Enhancing the NPO Start Recommendation System with Metadata

submitted in partial fulfillment for the degree of master of science

Eileen Kapel
12369462

master information studies
data science
faculty of science
university of amsterdam

YYYY-MM-DD

	External Supervisor	External Supervisor	Second External Supervisor
Title, Name	Dr Maarten Marx	Arno van Rijswijk	Robbert van Waardhuizen
Affiliation	UvA, FNWI, IvI	NPO	NPO
Email	maartenmarx@uva.nl	arno.van.rijswijk@npo.nl	robbert.van.waardhuizen@npo.nl

ABSTRACT

KEYWORDS

Recommendation systems; Video recommendation systems; Metadata

1 INTRODUCTION

Humans are tasked with making thousands of decisions daily which may range from selecting which outfit to wear to which television show to watch. These decisions can be made more efficient by recommendations, which are made by sorting through potentially relevant information and making recommendations customised to the individual user [8]. One system that employs recommendations is NPO Start[1]. It is a video-on-demand service from the NPO (Dutch Public Service Media) which gives people the ability to watch series, movies and documentaries as well as watch live television. At the moment there are personalised recommendations available for registered users, which is based on collaborative filtering. However, there is a lot of metadata available about the offered content which has been mostly unused until now. In this thesis, the metadata of broadcasts will be utilised to determine if it can improve the performance of the current video recommendation system.

This will be done by answering the main research question: *'Can a hybrid recommendation system using metadata perform better than the current recommendation system of NPO Start which uses collaborative filtering?'*. The research is split into three sub-questions:

(1) What is the current collaborative filtering performance?
(2) Can the performance of the current recommendation system be improved by implementing different hybrid recommendation methods?
(3) Which metadata features improve the performance the most?
(4) Are subtitles as a content feature an addition to the hybrid recommendation system?

The thesis is structured as follows: first, various related works of literature are outlined in section 2 that relate to the goal of the thesis. The methodology employed during the research is discussed in section 3, which is followed by its results in section 4. Subsequently, the conclusions of these results are presented in section 5 after which a discussion of the choices and possible future work follows in section 6.

2 RELATED LITERATURE

Recommendations are based on ratings which may be implicitly inferred from users their actions [8], like a click or a certain watch duration. There are three main approaches for building a recommendation system. First, content-based systems that utilize the information about items. Second, collaborative filtering systems which use user interactions with items. Lastly, hybrid recommendation systems which are a combination of the two previous approaches and exploits item and user interaction information to provide recommendations. The first two approaches each have their own shortcomings, like overspecialization, rating sparsity and cold-start [1], which hybrid systems aim to overcome to provide more accurate recommendations.

[1] www.npostart.nl

2.1 The NPO Start Recommendation System

NPO Start is a service that offers users the ability to watch video content on demand. This video content is displayed to users in so-called "ribbons" or rows that have a certain theme, like 'Populair', 'Nieuw' and 'Aanbevolen voor jou'. Each ribbon consists out of a ranked list of several items.

Users of the service who have an account have the ability to receive several personalized ribbons which contain items that are recommended for a specific user. These recommendations are materialized on the front page of the service in two ribbons, namely 'Aanbevolen voor jou' en 'Probeer ook eens'. The 'Aanbevolen voor jou' ribbon utilizes the history of user interactions with items to perform collaborative filtering. These user interactions are grouped on series level and evaluated by pairs of series which are frequently watched together, or coincide often, with the history of the user. Of these coincidences, the top 100 pairs are extracted which are ordered based on their frequency. A sliding window technique of the past 21 days is used for the interactions, which is based on the intuition that recent events are more relevant than older ones [1]. The items for the second personalized ribbon 'Probeer ook eens' are produced by multiplying the items of the 'Aanbevolen voor jou' list with their public value rating, and filtering out the items already displayed in the former ribbon. These recommendation lists are updated hourly at peak times and bihourly off-peak.

2.2 Hybrid Recommendation Systems

Hybrid recommendation systems are able to deal with scenarios where there is little data on new users and items, also called the cold-start problem, or when there is sparse interaction data available. This is done by joining content and collaborative data.

