12-2019

Improving Video Game Recommendations Using a Hybrid, Neural Network and Keyword Ranking Approach

Nicholas Crawford
Sheridan College

IMPROVING VIDEO GAME RECOMMENDATIONS USING A HYBRID, NEURAL NETWORK AND KEYWORD RANKING APPROACH

A Thesis

Presented to

The Faculty of Applied Science and Technology, School of Applied Computing

of

Sheridan College, Institute of Technology and Advanced Learning

by

CRAWFORD, NICK

In partial fulfillment of requirements

for the degree of

Bachelor of Computer Science (Mobile Computing)

December, 2019

ABSTRACT

IMPROVING VIDEO GAME RECOMMENDATIONS USING A HYBRID, NEURAL NETWORK AND KEYWORD RANKING APPROACH

Nick Crawford

Sheridan College, 2019

Advisors:

Professor Khaled Mahmud,

Professor Richard Pyne

Recommendations systems are software solutions for finding high-quality and relevant content for a given user type ranging from online shoppers, to music listeners, to video game players. Traditional recommendation systems use user review data to make recommendations, but we still want recommendations to perform well for new users with no review data. Currently, one of the problems that exists in recommendations is poor recommendation accuracy when only a small amount of data exists for a user, called the cold start problem. In this research we investigate solutions for the cold start problem in video game recommendations and we propose a solution that uses a hybrid neural network and keyword ranking approach. We evaluate this system with precision and recall metrics and compare the results to a traditional recommendation system. We present that this hybrid system offers performance gains when recommending to users who have low-medium previous reviews.

Keywords: Recommendation system, cold start problem, collaborative filtering, content-based filtering, embeddings, nearest neighbor.

ACKNOWLEDGMENTS

I give my deepest appreciation to my family and friends for supporting me on my academic journey. The author gratefully acknowledges Dr. Khaled Mahmud and Dr. Richard Pyne for their time, expertise and guidance on this journey.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Video game consumers are increasingly shopping online for new games on sites like Amazon.com. Recommendation systems on Amazon learn about user’s interests and they recommend similar video games that they may enjoy. Unfortunately, these systems have problems when trying to recommend to users who have few or no previous video game reviews, this is the cold start problem.

We investigate recommendation systems and methods to improve them, specifically our research will focus on hybrid recommendation systems. Our research area will be about recommending video games. The main goal of our research is to recommend video games to users in a way that aims to solve the cold start problem. Our research will be a significant contribution to the field of recommendations systems both because of the unique algorithm being used and also because the field of recommendations in video games is one that is not often explored.

Our research is a valuable contribution to the field of recommendations systems because we are contributing to the discussion of how we can solve the cold start problem in recommendations and also proposing an algorithm to do so.

1.1 Terms and Definitions

Recommendation System (RS)	An information filtering system that seeks to predict the rating that a user would give to an item. In order to suggest content to the user that is relevant to them.

Content-based filtering	A recommendation system algorithm where predictions are based on a description of the item and a profile of the user's preferences.
Collaborative based filtering	A recommendation system algorithm where the data of previous users is tracked and used to create predictions for the behavior of future users.
Cold start problem	When a software system cannot draw any inferences for users or items about which it has not yet gathered sufficient information.
Hybrid Recommendation Algorithms	A recommendation algorithm that combines collaborative based filtering and content-based filtering.
Embeddings	A categorical feature represented as a continuous-valued feature. An example is representing a video game with features as an n-dimensional vector.
Nearest Neighbors	A method for finding items that are closest to a given item. Calculated using cosine similarity of vectors.

t-Distributed Stochastic Neighbor Embedding (TSNE)	A dimensionality reduction machine learning algorithm used to visualize high dimensional data in a low dimensional space like 2 dimensions.
Latent	Latent means hidden. Eg. Latent features are features that are immediately visible, they must be learned.

1.2 Problem Statement

Traditional recommendation systems use the behavior data of previous users of the system to predict content for future users. Unfortunately, when new users try to use the website, there is no user review data to make recommendations with. Similarly, when new items are added to a website, no user has initially interacted with those items, so there is no user behavior data available to make recommendations for those items. This is called the cold start problem [7].

Hybrid recommendation systems are a potential solution to the cold start problem that work by only requiring small amounts of user data. Hybrid recommendation systems use a combination of collaborative filtering and content-based filtering algorithms. Content-based filtering uses the properties of the digital content itself and compares the properties with the properties of other content in order to find recommendations. Our research is to build and evaluate a hybrid recommender that will help to overcome the cold start problem.

1.3 Purpose of the Study

The purpose of the study is to progress the quality of recommendation systems. The goal of the research is come up with a more effective strategy for recommending relevant content than what is currently being used and ultimately to improve the content discovery experience for users.

1.4 Motivation

The motivation for this research was to try and improve the quality of recommendation systems. Recommendation systems are popular with different online web applications and e-commerce platforms but there are improvements that could be made to these systems, notably in the area of the cold start problem. We will offer those improvements with this research.

1.5 Proposed Work

Our research consists of the following steps.

- Download, clean and sort Amazon video game review data
- Train a collaborative filtering model to predict the rating a user will give a particular movie. This model subsequently generates a weight matrix of the latent features of each video, that we can represent as a vector.
- Create user vectors by averaging the video game vectors in their review history.
- Use a nearest neighbor search algorithm to find video games vectors nearby our user vectors. These neighbors are our recommendations.
- Implement a content-based filtering method into our nearest neighbor algorithm that ranks video games based on their category keywords.

- Recommend video games based on this new hybrid method.
- Benchmark our recommendations using average precision and recall.

1.6 Thesis Statement

Recommendation systems can be used to recommend video games to users based on their unique tastes and preferences. User review data can be used to train a collaborative filtering model and video game categories can be used build a category-similarity model. This research proposes a hybrid algorithm using both a collaborative and a category-similarity model and evaluates the performance using precision and recall metrics. This research aims to offer a solution to the cold start problem.

1.7 Significance of the Study

In the past, content-based recommendation algorithms have been shown to be less powerful than the popular collaborative filtering based algorithms. An early study of content-based filtering measured a 60% prediction accuracy, this paled in comparison to the 70% accuracy for collaboration filtering based filtering recommendation systems. [2] More recently, content-based filtering research is being done at YouTube. In a study by Google, the audio and video content of a video was processed with a content-based recommendation algorithm. Specifically, machine learning was used to classify the content and similar videos could be recommended from that information [31]. This method worked really well for recommending related videos and dealing with the cold start problem; though the researchers do admit there are weaknesses to their solution. What we can take away from this is that content-based filtering is still an algorithm under active development and further contributions are needed to improve the technique.

1.8 Organization of the Thesis

Our research begins with an overview of the current literature surrounding recommendation systems and its popular implementation methods. It discusses machine learning based recommendation methods and the details of the cold start problem. We then discuss our research methodology and specific algorithms and metrics that were used. This includes discussing selected training data, the software being used, illustrating our hybrid algorithm design, and how we calculate our metrics. Finally, we offer a conclusion to our research and suggestions for future exploration.

