An evaluation on potential Recommender Systems for “DU Kulturmagazin”

Delft University of Technology, Faculty EEMCS
Marc Juchli (m.b.juchli@student.tudelft.nl)
Mateusz Garbacz (m.garbacz@student.tudelft.nl)
Sharif Jacobino (s.e.jacobino@student.tudelft.nl)

Introduction

The DU Kulturmagazin is a long established Swiss cultural magazine, founded in 1941. The company publishes on a monthly basis, skipping one month in the summer and winter, resulting in 10 editions per year. As of today, 876 editions exist in total. The characteristic of this magazine is of broad interests, considering such a low number of editions. Although the topics are all related to art or culture to some extent, it remains hard to distinguish specific user interests.

In this report, we analyze various recommender systems for the e-commerce setup of the company. We prototype possible setups given the user- and content data which was made available to us for this study. The goal of this study is to evaluate which recommender systems would be an appropriate choice for an integration into the online shop or for targeted newsletter marketing campaigns. Considered are content-based filtering methods based on text, collaborative filtering based on e-commerce transactions as well as a hybrid approach which incorporates transactional as well as content related information.

The report is structured as follows. At first, an overview of the available data is given in Section *Data*. In Section *Prototypes* we present three approaches of possible recommender systems, followed by their evaluation in Section *Evaluation*. Finally, in Section *Conclusion* we conclude our results and provide suggestions to the company.

Data

Various types of data can be taken into account, all of which are a product of the website: http://www.du-magazin.com. We consider the data of customers that had at least 5 purchases, namely 140 accounts. Moreover, we limit the issues considered to issue numbers 800-876 in order to focus only on the newest editions, for which we have the most purchases. After filtering the dataset we get 1021 purchases, which indicates high sparsity of the later established User Item Matrix.

Textual Content

For editions 823-876 there are .pdf files available containing the entire content. For editions 800-822 we had to consult E-periodica (http://www.e-periodica.ch/) which archived all editions and provided us with .jpg files, one for every page of a magazine. The text from the .pdf files were extracted using the python package *textract* which uses *pdftotext* in the background. For the .png files we used the python implementation of *tesseract-ocr*. All the extracted text was normalized replacing special characters with regular characters for easier processing.

Transactions

E-commerce transactions were gathered from the productive database and preprocessed such that transaction records include informations of users and their purchases. The data type can therefore be described as follows:

```
completed_at | magazine_number | issue_date | email
```

As one recognizes, the *email* serves as the identifier for the user.

Prototypes

Content-based filtering

In order to solve the cold-start problem when a new customer visits the e-commerce website, the content-based filtering approach should provide the user with a recommendation based entirely on similarities between magazines. When the customer selects a magazine a list of similar issues would be shown.

Particularly, we describe a *TF-IDF (Term Frequency - Inverse Document Frequency)* implementation to parse through the entire textual content of a magazine to identify distinct phrases and find similar magazines. Our TF-IDF approach looks at *n-grams* (uni-, bi- and tri-grams) in the text of the magazines and divides them by the number of times of their appearance in the magazine. This way, the terms which are more distinct get a higher score, and terms which appear more often get a lower score. Eventually, *cosine similarity* is used to measure the similarities between the TF-IDF feature vectors of the magazines and identify which magazines are the closes to each other. The implementation used is based on the one presented on the website [2].

As a second similarity method we use *word Shingling* with *Jaccard similarity* similarity metric. Shingles of 4 and 5 characters are extracted from each magazine and the Jaccard similarity is computed between each set of shingles. As a result, again, similarities between each magazine can be identified.

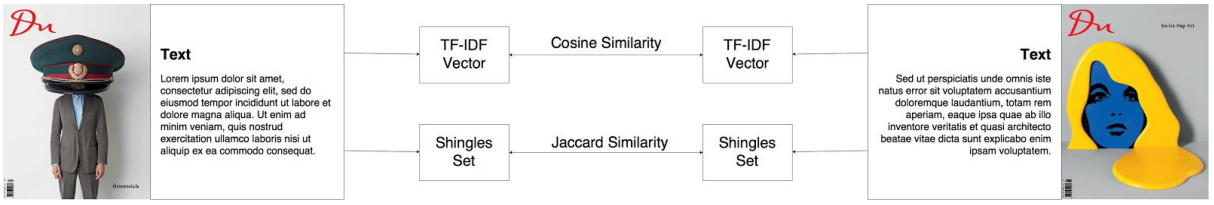


Figure 1: *Content-based similarity measure methods*

Figure 1 summarizes the two described processes. The similarity measures are further being used in *hybrid approach,* which is described in a section. However, we have not focused on a recommender system based only on the distances.