Different approaches exist for building a hybrid recommendation system, however most employ matrix factorization for the collaborative filtering part of the system. This approach was popularized due to solution of the Netflix Prize competition which employed matrix factorization using the alternating least square (ALS) algorithm [2, 4, 5]. Some works that have successfully employed matrix factorization for their hybrid recommendation system are Rubtsov et al. [9], Ludewig et al. [7] and Al-Ghossein et al. [1]. Rubtsov et al. used the LightFM library which is a Python implementation of a matrix factorisation model which can deal with user and item information [6] and uses a weighted approximate-rank pairwise loss. A combination of a k-nearest-neighbor technique and an ALS matrix factorization algorithm was employed by Ludewig et al. Content was incorporated by weighting the results with the IDF score of titles to produce the final list of recommendations. Lastly, Al-Ghossein et al. describes incorporating topics extracted using topic modeling with matrix factorization for recommendations.

Furthermore, neural networks have been combined into recommendation systems with matrix factorization, due to its recent popularity. Volkovs et al. [10] describes a two stage model for

playlist continuation which first employs weighted regularized matrix factorization to retrieve a subset of candidates and then uses convolutional neural networks and neighbour-based models for detecting similar patterns. In the second stage features of playlists and songs are combined with the items which produces the final ranking of the recommendations.

2.3 Personalized services

Recommender systems are frequently employed by services to provide a personalized experience to their users. This occurs in different domains, e.g. product recommendation by Amazon, music recommendation by Spotify and video recommendation by YouTube and Netflix.

Netflix is a special case where the whole experience of its service is defined by recommender systems [3]. This is primarily showcased on its homepage which consists out of rows of recommended videos with a similar theme. This is similar to the ribbons of NPO Start however NPO Start only employs recommended videos in a few ribbons. A few of the row themes are 'Top Picks', 'Trending Now' and genre rows, like suspenseful movies or action & adventures. Each row theme uses different data, signals and models to produce recommended videos which are ranked using a personalized video ranker. On top of that, pages are also personalized by selecting and ordering rows that are relevant and diverse for each user. Aside from the homepage, personalization is also offered while using search for which play data, search data and metadata is used. Netflix uses topic modeling, matrix factorization and probabilistic graphical models in its recommendation system. A/B testing is used to improve algorithms to enhance member retention and offline experiments are used for faster innovation.

The streaming service Spotify employs recommendations for automatic playlist continuation which is a task that adds one or more tracks to a playlist that fits the original playlist [11]. Spotify organized the ACM Recommender Systems Challenge 2018 with the purpose of improving this task. The most frequent approaches by teams for this challenge consisted out of two-stage architectures, matrix factorization, neural network models and learning to rank models.

3 METHODOLOGY

This section describes the methodology employed for answering the research questions. It consists out of a description of the data, the methods used and how these methods are evaluated.

3.1 Description of the data

The data for the recommendation systems consist out of interaction and item information.

3.1.1 Interaction information. Interaction information consists out of NPO event data. This data consists out of offers and choices, and describes the recommended items, the rank of each item, the total amount of items in a row and which is chosen.

3.1.2 Item information. Item information describes the content of each item which is provided by the Publieke Omroep Media Service (POMS) of the NPO and consists of 1.5 million media items, such as broadcasts, movies, or segments. Each item consists out of 37 columns describing metadata consisting out of a media id (mid), age rating, broadcaster(s), credits, descriptions, genres, images, locations, schedule events, etc. A focus is put on broadcasts which are available to stream on NPO start, resulting in n = 83.886 items. Each broadcast is part of a series and a total of 2490 unique series were identified. It should be noted that broadcasts are continuously updated, meaning that the total amount of items may differ. Eight metadata features were selected for the content recommender system which were derived from the POMS data, namely age rating, broadcaster, credits, description, genres, mid, series reference, subtitles and title. These features are described in Appendix 2. The metadata for each item is provided by program makers and the richness is shown in Figure 1. This shows that genres, description, and especially credits and subtitles have missing values.

