is to investigate how the principles of a content-based approach can be used to enhance a collaborative system used for film recommendations. The research question is therefore: How can film content be used to enhance the quality of predictions in a collaborative recommender system?
The dataset which will be used for user ratings is the MovieLens Dataset, which provides information about which users liked which films at specific points in time. The films in this dataset can be cross-referenced with the Open Movie Database in order to gather more meta-data about the films in question, which provides the meta-data for the content-based approach.

1.3 Limitations

While it may be ideal to create a whole recommender system from the ground up using a new fused approach of the collaborative and content-based approaches, this would be far outside the scope of this investigation. We will focus our efforts on evaluating how content parameters can be used to enhance an existing collaborative system. Specifically in regards to the selection of films analyzed during the prediction computation. In terms of content related to the films, we will limit ourselves to the information available from the MovieLens dataset and the Open Movie Database.

It is important to note that we will not focus on the performance of the algorithm. This report will be limited to evaluating the quality of the actual recommendations, not how quickly the algorithm runs.

1.4 Report Overview

This following section of the report will outline the underlying theory used in this investigation. This will be followed by a method section which will describe how the investigation was carried out in detail. The results this method yielded will then be listed followed by an in-depth discussion of the implications of these results and the report’s conclusions.

2 Background

2.1 Collaborative Recommender Systems

Collaborative Recommender Systems follow the idea that similar users will like similar items. There are two distinct strategies for approaching this problem. The traditional strategy is known as user-based collaborative filtering, which involves determining the similarity between users based on similar tastes in items. Items from this set can then be recommended to the user. While this approach can be effective in terms of finding accurate recommendations, it can often be inefficient in systems where users only rate a small subset of the total number of items in the system. A user-item matrix is often constructed to structure the data and correlations between users and items. This is pictured in figure 1 where all the users occupy one dimension and all the items occupy the other. Following this approach the user-item matrix (figure 1) generated in the described collaborative algorithm would be very sparse, which means the algorithm will spend the majority of its time sifting through empty datapoints. This presents a scalability problem in datasets where the number of users grows much faster than the number of items (this follows intuitively from the fact that there will always be more people watching films than films being made). In this case, the second strategy presents a more viable option [2].
\[ U_{u,i} = \begin{pmatrix} i_1 & i_2 & \cdots & i_n \\ u_1 & r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ u_2 & r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_m & r_{m,1} & r_{m,2} & \cdots & r_{m,n} \end{pmatrix} \]

Figure 1: The user-item matrix, where each cell contains the rating for an item from a specific user. Following either the rows or the columns gives us vectors of users or items.

The second strategy is called item-based collaborative filtering which uses correlations between items in terms of rating from users. Two items will be considered similar if they have been rated by similar users. From this the nearest items to those that a user has already liked can be suggested to the user. The difference to the other approach is that the similarity is computed between the columns of the matrix in figure 1 instead of the rows. One of the main benefits of this approach is that it is more scalable. Research within this field has also shown additional benefits of this approach. For example, Sarwar et al. found that these systems not only perform significantly better than user-based implementations, but also yield more accurate prediction than the best available user-based filtering systems.

A major issue associated with collaborative filtering is its inability to handle new items in the system, which is known as the Cold Start problem. A new film will only be recommended in a collaborative system once it has been rated enough times. In order for it to be rated, however, it often first has to be recommended. As such, new films will often be confined to the realm of obscurity unless its ratings are driven by external factors. The same idea can also be applied to new users in a system. The system cannot know which movies to recommend to a user who has not rated enough movies, or any movies whatsoever.

### 2.2 Content-Based Recommender Systems

Content-based recommender systems present a drastically different approach. They work by analyzing the various properties of items in a system, instead of looking at the item’s ratings. For example, a content-based system may recommend new films to users based on the film genres they have already rated. Lops et al. describes a content-based system as having three components:

**Content Analyzer**

The content analyzer attempts to structure information about items. This is especially important in the case on highly unstructured items such as web pages consisting of only raw text. The items are processed using feature extraction technique and transformed into a format usable by the following components.

