Neural Content-Collaborative Filtering for News Recommendation

by

Dhruv Khattar, Vaibhav Kumar, Manish Gupta, Vasudeva Varma

in

2018 European Conference on Information Retrieval
(*ECIR-2018*)

Grenoble, France

Report No: IIIT/TR/2018/-1

Centre for Search and Information Extraction Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
March 2018

Neural Content-Collaborative Filtering for News Recommendation

Dhruv Khattar, Vaibhav Kumar*, Manish Gupta†, Vasudeva Varma
Information Retrieval and Extraction Laboratory
International Institute of Information Technology Hyderabad
dhruv.khattar, vaibhav.kumar@research.iiit.ac.in, manish.gupta, vv@iiit.ac.in

Abstract

Popular methods like collaborative filtering and content-based filtering have their own disadvantages. The former method requires a considerable amount of user data before making predictions, while the latter, suffers from over-specialization. In this work, we address both of these issues by coming up with a hybrid approach based on neural networks for news recommendation. The hybrid approach incorporates for both (1) user-item interaction and (2) content-information of the articles read by the user in the past. We first come up with an article-embedding based profile for the user. We then use this user profile with adequate positive and negative samples in order to train the neural network based model. The resulting model is then applied on a real-world dataset. We compare it with a set of established baselines and the experimental results show that our model outperforms the state-of-the-art.

1 Introduction

A popular approach to the task of recommendation is called collaborative filtering (CF) (Bel07)(Ren05)(Sal07) which uses the user's past interaction with the item to predict the most relevant content. Amongst the various approaches for collaborative filtering, matrix factorization (MF) (Kor08) is the most popular one. However, it requires a considerable amount of previous history of interaction before it can provide high quality recommendations. It also drastically suffers from the problem of item cold-start, handling which is very crucial for news recommendation.

Another common approach is content-based recommendation, which recommends based on the level of similarity between user and item feature/profile. Although it can handle item cold-start, it suffers from the problem of over-specialization. Both, CF and content-based cannot directly adapt to the temporal changes in users interests.

In general, a news recommender should handle item cold start very well due to the overwhelming amount of articles published each day. It should also be able to adapt to the temporal changes in the users interests. In case of news, the content of the news article and the preference of a user act as the most important signals for news recommendation. In order to do this, we come up with a hybrid approach for recommendation.

Our model consists of two components. For the first component, we utilize the sequence in which the articles were read by the user and come up with a user profile. We do this as follows:

1. First, we learn the doc2vec (Le14) embeddings for each news article by combining the title and text of each article.

2. We then choose a specific amount of reading history for all the users.

3. Finally, we combine the doc2vec embeddings of each of the articles present in the user history using certain heuristics which preserves the temporal information encoded in the sequence of articles read by the user.

*Author had equal contribution.
†The author is also an applied researcher at Microsoft.

In: D. Albakour, D. Corney, J. Gonzalo, M. Martinez, B. Poblete, A. Vlachos (eds.): Proceedings of the NewsIR'18 Workshop at ECIR, Grenoble, France, 26-March-2018, published at http://ceur-ws.org

The second component then captures the similarity between the user profile and the candidate articles by first computing an element-wise product between their representations followed by fully connected hidden layers. Finally, the output of a logistic unit is used to make predictions. We pose the problem of news recommendation as that of binary classification in order to learn the parameters of the model. We only rely on the implicit feedback provided by the user. The first component enables us to understand the user preferences and model the temporal changes in their interest thereby giving us the advantages of a content-based recommendation system. While, the second component models the user-item interaction in a manner similar to that of matrix factorization giving us the advantages of a collaborative filtering based recommender system.

To summarize, the contributions of the work are as follows:

1. We use doc2vec embeddings of each news article in order to come up with user profiles for each user which encapsulates information about the changing interests of the user over time.

2. We use a deep neural architecture for news recommendation in which we utilize the user-item interaction as well as the content of the news.

3. We pose the problem of recommendation as that of binary classification in order to learn the parameters of the model by only using the implicit feedback provided by the users.

4. We perform experiments to show the effectiveness of our model for the problem of news recommendation.

2 Related Work

There has been a lot of work on recommender systems with a myriad of publications. In this section we attempt to review work that is closely associated to ours.

Collaborative Filtering Collaborative Filtering is an approach of making automatic prediction (filtering) about the interests of a user by collecting interests from many related users. Some of the best results are obtained based on matrix factorization techniques (Kor09). Collaborative Filtering methods are usually adopted when the historical records for training are scarce.

