Sequence-aware news recommendations by combining intra- with inter-session user information

Panagiotis Symeonidis · Dmitry Chaltsev · Chemseddine Berbage · Markus Zanker

Received: 30 May 2020 / Accepted: 20 August 2022 / Published online: 28 September 2022
© The Author(s) 2022

Abstract
There exist many research works that strive to answer the question “what news article is a user going to click next given his profile”. These works take into account the time dimension to reveal users’ preferences over time. However, few works exploit adequately the information that is hidden inside user sessions. User sessions include a list of user interactions with items within a short period of time such as 30 min, and can reveal her very last intentions. In this paper, we combine intra- with inter-session item transition probabilities to reveal the short- and long-term intentions of individuals. Thus, we are able to better capture the similarities among items that are co-selected inside a user session but also within any two consecutive sessions. We have evaluated experimentally our method and compare it against state-of-the-art algorithms on three real-life datasets. We demonstrate the superiority of our method over its competitors.

Keywords  Session-aware recommendation · Sequence-aware recommendation · Markov chain model

1 Introduction

Collaborative filtering (CF) helps the exploration of news articles that other users similar to the target user have also read in the past. That is, given a target user and her positively rated news articles, a CF algorithm will identify the most similar users of the target user, i.e.
user-based CF (UBCF). Another CF algorithmic variation is the item-based CF (IBCF), 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. Moreover, matrix factorization (MF) is a well-known CF method that learns the user and item vector representations and calculate their matching score based on their dot product. A significant improvement on the prediction accuracy of classic MF algorithm can be obtained through the incorporation of implicit feedback into the MF model, denoted as SVD++.

All the aforementioned methods are considered as non-sequential, since they learn a user’s preference on each individual item, and then they rank these items based on their score to provide recommendations. This functionality makes these methods good at capturing the general tastes of users by using the whole historical information (i.e. by aggregating a user’s complete log history). However, they are not able to capture the transition relationship between two or more adjacent items in user log sequences (i.e. user sessions) so as to learn the sequential dynamics among items. For example, many sophisticated sequence-based approaches were proposed that implement some form of sequence modelling based on Markov Chain Models (MCM) (Chen et al., 2012; McFee & Lanckriet, 2011; Paparrizos et al., 2011), which can capture the item transition probabilities and the very last user intentions inside a user session. In the contrary, we use in our contribution a transition probability matrix and take into consideration both the intra-session and the inter-session probabilities. In this paper, we use Markov chain modelling to provide session-aware news recommendations, by distinguishing the long-term preferences of the users from their very last intentions over single items. The fact that we take under consideration the time dimension is crucial for increasing the effectiveness of our algorithm. First, to reveal the very last user’s intention, we analyse the item interactions inside her latest sessions separately (i.e., intra-session item similarity). Next, to deal with the problem that in the beginning of each session, there is not enough information to learn much about user’s concrete intention, we also use the information that comes from other recent sessions (i.e., inter-session item similarity). Moreover, we track the evolution of these preferences by using a sliding time window to disregard articles that are outdated or old. Thus, our model is continuously updated with the latest user clicks, which makes it sensitive to adapt to the changes of the user preferences.

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 provides the problem formulation, whereas Sect. 4 presents our proposed methodology. Experimental results are given in Sect. 5. Section 6 discusses some limitations of our work and describes ways to overcome them. Finally, Sect. 7 concludes the paper.

2 Related work

A Markov chain is a stochastic process of possible events that satisfies the Markov property, which states that probability of each event depends only on the present state and not on the previous states. A variation of MCM, denoted as Markov Chain Model of Order $m$, states that the future state depends on the past $m$ states. Hidden Markov Model (HMM) is also a MCM with hidden states. Moreover, Markov Decision Processes (MDPs) extend MCMs, where at each timepoint $t$, when the process is in state $x_t$, the decision maker may choose any action $a \in A_x$. MDP responds at the next time step by randomly moving into a new state $x_{t+1}$, and it gives the decision maker a corresponding reward $R(x_t, a, x_{t+1})$. 

Springer
There are several works (Chen et al., 2012; Esiyok et al., 2014; Garcin et al., 2013; He et al., 2009; McFee & Lanckriet, 2011; Paparrizos et al., 2011) in recommender systems that use sequential modelling based on MCMs. Esiyok et al. (2014) studied the users’ behaviour in the context of news categories by building a Markov Chain Model based on the Plista data set and by describing patterns in the evolution of news categories while users browse news articles online. Moreover, approaches for recommender systems that use MDPs are published by Moling et al. (2012) and Shani et al. (2005). A hybrid model that combines MCM with MF is proposed by Rendle et al. (2010). Factorized Personalized Markov Chains (FPMC) is used for the next-basket recommendation problem. The task at hand is to predict user’s next basket content, given history of past shopping baskets. This approach combines MCM with MF using a three-dimensional tensor (user, current item, next item). Each entry in the tensor corresponds to an observed transition between two items performed by a specific user. The proposed method then uses pairwise factorisation to predict the unobserved entries in the sparse tensor (Ludewig & Jannach, 2018). However, this approach is computationally intensive, and does not scale well in real world online situations.

Session-aware recommendation usually refers to the scenario, where we have a set of sessions and we seek to build a user’s profile. Recently, session-based recommendations have been modelled with Recurrent Neural Networks (RNNs). Hidasi et al. (2015) 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 (Jannach et al., 2017) have shown that a simple \( k \)-nearest neighbor (session-kNN) scheme adapted for session-based recommendations often outperforms the GRU4REC model. The authors claim that best results are achieved when a session-based kNN model is combined with GRU4REC model in a weighted hybrid approach. Nevertheless, several new adjustments were proposed during last years that improve the performances of the initial RNN model (Hidasi & Karatzoglou, 2018; Hidasi et al., 2016; Quadrana et al., 2017; Smirnova & Vasile, 2017; Tan et al., 2016).