**Profile Learner**

The profile learner generalizes the users preference data, so that it can be used as to make predictions about other items the user might be interested in. In order to achieve this, machine learning techniques are usually employed.
Filtering Component
The filtering component computes the difference between the available items and items within the user profile. The result is a list of potentially interesting items. There are many techniques for calculating the similarity of items, a few of them outlined above.

In order to construct the filtering component, the system will need a training set so it can build the user’s profile. That is, it will need a set of items rated by the user. In order to construct the filter, the system needs to analyze the user’s feedback. This feedback can be explicit, for example a binary rating (like/dislike), a numerical scale (1-5 for example) or in some cases user inputted comments [6] can be used to decide if an item has been appreciated by the community.

The advantages of explicit feedback is that it is simple to understand and makes it more transparent to the user how the recommendation system operates. However, they require active engagement from the user, which is not always easy to get. On the other hand, implicit feedback systems try to make assumptions on user preferences based on how the user behaves in the system.

In the dataset used in this report, feedback is provided explicitly using a scale (from 1 to 5) that the users enter for each item. This means the system does not have to deal with implicit feedback.

2.2.1 Advantages of Content-Based Recommender Systems
There are several advantages of content-based recommender systems. They are user independent and, unlike collaborative recommender systems, they can make recommendations without any additional user input. Another advantage is that content-based systems are very amenable to new items added to the system. A common problem in the collaborative recommender system is that new items do not have any ratings from users. A content-based system does not rely on other users’ input and therefore overcomes this problem gracefully.

Transparency to the user is another advantage of content-based systems, since the user can follow their own input and to some extent understand why a recommendation has been made. This correlation is not as clear in collaboration-based systems.

2.2.2 Disadvantages of Content-Based Recommender Systems
There are, however, a number of shortcomings inherent to content-based systems. One of these is the limitations of the content analysis. Significant amounts of data are required if a system is to be able to effectively identify correlations between properties of items and a user’s preferences [6]. This is not required by the collaborative approach since it only needs to identify similarities between the users themselves. Another drawback is the tendency of content-based systems to be overspecialized [6]. Take, for example, the situation of a user only having liked movies directed by Quentin Tarantino; the recommender system would naturally just continue to recommend movies by this director. This problem is also known as the serendipity problem [6].

The concept of quality may also be lost to a content-based system. Two movies of the same genre may appear very similar to such a system, and thus
prompt a recommendation, even if the first movie was widely-regarded to be a cinematic masterpiece and the other a waste of time and money. Without ratings, the system may be operating under false assumptions.

Finally one of the greatest shortcomings is the fact that the content-based systems rely solely on user history in order to make predictions, meaning that a new user (i.e. one with no history of ratings) would not be able to benefit from the recommender system. This mirrors the Cold Start problem in collaborative systems.

2.3 Item and User Similarity Methods

In all versions of recommender systems some kind of measure is required to determine the similarity between items or users.

Cosine Similarity

A popular measure used to determine the similarity between two items is to use cosine similarity. By representing the items as vectors of ratings in a user space, the cosine angle of these vectors can be used to determine the similarity of the items. This is shown in figure 2.

![Figure 2: The similarity between two items, x and y, is calculated using the angle $\theta$ between the two item vectors.](image)

A flaw with this kind of measurement is however that it does not take into consideration the magnitude of the vectors, only the angle. Consider a set of two vectors of ratings for two different films. The first, seemingly beloved by all, has received the rating 5 by all users. Whereas the other has only received ones by all users. These two vectors are identical in all aspects except for magnitude. However, the system would consider these films equal. This is shown in figure 3.
Let $i, j$ represent two items, their cosine similarity $sim(i, j)$ is calculated as follows \[2\]

$$sim(i, j) = \cos(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\| \cdot \|\vec{w}\|}$$

(1)

