Content-Based Recommendation System

CAMILA LARANJEIRA (2020656790), Universidade Federal de Minas Gerais, Brazil

1 INTRODUCTION

The current document outlines an implementation of a content-based recommendation system to solve the movie recommendation problem presented as the second programming assignment of the Recommendation System 2020.2 course at PPGCC/UFMG. The chosen approach is a straightforward TF-IDF computation over movie genres.

1.1 Dataset

Three sets of data were provided for this assignment. The training set contains over 300 thousand interactions from around 34, 100 users with approximately 19, 400 items. It contains explicit feedback on a 0 to 10 rating scale, followed by a timestamp providing a temporal relationship among the samples. The test set however does not provide timestamps nor ratings, only a single column csv document containing 4, 872 users and 9, 698 items, amounting to 77, 276 interactions. The data is arranged such that the test set contains 40.8% of cold start samples, due to interactions in which either the user, item or both are unknown.

A third document is provided, with content information for all items in both training and test sets. It contains very diverse information, as shown in Fig 1.


```
{'Title': 'Edison Kinetoscopic Record of a Sneeze',
 'Year': '1894',
 'Rated': 'N/A',
 'Released': '09 Jan 1894',
 'Runtime': '1 min',
 'Genre': 'Documentary, Short',
 'Director': 'William K.L. Dickson',
 'Writer': 'N/A',
 'Actors': 'Fred Ott',
 'Plot': "A man (Edison's assistant) takes a pinch of snuff and sneezes. Th
is is one of the earliest Edison films and was the first motion picture to
be copyrighted in the United States.",
 'Language': 'N/A',
 'Country': 'USA',
 'Awards': 'N/A',
 'Poster': 'N/A',
 'Metascore': 'N/A',
 'imdbRating': '5.9',
 'imdbVotes': '988',
 'imdbID': 'tt0000008',
 'Type': 'movie',
 'Response': 'True'}
```


Fig. 1. Instance from the content set provided.

Posing as a competition on Kaggle [1], the assignment allows the submission of estimated ratings, which are automatically evaluated according to the ground truth, returning a single measure of Root Mean Square Error (RMSE).

[1] https://www.kaggle.com/t/7e88854cbdbf47968d45eb20a0f4fd3d

Author's address: Camila Laranjeira (2020656790), camilalaranjeira@ufmg.br, Universidade Federal de Minas Gerais, Av. Pres. Antônio Carlos, 6627, Belo Horizonte, MG, Brazil, 31270-901.

2 METHODOLOGY

Our approach consists in leveraging genre information as input to a TF-IDF approach, in order to represent both items and users as vectors in R^g with g being the number of genre categories on the dataset.

The first step of our approach is to load the content and preprocess genre information. Preprocessing consists in tokenizing genre description by splitting the comma separated content (refer to Fig 1), and converting each term to lower case. Then, we build two data structures: (1) a description for each item as a list of its genres and (2) a vocabulary with pairs $(genre, frequency)$ which will later feed TF-IDF computation. We have 29 genre categories, i.e., the vector space in which both items and users will be project is in R^{29}.

Figure 2 shows the vocabulary pairs. It is worth noting the large variability of frequencies, with categories such as drama belonging to over 10 thousand documents, while talk-show appears in a single document. That behaviour highlights the importance of a normalized metric such as TF-IDF to compute user and item vectors.

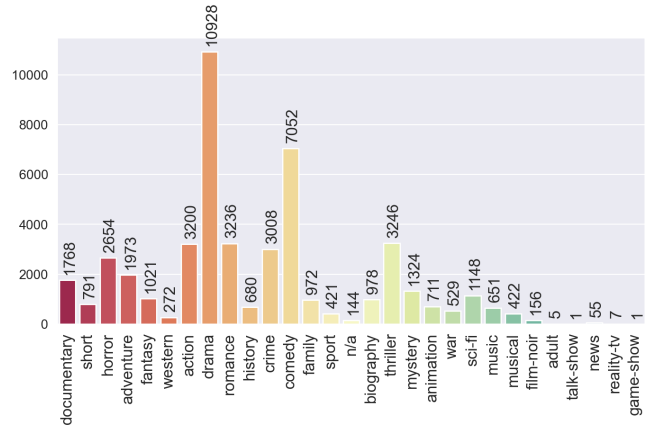

Fig. 2. Frequency of each term in the vocabulary.