For the news recommendation task, related work (Li et al., 2014) has shown that a way to increase accuracy is to consider the context of the user (i.e., time, location, mood, etc). For example, Das et al. (2007) generated recommendations based on collaborative filtering that takes under consideration the co-visitation count of articles, which is the number of times a news story was co-visited with another news stories in the user’s click-history. In other words, co-visitation is defined as an event in which two stories are clicked by the same user within a certain time interval (typically set to a few hours). This captures the following simple intuition: Users who viewed this item also viewed the following items. Liu et al. (2010) combined the content-based method with the collaborative filtering method previously developed for Google News (Das et al., 2007) to generate personalised news recommendations. The hybrid method develops a Bayesian framework for predicting users’ current news interests based on profiles learned from: (i) the target user’s activity and (ii) the news trends demonstrated in the activity of all users. Please notice that in our paper we will not compare to the work of Liu et al. (2010), because the exploitation of content is beyond the scope of this paper.

Ludmann’s recommender system (Ludmann, 2017), denoted as Ody4, won the CLEF NewsREEL 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-read articles based on a stream of impression events (i.e., most clicked articles of a 12-hour sliding time window). That is, by analyzing impression events of users, he calculated a set of
recommendations based on the item popularity in a given time window. Ludmann (2017) has underpinned in his research that a crucial parameter is the time-window size. Since he wanted to count the current views of each article, he had to find the correct time span that leads to a set of interaction events (i.e. article clicks) that represent the current popularity of articles. That is, if the size of the sliding window is too large the system is not sensitive to popularity changes (concept drifts). If it is too small, then there is not enough data to distinguish the popularity of articles.

An et al. (2019) have proposed a news recommender based on a neural approach to catch long- and short-term user representations. The authors proposed two different encoders. The first one concerns the news and learns the representations from the titles and topic categories, while the second one concerns the users and learns the representations from the behavioural data. Furthermore, Yu et al. (2019) have proposed an improved traditional RNN using a time-aware controller, and a content-aware controller. In their proposition, the authors use an attention-based framework to combine users’ long-term and short-term preferences.

3 Problem formulation

We are interested in building a recommender system for news media that provide news articles to interested content consumers. The news publisher updates a small personalised top-$N$ list of article recommendations (which are shown inside a widget), every time an article is selected by the user, because the publisher wants to engage the user more time in the website for reasons of advertisement and for fulfilling his reading desires. The publisher employs different algorithms (i.e. Markov chain modelling, collaborative filtering, matrix factorisation, hybrid methods, etc.) to provide recommendations. The system monitors how visitors react upon the received recommendations to drive better suggestions and try to predict their next click/item inside a session. Thus, our approach adapts to the user’s choices/clicks based on his reading behaviour inside his current session by combining his long- and short-term preferences. For example, by joining all the sessions that a user $u$ has interacted with the website, we can capture his long-term reading behaviour (i.e. the user’s reading profile is the set of all sessions $S_u$ of user $u$), but by focusing only in his late sessions with our system using a sliding window, we can capture only his short-term intentions.

Let $U$ denotes the increasing set of users that visit the online web site, and $I$ is the increasing set of incoming articles/items. We keep track of the users’ actions over items in the website. In particular, whenever a user reads one or more articles in a short time period (i.e., 30 min), we store these interactions in the database as a user’s session. These interactions with items have a sequence. That is, we know for every item that belongs in a session, if it is selected first, second or last, the time that it is selected by the user, and how much time he interacted with it inside the session. For example, session $S_1(user = u_1, TimeStarted = t_1 \{i_1, 20sec\}, \{i_2, 145sec\})$ indicates that within session $S_1$ that started at timepoint $t_1$ from user $u_1$, item $i_1$ was selected first and it was read for 20 s and $i_2$ was selected second and it was read for 145 s.

Our session-aware recommender consists of two modules. The first is the user profile updater module, which reads instances from the stream of sessions combining them with earlier recorded information. In particular, our user profile updater 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 respect all events not older than $t - w$. Therefore,
the profile updater sets a (half open) validity interval \((t - w, t]\) to an event that has been arisen at time \(t\). Then, the second module is the recommender that runs on top of the profile updater to deliver the top-\(N\) recommended items to each user. Table 1 summarises some basic symbols and notations that will be used later.

To better explain our approach, we will use as our running example the following graphical representation, which is shown in Fig. 1. In our running example, we have 3 users and we want to predict the news story that user 2 will click next in his unfinished session (i.e., session \(S_7\)).

For computing the similarities between the target user 2 with the other two users, please notice that sessions \(S_1\) and \(S_2\) cannot be considered, because they are outside of the sliding time window valid interval that we have set \((t - w, t]\). This sliding time window captures the notion of recency of news stories. That is, a news story has a life span, which obsoletes fast. Thus, we should try not to recommend stories to users that are not recent. When two or more items are selected within one session, these items can be considered as more similar compared to items that were selected in different sessions from the same user. For example by taking into account actions of user 1 (i.e., \(U_1\)) we can infer that item \(i_4\) is more similar to item \(i_6\) and \(i_7\) rather than item \(i_9\), since they were selected inside the same session \(S_4\) with the item \(i_4\).

In our running example, as depicted in Fig. 1, session \(S_7\) is still open and it is running for user \(U_2\). Thus, items \(i_2\) and \(i_3\) of session \(S_7\) can be matched in order to make item recommendations to user \(U_2\). As shown, user \(U_3\) has also selected the same items (inside the valid
window time interval) with $U_2$. He has also selected $i_8$, which could be a nice recommendation for $U_2$. Please notice that also $U_1$ has selected exactly the same items with those of session $S_7$ of $U_2$. However, we cannot use session $S_1$ for recommendations because it is not inside the valid sliding time window.