**Adjusted Cosine Similarity**

As stated, the cosine similarity does not take the magnitude of the vectors into account. If two users have rated the same items, but with sufficiently different magnitudes, this will not be taken into account. This is especially important to consider in item-based systems \[2\]. The adjusted cosine similarity takes this into account by subtracting the user average from each of the co-rated items. Let $U$ denote the co-rated items, $R_{u,i}$ is the rating from user $u$ to item $i$, and $\bar{R}_u$ is the average rating of user $u$. The similarity is calculated as \[2\]

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2 \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}}}$$

(2)
Figure 4: Illustration of the adjusted cosine similarity. The similarity between
two items, \( i \) and \( j \), is calculated by looking at the correlated items.

Figure 4 shows an example of how adjusted cosine similarity can be used
to compare two items based on user ratings. Each column is a film, and each
row represents one user, with the value being their rating for the film. The
highlighted rows show users which have rated both items. These rows are the
ones used in the calculation.

Correlation-based Similarity

In a correlation-based approach, the similarity of two items are measured using
the Pearson-r correlation. In order to do this, the co-rated items are isolated
and the similarity is calculated as follows [2]

\[
sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \overline{R}_i)(R_{u,j} - \overline{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \overline{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \overline{R}_j)^2}}
\]  

(3)

Which is similar to the adjusted cosine similarity, except we instead use \( \overline{R}_i \),
the average rating of item \( i \).

2.4 Prediction Computation

Simply knowing which items are similar is only half of the problem, some kind
of prediction as to how a user would rate an item must be made taking the
similarity into account. One such measure is the Weighted Sum, where the
predicted rating for item \( i \) for user \( u \) is given by

\[
P_{u,i} = \frac{\sum_N (s_{i,N} \cdot R_{u,N})}{\sum_N (s_{i,N})}
\]

(4)

Where \( N \) is the set of items similar to item \( i \). The weighted sum method
has been shown to be efficient in calculating predictions [2].
3 Method

This chapter will first describe the implementation of the collaborative filtering algorithm that laid the foundation for this investigation. It will then describe an overview of how a content-based ideology was used in an attempt to enhance the accuracy of the predicted ratings.

The implementation was written in the programming language R, version 3.2.3 [7], since it provided a large amount of tools for statistical analysis. It also allowed us to focus on testing improvements to the algorithm, instead of debugging the code. We focused solely on improving the quality of predictions made (optimizing the algorithm was outside the realm of this project). The R-dependencies used are hydroGOF [8] and jsonlite [9].

To measure the error in the generated predictions we used root-mean-square error (shortened RMSE) as well as mean absolute error (shortened MAE), two common measures of error in statistics.

3.1 Purely Collaborative Approach

As previously described, the collaborative approach used ratings across multiple users as a measure of similarity and to calculate the prediction. Our approach was based on the method described by Sarwar et. al. The similarity was calculated using adjusted cosine similarity (2), since previous reports have proven its accuracy [2]. We also used the Weighted Sum (4) method for calculating our predictions, as we outlined in section 2.4.

To calculate a user’s prediction for a film, we first removed the film from the set of all films rated by the user. We then calculated the similarity between the film and all other films in the remaining set using adjusted cosine similarity. The 10 most similar movies were used to calculate a predicted score for the film based on the weighted sum method. Our testing showed that this number was the most appropriate for our dataset. After repeating this method for multiple films across a range of users the RMSE and MAE were used to gauge the accuracy of the predictions. This method formed the foundation on which we implemented a content-based approach.

3.2 Content-Based Enhancement

Our approach to using content to enhance the quality of predictions underwent two stages. The first stage involved investigating how content meta-data about films could be used to enhance the selection of films in the purely collaborative approach. The second stage involved using the lessons this taught us to tune our content-based approach which significantly increased the accuracy of our predictions.