1.9 Conclusion

In this chapter we discussed the value of recommendation systems and the value that our research can bring to the field. Recommendation systems are software systems that are actively being used today in companies like Amazon and Google. They offer a lot of value but they still have room for improvement. These systems are under active research in the computer science community and our research will be a significant contribution to the library of research in this area. We hope that our contributions will bring a significant inquiry into the field.

CHAPTER 2

LITERATURE REVIEW

In this literature review we outline the current state of research in the field of recommendation systems (or RSs). This review covers different algorithms being used to optimize and build recommendation systems, different types of recommendations systems and different applications for recommendation systems. The purpose of this literature review is to develop an understanding of the current stage of research in recommendation systems. The strategy used to develop this literature review was a lot of investigation and discovery of relevant papers; especially highly acclaimed relevant papers. Resources such as Google Scholar and Google Research were used to discover relevant and industry leading papers.

In summary, the topics discussed in this literature review are: the differences between collaborative filtering and content-based filtering recommendation systems, implementation methods for RSs, applications of RSs and the cold-start problem of RSs.

2.1 Recommendation Systems

Recommendation systems provide value to websites of all kinds; they are currently used in e-commerce websites, online library databases, video streaming services and music streaming services. In this review we will discuss different RS implementation strategies in order to guide us to a new and valuable area of research. The first thing that will be discussed is the two major categories of recommendation systems, collaborative filtering RSs and content-based filtering RSs.

2.2 Collaborative Filtering

Recommendation systems are used to recommend new and relevant content to users, but in order to recommend relevant content the algorithm needs to learn which content to recommend. These two algorithms use two different types of data in order to make recommendations.

Traditional recommendation systems use collaborative filtering approaches [1] to suggest related content to a user, it works by using aggregate user data to make recommendations. For example, if the content being recommended is YouTube videos, the system will keep track of how many users have watched a particular candidate video right after watching the seed video and recommend these related videos to users. This approach works very well when there are plenty of users and the videos are relatively popular, but what about when there are no users? This is called the “cold-start problem” [21]; when there are not enough users to generate data to build quality predictions.

2.3 Matrix Factorization

A collaborative filtering model is a model that predicts the recommendations for future users based on the data of previous users.

Matrix factorization is a popular collaborative filtering algorithm that also uses user data to make predictions for future users. Matric factorization uses machine learning to learn to represent some content as a collection of features and how strongly each item contains that feature. For example, if it is recommending video games, a feature could be “an action game” or a “sports game”. The model would learn which features some particular game using training data, and the game is assigned a number between -1 and 1 for how strongly that game represents that feature.

After each video game learns their features we use that numeric information and represent each video game as an n-dimensional vector and compare the cosine similarity between vectors in order to find video games that are similar to each other. This is how music recommendations are done at Spotify [35] and this technique is a big inspiration for the collaborative filtering aspect of our model. A visual representation can be seen in Fig. 2-1.

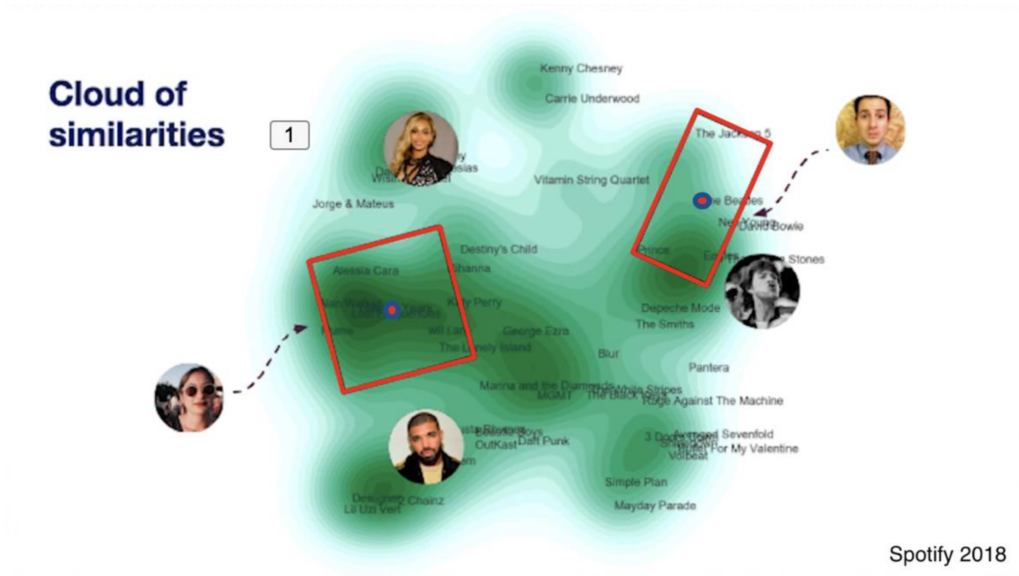

Figure 2-1: Music Recommendations Via Vectors at Spotify

Another term for these vector feature representations are “Embeddings”. Because these vectors represent hidden features, naturally video games that are similar or related will be close together in the vector space.

2.4 Content-based Filtering

A solution to the cold-start problem is content-based filtering RSs [22]. These RSs are used to analyze the content itself instead of analyzing the user behaviour around this content [19][20]. When the system analyzes only the data file itself, it does not need any other information about user behaviour and is therefore not affected by the cold-start problem. Joonseok Lee is a researcher at Google who has recently published two studies about using Content-based recommendations for YouTube videos using deep learning. His research revolves around analyzing thousands of video frames in order to make an intelligent recommendation network. Initially he started doing only video analysis [1] and then combined both audio and video analysis [2]. The content-based filtering system uses convolutional neural networks (CNN) to allow video understanding. Joonseok concluded in the study that this is a useful tool to work alongside traditional collaborative filtering approaches.

2.5 Cold-Start Problem

Several researchers have developed strategies to solve the cold-start program. One solution to the cold-start problem was suggested by Zi-Ke Zhang et al. [3]. The team suggested the use of "social tags" [24] which are explicitly defined keyword tags by users which classify the content into categories. As soon as two objects share the same tag, then they can be considered related and eligible to be recommended to users.

A study by Manasi Vartak et al. [4] uses neural networks in order to develop a "meta-learning strategy". Matrix factorization (MF) is one of the most popular techniques for product recommendation, but is known to suffer from serious cold-start problems. In Vartak's research, his team developed an algorithm for new items that arrive continuously, in this example tweets. The research uses two deep neural network architectures that implement a meta-learning strategy. The first architecture learns a linear classifier whose weights are determined by the item history while the second architecture learns a neural network whose biases are instead adjusted.

Another technique for dealing with the cold start problem is using Latent Factor Models. These models map users and items into a dense and reduced latent space that captures their most salient features [17][18]. These models provide recommendations results which are better than traditional neighborhood methods.