In summary, items that are selected in the same sessions (intra-session item similarity) can be considered as more similar than those items that are selected from the same user in different sessions (inter-session item similarity). Intra-session item similarity can reveal the short-term preference of the user and his intentions inside a session independently of other sessions. In addition, inter-session similarity is able to find item similarities even when sessions have a very small number of item interactions (i.e. low average number of item interactions per session).

4 Our proposed method

In this Section, we want to identify individual’s short-term (inter-session) preferences and his latest specific (intra-session) intentions. The inter-session similarity takes any two subsequent sessions and creates an item transition probability matrix based on the subsequences between any two items of any two subsequent sessions. The intra-session similarity algorithm goes inside each session separately and creates an item transition probability matrix based on the subsequences between items inside each session.

Based on the Bayesian inference that considers independence among evidences, we can predict the items that will be included in a last session $S_N$ of a user $u$ based on the items that are already included in $S_N$. In particular, we can use the following formula to build the intra-session transition probabilities between any two subsequent items in each distinct session in time window $t_p$ as follows:

$$p(j \in S_N|i_{1:m} \in S_N) \propto \prod_{k \in S_N} p(j \in S_N|i_k \in S_N),$$

(1)

where $i_k$ is the set of items that user $u$ already has clicked in current session $S_N$, and $j$ is the item to be predicted as next recommended item in $S_N$. However, to deal with the problem that at the start of the session there is not much information about the user’s current interests, we can also learn from other recent sessions (inter-session) item transition similarities, and predict the user’s interest in the current session. We can extend the above formula by taking into account also the inter-session transition probabilities among items of any two subsequent sessions:

$$p(j \in S_N|i_{1:m} \in S_N \land (S_{n+1}, S_n)) \propto \prod_{i_k \in S_N} p(j \in S_N|i_k \in S_N) \prod_{i_k \in S_N} \frac{p(j \in S_{n+1} \land i_k \in S_n)}{p(i_k \in S_n)},$$

(2)

where $n$ is a time point, $n + 1$ is the next time point, and $N$ is the number of recommendations.
To capture user’s behavior when he interacts with the system, a transition probability matrix $T$ that expresses the transition from one item that belongs to an old session $S_n$ to another item that belongs to the next session $S_{n+1}$ is constructed as follows:

$$T_{i_1,i_2} = p(i_2 \in S_{n+1} | i_1 \in S_n),$$

where $T_{i_1,i_2}$ is an element of $T$, and represents the transition probability between $i_1$ and $i_2$, $i_1$ and $i_2$ are items that belong to $S_n$ and $S_{n+1}$, respectively. The probability that a user will be interested in a news article $j$ given the previous items of session $S_N$ can be defined as the mean over all transition probabilities from the previous items of this session to this article:

$$p(j|i_1:m \in S_N) = \frac{1}{m} \cdot \sum_{i_k \in S_{n+1}} p(j \in S_{n+1} | i_k \in S_N),$$

where $m$ is the number of items in the current session. Next, using the maximum likelihood estimator we can compute the transition probability between any two articles $T_{i_1,i_2}$ that belong in subsequent sessions as follows:

$$p(i_2 \in S_{n+1} | i_1 \in S_n) = \frac{|\{(S_{n+1},S_n) : i_2 \in S_{n+1} \land i_1 \in S_n\}|}{|\{(S_{n+1},S_n) : i_1 \in S_n\}|},$$

where the numerator expresses the number of times item $i_1$ was included in $S_n$ and $i_2$ in $S_{n+1}$. The denominator expresses the number of times that a session contains item $i_1$ in time period $t_p$. Based on Equation 5, in our running example of Figure 1, the transition probability from item $i_4$ to item $i_2$ is equal to $\frac{1}{2}$, where the numerator is equal to one, since there is only one instance of the two consecutive sessions where $i_4$ belongs to the first session ($S_3$) of the two sessions and $i_2$ belongs to the second session ($S_7$) of the two sessions; and the denominator is equal to two since there are two sessions with $i_4$ (sessions $S_3$ and $S_4$). The inter-session transition probability matrix is presented in Table 2, whereas the zero values refer to zero probability to transfer from one item to another item.

As far as the intra-session item transition probability is concerned, by using a first-order Markov Chain, we can describe the transition probability between two subsequent events in a session. That is, we can simply count how often users viewed item $i_b$ immediately after viewing item $i_a$.

Let a session $S_n$ be a chronologically ordered set of item click events $S_n = (i_1, i_2, ..., i_m)$ and $S$ be a set of all sessions $S = \{S_1, S_2, ..., S_N\}$. Given a user’s current session $S_N$ with $i_m$ being the last item in $S_N$. As reported in (Ludewig & Jannach, 2018), we can define the score for a recommendable item $j$ as follows:

| Table 2 | Inter-session transition probability matrix of our running example |
|---------|---------------------------------------------------------------|
| Inter-session TPM | $i_2$ | $i_3$ | $i_9$ |
| $i_1$ | 1 | 1 | 0 |
| $i_4$ | 0.5 | 0.5 | 0.5 |
| $i_6$ | 0 | 0 | 1 |
| $i_7$ | 0 | 0 | 1 |
where the function \( \text{isSame}(i_a, i_b) \) indicates where \( i_a \) and \( i_b \) refer to the same item as follows:

\[
\text{isSame}(i_a, i_b) = \begin{cases} 
1, & \text{if } i_a = i_b; \\
0, & \text{if } i_a \neq i_b.
\end{cases}
\]

Based on Eq. 6, in our running example of Fig. 1, transition probability from item \( i_4 \) to item \( i_6 \) is equal to \( \frac{1}{2} \), and it is so since in all the sessions of time window \( t_p \) there is only one case where \( i_4 \) is followed by \( i_6 \) (session \( S_4 \)); and the denominator is equal to two, since there are two sessions where \( i_4 \) is followed by any other item (sessions \( S_3 \) and \( S_4 \)). The intra-session transition probability matrix is presented in Table 3 (rows and columns with zeros are not shown).

