Thesis Design
Enhancing the NPO Start Recommendation System with Metadata

Eileen Kapel

March 15, 2019

Abstract

NPO Start is a video-on-demand service from the NPO (Dutch Public Service Media). It gives people the ability to watch series, movies and documentaries as well as watch live television. At the moment there are personalized 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 unused until now. In this thesis, the metadata of broadcasts will be utilized to determine if it can enhance the current recommendation system.

1 Personal details

My email eileen.kapel@student.uva.nl

My supervisors email (UvA) maartenmarx@uva.nl

My supervisors email (company) arno.van.rijswijk@npo.nl

The wiki on my github account `github.com/ekapel22/masterthesis/wiki`

2 Research question

The main research question is *"Can content-based filtering using metadata enhance the current recommendation system of NPO Start which uses collaborative filtering?"*. The research is split into four sub-questions:

1. What is the current collaborative filtering performance?

2. How much can the recommendation system be improved by using different content-based filtering methods?

3. Which metadata features improve the recommendation the most?

4. Can the recommendation system be improved by using pre-processing steps?

3 Related literature

Humans are tasked with making thousands of decisions daily which can range from selecting which outfit to wear to which television show to watch. These decisions can be made more efficient by recommendations. These recommendations are made by sorting through potentially relevant information and making recommendations customized to the individual user [4]. These users may give their ratings explicitly or they may be inferred implicitly from the user's actions. The two main approaches for recommendations are content-based filtering and collaborative filtering. However, another possible approach is a hybrid recommender system which combines these two. These hybrid recommender systems try to overcome the shortcomings of both individual approaches and make use of more of the available information, with the consequence that the recommendations become more precise.

3.1 Similar research

There are two types of methods for a hybrid recommender system, one that combines the ratings of a separate content-based and collaborative filtering predictor, and one that incorporates a content-predictor into the collaborative filtering. These two hybrid approaches were researched by Melville, Mooney and Nagarajan in the domain of movie recommendations, and used ratings from the EachMovie dataset and content from the Internet Movie Database (IMDb) [3].

For the content-based prediction, a bag-of-words naive Bayesian text classifier was used that uses a vector of bag-of-words, where each bag corresponded to a movie feature. These features include the title, director, cast, genre, plot summary, plot keywords, user comments, external reviews, newsgroup reviews, and awards.

For the collaborative filtering, a neighborhood-based algorithm was used. The predictions were made by choosing a subset of users who are similar and using a weighted combination of their ratings.

The first hybrid method gets the ratings of the content-based and collaborative filtering predictor and outputs the average of them.

The second hybrid method designed by Melville, Mooney and Nagarajan where content was incorporated into a collaborative filtering approach is called Content-Boosted Collaborative Filtering (CBCF). First, a pseudo user-rating matrix was created using content-based prediction. This matrix was filled in with the actual rating when available, otherwise a predicted rating was used. Afterwards, the matrix was used for collaborative filtering.

The evaluation of the two methods was done by removing a subset of the ratings, and using the leftover ratings for training the recommender systems. These recommender systems were then tested on the removed ratings. The accuracy of the methods was evaluated by two metrics: statistical accuracy metrics and decision-support metrics. The first metrics are evaluated by comparing the predicted values to the user-provided values, which is valued by the mean absolute error. The decision-support metrics evaluate if the predictions are good and bad, by looking at the ratings from 1 to 5 and if a rating of 4 or above is given the item is considered good, otherwise it is considered bad.

The result was that the CBCF method resulted in predictions with the best accuracy for both metrics.

3.2 Similar domain

The most popular video-on-demand service currently is Netflix. The recommendation algorithm for Netflix was developed by an open competition called the Netflix Prize, where the best algorithm to predict user ratings for films was rewarded a grand prize. This was won by the "Bell-Kor's Pragmatic Chaos" solution [2]. The data that was made available for the development of the algorithm was one file of movie information, with an identifier, year of release and title, and another file of ratings, with a customer identifier, rating and a date. The winning algorithm used collaborative filtering by using the alternating least square (ALS) algorithm which uses matrix factorization [1].

4 Methodology

Content will be used to enhance the current recommendation system which uses collaborative filtering, which will result in a hybrid recommender system. The current algorithm enhances the user experience by giving recommendations based on their viewing behavior, insight into their viewing history, the ability to continue watching on different devices and the ability to watch programs later or follow them.

