

# Session-based News Recommendations

Gabriele Sottocornola  
Free University of Bolzano  
Bolzano, Italy  
gsottocornola@unibz.it

Panagiotis Symeonidis  
Free University of Bolzano  
Bolzano, Italy  
psymeonidis@unibz.it

Markus Zanker  
Free University of Bolzano  
Bolzano, Italy  
markus.zanker@unibz.it

## ABSTRACT

In the context of news recommendations, many time-aware approaches were proposed. These approaches have tried to capture the recency of news with respect to their short life span, by using either decaying weights on past articles or even forgetting them. However, most of these approaches have missed to consider sessions, which encapsulate inside them the articles that a user has interacted with in a short time period. In this paper, we provide news recommendations based on user sessions to reveal their short-term intentions. We also combine content-based with collaborative filtering to deal with the severe data sparsity problem that exists in our real-life data set. We have experimentally seen that the users' interests evolve over time and that our strategies can adapt fast to these changes.

## CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; *Data streams*;

## KEYWORDS

Session-based recommendation; Time-aware recommendation; News recommendation

### ACM Reference Format:

Gabriele Sottocornola, Panagiotis Symeonidis, and Markus Zanker. 2018. Session-based News Recommendations. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3184558.3191582>

## 1 INTRODUCTION

Content-based (CB) news recommendations build a user's profile based on the news articles that she has read in the past, and recommend to her similar articles. The main advantage of CB is that it alleviates the “cold start” problem (i.e. recommendations for new incoming items/users without ratings). However, CB cannot provide diversified recommendations, because it favors only articles similar to the user's historical profile. To overcome the aforementioned problem, Collaborative Filtering (CF) recommends articles that other similar users to the target user have read in the past. Another CF variation is the item-based CF algorithm [8, 12], where given a target user and her positively rated news articles, the algorithm relies on the items' similarities for the formation of a neighborhood of nearest items. For news recommendation, there are hybrid approaches that combine CB with CF algorithms [9] to deal with the

aforementioned problems. However, even if these works take into account the time dimension (e.g., decay the weight of articles that are old, etc.), they miss to take under consideration the sessions, which can reveal the short term preferences of a user.

In this paper, we provide news recommendations that use these short user sessions. The advantage of the session-based recommendation is that we concentrate mostly on the short-term intentions of a user. First, we formulate the problem of news recommendation and we present the main dynamics that govern it, such as recency of news, the life span of a news article or topic category, and the usage of a sliding time window to forget news articles that are old. Our proposed method builds a CB user profile, by identifying the main categories of news articles that a user is interested in (e.g., politics, sports, etc.). We also use a sliding time window to reduce the impact of news articles that have been clicked earlier in the past. Moreover, to reveal the short-term user's profile, we analyze the latest news articles that a user has read (i.e. in his last sessions).

The rest of this paper is organized as follows. Section 2 summarizes the related work. Section 3 formulates the problem and describes our proposed algorithm. Experimental results are given in Section 4. Finally, Section 5 concludes the paper.

## 2 RELATED WORK

In this Section, we will present the main work in session-based recommenders and news recommender systems.

Session-based recommendation usually refers to the case scenario, where we have only anonymous sessions and we are not able to build a user's profile [3, 13]. Recently, session-based recommendations have been modeled with Recurrent Neural Networks (RNNs). Hidasi et al. [4] presented a recommender system based on Gated Recurrent Unit (GRU), which learns when and how much to update the hidden state of the GRU model. However, a more recent study [6] have shown that a simple  $k$ -nearest neighbor (kNN) scheme adapted for session-based recommendations often outperforms the GRU model. The authors claim that best results are achieved when a session-based kNN model is combined with GRU model in a weighted hybrid approach. Nevertheless, several adjustments were proposed during last years in order to improve the performances of the initial GRU model [5, 11, 14, 15].

For the news recommendation task based on short user sessions, Epure et al. [2] considered three levels of reading interests based on time dimension: short-, medium-, and long-term. They also explored different combinations of these three levels to increase either the accuracy of stories recommendations or the diversity of the recommendation list. They have experimentally shown that recommendations based on combined short-term and long-term interests result in increased accuracy while recommendations based on combined short-term and medium-term interests can increase the diversity of the recommendations.