To summarise, intra-session TPM infers similarity among items inside each session independently from other sessions, whereas inter-session similarity captures the notion of similarity between any two consecutive sessions. As will be shown experimentally, the inter-session similarity is more effective when we increase the size of the sliding time windows, which means that it can better capture the long-term user preferences, whereas the intra-session similarity is more effective with smaller window sizes, which makes it more suitable to capture the short term preferences.

### 4.1 Combining intra- with inter-session item transition probabilities

Intra-session transition probability matrix (TPM) captures the relevance of articles inside a session. However, when sessions do not have many items or in the beginning of a session, it is difficult to provide accurate recommendations. On the other hand, inter-session TPM captures the relevance of articles among different sessions. Thus, it is able to detect those items that belong to other possible similar sessions with those of the target item. As our objective is to provide more accurate news recommendations, we combine the two TPMs into a single one, which is given in Eq. 7:

\[
intra - inter(t_p, i, j) = \alpha \ast intra(t_p, i, j) + (1 - \alpha) \ast inter(t_p, i, j),
\]

where \( t_p \) is the valid time period. Please notice that in several cases the distribution of the item transition probability values between intra-session and inter-session TPM may
differ significantly. Therefore, in that case, we have to normalise in the interval \([0,1]\) the values of the two TPMs before combining them. Coefficient \(\alpha\) give us flexibility to boost one prediction model over the other.

### 4.2 Recommendation list creation

Our recommender module provides recommendations based on the combined TPMs presented in previous section. For each target user \(u\), the recommender checks the set of her recently viewed items \(I_{tp,u}\) (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 \(I_{tp,u}\). Same as in Ludewig and Jannach (2018), for each target user \(u\) in \(t_p\) and for each item \(j\) we compute a ranking score \(score(t_p,u,j)\) as follows:

\[
score(t_p,u,j) = \sum_{i \in I_{tp,u}} \text{intra} - \text{inter}(i,j) \times 1_{(j,K_i)}
\]

where \(1_{(j,K_i)}\) 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.

### 5 Experimental evaluation

In this Section, we will describe the basic characteristics and statistics of three real-life data sets (acquired from two news providers’ - an Italian and a German- that operate in the region of Alto Adige in Italy and a Norwegian news provider which operates in Norway (i.e., the well-known Adressa data set). These data sets will be later used to evaluate the effectiveness of our method against state-of-the-art methods (i.e., GRU4REC, Session-knn, MF, etc.) .

#### 5.1 Italian news provider data set

For the Italian news provider, the data set accommodates 14367 interactions/events on 2081 articles of 14047 unique users in one year (i.e. from 1st April 2016 to 30th March 2017). 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. User sessions have an average number of interactions equal to 2.78 after removing sessions that had only one item, as shown in Fig. 2a.

Detailed general statistics for the Italian and the other two news providers are summarized in Table 4, where the cleaning procedure lies in removing the sessions that contain only one article, as no predictions can be tested on such sessions, and no article co-occurrence patterns can be identified. Thus, there need to be at least two items within a session to use it for for experimental testing. Please notice that most of the users have only a small number of sessions (i.e., 1.23 or 1.17 or 1.03 sessions per user in the Italian, the German and the Norwegian/Adressa news provider, respectively). For the Italian news provider, there are 2681 article to another article transitions within 1126 sessions made by 918 unique users. However, the number of sessions per user, and the number of sessions by item is very close in all datasets.
Fig. 2 A visual representation that shows the number of interactions per session for the a Italian news provider, b German news provider, and c The Adressa

Table 4 General statistics of datasets

|                     | Italian news provider | German news provider | Norwegian news provider (Adressa) |
|---------------------|-----------------------|----------------------|-----------------------------------|
|                     | No cleaning | After cleaning | No cleaning | After cleaning | No cleaning | After cleaning |
| Users               | 10,421   | 918         | 3626       | 600         | 238,124     | 9141         |
| Articles            | 2081     | 1097        | 4078       | 704         | 6091        | 2314         |
| Sessions            | 11,686   | 1126        | 5536       | 2161        | 1,356,987   | 32,281       |
| Views               | 14,367   | 3130        | 468        | 169         | 6091        | 2314         |
| Avg # views per session | 1.23     | 2.78        | 1.36       | 3.07        | 2.92        | 3.42         |
| Avg # views per user | 1.38     | 3.41        | 1.53       | 3.6         | 5.7         | 3.53         |
| Avg # sessions per user | 1.12     | 1.23        | 1.2        | 1.17        | 1.95        | 1.03         |
| Avg # sessions per article | 5.61     | 2.53        | 9.94       | 6.68        | 76.32       | 4.08         |
| Avg # views per article | 6.9      | 2.85        | 11.8       | 12.78       | 222.79      | 13.95        |
5.2 German news provider data set

For the German news provider, the data set accommodates 5536 interactions on 468 articles of 3626 unique users in one year. User sessions have an average number of interactions equal to 3.07, as shown in Figure 2b. For the German news provider there are 1458 article to another article transitions within 704 sessions made by 600 unique users, as shown in Table 4. Moreover, news articles on German news provider’s website are viewed approximately twice as much in comparison to the news articles on Italian news provider website, which means that its visitors are more engaged. The german-speaking population is more engaged in the German’s news provider website because it may not have many choices of other local news sources in the region of Alto Adige in Italy. It is also interesting to note that though there are less news articles in German’s news provider website site, these articles are viewed almost twice as much comparing to the news articles on the Italian News provider web site.