Our content-based system was based on a vector space approach. The similarity between two films was determined by mapping the features of these films to vectors. Each position in the vector represented a characteristic of the film, with a binary value. Once represented in this format, the similarity of these two vectors could be determined using cosine similarity (1). From this set of similarities, the best matches were used in combination with weighted sum (4) to calculate a prediction. A system overview is provided in figure 5.
The critical section of the collaborative approach which was improved upon is the choice of films to analyze. In the purely collaborative approach, the films were sorted based on the results of the adjusted cosine similarity results, and the closest matches were used in the weighted sum calculation. For our content enhanced implementation this was taken a step further by first sorting the films based on the results of their content similarity, and then sending the closest matches to be compared using the collaborative approach.

![Figure 5: The architecture of the content enhanced prediction algorithm, that takes the set of movies a user has rated and the movie to predict and produces a predicted rating between 0.5 to 5.0](image)

The first step in the content enhanced algorithm was to choose a movie to predict and a user whom the rating will be predicted for. The film was compared to all the films the user has rated based on our content analysis method. This involved using either the directors, actors, genres, or a combination of all of these for the films. The 20 most content-similar films were sent to the collaborative
algorithm which calculated the collaborative similarity between these films using adjusted cosine similarity \(^2\). Weighted sum \(^4\) was then used to calculate a prediction based on the 10 most similar films from this step.

The meta-data used for the content-based approach came from the Open Movie Database (OMDb). The parameters in this dataset that were investigated in this report were the actors, directors and genres. The effects were of these parameters were first examined individually, then combined as a group. We used RMSE and MAE as an objective measure of the accuracy of our predictions.

The effects of the content-based parameters’ influence on the accuracy of the prediction were identified by comparing the results from these approaches over a range of predictions for films. The results clearly highlighted the strengths and weaknesses of using content in this manner to generate predictions. Due to the limited resources and time available, experiments were not conducted on all aspects of the dataset. This would have required a significantly optimized implementation, and as was stated in the introduction the aim was to assess the accuracy of predictions and not to optimize the code. The tests were therefore limited to a subset of the films and users in our dataset. This subset was kept consistent across all tests to ensure the validity of any comparisons.

### 3.3 Improvements to the Content Enhanced Algorithm

During the construction of the prediction algorithm it was discovered that in some cases the content enhanced approach produced substantially worse predictions than the purely collaborative approach. This was caused by a lack of data in the generated set of films, which had a negative impact on the overall RMSE. Although the films were similar on a content level, there were not enough users who had rated both films. These findings were used to revise the content-based algorithm. In the improved algorithm, films chosen based on content which had an adjusted cosine similarity below 0 were replaced by the films from the purely collaborative-based approach with the highest similarity scores. By doing this, the algorithm was able to gracefully handle these extreme results. The effect of this improved algorithm on the RMSE is discussed further in the result and discussion chapters of this report.
Figure 6: To ensure the prediction is more accurate the films used in the prediction were replaced if not enough data was available from the collaborative analysis

In the improved version of the algorithm both the content-based similarity and the collaborative similarity were calculated. Just like in the other version the top 20 most similar films from the content-based similarity were initially selected. The collaborative similarities of these 20 films were then analysed
and any films with a collaborative similarity less than 0 were replaced. The replacements came from the most similar films from the collaborative approach. This set is then used for prediction computation.

### 3.4 Dataset

The dataset used for the ratings in this study was the most recently updated (at the time of writing) version of the MovieLens small dataset, consisting of 100,000 ratings on 10,000 movies by 700 users, last updated 1/2016. One problem with the dataset was that the scale for the ratings system was changed. The newer system allows for half-steps, i.e. ratings of 0.5, 1.0, 1.5, 2.0 and so on up to 5.0, as opposed to the old 1, 2, 3, 4, 5 system. This change may have affected the perceived accuracy of predictions made.