Collaborative filtering

The collaborative filtering approach focuses on making recommendations based on past user transactions. Thereby, we hope to find patterns in people's buying behaviour and take this knowledge to our advantage in order to recommend people with similar buying behaviour those magazines which they have not bought yet but do fit into their behavioural pattern.

The method we have chosen to accomplish collaborative filtering is the graph-based model presented in Z. Huang et al. [1], which is an associative retrieval techniques that has been shown to perform well for sparse data sets. Figure 2 therefore shows an illustration of this graph approach containing the customers and magazines they bought.

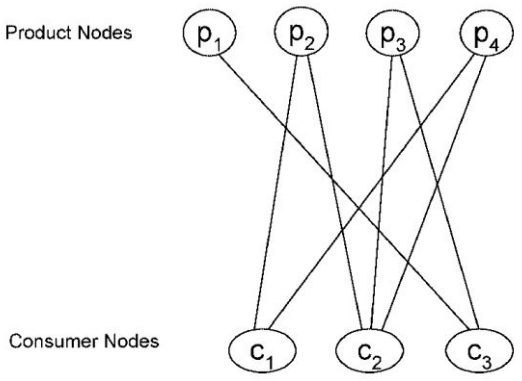


Figure 2: *Graph-based collaborative filtering approach, Z. Huang et al. [1]*

Given 1021 transactions from 141 users who have bought at least 5 magazines, a graph was built that included 63 magazines, a subset of [800 … 876].

Hybrid approach

In order to take advantage of both transactional and content data at once, we further present a hybrid approach.To incorporate the collaborative and content-based recommendation we apply the advanced feature engineering, as in Factorization Machines. For each transaction we generate a feature vector consisting of: the user features with 1 at the user's dedicated feature and 0 elsewhere, item features similarly representing the issues bought and finally the distances between issues purchased and all the issues in the dataset taken from the content-based approach. The first two types of features represent the collaborative side, while the rest of the features represent the context. The label for each transaction is 1, since we can only model the class of purchases, a single class problem. The complete set of features is visualized in the Table 1.

	User				**Item**				**Distance from items**				
ID	U1	U2	...	Un	I1	I2	...	In	D1	D2	...	Dn	Label
1	0	0	...	1	1	0	...	0	0	0.2	...	0.8	1
2	1	0	...	0	0	0	...	1	0.5	0.4	...	0	1
3	1	0	...	0	0	1	...	0	0.3	0	...	0.6	1

Table 1: *Features extracted and labels for each transaction*

Then, on the training data we fit a one class SVM [3] model, which learns the distribution of the training distances and finds a nonlinear boundary separating the representative training samples from the outliers. The recommendation can be done by comparing the distance to the class boundary of the generated transaction data. The distance is multiplied by -1 if the data is on the outliers' side and positive on the representative samples' side. Therefore, the most recommendable issues would be those having the maximum value of the modified distance.

Evaluation

For the evaluation of the recommender systems, we have decided to measure the *recall* depending on the number of recommendations produced by a system. In that sense, the more magazines the user actually bought from the recommended ones, the better the system will be rated.

Method

For each user we extract 20% of his/her transactions as a *testing set*, i.e. if a user has only 5 purchases his test set would consist of only one sample, which is not used in *training*. Then, for each user we find the *N* (e.g. 10) highest recommendations from all the issues not present in the training set and measure recall. Further, we average the recall over all users to find the percentage of how likely it is that an average user would purchase a magazine given the set of recommended items. Therefore 10 runs of this method were performed to again produce averaged results in order to find a stable measure for a given model. Since the DU Kulturmagazin is most interested in convincing readers to purchase as many issues as possible, while not disturbing them with too many recommendations, we find this evaluation method to meaningful for this analysis.

Results

To summarize our findings, we have tested different recommendation models with different parameters and distance metrics such as *TF-IDF* and *Shingles*. The mean recall for different number of recommendations can be seen in Figure 3.