5.3 Norwegian news provider data set (Adressa)

For the Adressa news data set, we have used the data from two days (i.e., 5/1/2017 and 6/1/2017) of the light version (1.4 GB) to speed up the evaluation’s process. It is a Norwegian company and its data set includes 1356987 views/interactions on 6091 articles of 238124 unique users. We have identified 18 article categories (e.g., 100sport, nyheter, ‘pluss’, etc.) and 66 article sub-categories (e.g., nyheter—okonomi, nyheter—trondheim, ‘pluss—okonomi’, ‘pluss—magasin’, etc.) Please notice that we have built the graph by considering subcategories instead of categories, based on the fact that this is more detailed information. User sessions have an average number of interactions equal to 2.64, as shown in Fig. 2c.

5.4 Prequential evaluation protocol

In this Section, we present our evaluation protocol, which is in the same direction with the one introduced by Jannach et al. (2017), Ludewig and Jannach (2018) for predicting the

---

1 https://www.panagiotissymeonidis.com/?page_id=37
2 http://reclab.idi.ntnu.no/dataset/
1.3 next item inside a session, known also as prequential evaluation in stream mining (Quadranza et al., 2018; Vinagre et al., 2014), whereas we divided the data into three splits, so we have 80% of data for training the prediction model, 10% of the data for tuning the parameters, and the rest 10% of data for evaluating the model. As shown in Fig. 3, in prequential evaluation, future articles are first predicted by the model, so that the quality of the model is evaluated; then articles with their true labels are used for model learning, which means that approaches adapt to the user’s every next click. As also shown in Fig. 3, results are obtained when applying a sliding-window protocol, where we split the data into several slices of equal size.

An important parameter of this protocol is the sliding time window size of the training data. 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 predicting the next items in a session.

Finally, we evaluate the precision (i.e., the number of hits divided by the number of recommended items) we get when we recommend top-5 articles for each next item prediction inside a session. We split time in $N_t$ time periods, so that we can aggregate the precision results for each different time period $t_p$.

5.5 Sensitivity analysis of the proposed method

In this Section, we study the accuracy performance of the (i) intra-session TPM, (ii) inter-session TPM, (iii) and their combination intra-inter session TPM. We will explore, how the precision accuracy of the aforementioned methods changes as we vary different parameters such as (i) different time period splits: $N_t = 12, 45, 90, 183, 365, 730$ (ii)
various time window sizes: $w = 1, 5, 10$, (iii) different number of recommended top-$N = 1, 2, 3, 4, 5$ items and (iv) the linear combination of intra-session TPM with inter-session TPM, where both methods are considered to have equal weight (0.5).

For the Italian news provider data set, in Fig. 4a, we set the sliding time window $w = 1$ and change the number of time period splits $N_t = 183, 365, 730$, which consider a time slot $t_p$ equal to 2 days, 1 day, and 12 hours, respectively. The number of time period splits $N_t$ controls the size of the future that we want to predict. Please notice that for all $N_t$ values we aggregate the results in the level of months and show their average precision score also over a month (12 two-hour time slots for Adressa). The reason is two-fold: (i) we present results that are statistical significant and (ii) we show more meaningful aggregated visual analytics. All presented measurements (i.e., the average of the reported values between the runs over a month), based on two-tailed t-test, are statistically significant at the 0.05 level. The null hypothesis was that the average of the reported results of precision will be outside the confidence interval. However, the null hypothesis is rejected with 0.05 statistical significance. Thus, we can generalize our found results and claim that precision averages reported results will be always the same as many times we perform a new run of the experiment.

As shown in Fig. 4a, when we set $N_t = 365$, we get the best precision. This means that we should focus in predicting the next day and not longer time periods (e.g., week). Henceforth, we perform experiments with $N_t = 365$. Next, as shown in Fig. 4d, we want to measure how the performance of the “intra-session TPM” changes as we use different time window sizes: $w = 1, 5, 10$, for time slots $N_t = 365$. That is, we want to see how our model is affected as we look more back into the past. As shown in Fig. 4d the window size $w=10$ attains the best average precision. Please notice that as we decrease $w$ precision drops. For further experiments, we fix our method for $N_t = 365$ and $w = 10$, since when we set $w > 10$, precision also drops. The reason for this phenomenon is the recency of news articles and the fact that they have a very short life span (i.e., only some days). Finally, as shown in Fig. 4g, we want to measure the performance of “intra-session TPM” as we change the number of top-$N$ recommended items: $N = 1, 2, 3, 4, 5$. As expected, as we increase the number of top-$N$ recommended items, precision drops. As the data sets are sparse and few recommendation items are available for every user in the test set, we will measure precision by setting $N=1$ recommended items in the rest of experiments.

For the German news provider, results are shown in Fig. 4b, e, and h. As shown in Fig. 4b, when we set $N_t = 45$, we get the best precision. Henceforth, we perform experiments with $N_t = 45$ for this data set. As shown in Fig. 4e the windows size $w=1$ attains the best average precision and as we increase $w$ precision drops. This contradicts the trend that we have seen in the Italian news provider data set. The reason is that in the German news provider data set, we have less time intervals/splits (i.e., 45 instead of 365 which we had for the Italian news provider). Thus, by having time intervals of 8 days, we do not need to increase the window size $w$ (i.e., the time period that we look back into the past), as we did for the Italian new provider data set by setting $w = 10$ days. Finally, as shown in Fig. 4h, again as we increase the number of top-$N$ recommended items, precision drops.