2.6 Hybrid Systems

We have seen good performance from both content-based filtering and collaborative filtering recommendation system algorithms. In industry, instead of using one algorithm or the other, it is often the case that a hybrid of both algorithms is used [12]. A system that uses both algorithms is called a hybrid recommendation system and they are useful because the combination of both algorithms helps to make up for the limitations that each other algorithms have [16].

Content-based filtering algorithms are useful when building a user profile. We can look at the past purchases/transactions that a user has had on their account and we can use content-based filtering in order to infer the general categories of content that the user is interested in. [11] However content-based filtering has a limitation that the algorithm cannot differentiate between good and bad content, it simply finds a similarity between content. [15]

Collaborative filtering systems are powerful because they are better at discovering high popular content instead of just content that is similar. Furthermore, collaborative filtering algorithms can deal with any kind of content. Where these systems are limited is in regard to the cold-start problem that we mentioned earlier. Collaborative filtering algorithm performance also degrades as the number of users increases [13].

The combination of both of these algorithm makes for a much more powerful and rounded out algorithm [14].

2.7 Implementation Methods

Looking beyond the cold-start problem we will describe popular and significant algorithms used to implement RSs of different kinds.

Amazon was an early pioneer of recommendation systems, they introduced RSs to their site to personalize their online store for each customer [5]. Most recommendation algorithms start by finding a set of customers whose purchased and rated items overlap the user's purchased and rated items (ex. collaborative filtering and cluster models). Instead of using those systems, Amazon developed an item-to-item collaborative filtering algorithm — that focuses on finding similar items, not similar customers. Unlike traditional collaborative filtering which uses a very computationally heavy algorithm, the Amazon item-to-item algorithm uses online computation that scales independently of the number of customers and number of items in the product catalog.

Amazon's algorithm set the stage in 2003 with their algorithm which made a big impact in the technological community. Since then there have been many new innovations with RSs, in particular with the integration of machine learning technology. Another recommendation system that was developed by Google for youtube.com recommends new videos to users [6] using 2 neural networks. The neural networks use user-to-user collaborative filtering (unlike amazon's item-to-item filtering algorithm). For the ranking neural network, they use an extreme multi-class classification algorithm where the prediction problem is accurately classifying a specific video that is interesting to a given user.

Lastly, in another approach Andrew Schein et al. [7] used both collaborative and content-based filtering to make predictions to users. Instead of using a neural network for classification, they used a naive Bayes classifier.

2.8 Applications of Recommendation Systems

We discussed popular areas where RSs are useful, such as YouTube videos, Amazon products and tweets, but RSs have many more applications, some of which we'll discuss here. In the process of investigation, we found different innovative application areas for RSs. One example is research at Google about personalized news recommendations based on click behaviour [8][23]. They use a Bayesian framework for predicting user's current news interests, which considers both the activities of that particular user and the news trends demonstrated in activities of a group of users.

MovieLens is an item-to-item collaborative filtering RS [9] inspired by Amazon's algorithm, that recommends relevant movies to users based on user behaviour data. Netflix similarly invests a lot of resources into collaborative filtering algorithm [10]. in 2006 Netflix ran a worldwide competition asking researchers to develop a better collaborative filtering recommendation system algorithm for their platform. The winning team was able to produce an algorithm that improved performance by 10%.

There are many more applications of recommendations systems for all types of different industries. There is research about developing a RS for RSS feeds and Blogs [25], research to replace the need for word of mouth recommendations [26], and an outdoor recommendation system based on user location history [27]. Furthermore, there is research about RSs for locations for location-based social networks [28], a graph-based friend recommendation system using genetic algorithms [29] and a context aware personalized travel RS that recommends activities while traveling [30]. We can see that the application areas for RSs are limitless.

2.9 Conclusion

Throughout this research we have outlined modern innovations in the field of recommendations systems. We discussed machine learning based and non-machine learning based recommendation algorithms, contrasted the two popular RS types and illustrated some popular applications for recommendation systems.

The research done in this paper will be on hybrid recommendation algorithms, which is a combination of content-based filtering and collaborative filtering

For our research, the area of interest and opportunity for a unique contribution is in the area of hybrid recommendations algorithms, similar to [1] [2] is an area that can be easily researched given the resource and time limitations of our research. Additionally, using new machine learning techniques can provide new innovations to the field.

CHAPTER 3

METHODOLOGY

As discussed previously there are two commons methods for recommendations in literature, collaborative filtering and content-based filtering. This research integrates both of these and develops a unique hybrid recommendation system for video games that seeks to overcome the limitations of the cold start problem.

Our recommendation system will recommend video games to a specific user, trying to personalize the recommendations for the user. We will be focusing on solving the cold start problem, which in our case means when a new user to the system has a small number of previous item interactions. Such a problem is typically hard for recommendation systems to solve.

In this chapter we will discuss the details of the proposed model including how to build it, what metrics to use to benchmark it and the testing strategy. Fig. 3-1 gives a brief overview of the different steps of our research and also an outline of what will be discussed in this chapter.

Figure 3-1: An Overview of All Research Steps

3.1 Data Source

In order to do training and testing we will be using the 2018 Amazon product data set [33] which contains video game purchase ratings from May 1996 - July 2014.

The two files we are using are Video_Games.json.gz (review data) and meta_Video_Games.json.gz (product metadata). Fig. 3-2 shows the four columns we will be using, “overall” is a user rating out of 5.

[87]:

	overall	reviewerID	title	category
312259	4.0	A26BGHYLTCD5SM	Halo 4 - Xbox 360 (Standard Game)	['Video Games', 'Xbox 360', 'Games', '']
175973	1.0	A3EP0KWNRV2GI2	Time Splitters: Future Perfect - PlayStation 2	['Video Games', 'Retro Gaming & Microconsoles'...
151139	5.0	A3V6Z4RCDGRC44	Yu Yu Hakusho Dark Tournament - PlayStation 2	['Video Games', 'Retro Gaming & Microconsoles'...
5186	5.0	A3JUMMIGJ7B38X	Final Fantasy VII	['Video Games', 'Retro Gaming & Microconsoles'...
424330	5.0	A1C5RRPBGRDF2Z	Phantasy Star III - Sega Genesis	['Video Games', 'Retro Gaming & Microconsoles'...
...	...	...	...	...
318418	5.0	A1WLAQOT4R9RFK	Oblivion (Game of the Year Edition) -Xbox 360	['Video Games', 'Xbox 360', 'Games', '', 'Incl...
387097	5.0	A3PLN1BLX8U9CY	Farming Simulator 15 [Download]	['Video Games', 'PC', 'Games', '', 'New graphi...
218691	5.0	ANMS35SC8X6VX	Call of Duty 4: Modern Warfare - Xbox 360	['Video Games', 'Xbox 360', 'Games', '']
351605	5.0	AHMLUEQJI0EJG	Mario Kart 8 - Nintendo Wii U	['Video Games', 'Kids & Family', 'Wii U', 'Gam...
426471	5.0	A1TVD594BZ421P	Machinehead - Sega Saturn	['Video Games', 'Retro Gaming & Microconsoles'...