### 3.5 Content Fetching System

The OMDb api was used to collect metadata about the film. The IMDb ID provided in the MovieLens dataset [10] was used, appended with the characters “tt” since OMDb required that format. The response was in JSON format (JavaScript Object Notation). The JSON objects were parsed and stored in DSV format (Delimiter Separated Values, with “ˆ” as delimiter) to conform with the data from the MovieLens dataset. The reason a “ˆ” was used symbol instead of the standard “,” as a delimiter was that fields in the response contained commas, but the caret symbol was never used. This provided a range of meta-data parameters to work with, but the scope of this investigation is limited to the genre, director and actor fields.
4 Results

4.1 Purely Collaborative

The purely collaborative algorithm yielded an RMSE of 0.7864 and a MAE of 0.5423 across an arbitrary selection of 200 films across 200 users in the dataset. Figure 7 shows a subset of these results. The results of the predictions are marked with squares and bold lines, whereas the actual ratings are marked with circles and dashes.

Figure 7: Collaborative algorithm results compared to the actual ratings

This initial test shows the accuracy of the purely collaborative approach and provides a benchmark for the content-enhanced approach. The proximity of the two lines is noteworthy. This reflects the accuracy of each individual rating.

4.2 Preliminary Content Enhanced Approach

The genre-based content enhanced approach yielded an RMSE of 1.0515 and an MAE of 0.73895 across the same dataset. Similarly the director-based approach
yielded an RMSE and MAE of 1.27116 and 0.89199 respectively, the actor-based approach scoring 1.19948 and 0.84984 respectively, and finally a mix of all three parameters resulting in 1.04003 and 0.73739. All of the content enhanced algorithms yielded worse predictions than the purely collaborative approach, with the best of the four showing a 32% decrease in accuracy. The following graph shows a subset of the results from the director-based approach.

Figure 8: Director filtered Content-based filtering combined with Collaborative filtering

4.3 Content Enhanced Improvements

Figure 8 highlights a clear disadvantage of the content enhanced approach. Two of the points on the graph show that the algorithm predicted a rating of 0.5 for two films for which actual ratings given were far higher. Interestingly, the purely collaborative approach proved far superior in these cases. A closer analysis of cases like these showed that the problem with the prediction was due to a lack of co-ratings for these films, i.e. there were not enough users who had rated both films for the adjusted-cosine algorithm to compute a similarity for the two. As
was discussed in the method chapter, this led to a revision the content enhanced algorithm.

In our improved version of the algorithm, the genre-based approach had an RMSE of 0.73920 and MAE of 0.52251, the director-based approach had 0.70017 and 0.50558, the actor-based had 0.69546 and 0.50210, and the combination of all had 0.74057 and 0.52318. These results show a significant improvement over both the initial content enhanced approach and even the purely collaborative approach. The following graph shows the effect that this updated version of the algorithm had on the director-based approach. Lay note to the improvement in the accuracy of the predictions as compared to figure 8 (especially the previously noted outliers).

Figure 9: Improved director-based results using content-based fallback if no matches are found

The results from our improved algorithm show that the actors in a film were the best content-based indicator, with the directors in a close second. Both of these showed an improvement of over 10% in RMSE compared to the purely collaborative approach, and over 40% compared to the initial content-based
algorithm. Figure 10 shows a subset of the error comparison of the actor based approach and the purely collaborative approach. The actor-based approach is shown by the bold line and squares, and the purely collaborative in by the circles and dashed line (note that lower error ratings are better).

![Actor to Collaborative Error Comparison](image)

Figure 10: Comparison between the improved actor-based algorithm and the purely collaborative algorithm

Figure 10 shows that the actor-enhanced approach’s prediction accuracy tends to be equivalent or very close to the purely collaborative approach in most cases. This is to be expected, as our preliminary testing showed that data tended to be missing in the average case, which would cause the algorithm to fall back to the purely collaborative results. The actor-enhanced approach also shows a clear advantage in many edge cases, and rarely performs worse than the purely collaborative approach. The same holds true for the genre and director-based approaches, albeit with slightly worse RMSEs than the actor approach.
4.4 Summary of Results

Figure 11: An overview of the results, showing the RMSE before the improvement in striped diagonal pattern and after the improvement in dotted pattern compared to the pure collaborative algorithm on the far right.