This paper is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

*WWW '18 Companion, April 23–27, 2018, Lyon, France*

© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.

ACM ISBN 978-1-4503-5640-4/18/04.

<https://doi.org/10.1145/3184558.3191582>

Liu et al. [9] combined the content-based method with the collaborative filtering method to generate personalized news recommendations. The hybrid method develops a Bayesian framework for predicting users' current news interests based on profiles learned (i) from the target user's activity and (ii) the news trends demonstrated in the activity of all users. Li et al. [7] proposed a novel recommendation approach, in which the long-term and short-term reading preferences of users are combined to recommend news stories. In particular, their method is based on a hierarchy of news stories (i.e. news categories) that a user might prefer. Ludmann's recommender system [10], denoted as Ody4, won the CLEF News-REEL 2017 contest, which was about recommending effectively and efficiently news articles. Ody4 is a stream-based recommender system, which relies on the open source data stream management system Odysseus. Ody4 continuously calculated the most popular and recent articles based on a stream of interaction events.

### 3 PROPOSED METHOD

We are interested in providing recommendations for news web sites. These media provide news articles to interested content consumers (anonymous or registered users). For anonymous users, we do not have a user profile, instead, we store anonymous sessions together with the articles accessed during them. For registered users or those who have accepted cookies requests, we can keep track of all the interactions that they did in the past (i.e., we can build a user profile).

Our recommender consists of two modules. The first is the profile updater, and the second is the recommender that runs on top of the profile updater to deliver the top- $N$  recommended items to each user. The profile updater module reads instances from the stream of user sessions combining them with earlier recorded information on the participating entities (users, items, sessions). Thus, it assigns validity intervals to elements of the sessions stream  $S$ . Then, a sliding time window of size  $w$  states that the processing at a point in time  $t$  should consider all events not older than  $t - w$ . Therefore, the profile updater sets a validity interval  $[t - w, t)$  in which it computes the similarity among items based on either articles-sessions interactions (CF) or articles' topic categories (CB). Then, this information is provided to the recommender module, to suggest top- $N$  items to each user.

For the news recommendation scenario, whenever a user reads one or more articles in a short time period (i.e., 30 minutes), we store these interactions as a user's session. Formally, a session  $S_n^u$  for a user  $u$  is defined as a set of items, that the user has interacted with, within a short time period,  $S_n^u = (i_1, i_2, \dots, i_m)$ . Since user-item interactions arrive at any time, a session is a stream. New users and new items may also arrive at any time. To further clear out our idea, we will use the running example shown in Figure 1. In this running example, we have 4 users and we want to recommend the next story to user 3, who has an unfinished session (i.e. session  $S_8$ ). For computing the similarities between the session of the target user  $U_3$  with the sessions of the other 3 users, please notice that sessions  $S_1$ ,  $S_2$ , and  $S_3$  cannot be considered, because they are outside of the sliding time window that we have set  $[t - w, t)$ . This sliding time window captures the notion of recency of news stories. That is, as a news article has a short life span, which obsolesces fast, then, we

should try to recommend more recent articles. The main idea of our paper is that items which are selected in the same sessions can be considered as more similar than those items that are selected from the same user in different sessions. For example, taking into account actions of user 3 (i.e.,  $U_3$ ) we can infer that similarity between item  $i_6$  with  $i_3$  is bigger than the similarity between item  $i_6$  with  $i_1$  since they were selected within the same session  $S_5$ . This methodology reveals also the short-term preference of the user and his intentions inside a session. We consider two aspects of similarity between item-session interactions and item-category correlations. In the following, we describe these two different types of similarity among items, and we explain how we can combine them into a single similarity function.

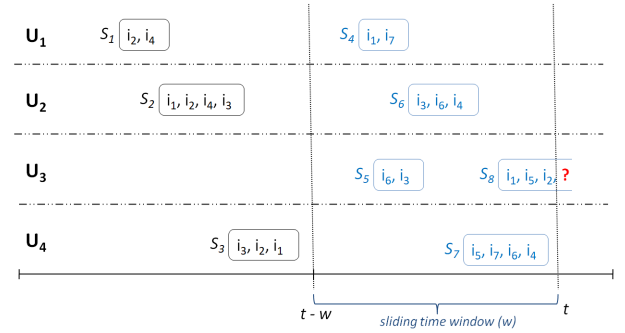


Figure 1: A visual representation of our running example.