The conditions for this method, that fuses content into the recommendation system, to work is that good features should be chosen that correctly reflect the content for recommending. It should be consistent and dividable into categories.

Shortcomings of content-based recommendations is that it only recommends content that it has seen before, meaning that something that hasn't been encountered will not be recommended. A big shortcoming of collaborative filtering is that it has a cold-start problem, meaning that in the start less appealing recommendations are given to an user because less is known about that user. Sparsity is another shortcoming for collaborative filtering, which is caused by a very small amount of items being rated making training more difficult.

Advantages of a hybrid recommender system is that it utilizes the available information more, which consequently leads to more precise recommendations. The hybrid system overcomes the cold-start problem, since similar content can be used for recommendation when less is known about an user. Sparsity is also overcome since metadata can help in making more reliable predictions for users which can then be used in the collaborative filtering, meaning that more ratings can be used in the system.

The recommendations that come out of hybrid recommender system can be determined in two ways: by incorporating content into the collaborative filtering algorithm, or by combining the results of the two algorithms in the end. In this thesis, the content will be incorporated in the existing collaborative filtering algorithm. The exact way how this is done will be investigated during the thesis.

4.1 Data

The data comes from the POMS (Publieke Omroep Media Service) of the NPO and consists out of all the content that is offered on the NPO Start platform, but in this project we will focus on the broadcasts. These broadcasts are about 30% of the total content and they consist out of seasons and episodes which increase every single day. Metadata is saved for each broadcast, which consists out of the id, season number, episode number, date, time, title, description, subjects, subtitles, entities, etc. The metadata is quite inconsistent with quite a lot of NA's values, which are most common in the features subtitle and image information. The mean length of a title is about 14 words, about 113 for the short description and about 209 for the description. The determination of exactly which features of the data will be used in the method is going to be investigated.

The infrastructure of the company is mainly Google Cloud based and projects are almost exclusively executed in Python and Spark.

4.2 Evaluation

The beforementioned method(s) will be evaluated using online evaluations or so-called A/B tests. The company has experience with these tests and mainly compares these two versions using the CTR (Click-Through Rate). The amount of minutes that a user watches is also used for the evaluation. A higher CTR and amount of watch minutes using the researched method will result in a positive evaluation.

5 Risk assessment

5.1 Inconsistent metadata

The overall data is clean and has been preprocessed by the company, however it is quite inconsistent. Program makers are tasked with filling out the description and other necessary information for a broadcast, which is done differently in each broadcasting department and by each person. If the current variables are not consistent or rich enough, steps could be taken to utilize the subtitles of a broadcast as a feature by extracting important topics out of them.

5.2 Evaluation risks

The wished evaluation method to be used is A/B testing. Since this is an online evaluation method that is tested on customers, it needs to be ensured that the developed recommender system does not create latency or errors that might badly influence the image of NPO Start. The back-up for the A/B testing is an offline evaluation that uses the ratings of a subset of broadcasts. From this subset a percentage will be removed, on which the current and developed recommender system will make predictions based on the ratings that were not removed. The results of these systems will then be compared using their root-mean-square error.

6 Project plan

The project plan consists out of a total of 12 weeks and is shown in Table 1.

Week		Achievements
1	01/04 - 05/04	Introduction done. Data cleaned. Data richness investigated per broadcaster. Key topics extracted from subtitles.
2	08/04 - 12/04	Literature done. Pre-processed and described data. Investigated and described current performance.
3	15/04 - 19/04	Method section done. Features investigated and selected. First content-based filtering method was set up.
4	22/04 - 26/04	Worked out results for the first method. Second content-based filtering method was set up. Worked out results for the second method.
5	29/04 - 03/05	Compared performance of content-based filtering methods. Investigated integrating content-based filtering and collaborative filtering.
6	06/05 - 10/05	Mid-presentation: preliminary results evaluated. Hybrid recommender system was set up.
7	13/05 - 17/05	A/B testing was set up.
8	20/05 - 24/05	Checked results of A/B testing. Experimented and tuned hyperparameters.
9	27/05 - 31/05	Checked results of A/B testing. Experimented and tuned hyperparameters.
10	03/06 - 07/06	Results done.
11	10/06 - 14/06	Conclusion & discussion done.
12	17/06 - 21/06	Thesis done.

Table 1: Project plan

References

[1] Robert Bell, Yehuda Koren, and Chris Volinsky. 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*, pages 95–104. ACM, 2007.

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

[3] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. *Aaai/iaai*, 23:187–192, 2002.

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