Content-based Filtering Content-based recommender systems try to recommend items similar to those a given user has liked in the past (Lop11)(Sai14)(Kum17). The common approach is to represent both the users and the items under the same feature space. Then similarity scores could be computed between users and items. The recommendation is made based on the similarity scores of a user towards all the items. The Content-based Filtering methods usually perform well when users have plenty of historical records for learning.

Hybrid of CF and Content-based Filtering As a first attempt to unify Collaborative Filtering and Content-based Filtering, (Basilico and Hofmann 2004) proposed to learn a kernel or similarity function between the user-item pairs that allows simultaneous generalization across either user or item dimensions. This approach would do well when the user-item rating matrix is dense (Bas04). However in most current recommender system settings, the data is rather sparse, which would make this method fail.

Neural Network based approaches Early pioneer work which used neural network was done in (Sal07), where a two-layer Restricted Boltzmann Machine (RBM) is used to model users' explicit ratings on items. Recently autoencoders have become a popular choice for building recommendation systems (Che12)(Sed15)(Str15). In terms of user personalization, this approaches shares a similar spirit as the item-item model (Nin11)(Sar01)(Kum17) that represents a user using features related to her rated items. While previous work has lent support for addressing collaborative filtering, most of them have focused on observed ratings and modeled observed data only. As a result, they can easily fail to learn users' preferences accurately from the positive-only implicit data. However, all these models are based on either user-user or item-item interaction whereas our method is based on user-item interaction. Hence, we leave out comparison with such methods as there might be differences caused due to user personalization.

Implicit Feedback Implicit Feedback originated from the area of information retrieval and the related techniques have been successfully applied in the domain of recommender systems (Kel03)(Oar98). The implicit feedbacks are usually inferred from user behaviors, such as browsing items, marking items as favourite, etc. Intuitively, the implicit feedback approach is based on the assumption that the implicit feedbacks could be used to regularize or supplement the explicit training data.

3 Dataset

For this work we use the dataset published by CLEF NewsREEL 2017. CLEF NewsREEL provides an interaction platform to compare different news recommender systems performance in an online as well as offline setting (Hop16). As a part of their evaluation for offline setting, CLEF shared a dataset which cap-

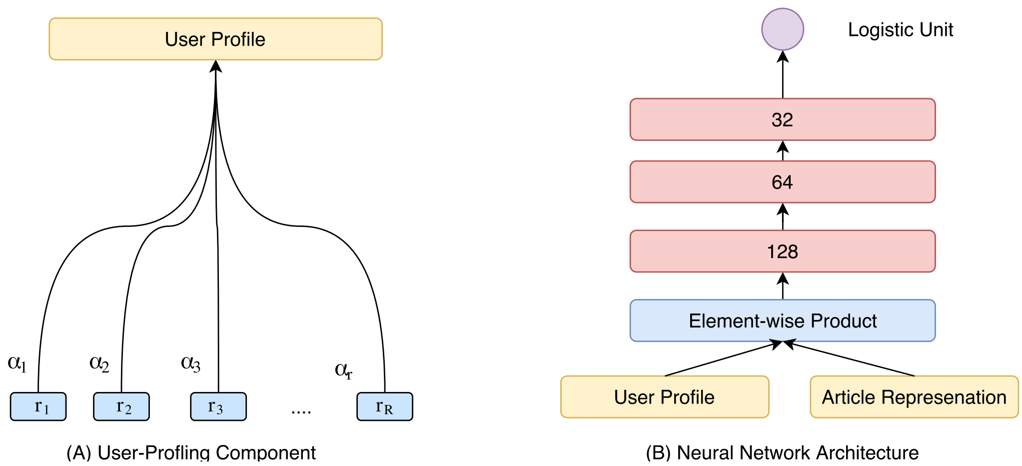

Figure 1: Model Architecture

tures interactions between users and news stories. It includes interactions of eight different publishing sites in the month of February, 2016. The recorded stream of events include 2 million notifications, 58 thousand item updates, and 168 million recommendation requests. The dataset also provides other information like the title and text of each news article, time of publication etc. Each user can be identified by a unique id. For our task, we needed to find out the sequence in which the articles were read by the users along with its content. Since, we rely on implicit feedback we only need to know whether an article was read by a user or not.