#### 3.1 Session-based CF Similarity

In this Section, instead of using the classical user-item matrix we use the session-item matrix. The advantage of inferring item similarities from session interactions, instead of user interactions, is that we focus in the short term intentions of the user. Since we perform collaborative filtering on a stream of sessions and items, we slide a window of length  $w$  over the stream of interactions and consider at each time point  $t$  only the set of interactions inside the time window  $t_p = [t - w, t)$ . We encode the session as a binary vector: For each session which is active in time period  $t_p$ , we set  $r_{s,i} = 1$  if item  $i$  is accessed inside section  $s$ , otherwise  $r_{s,i} = 0$ . Thus, the similarity between item  $i$  and  $j$  is computed as follows:

$$\text{simCF}(t_p, i, j) = \frac{\sum_{s \in S_{t_p, i} \cap S_{t_p, j}} (r_{s,i} \cdot r_{s,j})}{\sqrt{\sum_{s \in S_{t_p, i}} (r_{s,i})^2} \sqrt{\sum_{s \in S_{t_p, j}} (r_{s,j})^2}} \quad (1)$$

where  $S_{t_p, i}$  and  $S_{t_p, j}$  are the set of sessions in which item  $i$  and  $j$  appeared, respectively, within time period  $t_p$ . In our running example,  $i_7$  would be recommended because it is more similar than  $i_4$  to the items of the last session ( $S_8$ ) of  $U_3$ .

#### 3.2 Content-based Similarity

In this Section, we describe a content-based method to recommend news articles according to their topic categories. Based on the articles' text, we train a topic model and we create for each article its topic category profile. That is, an article is represented by a probability distribution vector over the set of topic categories. Based on

the user's news reading history within a specific time period  $t_p$  the profile updater computes the similarity matrix among items, based on the cosine similarity of each pair of article-category profiles. Let  $f_{c,i}$  be the probability that an item  $i$  belongs to a topic category  $c$ . Then, the similarity between items  $i$  and  $j$  is computed as:

$$\text{simCB}(t_p, i, j) = \frac{\sum_{c \in C_{t_p,i} \cap C_{t_p,j}} (f_{c,i} \cdot f_{c,j})}{\sqrt{\sum_{c \in C_{t_p,i}} (f_{c,i})^2} \sqrt{\sum_{c \in C_{t_p,j}} (f_{c,j})^2}} \quad (2)$$

where  $C_{t_p,i}$  and  $C_{t_p,j}$  are the set of categories that articles  $i$  and  $j$  may belong to within time period  $t_p$ .

### 3.3 Combining CB with Session-based CF

$\text{simCF}$  matrix captures the interactions of sessions with articles. However, when sessions do not have many items, it is difficult to provide accurate recommendations. On the other hand,  $\text{simCB}$  matrix captures the topic categories profile of the articles. Thus, it is able to detect those items that belong to the same categories with those of the target item, alleviating the aforementioned problem. As our objective is to provide more accurate news recommendations, we combine the two similarity measures into a single one, which is given in Eq. 3:

$$\text{simCBCF}(t_p, i, j) = \alpha * \text{simCB}(t_p, i, j) + (1 - \alpha) * \text{simCF}(t_p, i, j) \quad (3)$$

Please notice that in several cases the distribution of the similarity values between  $\text{simCF}$  and  $\text{simCB}$  may differ significantly. Therefore, in that case, we have to normalize in the interval  $[0,1]$  the values of the two similarity matrices before combining them.

### 3.4 Providing Recommendations

Our recommender module provides recommendations based on the similarity matrices presented in previous sections. For each target user  $u$ , the recommender checks the set of her recently viewed items  $I_{t_p,u}$  (i.e., the ones she has interacted with in the current time period  $t_p$ ) and computes  $K_i$ , which is the set of the  $k$  nearest items to each item  $i$  that belongs in  $I_{t_p,u}$ . Next, for each target user  $u$  in  $t_p$  and for each item  $j$  we compute a ranking score  $\text{score}(t_p, u, j)$  as follows:

$$\text{score}(t_p, u, j) = \sum_{i \in I_{t_p,u}} \text{simCBCF}(i, j) * \mathbf{1}_{(j, K_i)} \quad (4)$$

where  $\mathbf{1}_{(j, K_i)}$  is an indicator function that is equal to 1 if the item  $j$  is present within the  $k$ -nearest neighbors of item  $i$ , and 0 otherwise. Then, for each user we sort the items in decreasing score and recommend to her the top- $N$  ones.