Figure 11 shows the RMSE of the algorithms before and after the improvements were implemented along with the purely collaborative algorithm for comparison. This highlights the benefits of the improved algorithm over the initial content-based approach and also shows the overall improvement as compared to the purely collaborative approach.
5 Discussion

5.1 Initial Content-Based Approach

The goal of the initial content-based approach was to determine how various content parameters could influence the selection of films in a purely collaborative approach. The expectation was that the content enhanced approach would outperform the collaborative approach in a few cases, especially in cases where there was enough data to go by. Similarly, the results were expected to turn out worse in the cases where data was sparse. The results very clearly mirrored this expectation. In most cases the content enhanced approach performed as well as (and sometimes even much better than) the purely collaborative approach, but the RMSE was generally greatly increased as a result of a few very poor predictions.

The selection of the actual parameters used proved insightful. The genre of a film generally tended to be a more reliable source of similar movies than actors or directors. This is not surprising as the number of films in a certain genre will naturally greatly exceed the number of films created by any specific director or acted in by a specific actor.

5.2 Improvements the Content-Based Approach

Although the initial content-based results were, in many cases, far worse than the purely collaborative approach, they did highlight some clear benefits. Analyzing the data showed that the extremely poor prediction cases could generally be attributed to a lack of collaborative similarity between films that were otherwise similar in terms of content, i.e. there simply were not enough ratings to go by in these cases. This problem was addressed by allowing the algorithm to fall back on results from the purely collaborative approach when necessary.

The results of this amendment to the collaborative approach suggest that there is an advantage to mixing the approaches in this way. All of the content enhanced approaches outperformed the purely collaborative method to varying degrees.

5.3 Effects of Individual Content Parameters

The scope of this investigation did not take all possible parameters into account, let alone various possible combinations of such parameters. However, the results show a clear advantage to taking the parameters used into account. All of the parameters investigated resulted in more accurate predictions than the purely collaborative approach when tested individually and when combined. Although actors yielded the best results, it would be difficult to draw the conclusion that this indicator is somehow objectively better than the directors, since the difference between them was small. It is noteworthy that the genre-approach performed considerably worse than both of these indicators, whereas the opposite was true in the first content-based approach. Even when mixing the three indicators, the results were closer to the genre-approach. This would suggest that the genre is a comparatively poor indicator of general user preferences for movies.
5.4 Strengths of Content-Enhanced Filtering

The first content-enhanced algorithm showed that using content can yield significantly better results in many edge cases, but that the overall RMSE suffered due to a lack of data. The improved version of this algorithm showed that the lack of data in this general case could be corrected to greatly improve the accuracy of predictions. This suggests that users often exhibit a bias towards certain characteristics of films, especially with regards to directors and actors. It is rather uncommon for two movies to share the same actor, causing the similarity computation in the content analysis to result in movies being completely different a majority of the time. However the actor does affect the experience of the movie, so when there actually are movies that share actors, the bias introduced by the content enhancement improve the prediction. An approach purely based on ratings will be blind to such biases, which is why this is the main strength of content-enhanced filtering.

5.5 Limitations of Content-Based Filtering

The content-enhanced approach in this report relies on the existence of a user bias for certain characteristics of films. Or, in a more abstract sense, a bias for certain characteristics of items in a set. The results suggest that this bias not only exists but is also exploitable. However, the dataset is too small to draw general conclusions about the validity of such an approach for film predictions as a whole. Perhaps a collaborative approach will be better in real world applications, since in addition to generating accurate predictions it also generates diverse predictions. One goal of recommender systems is user-retention, and a system which consistently recommends movies of similar genres or with similar actors may lead to users losing interest in using the system due to the lack of variety.