432782 rows × 4 columns

Figure 3-2: Sample of Filtered Video Game Review Data

3.2 Data Processing and Filtering

We have to do multiple operations to the original dataset in order to use it efficiently. Here is a list of those operations:

- Drop table rows that consist of irrelevant video game subcategories like: Accessories, Cases, Chargers, and Controllers.
- Drop video games with reviews from less than 5 people, so that video game data will be both in our training and validation set.
- Drop reviewers with less than 5 reviews, so that user review data will be both in our training and validation.

- Drop columns that we did not use. From Video_Games.json.gz the columns that we didn't use are: verified, reviewTime, asin, reviewerName, reviewText, summary, unixReviewTime, vote, style and image. From meta_Video_Games.json.gz the columns we did not use are: brand, rank, main_cat, description, also_buy, also_view, price, feature, date, tech1, details, similar_item and tech2.

After dropping all unneeded data, the data was reduced from 2,848,246 rows of video game reviews, to 432,782 rows.

3.3 Proposed Hybrid Model

The way our hybrid algorithm works is illustrated simply in Fig. 3-3. 1)Pick a user. 2)Calculate user vector from embedding matrix. 3)Extract top 5 most commons video game category keywords for user. 4)Find nearest neighbors of user. 5)Use category keywords to look for neighbours with similar keywords. 6)Choose n number of recommendations from these neighbours.

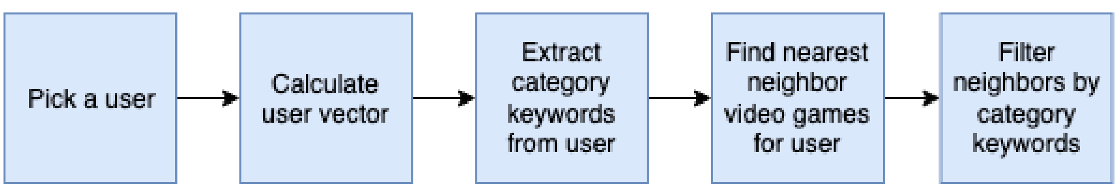

Figure 3-3: Steps of Hybrid Recommendation Algorithm

3.3.1 Collaborative Filtering Details

For our collaborative filtering model, we are using matrix factorization, a collaborative filtering algorithm that was described in the literature review. Most importantly we are using matrix factorization because of the video game embeddings that get trained from our model. The embeddings contain latent features of each video game and they give us numeric values that we use to represent each video game as a vector.

We train a neural network using the Amazon review data, to predict the rating out of 5 that some user would give for some video game. We split the review data into 90% for training and 10% for validation. After training our model, the neural network weight layer (also known as video game embedding layer) gives us the information we need to build our collaborative filtering model. Specifically, the video game embedding is an n x m matrix, where n is the number of video games in our data and m is the arbitrary embedding size, in our case we chose 40. Each row of the matrix represents a video game and the features of the game are represented using 40 different numbers. We use the 1 x m matrix given for each video game and we use it as a vector that we can compare with other video game vectors. The video game vectors are trained in our neural network which will be discussed in the following section.

After each video game is represented as a vector, then we can use the relationship between video game vectors to make recommendations. For example, if each number in our m dimensional video game vector represents some feature, then different video games that have similar feature values should be related in some way. We use this assumption to make recommendations.

Our collaborative filtering model is a nearest neighbor search of thousands of vectors. In order to generate recommendations, we first represent a user as a vector, then we use a nearest neighbor algorithm to find nearby vectors to that user which will find video games that should be relevant for that user and their vector.

Fig. 3-4 shows two examples of this nearest neighbor algorithm, it shows hundreds of blue dots which are vector representations of video games. The closer some dots are to each other, the more related they will be.

In Ex. 1 we can see a red arrow pointing to one location and the location is showing different games from "The Sims" video game series concentrated nearby each other. This is because these games are related. It is to because expected that different games from The Sims series have similar vector values, because if the games are from the series they should have very

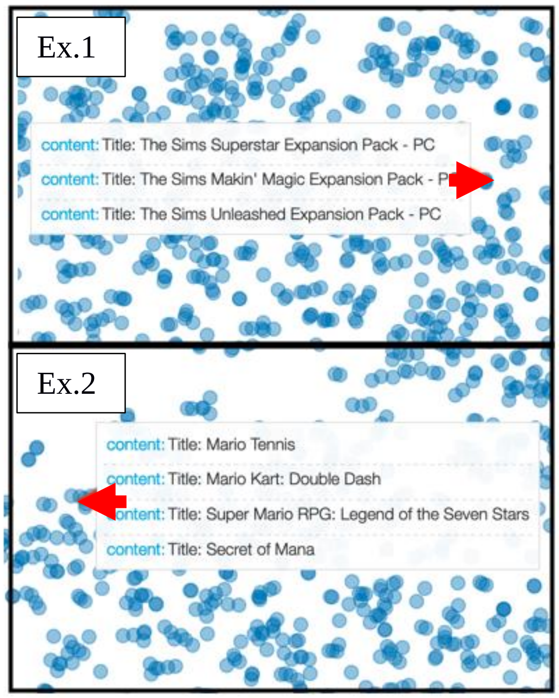

Figure 3-4: Our Vector Representation of Video Games

similar characteristics. In Ex. 2 we see again an arrow pointing to a region, and this time we see different video games from the "Mario" series located nearby each other.

This is the first element of our recommendation system. By combining this commonly practiced technique with a keyword-based content filtering system, the result is a unique algorithm that generates recommendations that work effectively even with new users that have very few reviews (the cold start problem).

3.3.2 Collaborative Filtering Architecture and Training

The collaborative filtering uses a neural network that is trained with the PyTorch Python library and a helper library called fast.ai. The feedforward structure of the network is as follows. There are two embedding layers for the users and the items, these are the weights of the neural

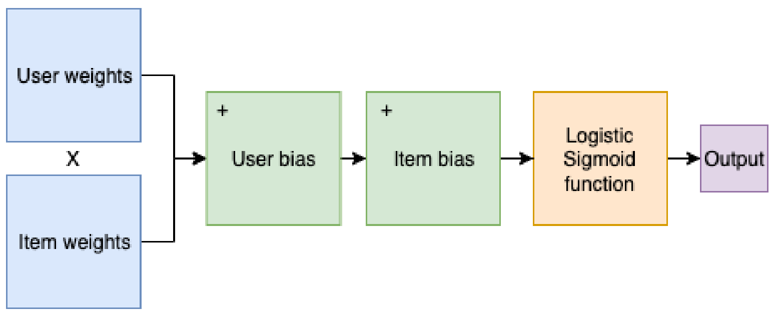

Figure 3-5: Neural Network Architecture

network. Each row of the user embedding matrix is multiplied (dot product) with each column of the item embedding matrix. Afterwards a user bias value and an item bias value is added to the resulting value. Lastly the value is put through a sigmoid layer to force the output between values 0 and 1, this purely makes the calculation more simplified for the hardware and therefore faster. We can see the whole network architecture in Fig. 3-5.