## 4 EXPERIMENTAL EVALUATION

### 4.1 Data Description and Log Analysis

The data set is provided by an Italian news publisher that accommodated 14367 interactions/events on 2081 articles of 14047 unique users in one year (i.e. from 1st April 2016 to 30th March 2017). An interaction is logged when an article is accessed on the news publisher's website. The interactions of each session are logged with the following information: the user session's identifier, the interaction's time stamp and duration, the article's textual content. Based on the articles' text, we train a Latent Dirichlet Allocation

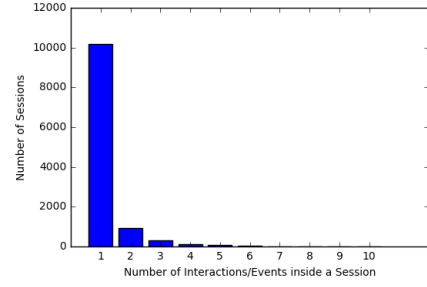


Figure 2: A visual representation that shows the number of interaction per Session.

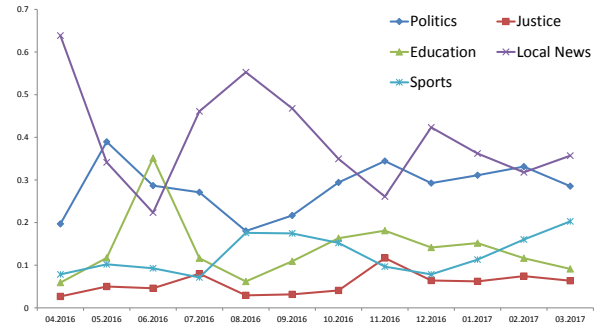
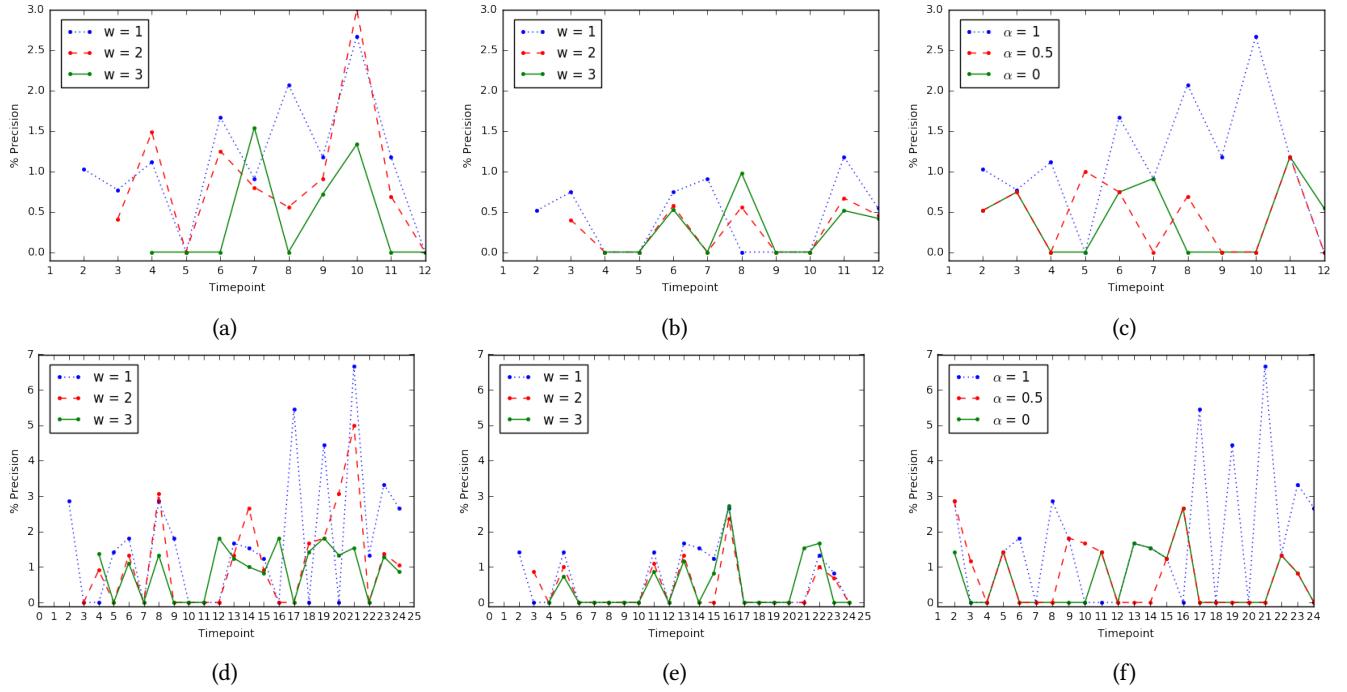