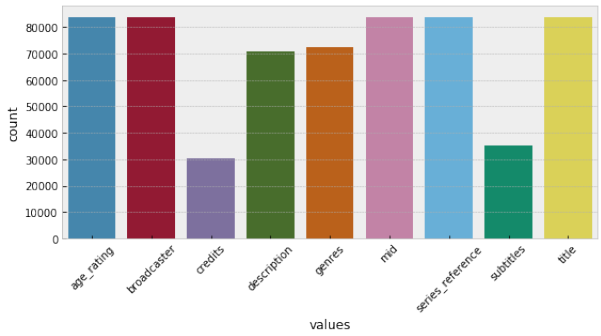

Figure 1: Richness of Content Features

The two categorical features are age rating and broadcaster. The distribution of these features are displayed in Figure 2 & 3. A majority of the items have the age rating 'ALL', however it should be noted that missing age ratings were filled by this particular rating category. Of the 30 broadcasters most of the items are broadcasted by the NPO. This is followed by the broadcasters EO, NCRV and NTR with about a third of the NPO count. The other broadcasters broadcast about 3000 items and there are even few who broadcast only tens of items.

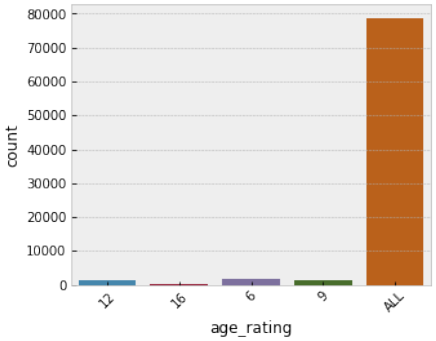

Figure 2: Distribution of Age Rating

Additionally, there are three textual features, namely title, description and subtitles. The average word length of these three

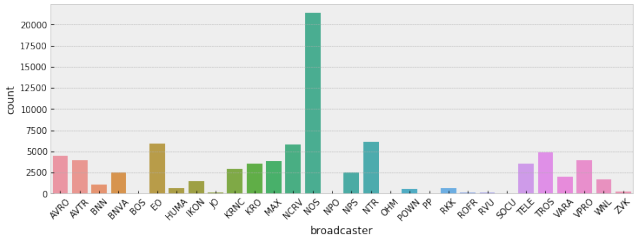

Figure 3: Distribution of Broadcaster

features and their mean on series and items level is described in Table 1. The length of the features is overall greater on series level than on items level. Titles have on average the fewest amount of words, followed by description and subtitles have the biggest amount of words on average. All the medians of the features lie below the mean, indicating that the lengths are not evenly distributed. The distribution of average title, description and subtitles length on series level is displayed in Fig. 4. Items with no description or subtitles were rewarded a length of zero. The three distributions all indicate a skewed distribution to the left, with the majority of the series having a short length which is accompanied by several big values. The subtitles length distribution indicates that a majority of the series have subtitles with a length of zero, thus not being available. Description length also has a few series with an amount of zero, but most seem to have a description present.

Table 1: Length of Textual Features

Feature	Series		Items	
	Mean	Median	Mean	Median
Title	3.3	3.0	2.5	2.0
Description	27.6	22.0	20.7	18.0
Subtitles	1687.0	500.7	1038.6	0.0

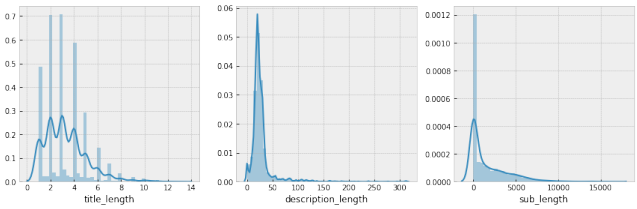

Figure 4: Distributions of Textual Features Length

Futhermore, there are two variables that consists out of lists, namely genres and credits. Each item either has multiple genres, one genre, or no genre assigned to it. A total of 53 different genres were identified. Genres have a main type indicated by an id of four integers, e.g. 3.0.1.7 'Informatief', and may have subtypes which are indicated by an id of five integers, e.g. 3.0.1.6.26 'Informatief, Religieus'. Additionally, the credits of each item also consists out of multiple credits, one credit or no credits. There were 5484 different unique credits, which have different roles such as presenter, guest or actor. The unique genres and unique people that are accredited have been one-hot encoded.

Lastly, TF-IDF on subtitles has been employed on series level of items, so on items with the same title, and the top-? words were also one-hot encoded.

3.2 Methods

The hybrid recommendation system incorporates metadata into a matrix factorization collaborative filtering model. The users are denoted by U, the items by I and the item features by F^I. The item features represent item metadata on series level (see Table 2). Interaction information is denoted by user-item interaction pairs $(u, i) \in U \times I$ which displays the interactions of users with items.