PyTorch looks at the output value and compares it with the ground truth value of our training data and computes the error. The error is back-propagated in our network and our weights and bias values are changed along the direction of the gradient. We train for the number of epochs in order to most minimize our loss function. The optimization function we use is AdamW.

In Fig. 3-5 we see the “User Weights” layer, this is the embedding matrix layer that was discussed in section 3.3.1. This is the main output that we care about from the neural network model. This weight/embedding matrix is what we use to represent each video game as a vector.

3.3.3 Content-based Filtering

For the content-based filtering aspect of our algorithm, we will be using category data for each Amazon video game. For example, in the Amazon dataset, the video game “Grand Theft Auto 3” has the following category keywords “Video Games”, “Retro Gaming & Microconsoles", “PlayStation 2”, and “Games”.

As discussed earlier, a collaborative filtering algorithm uses only past user behaviour to make recommendations. Whereas content-based filtering uses only item qualities such as video game title, video game publisher, and the year it was published in order to recommend video games with similar qualities.

By combining both of these recommendation algorithms we can build a powerful recommender that is still effective even for users with few item interactions.

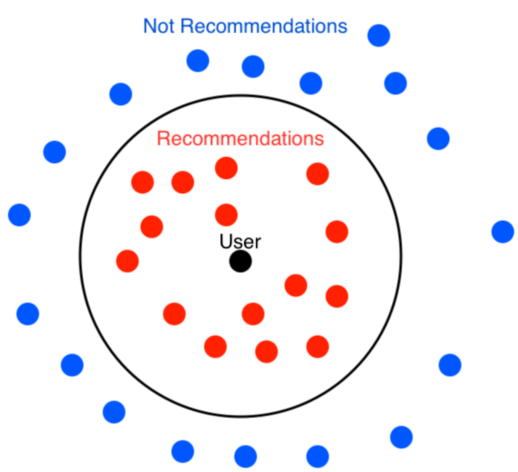

Figure 3-6: Collaborative Filtering Only

In order to illustrate how this hybrid system will work, we will present here two diagrams. Fig. 3-6 is a pure collaborative filtering algorithm. The diagram illustrates a 2D vector space containing video games and a user. Given a user vector, the algorithm uses a nearest neighbours search to recommend nearby video game vectors.

We can see the recommended items in red. This algorithm performs well but it suffers when the given user has a small number of item interactions, then the algorithm does not have enough information to offer accurate recommendations.

In Fig. 3-7, we can see a given user is given nearest neighbor recommendations just as before, but this time the user is also given recommendations outside the nearest neighbor boundary. These new recommendations are discovered using our content filtering keyword search algorithm.

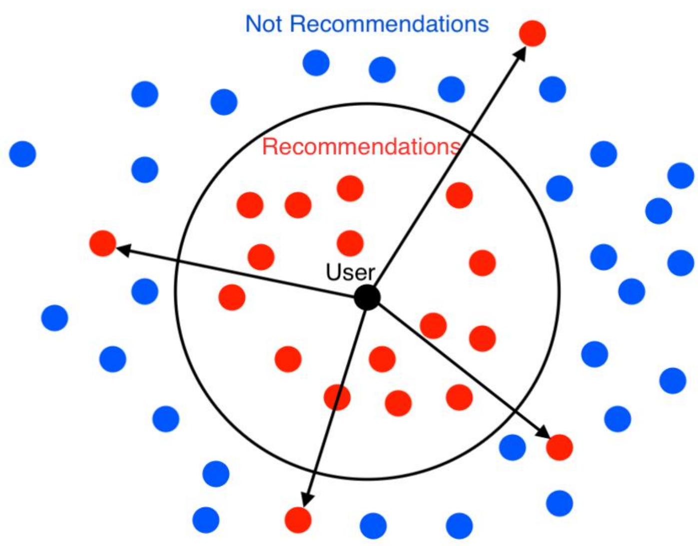

Figure 3-7: Hybrid Model

3.3.4 Hybrid Collaborative Filtering and Keyword Search Algorithm

Our hybrid algorithm combines the matrix factorization approach for collaborative filtering and a category keyword content-based filtering approach. The step by step process of recommending video games to users with our hybrid recommendation system is described below.

Algorithm 1.

1. Choose a user that has 1 or more video game reviews. We will recommend video games for this user.
2. For each video game in user's review history, look at our video game embedding matrix and calculate the average vector for all of the video games. This average vector will represent the average video game types this user is interested in.
3. Extract the top 5 most common video game category keywords from all video games in the user's review history.
4. If we want to find k recommendations for a user, we will use n candidate neighbors, where n = k*2. We use "Annoy" nearest neighbor library to find some n candidate nearest neighbour video games for this user vector, using cosine similarity vector comparison.
5. Using our top 5 most common category keywords, search within the nearest neighbours for video games that share similar category keywords.
6. Rank each video game in our n candidate nearest neighbors by using a combination of video game vector cosine similarity and keyword set similarity using the user's top 5 most common category keywords. The top k scoring rankings will be k recommendations for a user.

The specific function for ranking a video game, as mentioned in step 6 is as follows. Where R is a ranking value between 0 and 1, 1 the most relevant recommendation for a user. Where d is a measure of cosine similarity between 2 vectors that returns between 0 and 1. Where k is a keyword set similarity value between 0 and 1, that compares the similarity of keywords between a user keywords and each video game set of category keywords. And where w1 is a constant = 2 and w2 is a constant = 1.

$$R = d * w_1 + k * w_2$$

We believe that a hybrid model such as this will provide the benefits of collaborative filtering along with the benefits of content-based filtering and most notably improving cold start situations when new users have a small amount of reviews.

3.4 Metrics

To prove the effectiveness of our hybrid recommender model we will compare performance between a pure collaborative filtering model versus our hybrid (collaborative and content-based recommender model).

The two main metrics we will be using are Precision and Recall, these are common metrics used for recommendation system performance measurement. They both give values between 0 and 1. These metrics can also be described as a function of true positive (TP), false positive (FP) and false negative (FN) values. A model that produces no false positives has a precision of 1.0, and a model that produces no false negatives has a recall of 1.0.

Precision and recall [32] can be defined as,

$$\boldsymbol{Precision} = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of items we recommended}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}}$$

$$\boldsymbol{Recall} = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of all possible relevant items}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}}$$

They give us a measure of how good our recommendations are.