For the Norwegian news provider (Adressa data set), as shown in Fig. 4c, we get the best precision with $N_t = 12$, which means that we attain better precision when we try to predict the next 2 hours. As expected, different values of the time window $w$ do not give different precision levels, as shown in Fig. 4f. The reason is that our Adressa sub-set consists only of 24 hours of user-item interactions. Thus, the fact that we increase the time
window \( w \) for some hours does not contribute so much in increasing precision. Henceforth, we will set \( w = 5 \) since it attains the best precision.

Next, we want to see if the combination of intra- with inter-session TPM can give us better results. For the Italian news provider data set, as it is shown in Fig. 5a, the intra-inter session TPM achieves better results than intra-session TPM and inter-session TPM separately, for almost all time points, when we set \( N_t = 365 \), top-\( N = 1 \) and \( w = 10 \) with \( \alpha = 0.5 \). That is, by forgetting faster older news is better. In other words, when we consider for our prediction model only the articles of the previous ten days before the target session for which we want to make article predictions, then we get the best precision. This is as expected, since the life span of articles is short and news stories become the focus of interest quickly and disappear just as quickly. Please notice that when we run experiments

| \( \alpha \) parameter | Italian news portal, precision (%) | German news portal, precision (%) | Norwegian portal, precision (%) |
|------------------------|-------------------------------|-------------------------------|-------------------------------|
| 0                      | 1.28                          | 7.86                          | 15.78                         |
| 0.1                    | 2.37                          | 16.70                         | 16.05                         |
| 0.2                    | 5.65                          | 17.27                         | 15.69                         |
| 0.3                    | 8.75                          | 17.89                         | 14.39                         |
| 0.4                    | 9.67                          | 18.22                         | 13.73                         |
| 0.5                    | 11.4                          | 19.46                         | 13.10                         |
| 0.6                    | **12.37**                     | 20.23                         | 12.58                         |
| 0.7                    | 10.58                         | 21.56                         | 11.88                         |
| 0.8                    | 8.45                          | **22.88**                     | 11.56                         |
| 0.9                    | 7.7                           | 21.19                         | 11.34                         |
| 1                      | 6.54                          | 20.73                         | 6.85                          |

Bold values indicate the highest average precision that we get for each dataset, when we hypertune parameter \( \alpha \).

Fig. 5 Comparison of “intra-session TPM”, inter-session TPM, and intra-inter session TPM for top-\( N = 1 \) recommended items
with $N_t = 730$ and $w = 10$, precision is again decreased, which means that when we predict recommendations based on half day (12 hours) is more effective than making recommendations based on just 6 hours. For the German news provider data set, again the intra-inter session TPM achieves better results, as it is shown in Fig. 5b, when we set $N_t = 45$, top-$N$ = 1 and $w = 1$. Finally, for the Norwegian news provider, as shown in Fig. 5c, the combination of intra- with inter-session TPM does not give better results than the intra-session TPM alone. The main reason is the very small number of sessions per user. That is, each user has on average only 1.03 sessions, which makes the contribution of inter-session TPM extremely marginal.

Finally, we want to identify the best performance of intra-inter session TPM by tuning the $\alpha$ parameter. We measured precision for the Italian, the German and the Norwegian News Providers, for $\alpha = 0.1, 0.2, 0.3..0.8, 0.9, 1$. The results are summarised in Table 5. As it is shown, the best results are attained when we set $\alpha$ parameter equal to 0.6, 0.8 and 0.1 for the Italian, the German and the Norwegian news provider, respectively. Henceforth, we will use these $\alpha$ parameter values for the comparison with other state-of-the-art methods.

### 5.6 Comparison with other methods

In this Section, we compare our method with the following baseline and state-of-the-art comparison partners, which are representatives of different algorithmic families such as collaborative filtering, recurrent neural network, Markov chain model, matrix factorisation, and a hybrid method of the last two ones.

(i) **Most Popular Recent Items (Recently POP):** Recently POP recommends the top-$N$ most clicked articles of the active/valid time period $t_p$.

(ii) **Item-based Collaborative Filtering (IBCF)** (Das et al., 2007): Based on IBCF, two items are considered similar, if they are selected by similar users. In Das et al. (2007), IBCF considers the co-visitation count of news articles, which counts the number of times an item was co-visited (clicked before or after) with another item.

(iii) **Session-knn** (Jannach & Ludewig, 2017): Session-knn method takes the set of the target user actions in the current session, e.g. two view events for certain items, and then finds the $k$ most similar past sessions in the training data, and recommends items from this neighborhood of similar sessions. In other words, it takes the set of user actions in the current session, e.g. two view events for certain items, and then in a first step determines the $k$ most similar past sessions in the training data. Then, given the current session $s$, the set of $k$ nearest neighbors $N_s$, and a function $sim(s_1, s_2)$ that returns a similarity score for two sessions $s_1$ and $s_2$, the score of a recommendable item $i$ is:

$$score_{KNN}(i, s) = \sum_{n \in N_s} sim(s, n) \times 1_n(i),$$

where $1_n(i) = 1$ if $n$ contains $i$ and 0 otherwise. The similarity measure used by Jannach and Ludewig (2017) in experiments is cosine similarity, as it was found out that the best results are achieved when encoding sessions as binary vectors of the item space.

(iv) **GRU4REC** (Hidasi et al., 2015): GRU4REC is a neural network-based recommender system that uses a Gated Recurrent Unit (GRU), which learns when and how much to update the hidden state of the GRU model. In particular, GRU4REC is a recurrent neural network, which modifies the basic GRU to fit the prediction task better by introducing session-parallel mini-batches, mini-batch output sampling and the ranking loss function.
Matrix factorization (MF) (Symeonidis & Zioupos, 2016): MF factorises the user-article rating matrix $R$ into two matrices, one $U$ with $n$ rows and $k$ columns and one $V$ with $m$ rows and $k$ columns, such that $UV^\top$ produces $R$ with the blank entries filled and a small deflection of the initial values.