4 Model Architecture

In this section we briefly provide the description of our model. We first discuss user profiling, followed by the neural network architecture. We then provide the training criteria for our model.

4.1 User Profiling

The overview of this can be seen from Figure 1(A). We first define a set of notations useful in understanding the creation of user profile. We define the number of articles in the user reading history to be R. The doc2vec embeddings of each article in the history is represented by r_h where $1 \leq h \leq R$. Each vector is of size 300. The user profile for a user is denoted by U. We now discuss three kinds of operations using which we create the user profiles.

1. **Centroid**

   In this method, we find the centroid of the embeddings of the articles present in the reading history of the user. The centroid then represents the user profile.

$$U = \frac{1}{R}\sum_{h=1}^{R} r_h \tag{1}$$

2. **Discounting**

   In this we first discount each of the vectors present in the user reading history by a power of 2 such that an article read at time $t-1$ carries half the weight compared to an article read at time t. We then take an average of all the vectors.

$$U = \frac{1}{R}\sum_{h=1}^{R} \frac{r_h}{2^{R-h}} \tag{2}$$

3. **Exponential Discounting**

   In this we discount each of the vectors present in the user reading history by a power of e such that an article read at time $t-1$ carries $1/e$ the weight compared to an article read at time t. We then take an average of all the vectors.

$$U = \frac{1}{R}\sum_{h=1}^{R} \frac{r_h}{e^{R-h}} \tag{3}$$

Using such a method, allows us to understand the preferences of the user based on the content of the articles read by the user. It also helps us to understand the temporal changes in users interests.

4.2 Neural Network Architecture

After the user profile is obtained, we then perform an element-wise product between the profile and the embedding of the candidate article as can be seen from Figure 1(B). These candidate articles are basically the positive and the negative samples used for training the model. We then feed the element-wise product as inputs to a hidden layers of size 128. This is then followed by two subsequent fully connected hidden layers of sizes 64 and 32. Finally we use the logistic unit to make predictions. A careful reader might have noticed that, such an architecture gives us the capability to learn an arbitrary similarity function instead of traditional metrics such as cosine similarity etc. which has been normally used for calculating relevance. Typically, in matrix factorization, in order to make predictions, a dot product between the user and the item representation is computed i.e $u^T q$ where u is the user representation and q is the item representation. However, in our case we compute $a_{out}(h^t(\phi(u) \odot \phi(i))$ where a_{out} and h represent the activation function (logistic function) and the edge weights of the output layer and $\phi(u), \phi(i)$ represent non-linear transformation for user and item respectively. An astute reader might notice that, if we use an identity function for a_{out} and enforce h to be a uniform vector of 1, we will be able to recover the Matrix Factorization model. Hence, using such an architecture helps us to retain the advantages of collaborative filtering associated with news recommendation.

4.3 Training

Since we only utilize the implicit feedback of users available at our disposal, we pose the problem of recommendation as that of binary classification where label 1 would mean highly recommended and 0 would mean not recommended. We use the binary cross entropy loss, also known as log loss, to learn the parameters of the model.

5 Experiments

As mentioned earlier we use the data provided by CLEF NewsReel 2017. We choose users who have read in between 8-15 (inclusive) articles for training and testing our model for item recommendation. The frequency of users who have read more than 15 articles varies extensively and hence we restrict ourselves to the upper bound of 15. We set the lower bound to 8 since we need some history in order to capture the changing user interests. However, for future work we would like to investigate how changing the lower bound affects the performance of our model.

Evaluation Protocol: For each user we held-out her latest interaction as the test set and utilized the remaining data for training. We then recommend a ranked list of articles to each user. The performance of a ranked list is judged by Hit Ratio (HR) and Normalized Discounted Cumulative gain (NDCG). Without special mention we truncate the ranked list at 10 for both metrics.