3.5 User Group and Testing Strategy

In order to generate these precision and recall metrics we will use our user review data for testing. We will use a specific user group subset of our data for testing. The user group is users have more than 100 item reviews, we choose this group so that we have a lot of data for us to work with. From this user group, we will use a sample of 100 users for testing.

We want to recommend video games to users and we need a way to measure whether the recommendations we made were good or not. The way we do this is for every user in our user group, we will take their review history and split it in half. Half of their reviews will be used to create a model of their video games interest, and the other half of their reviews we will use for validation. We can use this data to calculate our metrics. To be more specific I will outline the steps of gathering these metrics:

For each user in our sample

1. Split user's review data in half
2. Create a model for the user's video game preferences. This is just a vector we can put into our vector space.

3. Suggest recommendations for this user, using either the collaborative filtering or our hybrid recommendation system.
4. Use the data from our recommendations and from their personal review data in order to calculate Precision and Recall from the given formulas.

Then we can average the Precision and Recall values for each user.

3.6 Time Complexity Analysis

Refer to 3.3.4 for the steps of our algorithm. We will describe the time complexity of recommending 100 video games to 1 user.

$$Y = n + n + c + f$$

N is number of reviews in the user's history. The first n in the equation represents calculating a user's average vector, n the second time represents gathering all the keywords from a user's item reviews. C is the cost of using the "Annoy" nearest neighbor library and reading the index that was generated of all the different vectors in order to find 100 nearest neighbors. This c operation is doing a search through an optimized index of vectors, it is a constant time, very fast operation, nowhere near brute force search. F is the cost of looking at the category keywords of each video game in our nearest neighbors in order to find the highest ranking video games.

3.7 Software Used

The different software packages we used are: PyTorch, fasti.ai (a helper library on top of PyTorch, "Annoy" Python library for finding nearest neighbors and Google Colaboratory for training our neural network models.

CHAPTER 4

RESULTS AND ANALYSIS

Our research goal is to show that by building a hybrid collaborative filtering and content-based filtering model we can improve recommendation performance, especially for user's that have few reviews (the cold start problem).

We described the algorithms, data and metrics we would use to compare performance between a pure collaborative filtering model versus our novel hybrid model (collaborative and content-based filtering). We will now discuss the results of our research.

4.2 Results

We will call the pure collaborative filtering model "Collab" and we will call our unique hybrid model "Hybrid". After testing 100 users with more than 100 video game reviews each and averaging the results, here are the results we obtained.

Table 1: Precision Comparison

	Collab	**Hybrid (our model)**
Mean Precision	0.014640	**0.015960**
Max Precision	0.094000	**0.110000**

Table 2: Recall Comparison

	Collab	**Hybrid (our model)**
Mean Recall	0.147731	**0.158242**
Max Recall	0.523810	**0.476190**

4.3 Analysis

As we can see from the data in Fig. 4-1 and Fig. 4-2, while using our hybrid recommendation model, both Precision and Recall values were improved. Precision improved by

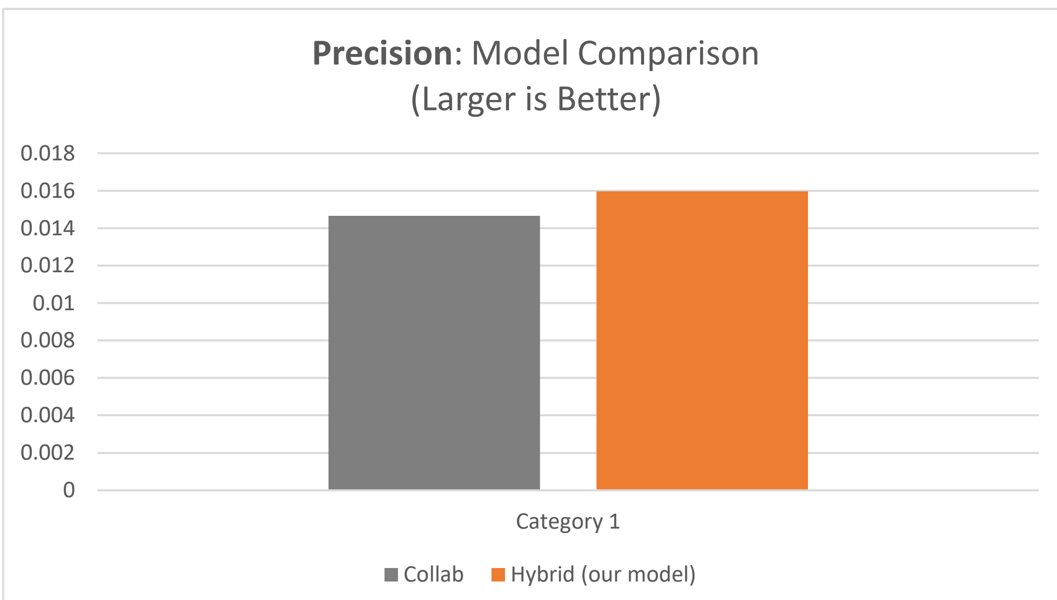

Figure 4-1: Precision Plot

nearly 9% and recall improved by 7%. Additionally, for our max precision, we see that our model returns a larger value. It is only in the case of max recall, where we see our model have a lower value.

In the last paragraph we were discussing only average precision and recall values. In Fig. 4-3 we can see a sample of individual users and their metrics. In Fig. 4-3 we can see that almost without fail, when we recommend to users with our hybrid recommendation model, the precision and recall scores improve.