DeepMF (Guo et al., 2017) DeepMF is a new neural network algorithm that combines matrix factorization and deep learning to improve the model’s prediction effectiveness.

Neural collaborative filtering (NeuralCF) (He et al., 2017a) NeuralCF is a new neural network architecture that models latent features of users and items and devise a general framework for collaborative filtering based on neural networks.

Factorized personalized Markov chain (FPMC) (Rendle et al., 2010): FPMC factorizes a transition matrix $P_{ij}$ of a Markov Chain Model $\{X\}$, where each row contains the probability of transition between states.

$$P_{ij} = \mathbb{P}(X_{t+1} = x_j | X_t = x_i)$$  (10)

Each row of the matrix is a probability vector, and the sum of its components is equal to 1. Thus, FPMC is a first-order Markov chain whose transition matrix is jointly factorised using a standard 3-dimensional tensor (i.e., user, current item, next item). This joint factorisation at the end makes it possible to infer the unobserved transitions in the Markov chain from the transition pairs of other users. By limiting the basket size to one item and by observing the current session as the history of transactions, the method can be directly applied for computing article recommendations.

The parameters we used to evaluate the performance of the comparison partners are similar to those reported in the original papers, since after tuning we got the best results for our data sets.

Table 6 reports the average precision over all users of the comparison year for all the under comparison algorithms, when $N_t = 365$ and $w = 10$ for the Italian news provider, $N_t$
= 45 and \( w = 1 \) for the German news provider, and \( N_t=12, w = 5 \) for the Norwegian news provider. We run experiments with top-5 recommended articles in all three data sets. As shown in the last row of Table 6, our proposed approach has the best average precision over the year among all comparison partners. The reason is that when we combine the two models together (i.e., inter-, with intra-session TPM), we are able to capture the short and long preferences of individuals. In our combined approach, we incorporate 2 different transition probabilities matrices to capture the individual’s long and short preferences (see Eq. 7).

As far as the rest comparison partners is concerned, as expected, IBCF does not attain good results because there are not enough data to build a prediction model. That is, many users re-appear irregularly and very rarely at time points that are far apart, which means that collaborative filtering cannot build always a model, since users should appear in two consecutive time slots. MF attains very good results since it is the most successful non-sequential method in collaborative filtering. However, it fails to outperform our method because it builds only a long-term general model to capture the user preferences. FPMC, NeuralCF and DeepMF attain a better performance than MF alone. However, they do not consider the intra-session transition probability of items and thus, fails to capture the very last user intentions. Furthermore, as expected, Session-knn outperforms GRU4REC (Jannach & Ludewig, 2017). However, Session-knn is far worst than Intra-TPM, because it cannot capture adequately the latent associations among items inside the same session. Please notice that the Italian, the German and the Norwegian news data sets have avg. number of items per session 2.78, 3.07, and 3.42, respectively, as can be seen in Table 4, which means that there is a severe data sparsity.

In contrast, the performance of all methods for German and Norwegian news providers are twice better, than the performance for the Italian news provider. The reason is that articles in the German and the Norwegian news providers’ web sites are viewed many times more comparing to the articles of the Italian news provider web site. In particular, the average views per article for the German and the Norwegian news providers is 12.78 and 13.95, whereas only 2.85 for the Italian news provider, as can be seen in Table 4.

### 6 Discussion

Markov chain-based algorithms have scalability issues, when the number of possible actions (recommendation coverage) increases dramatically together with the number of users and items increase. In other words, Markov chain-based algorithms that operate in closed deterministic systems, where the number of states and the number of actions are strictly predetermined (Finite Markov Decision Processes) have serious problems in terms of scalability. That is, there are real-life problems, such as recommender systems where both the number of states (e.g., different user interactions with items, i.e., different user profiles) and the number of actions (e.g, the number of items that are candidates for recommendation) are too large. In such complex stochastic environments, reinforcement learning algorithms such as Q-learning and A2C, which do not have explicitly defined transition probabilities from one state to another, are able to overcome those scalability issues. Another solution could be the usage of a distributed reinforcement learning system such as Federated learning and the A3C algorithm, which splits the work on many servers and then aggregates their results to speed up the process of computing the transition probability matrix, by having also improvement in terms of the effectiveness of the prediction model.
In recent years, many deep learning algorithms, such as Deep Matrix Factorization (DeepMF) (Guo et al. 2017), Recurrent Neural Networks (GRU4Rec) (Hidasi et al., 2015), and the Neural Collaborative Filtering (He et al., 2017b), have been applied to recommender systems. The effectiveness of the aforementioned deep machine learning methods in recommender systems is high because they have multiple neural layers to process the data in great detail and analysis to reveal the hidden user interactions. However, we have shown experimentally that our method outperforms these state-of-the-art methods in terms of effectiveness. The main reason is that we exploit effectively the time dimension by combining intra- with inter-session user information to reveal the hidden interactions of individuals with items. In contrast to the neural network-based approaches, our proposed method has also the advantage of supporting explainable recommendations. Furthermore, our experiments have shown that matrix factorization algorithms, which use separately either long- or short-term time data, were not able to perform well. This was clearly observed by comparing our method with the state-of-the-art factorization algorithms such as MF and FPMC. These limitations concern also collaborative filtering approaches such as IBCF, where building a reliable prediction model was impossible because IBCF is incapable of processing data related with the time dimension. We can conclude that our approach has a superior performance as it allows to use both short- and long-term user preferences, and controls their adequate combination.