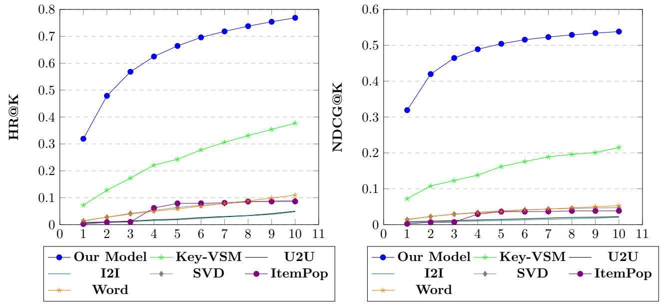

Figure 2: Performance of our model vs some state-of-the-art models

K	Avg		Discounting		Exponential	
	HR	NDCG	HR	NDCG	HR	NDCG
1	0.319	0.319	0.258	0.258	0.237	0.237
2	0.478	0.419	0.404	0.350	0.384	0.330
3	0.568	0.464	0.506	0.4013	0.483	0.379
4	0.624	0.489	0.573	0.430	0.550	0.408
5	0.664	0.504	0.619	0.447	0.595	0.426
6	0.696	0.515	0.654	0.460	0.631	0.439
7	0.718	0.522	0.678	0.468	0.658	0.448
8	0.737	0.529	0.696	0.474	0.680	0.454
9	0.754	0.533	0.712	0.478	0.697	0.460
10	0.768	0.538	0.724	0.482	0.713	0.464

Table 1: Performance with different user profiles

Baselines: We compare our method with several others. First we look at item popularity based method (ItemPop). In this we recommend the most popular items to the user. We then evaluate User-to-User (U2U-KNN) and Item-to-Item (I2I-KNN) by setting the neighbourhood size to 80. We then compare it with Singular Value Decomposition (SVD). We also implement Word Embeddings based Recommendations as in (Mus16) and Keyword based Vector Space Model (Key-VSM) as mentioned in (Lop11).

Parameter Settings: We implemented our proposed model using Keras (Cho15). We then construct our training set as follows:

1. We first define the reading history. We denote the reading history by h.
2. Leaving the latest article read by each user, the remaining articles are used as positive samples.
3. Corresponding to each positive sample, we randomly sample 4 negative instances (articles which the user did not read).

We then randomly divide the training set into training and validation set in a 4:1 ratio. This helps us to ensure that the two sets do not overlap. We tuned the

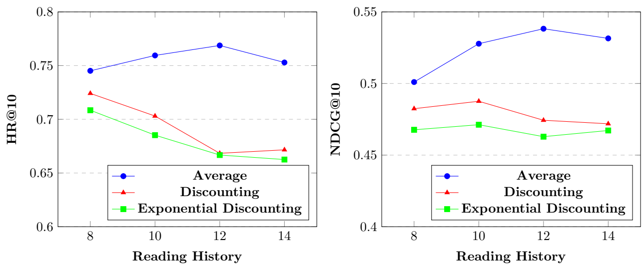

Figure 3: Performance of our model w.r.t Reading history of user

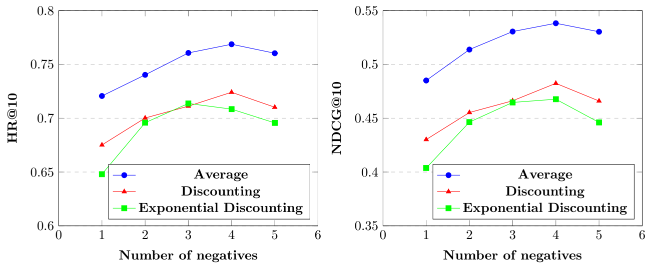

Figure 4: Performance of our model w.r.t number of negative samples

hyper-parameters of our model using the validation set. We use a batch size of 256.

6 Results

From Figure 2 we can see the results of our model as compared with the baselines. Our model outperforms the baselines by a significant margin in terms of both HR and NDCG across all positions. This clearly shows the effectiveness of our model in understanding the user preferences and making predictions accordingly. Further it can be clearly noticed that U2U, I2I and SVD do not perform well. One reason for this could be the sparsity of the data. In presence of sparse data these methods fail to capture relevant information. The low performance of Word Embedding based Recommendations suggests that a representation of words alone is not effective in profiling the user. The model also outperforms Key-VSM (Lop11) which suggests the effectiveness of the user profile component used in our model.

In Table 1, we compare the results obtained by using different sorts of profiling method. The trend in the performance can be seen as follows : Avg >Discounting >Exponential. This suggests that all the articles read by the user in a particular window have some importance in predicting the article that the user would be reading next.

Further we experiment on the size of reading history used as inputs to our model, the results for which are depicted in Figure 3. We see that choosing a size of 12 performs the best when using the averaging method for profiling. While for the other two, a size of 8 performs the best. We then also experiment with the number of negative samples for training the model parameters. From Figure 4, we can see that increasing the number of negative samples improves the performance of the model but only up to a certain point, after which the performance of the model deteriorates.