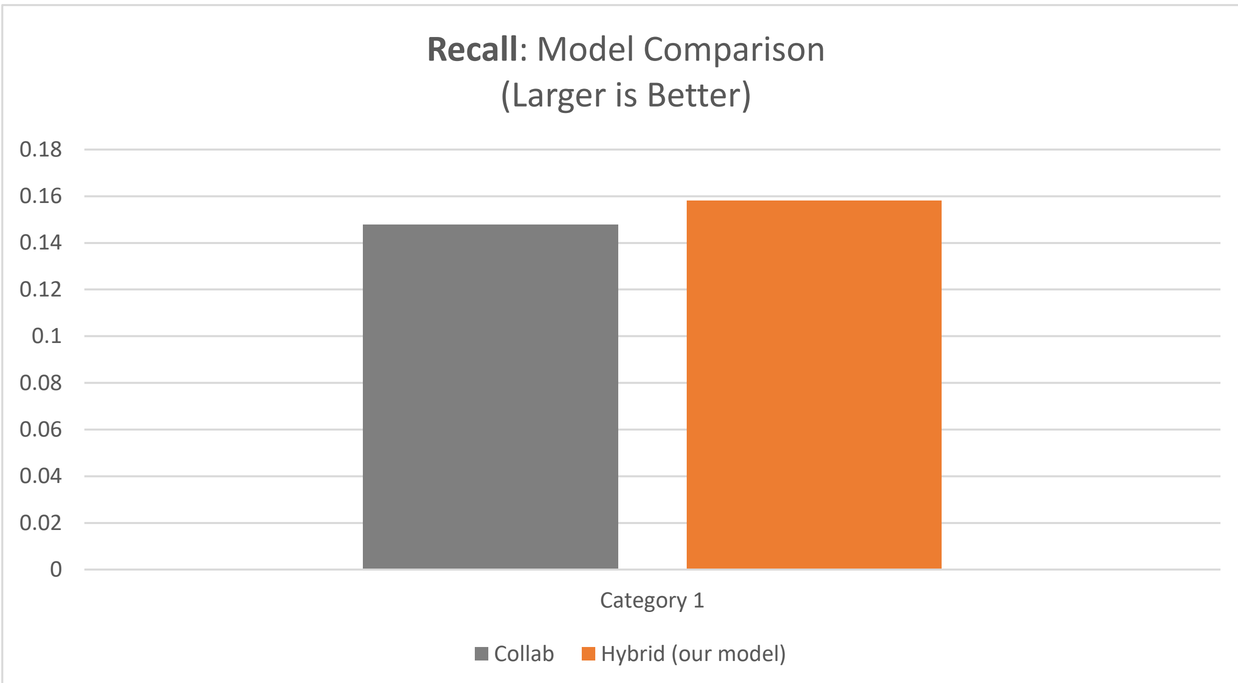

Figure 4-2: Recall Plot

What do these results tell us? We can see that with this user group, our recommendation performance improves when we use our hybrid system that utilizes the keywords from video game categories.

[89]:

	precision	recall	precision_kw	recall_kw
0	0.094	0.119593	0.110	0.139949
1	0.068	0.142259	0.078	0.163180
2	0.046	0.112745	0.048	0.117647
3	0.028	0.066667	0.036	0.085714
4	0.044	0.211538	0.048	0.230769
...	...	...	...	...
95	0.012	0.240000	0.014	0.280000
96	0.002	0.037037	0.004	0.074074
97	0.008	0.074074	0.010	0.092593
98	0.004	0.083333	0.006	0.125000
99	0.010	0.357143	0.010	0.357143

100 rows × 4 columns

Figure 4-3: Precision and Recall Model Comparison Per User

4.4 Literature Comparison

We can see that precision and recall improved with our hybrid model compared to the baseline collaborative model, but we still must consider if these results are good in terms of other research studies. For that question we look at literature comparing the performance of different collaborative filtering algorithms [34].

	Data set		
	Apparel	Book	Movie
Measure			
Precision	0.0088	0.0202	0.0648

Figure 4-4: Literature Collaborative Precision

In Fig. 4-4 we look at precision values for three different datasets that were tested and we can compare it against our precision value of **0.015960.** We see that our precision value belongs somewhere in the middle of these example values.

Measure	Data set		
	Apparel	Book	Movie
Recall	0.0356	**0.1320**	0.0468

Figure 4-5: Literature Collaborative Recall

In Fig. 4-5 we similarly look at recall values for three different datasets that were tested and we see how it compares against our model recall value of **0.158242.** In this example we see our recall value being higher in all three dataset cases.

What we can conclude from this comparison is that our research has the potential to be competitive against current literature metrics regarding recommendation systems.

4.5 The Cold Start Problem

In our thesis we state that this model is intended for use with users that have low-medium user reviews. The reason for this is unlike collaborative filtering, our model does not depend solely on user data, it also uses keywords of video games to make recommendations.

A clarification is needed, the term the cold start problem is sometimes used to refer to the situation with recommendation systems where there is a new user who has no previous user reviews. This is not what we are referring to in our research, we are strictly doing research to aid in the cold start problem with users that have low-medium reviews. It is not possible to make accurate recommendations to a user when we have absolutely no information about that user.

This model can be helpful for recommending to users with low-medium previous reviews because this method does not depend solely on user review data. Even if a user only has 1 or 2 user reviews, our algorithm will still compute a vector for this user and provide reasonable results based using out category keyword comparison. This is the power of content-based filtering algorithms, that make use of item content like category keywords. Therefore, this model can be helpful to solve the cold start problem for users with low-medium previous reviews.

Our research focused improving the cold start problem specifically when there were new users with a “small amount” of previous reviews. That is to say we were researching users who did have at least 1 previous item review.

4.6 Limitations

The problem with researching the cold start problem with a static dataset, is it can be harder to validate if our recommendations were of good quality. With a static dataset all we have is historical data. If we make a recommendation to a user, it may be a good recommendation but if the recommendation was not in their item history, then we could not validate it as relevant.

If we built an online recommendation system, then we would be able to more easily validate if a recommendation was good or not. We could validate to see if the user interacted with that recommendation or not.

4.7 Training Results

Fig. 4-6 and Fig. 4-7 provide us with statistics about the training phase of our neural network model. We see our validation loss decrease over 9 epochs.

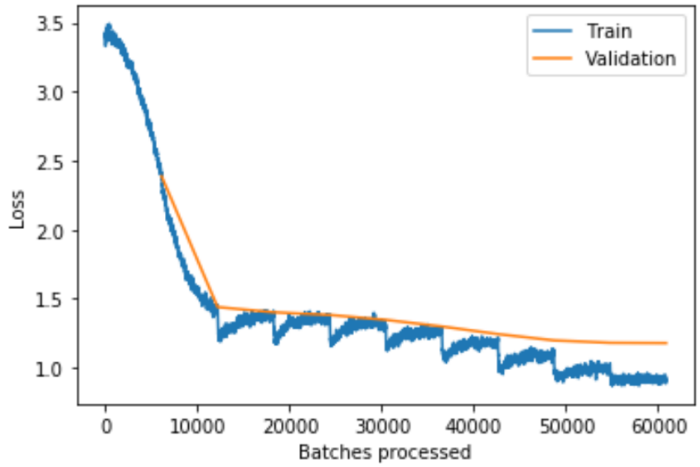

Figure 4-6: Validation Loss Over Time

epoch	train_loss	valid_loss	precision
0	2.393258	2.388412	0.014966
1	1.424343	1.439230	0.015712
2	1.398455	1.401489	0.016319
3	1.354882	1.382251	0.015741
4	1.309885	1.346496	0.015197
5	1.258569	1.298577	0.015124
6	1.216781	1.243524	0.015890
7	1.080528	1.197760	0.016257
8	1.018205	1.180618	0.015727
9	0.913583	1.178464	0.015823

Figure 4-7: Validation Loss Over Epochs

CHAPTER 5

CONCLUSION, FUTURE IMPROVEMENTS, DISCUSSION

5.1 Conclusion

This research proposes a novel technique for dealing with the cold start problem. By combining a neural network trained matrix factorization approach and a category keyword content-based filtering approach, our model provides better precision and recall metrics than a standard collaborative filtering approach and comparable results to literature. We also presented an argument that this hybrid system can be beneficial to users who have low-medium previous reviews because of less dependence on user review data.