Following, we project both items and users in the vector space of genres. For an item i, we calculate TF-IDF for each term t in the vocabulary, defined as

$$\vec{i}_t = tf\_idf_{i,t} = tf_{i,t} \times log(\frac{n}{n_t}),$$

such that each item is represented as a vector $\vec{i} \in R^{29}$. Since the genre description of each item has only a single occurrence of each corresponding term, for every i and t on the dataset, $tf_{i,t} = 1$. In other words, the components of an item vector are either zero for non-occurring terms, or the inverse document frequency of the remaining terms.

Each user u is represented as a linear combination of every item they have rated, i.e.,

$$\vec{u} = \sum_{i \in |I_u|} r_i \vec{i},$$

with I_u as the set of items the user u has rated r_i. Finally, to estimate the similarity between a user u and an item i, first we calculate the cosine similarity as follows

$$cossim = \frac{\vec{u}\vec{i}}{||\vec{u}||||\vec{i}||}.$$

Then, our similarity estimation is calculated w.r.t. the user's range of ratings. We consider *cossim*, which varies within the range $[0, 1]$, as a weighting factor as follows

$$\hat{r_{ui}} = \bar{r_u} + r_u^{min} + (r_u^{max} - r_u^{min}) \times cossim$$

with r_u^{min} and r_u^{max} being the deviation from the mean for a user's minimum and maximum ratings respectively. For example, a user with $\bar{r_u} = 7.9$, $r_u^{min} = -0.9$ and $r_u^{max} = 2.1$ has rated within the range $[7, 10]$. Cosine similarity weights how much the users deviates from their mean rating.

We consider three different rating scenarios, mainly since content-based recommendation still does not handle user cold starts. They are:

- User cold start + Rated item: Since we know nothing about the user, a statistically balanced average of the item's rating is computed as

$$\hat{r}_{u,i} = \alpha \bar{r}_i - z \frac{\sigma_i}{\sqrt{|U_i|}}.$$

- User cold start + Unrated item: A global average of ratings $\hat{r}$ is computed.
- An estimated rating $\hat{r}_{ui}$ is computed based on the similarity measure previously established.

3 EXPERIMENTS AND RESULTS

The main goal of our experiment is to validate the rating estimation as a function of the user's previous ratings, using similarity as a weighting factor of how much the user deviates from their mean rating. Additionally, we reinforce previous experiments from the first programming assignment, experimenting with filtering users with few interactions, i.e., considering them as cold start. The number of interactions used as threshold for filtering was set to 20, the best performing value from our previous work.

Table 1 compiles all experiments of the present programming assignment. Note that filtering users with few interactions proved once again to be a valuable pre-processing step. Besides, using the similarity as weight also appears to be a beneficial addition to the methodology, decreasing the distance metric RMSE considerably.

Setup	RMSE
Raw Similarity + All Users	4.46
Raw Similarity + Filtered Users	3.46
Similarity as Weight + Filtered Users	2.28
Random	4.10
User Average	2.08
Item Average	2.00

Table 1. Experiments.

4 CONCLUSIONS

This assignment proposed a movie recommendation challenge based on a content-based filtering. We suggested TF-IDF estimation over genre information, using the estimated similarity between user and item as a weighting factor of how much a given user would deviate from their mean rating. By exploring multiple settings of the proposed approach, we found that filtering users with few interactions remains a valid pre-processing step, while also validating our proposal of using similarity as a weighting factor.

Although results were not exceptional, since it still performs below user average and item average, we believe the results are very positive given that we explore only a single information from the item and propose a computationally light approach.