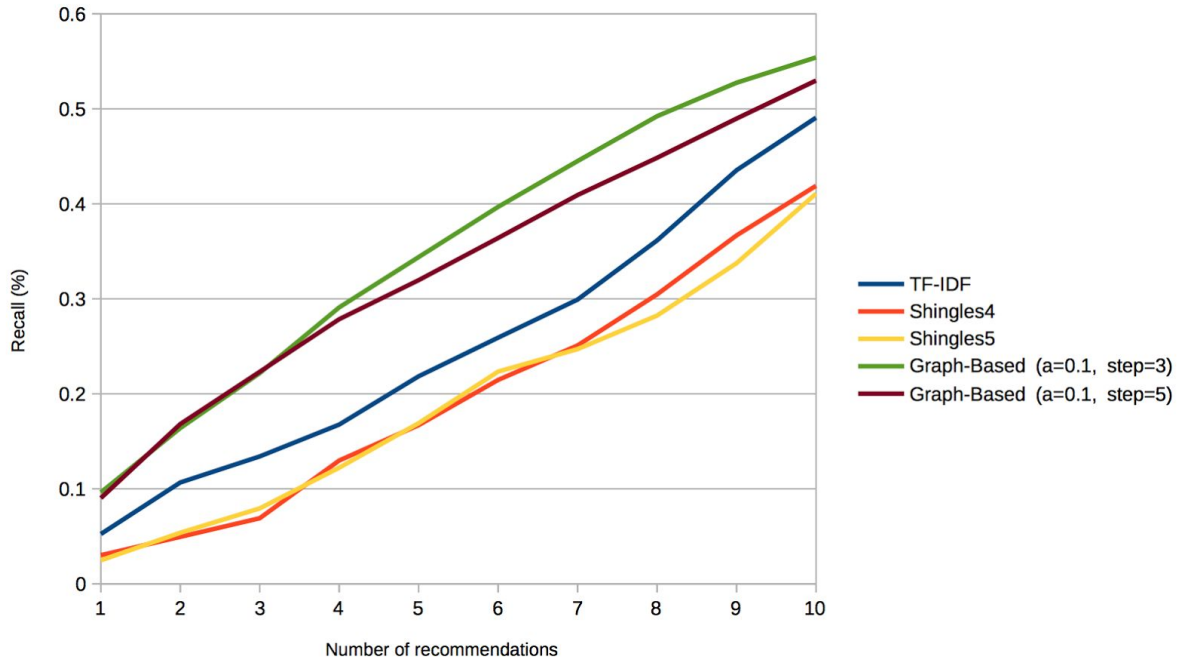


Figure 3: *Comparison of recall for collaborative and hybrid recommender methods*

Clearly, the graph based approach yields better results than the hybrid one. We see that the maximum step size of 3 has the highest recall for different all the numbers of recommendations. When it comes to the hybrid approach, the Tf-idf distance features allow

the classifier to give better results than the shingles-based metrics. We suppose that graph-based approach works better, because there are not many transactions for each user. The graph-based approach is designed in such a way that it can handle the sparsity, however, the classifier requires large amounts of data. Moreover, we find that most of the users are not consistent in buying the magazines by topic, instead they tend to purchase a couple of consecutive issues at once. This is the reason why the first couple of recommendations are worse for the hybrid approach, which tries to propose similar issues. When it comes to the graph-based approach, it mostly finds customers that bought similar consecutive items and therefore there is a high chance of getting a high recall results.

Having in mind the results, we see that the graph-based approach can match the user's buying behaviour better. However, we are still not sure whether a user would rather buy an item that is similar to what he is interested in or issues that other people with similar buying behaviour buy. Therefore, in the future, we would like to evaluate this by making actual recommendations to the magazine readers using these two approaches and testing which one on increased the sales more on average.

Conclusion

In this project we have created two distinct recommender systems for the "Du Kulturmagazin". The first one is based purely on the collaborative data, while the other takes into account the topical interests of the users based on the content of the magazines. Based on our evaluation metric we find that the former approach matches the user's buying behaviour more accurately. Although the results are promising, we suppose that the system requires further testing on a bigger set of actual users to see which approach makes the user more willing to make a purchase.

Furthermore, the content-based filtering method was not yet evaluated as an independent recommender system but rather as a feature for the hybrid approach as purely content-based recommender systems are rarely seen in practice. As this helped to improve the results, we suggest to evaluate the current implementation in more detail and perhaps make use of it on the company's e-commerce environment directly.

Finally, the question that remains to be answered is whether a recommender system in general is an appropriate suggestion to the e-commerce system of the "DU Kulturmagazin" given the relatively low amount of data available for a recommender system. We conclude that indeed the performance could be improved with more data available as the dimensionality is relatively large with respect to the samples available. However, the accomplished recall rate shows that sales can be improved in the current context.

References

[1] Huang, Zan, Hsinchun Chen, and Daniel Zeng. "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering." ACM Transactions on Information Systems (TOIS) 22.1 (2004): 116-142.

[2] [Online]. Available: http://blog.untrod.com/2016/06/simple-similar-products-recommendation-engine-in-python.html. [Accessed: 30.06.2017].

[3] [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html. [Accessed: 30.06.2017].