Figure 3: A visual representation that shows how the public interest evolves among the 5 topics over the last 12 months (1/4/2016 - 30/3/2017).

(LDA) [1] model and found five topic categories in which an article may belong to. The topic categories that were found are “education”, “politics”, “sports”, “local news” and “justice”. Most of the user sessions have a very small number of interactions (i.e., 84% of sessions have only one interaction) as shown in Figure 2, which should have a negative impact in the effectiveness of all recommendations algorithms. This explains the poor results for all methods in terms of  $\text{precision@N}$  that will be shown later. In addition, we have noticed big fluctuations of the crowd's reading trends during the year. Figure 3 shows how the public interest evolves over the year, for the 5 topic categories (politics, education, sports, justice and local news). Please notice that some news categories (e.g., local news, education and politics) have greater variation than others (e.g. Sports and Justice). This happens because of some big events, e.g., the local elections of Alto Adige region in Italy was performed on May 2016, and politics attracted the people's interest, as shown in Figure 3. Based on the above considerations, by using a smaller sliding time window size, we may identify faster when a topic category will attract the crowd's interest (concept drift), as will be later experimentally shown.

### 4.2 Recommendation Results

In this Section, we evaluate the  $\text{precision@5}$  (i.e., the ratio of correctly predicted articles to the number of recommended ones) we get when we recommend top-5 articles to each user that was active



**Figure 4: Precision vs. different window sizes ( $w = 1, 2, 3$ ) for (a) item-based CB with 12 timepoints, (b) item-based CF with 12 timepoints, (d) item-based CB with 24 timepoints and (e) item-based CF with 24 timepoints. Moreover, Precision for CBCF vs. different weighting schemes ( $\alpha = 1, 0.5, 0$ ) for (c) 12 timepoints, (f) for 24 timepoints.**

in  $t_p$  and has at least one session in time period  $[t, t + 1)$ , denoted also as  $t_{p+1}$ . In other words, we evaluate item recommendations against the interactions appearing in  $t_{p+1}$ , which we consider as future to evaluate our experiments. An important parameter for news recommendation is the window size of the interaction events. If this sliding time window is too large the system is not sensitive to changes (concept drifts), if it is too small there is not enough data to build a model for items or sessions.

Next, we study the accuracy performance of CB and CF methods with different time period splits:  $n_t = 12, 24$  and with various time window sizes:  $w = 1, 2, 3$ . We set the nearest neighbors  $k$ -nn = 20 for CF and  $k$ -nn = 30 for CB, according to the best results achieved during the preliminary evaluation. In Figures 4a, 4b, 4d, and 4e, we can see that the smallest window size  $w = 1$  achieves the best result for almost all timepoints. As far as the time period splits is concerned, the best results are attained with 24 time splits, which means that forgetting fast older news is better. Please notice that when we run experiments with  $n_t = 52$ , precision decreased. As expected, CB is on average 40% better than CF, due to the extreme data sparsity (i.e. cold start problem) that exists in our real-life data set, as it is also shown in Figure 2. Next, we evaluate the combination of CB with CF (i.e. CBCF) with different weighting schemes:  $\alpha = 1, 0.5, 0$ . As shown in Figures 4c and 4f, for the 12 and 24 time splits, respectively, CBCF (i.e. the red dotted line) give us better results than those of simple CF and CB at least in some time points. This is an encouraging result, which could be further optimized by learning a model for parameter  $\alpha$  using either a regression model or

a neural network. Finally, in Table 1 we report the average precision results over the whole year for CBCF against the popularity baseline (POP), which recommends the most popular articles in the time window  $t_p$ . To include also a static variation of both algorithms, we run experiments with  $n_t = 2$ , in addition to the  $n_t = 12$  and  $n_t = 24$ . As shown in Table 1, CBCF outperforms POP baseline in all cases. As expected, both static models perform poorly, because they recommend articles published during a time period of six months, ignoring the fact that articles have a short lifespan (e.g., some weeks).