We also evaluate the model on item cold-start and find out that our model achieves an HR@10 score of around 0.32. While the typical collaborative filtering models would fail to do, using content vectors for articles provides our model the flexibility to account for these cases as well.

7 Conclusion and Future Work

In this work, we come up with a neural model for content collaborative filtering for news recommendations which incorporates both the user-item interaction pattern as well as the content of the news articles read by the user in the past. In future, we would like to explore more on deep recurrent models for user profiling.

Acknowledgement
We thank Kartik Gupta of Data Science and Analytics Centre at International Institute of Information Technology Hyderabad for helping us in making a presentation of this work.

References

[Bas04] Basilico, Justin, and Thomas Hofmann. "Unifying collaborative and content-based filtering." Proceedings of the twenty-first international conference on Machine learning. ACM, 2004. APA

[Bel07] Bell, Robert M., and Yehuda Koren. "Improved neighborhood-based collaborative filtering." KDD cup and workshop at the 13th ACM SIGKDD international conference on knowledge discovery and data mining. sn, 2007. APA

[Che12] Chen, Minmin, et al. "Marginalized denoising autoencoders for domain adaptation." arXiv preprint arXiv:1206.4683 (2012).

[Cho15] Chollet, Franois. "Keras." (2015): 128.

[Hop16] Hopfgartner, Frank, et al. "Benchmarking news recommendations: The clef newsreel use case." ACM SIGIR Forum. Vol. 49. No. 2. ACM, 2016.

[Kel03] Kelly, Diane, and Jaime Teevan. "Implicit feedback for inferring user preference: a bibliography." Acm Sigir Forum. Vol. 37. No. 2. ACM, 2003.

[Kha17] Khattar, Dhruv, Vaibhav Kumar, and Vasudeva Varma. "Leveraging Moderate User Data for News Recommendation." Data Mining Workshops (ICDMW), 2017 IEEE International Conference on. IEEE, 2017.

[Kor08] Koren, Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008.

[Kor09] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 42.8 (2009).

[Kum17] Kumar, Vaibhav, et al. "Word Semantics based 3-D Convolutional Neural Networks for News Recommendation." 2017 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE, 2017.

[Kum17] Kumar, Vaibhav, et al. "Deep Neural Architecture for News Recommendation." Working Notes of the 8th International Conference of the CLEF Initiative, Dublin, Ireland. CEUR Workshop Proceedings. 2017.

[Kum17] Kumar, Vaibhav, et al. "User Profiling Based Deep Neural Network for Temporal News Recommendation." Data Mining Workshops (ICDMW), 2017 IEEE International Conference on. IEEE, 2017.

[Le14] Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." International Conference on Machine Learning. 2014.

[Lop11] 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.

[Mus16] Musto, Cataldo, et al. "Learning word embeddings from wikipedia for content-based recommender systems." European Conference on Information Retrieval. Springer, Cham, 2016.

[Nin11] Ning, Xia, and George Karypis. "Slim: Sparse linear methods for top-n recommender systems." Data Mining (ICDM), 2011 IEEE 11th International Conference on. IEEE, 2011.

[Oar98] Oard, Douglas W., and Jinmook Kim. "Implicit feedback for recommender systems." Proceedings of the AAAI workshop on recommender systems. Vol. 83. WoUongong, 1998.

[Ren05] Rennie, Jasson DM, and Nathan Srebro. "Fast maximum margin matrix factorization for collaborative prediction." Proceedings of the 22nd international conference on Machine learning. ACM, 2005.

[Sai14] Saia, Roberto, Ludovico Boratto, and Salvatore Carta. "Semantic Coherence-based User Profile Modeling in the Recommender Systems Context." KDIR. 2014.

[Sal07] Salakhutdinov, Ruslan, Andriy Mnih, and Geoffrey Hinton. "Restricted Boltzmann machines for collaborative filtering." Proceedings of the 24th international conference on Machine learning. ACM, 2007.

[Sar01] Sarwar, Badrul, et al. "Item-based collaborative filtering recommendation algorithms." Proceedings of the 10th international conference on World Wide Web. ACM, 2001.

[Sed15] Sedhain, Suvash, et al. "Autorec: Autoencoders meet collaborative filtering." Proceedings of the 24th International Conference on World Wide Web. ACM, 2015.

[Str15] Strub, Florian, and Jeremie Mary. "Collaborative filtering with stacked denoising autoencoders and sparse inputs." NIPS workshop on machine learning for eCommerce. 2015.