5.2 Future Improvements

Due to lack of time we did testing on only one user group, on users with a high number of video game reviews. This made our results more straightforward as there was a lot of data for us to use for validation and for calculating our metrics of precision and recall. In the future we would like to do further testing with different user groups. The first group that comes to mind is a group of users who have only a small-medium number of video game reviews. This would correlate more closely to the discussion of cold start problem and would be valuable data. We could do a comparison against different user groups.

Also, in the future it would be beneficial for testing our recommendation relevancy if we deployed this recommendation onto the internet.

A final improvement we could implement is adding additional information in our content-based ranking. For example, instead of just using category keyword comparisons, we could also integrate user features like a user's age, gender, country of origin or analyze a user's written review for keywords.

5.3 Discussion

Based on our research I would recommend that more research should be done into the area of hybrid-based recommendation systems. Collaborative filtering is the most popular algorithm for recommendations systems that is widely covered in literature [7], but it is valuable for research to be done in other content-based fields of recommendation systems. There are clearly benefits to be had when combining collaborative filtering with other methods, such as with a keyword-based content filtering algorithm like our case. Benefits like dealing with the cold start problem and less of a dependence on user data.

REFERENCES

[1] Lee, Joonseok, Nisarg Kothari, and Paul Natsev. "Content-based Related Video Recommendations." (2016).

[2] Lee, Joonseok, and Sami Abu-El-Haija. "Large-Scale Content-Only Video Recommendation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[3] Zhang, Zi-Ke, et al. "Solving the cold-start problem in recommender systems with social tags." EPL (Europhysics Letters) 92.2 (2010): 28002.

[4] Vartak, Manasi, et al. "A Meta-Learning Perspective on Cold-Start Recommendations for Items." Advances in Neural Information Processing Systems. 2017.

[5] Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering." IEEE Internet computing 7.1 (2003): 76-80.

[6] Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016.

[7] Schein, Andrew I., et al. "Methods and metrics for cold-start recommendations." Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2002.

[8] Liu, Jiahui, Peter Dolan, and Elin Rønby Pedersen. "Personalized news recommendation based on click behavior." Proceedings of the 15th international conference on Intelligent user interfaces. ACM, 2010.

[9] Harper, F. Maxwell, and Joseph A. Konstan. "The movielens datasets: History and context." ACM Transactions on Interactive Intelligent Systems (TiiS) 5.4 (2016): 19.

[10] Bell, Robert M., and Yehuda Koren. "Lessons from the Netflix prize challenge." Acm Sigkdd Explorations Newsletter 9.2 (2007): 75-79.

[11] Basilico, Justin, and Thomas Hofmann. "Unifying collaborative and content-based filtering." Proceedings of the twenty-first international conference on Machine learning. ACM, 2004.

[12] Salter, James, and Nick Antonopoulos. "CinemaScreen recommender agent: combining collaborative and content-based filtering." IEEE Intelligent Systems 21.1 (2006): 35-41.

[13] Barragáns-Martínez, Ana Belén, et al. "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition." Information Sciences 180.22 (2010): 4290-4311.

[14] Shahabi, Cyrus, et al. "Yoda: An accurate and scalable web-based recommendation system." International Conference on Cooperative Information Systems. Springer, Berlin, Heidelberg, 2001.

[15] Tewari, Anand Shanker, Abhay Kumar, and Asim Gopal Barman. "Book recommendation system based on combine features of content-based filtering, collaborative filtering and association rule mining." Advance Computing Conference (IACC), 2014 IEEE International. IEEE, 2014.

[16] Popescul, Alexandrin, David M. Pennock, and Steve Lawrence. "Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments." Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 2001.

[17] Lin, Jovian, et al. "Addressing cold-start in app recommendation: latent user models constructed from twitter followers." Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2013.

[18] Bobadilla, JesúS, et al. "A collaborative filtering approach to mitigate the new user cold start problem." Knowledge-Based Systems 26 (2012): 225-238.

[19] Lops, Pasquale, Marco De Gemmis, and Giovanni Semeraro. "Content-based recommender systems: State of the art and trends." Recommender systems handbook. Springer, Boston, MA, 2011. 73-105.

[20] Van Meteren, Robin, and Maarten Van Someren. "Using content-based filtering for recommendation." Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop. 2000.

[21] Mooney, Raymond J., and Loriene Roy. "Content-based book recommending using learning for text categorization." Proceedings of the fifth ACM conference on Digital libraries. ACM, 2000.

[22] Blanco-Fernandez, Yolanda, et al. "Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems." IEEE Transactions on Consumer Electronics 54.2 (2008).

[23] Das, Abhinandan S., et al. "Google news personalization: scalable online collaborative filtering." Proceedings of the 16th international conference on World Wide Web. ACM, 2007.

[24] Tso-Sutter, Karen HL, Leandro Balby Marinho, and Lars Schmidt-Thieme. "Tag-aware recommender systems by fusion of collaborative filtering algorithms." Proceedings of the 2008 ACM symposium on Applied computing. ACM, 2008.

[25] Rose, Ian, et al. "Cobra: Content-based Filtering and Aggregation of Blogs and RSS Feeds." NSDI. 2007.

[26] Shardanand, Upendra, and Pattie Maes. "Social information filtering: algorithms for automating "word of mouth"." Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press/Addison-Wesley Publishing Co., 1995.

[27] Takeuchi, Yuichiro, and Masanori Sugimoto. "CityVoyager: an outdoor recommendation system based on user location history." International Conference on Ubiquitous Intelligence and Computing. Springer, Berlin, Heidelberg, 2006.

[28] Berjani, Betim, and Thorsten Strufe. "A recommendation system for spots in location-based online social networks." Proceedings of the 4th Workshop on Social Network Systems. ACM, 2011.

[29] Silva, Nitai B., et al. "A graph-based friend recommendation system using genetic algorithm." Evolutionary Computation (CEC), 2010 IEEE Congress on. IEEE, 2010.

[30] Majid, Abdul, et al. "A context-aware personalized travel recommendation system based on geotagged social media data mining." International Journal of Geographical Information Science 27.4 (2013): 662-684.

[31] Pazzani, Michael J. "A framework for collaborative, content-based and demographic filtering." Artificial intelligence review13.5-6 (1999): 393-408.

[32] Ting, Kai Ming. "Precision and recall." Encyclopedia of machine learning (2010): 781-781.

[33] Jianmo Ni, Jiacheng Li, Julian McAuley. "Justifying recommendations using distantly-labeled reviews and fined-grained aspects." Empirical Methods in Natural Language Processing (EMNLP), 2019.

[34] Huang, Zan, Daniel Zeng, and Hscinchun Chen. "A comparison of collaborative-filtering recommendation algorithms for e-commerce." IEEE Intelligent Systems 22.5 (2007): 68-78.

[35] Jacobson, Kurt, et al. "Music personalization at spotify." Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016.