(Not finised)

3.3 Evaluation

To assess the quality of recommendations three metrics for evaluation were computed: precision@k (precision at k), AP@k (average precision at k) and CTR (click-through rate).

3.3.1 Precision@k. Precision@k is a metric that evaluates the proportion of top-k recommended items that are relevant to the user. Relevant items are denoted as a true positive (TP) which are positive predicted values. Precision is then given as the total number of predicted positives out of all predicted items (see equation 1).

$$Precision@k = \frac{TP}{k} \tag{1}$$

3.3.2 AP@k. AP@k evaluates the quality and rank of the recommended items. This metric is lower when positive predicted values do not appear at the top of the item list. It assesses the precision at each rank and multiplies it with the total amount of TPs (see equation 2.

$$AP@k = \frac{1}{TP} \sum_{i=1}^{k} \frac{TP\ seen}{i} \tag{2}$$

3.3.3 CTR. The CTR measures the proportion of users who choose to click on the ribbon with recommended items, opposed to times the ribbon was offered. The equation of the click-through rate is shown in equation 3.

$$CTR = \frac{number\ of\ click\text{-}throughs}{number\ of\ offers} \tag{3}$$

The higher the value of the metrics, the better. The version with the highest precision and CTR has the most success of recommending items that users are interested in, and the version with the highest AP@k is most successful in ranking the recommendations in a personalized manner.

4 RESULTS

5 CONCLUSIONS

6 DISCUSSION

ACKNOWLEDGMENTS

Acknowledgements

REFERENCES

[1] Marie Al-Ghossein, Pierre-Alexandre Murena, Talel Abdessalem, Anthony Barré, and Antoine Cornuéjols. 2018. Adaptive collaborative topic modeling for online recommendation. *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), 338–346.

[2] Robert Bell, Yehuda Koren, and Chris Volinsky. 2007. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 95–104.

[3] Carlos A Gomez-Uribe and Neil Hunt. 2016. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)* 6, 4 (2016), 13.

[4] Yehuda Koren. 2009. The bellkor solution to the netflix grand prize. *Netflix prize documentation* 81, 2009 (2009), 1–10.

[5] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[6] Maciej Kula. 2015. Metadata embeddings for user and item cold-start recommendations. *arXiv preprint arXiv:1507.08439* (2015).

[7] Malte Ludewig, Iman Kamehkhosh, Nick Landia, and Dietmar Jannach. 2018. Effective Nearest-Neighbor Music Recommendations. In *Proceedings of the ACM Recommender Systems Challenge 2018*. ACM, 3.

[8] Michael J Pazzani. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review* 13, 5-6 (1999), 393–408.

[9] Vasiliy Rubtsov, Mikhail Kamenshchikov, Ilya Valyaev, Vasiliy Leksin, and Dmitry I Ignatov. 2018. A hybrid two-stage recommender system for automatic playlist continuation. In *Proceedings of the ACM Recommender Systems Challenge 2018*. ACM, 16.

[10] Maksims Volkovs, Himanshu Rai, Zhaoyue Cheng, Ga Wu, Yichao Lu, and Scott Sanner. 2018. Two-stage model for automatic playlist continuation at scale. In *Proceedings of the ACM Recommender Systems Challenge 2018*. ACM, 9.

[11] Hamed Zamani, Markus Schedl, Paul Lamere, and Ching-Wei Chen. 2018. An Analysis of Approaches Taken in the ACM RecSys Challenge 2018 for Automatic Music Playlist Continuation. *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), 527–528.

A DATA

Table 2: Description of the Data Features

Feature	Type	Description
age rating	categorical	The viewer guide age rating consisting out of 6, 9, 12, 16 or ALL. Missing age ratings were filled by the rating ALL.
broadcaster	categorical	Broadcaster of the broadcast, e.g. NOS.
credits	list	The people accredited in the broadcast, such as presenters or guests.
description	string	Description of the broadcast. This is either the main description, otherwise the short description or the kicker.
genres	list	Genres of the broadcast denoted by a genre id and name, e.g. (3.0.1.6, [Amusement]).
mid	string	Unique media identifier.
series reference	string	A reference to the series of which the item is part of.
subtitles	string	The subtitles of the broadcast, which were extracted using the POMS subtitles API.
title	string	The main title of the broadcast.