Another clear disadvantage is the lack of data. This was shown by the extremely poor predictions generated by the first content-enhanced algorithm. Fixing this by allowing the collaborative results to have more of an impact did fix the general cases, but also somewhat defeats the purpose of using content. If the system is only able to take advantage of biases in edge cases, it might not be worth implementing such a system at the risk of removing an element of variety in recommendations.

5.6 Evaluation of This Study

A key takeaway point from this study is that the focus was placed on improving the accuracy of predictions, not the speed at which predictions can be generated. In a real world scenario the efficiency of the algorithm is key, and the algorithm implemented in this report does not come close to being performant enough to be useable at scale. This placed a restriction on the number of films we were able to generate predictions for, which may compromise the reliability of the results.

However, there are many ways in which the speed of the algorithm could be improved drastically. For example, the majority of the time taken by the algorithm was used calculating the adjusted cosine similarity between films. This could quite simply be fixed by memoizing the results. Furthermore, the vast
majority of the calculations can be parallelized. Finally, the implementation lan-
guage chosen was R, which is too high-level to achieve the kind of performance
which a language such as C++ could attain. Even with such improvements it is
difficult to say whether this algorithm could be run at an industrial scale with
millions of users and hundreds of thousands of films.

It is also important to note that this study investigated the effect of three
content parameters on the accuracy of predictions: genres, actors and directors.
There are, however, far more potential parameters to investigate. For example
there are the runtime, year of release, budget, language, rating, recording tech-
niques, and user-generated tags just to name a few. Different users will have
different preferences and biases, and an in-depth analysis of a larger range of
parameters may yield far more accurate predictions and, perhaps more impor-
tantly, more meaningful and personal film recommendations.

Finally, it is important to consider that this algorithm does not solve some
of the problems inherent to collaborative recommender systems. For instance,
because of the fact that the algorithm will always use explicit user ratings in
in some way, it cannot not address the Cold Start problem discussed in the
introduction.

5.7 Conclusion

The goal of this investigation was to investigate how content parameters can
be used to enhance the accuracy of predictions made by a collaborative rec-
ommender system. Specifically with regards to the selection of films to be
used in the predictions calculation. The initial results proved unpromising.
They yielded significantly poorer RMSE scores than the purely collaborative
approach. However, they showed that the amount of data available needs to be
taken into account when analyzing content. After this was corrected for, the
content-enhanced approach showed dramatic improvements, even compared to
the purely collaborative approach. Despite the limitations brought up in the
Evaluation section (section 5.6), the results strongly suggest that the accuracy
of a collaborative recommender system can be improved by adjusting for content
biases.

It appears that the actors were the best indicator (out of the parameters
investigated) to consider when creating a content-enhanced algorithm for films.
However, the comparably identical efficacy of the director-based approach cer-
tainly does not eliminate it as a candidate. Out of the three, the genres approach
was to be the only one with a clear disadvantage. The improvements of the im-
proved approach ranged from a 5% improvement when using a combination of
all parameters to an over 11% improvement when looking at just the actors.

5.8 Future Works

Investigating the effects of more types of content parameters would perhaps be
the most interesting extension to the work done in this report. Film meta-data
comes in a seemingly limitless number of forms, many of which may prove far
more effective for gauging predictions. The effects of different combinations of
these parameters would also be an interesting area to explore.

It would also be critical to test the effects of these parameters on a much
larger dataset to achieve more conclusive results. Doing this would require
optimizing the algorithm significantly (discussed at length in section 5.6).
References


[7] The R Foundation. The R programming language, version 3.2.3. [https://cran.r-project.org/bin/windows/base/old/3.2.3/](https://cran.r-project.org/bin/windows/base/old/3.2.3/) Released 2015-12-10.

[8] Mauricio Zambrano-Bigiarini. hydroGOF: Goodness-of-fit functions for comparison of simulated and observed hydrological time series. [https://cran.r-project.org/web/packages/hydroGOF/index.html](https://cran.r-project.org/web/packages/hydroGOF/index.html), Published 2014-02-04, version 0.3-8.