time splits	CBCF	POP
$n_t = 2$	0.2	0.0
$n_t = 12$	0.44	0.21
$n_t = 24$	0.71	0.39

**Table 1: Average precision@5 results over the whole year (%) for  $n_t = 2$ ,  $n_t = 12$ , and  $n_t = 24$ , with  $w = 1$ .**

## 5 CONCLUSION

In this paper, we have used short user sessions and topic categories to infer similarities among news stories. We have performed experiments on a real-life data set and proved that the combination of CB with CF could be effective when data sparsity is huge. In future, we want to predict the next item inside a session, which is more suitable for evaluating online recommender systems.

## REFERENCES

- [1] David M Blei. 2012. Probabilistic topic models. *Commun. ACM* 55, 4 (2012), 77–84.
- [2] Elena Viorica Epure, Benjamin Kille, Jon Espen Ingvaldsen, Rebecca Deneckere, Camille Salinesi, and Sahin Albayrak. 2017. Recommending Personalized News in Short User Sessions. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 121–129. DOI: <https://doi.org/10.1145/3109859.3109894>
- [3] Asnat Greenstein-Messica, Lior Rokach, and Michael Friedman. 2017. Session-Based Recommendations Using Item Embedding. In *Proceedings of the 22Nd International Conference on Intelligent User Interfaces (IUI '17)*. ACM, New York, NY, USA, 629–633. DOI: <https://doi.org/10.1145/3025171.3025197>
- [4] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR '16)*. arXiv:1511.06939 <http://arxiv.org/abs/1511.06939>
- [5] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 241–248. DOI: <https://doi.org/10.1145/2959100.2959167>
- [6] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 306–310. DOI: <https://doi.org/10.1145/3109859.3109872>
- [7] Lei Li, Li Zheng, Fan Yang, and Tao Li. 2014. Modeling and broadening temporal user interest in personalized news recommendation. *Expert Systems with Applications* 41, 7 (2014), 3168 – 3177.
- [8] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.Com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing* 7, 1 (Jan. 2003), 76–80. DOI: <https://doi.org/10.1109/MIC.2003.1167344>
- [9] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces (IUI'10)*. ACM, 31–40.
- [10] Cornelius A Ludmann. 2017. Recommending News Articles in the CLEF News Recommendation Evaluation Lab with the Data Stream Management System Odysseus. In *8th International Conference of the CLEF Initiative, Dublin, Ireland. CEUR Workshop Proceedings*.
- [11] Massimiliano Ruocco, Ole Steinar Lillestøl Skrede, and Helge Langseth. 2017. Inter-Session Modeling for Session-Based Recommendation. In *Proceedings of the 2Nd Workshop on Deep Learning for Recommender Systems (DLRS 2017)*. ACM, New York, NY, USA, 24–31. DOI: <https://doi.org/10.1145/3125486.3125491>
- [12] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM, New York, NY, USA, 285–295. DOI: <https://doi.org/10.1145/371920.372071>
- [13] Guy Shani, David Heckerman, and Ronen I. Brafman. 2005. An MDP-Based Recommender System. *J. Mach. Learn. Res.* 6 (Dec. 2005), 1265–1295. <http://dl.acm.org/citation.cfm?id=1046920.1088715>
- [14] Elena Smirnova and Flavian Vasile. 2017. Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks. In *Proceedings of the 2Nd Workshop on Deep Learning for Recommender Systems (DLRS 2017)*. ACM, New York, NY, USA, 2–9. DOI: <https://doi.org/10.1145/3125486.3125488>
- [15] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016)*. ACM, New York, NY, USA, 17–22. DOI: <https://doi.org/10.1145/2988450.2988452>