7 Conclusion

In this paper, we combined intra- with inter-session TPM to reveal the short- and long-term intentions of individuals, respectively. We have evaluated experimentally our method and compare it against state-of-the-art algorithms on three real-life datasets. We have shown the superiority of our method over its competitors. As future work, we will combine text mining and recommendation techniques to process data from multiple heterogeneous data sources (i.e., the text of news articles, the usage log data from user preferences on news articles, etc.), to better model the fact that articles’ life span is also depended on the news topic category that they belong to.

Funding Open access funding provided by HEAL-Link Greece.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.
References

An, M., Wu, F., Wu, C., Zhang, K., Liu, Z., & Xie, X. (2019). Neural news recommendation with long-and short-term user representations. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 336–345.

Chen, S., Moore, J.L., Turnbull, D., & Joachims, T. (2012). Playlist prediction via metric embedding. In The 18th ACM International Conference on Knowledge Discovery and Data Mining, KDD ’12, Beijing, China, ACM, pp. 714–722. https://doi.org/10.1145/2339530.2339643

Das, A.S., Datar, M., Garg, A., & Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In Proceedings of the 16th international conference on World Wide Web, ACM, pp. 271–280.

Esiyok, C., Kille, B., Jain, B. J., Hopfgartner, F., & Albayrak, S. (2014). Users’ reading habits in online news portals, In Fifth Information Interaction in Context Symposium, IIIix ’14, Regensburg, Germany, ACM, pp. 263–266. https://doi.org/10.1145/2637002.2637038

Garcić, F., Dimitrakakis, C., & Faltings, B. (2013). Personalized news recommendation with context trees, In Proceedings of the 7th ACM conference on Recommender systems, RecSys’13, Hong Kong, China, ACM, pp. 105–112. https://doi.org/10.1145/2507157.2507166

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). Deepfm: A factorization-machine based neural network for CTR prediction, arXiv arXiv:1703.04247.

He, Q., Jiang, D., Liao, Z., Hoi, S.C.H., Chang, K., Lim, E., & Li, H. (2009). Web query recommendation via sequential query prediction, In Proceedings of the 25th International Conference on Data Engineering, ICDE’09, Shanghai, China, IEEE, pp. 1443–1454. https://doi.org/10.1109/ICDE.2009.71

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web, pp. 173–182.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, WWW ’17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, p. 173-182.

Hidasi, B., & Karatzoglou, A. (2018). Recurrent neural networks with top-k gains for session-based recommendations, arXiv preprint arXiv:1706.03847.

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks, CoRR arXiv:1511.06939.

Hidasi, B., Quadrana, M., Karatzoglou, A., & Tikk, D. (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, pp. 241–248. https://doi.org/10.1145/2959100.2959167

Jannach, D., & Ludewig, M. (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, pp. 306–310. https://doi.org/10.1145/3109859.3109872

Jannach, D., Lerche, L., & Jugovac, M. (2017). Adaptation and evaluation of recommendations for short-term shopping goals, In Proceedings of the ninth ACM Conference on Recommender Systems, RecSys ’15, ACM, New York, NY, USA.

Liu, J., Dolan, P., & Pedersen, E.R. (2010). Personalized news recommendation based on click behavior, In Proceedings of the 15th international conference on Intelligent user interfaces, ACM, pp. 31–40.

Li, L., Zheng, L., Yang, F., & Li, T. (2014). Modeling and broadening temporal user interest in personalized news recommendation. Expert Systems with Applications, 41(7), 3168–3177.

Ludewig, M., & Jannach, D. (2018) Evaluation of session-based recommendation algorithms, in: Journal Computing Report Repository, arXiv:1803.09587.

Ludewig, M., & Jannach, D. (2018). Evaluation of session-based recommendation algorithms, arXiv preprint arXiv:1803.09587.

Ludmann, C. (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.

McFee, B., & Lanckriet, G.R.G. (2011). The natural language of playlists. In Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR’11, Miami, Florida, USA, ISMIR, pp. 537–542. http://ismir2011.ismir.net/papers/PS4-11.pdf

Moling, O., Baltrunas, L., & Ricci, F. (2012). Optimal radio channel recommendations with explicit and implicit feedback, In Sixth ACM Conference on Recommender Systems, RecSys’12, Dublin, Ireland, ACM, pp. 75–82. https://doi.org/10.1145/2365952.2365971
Paparrizos, I.K., Cambazoglu, B.B., & Gionis, A. (2011). Machine learned job recommendation, In Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys’11, Chicago, IL, USA, ACM, pp. 325–328. https://doi.org/10.1145/2043932.2043994

Quadrana, M., Cremonesi, P., & Jannach, D. (2018). Sequence-aware recommender systems, arXiv preprint arXiv:1802.08452.

Quadrana, M., Karatzoglou, A., Hidasi, B., & Cremonesi, P. (2017). Personalizing session-based recommendations with hierarchical recurrent neural networks, In Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys ’17, ACM, New York, NY, USA, pp. 130–137. https://doi.org/10.1145/3109859.3109896

Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). Factorizing personalized markov chains for next-basket recommendation, In Proceedings of the 19th International Conference on World Wide Web, WWW’10, Raleigh, North Carolina, USA, ACM, pp. 811–820. https://doi.org/10.1145/1772690.1772773

Shani, G., Heckerman, D., & Brafman, R. I. (2005). An mdp-based recommender system, Journal of Machine Learning Research (2005), Vol. 6, JMLR, pp. 1265–1295. http://www.jmlr.org/papers/v6/shani05a.html

Smirnova, E., & Vasile, F. (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, pp. 2–9. https://doi.org/10.1145/3125486.3125488

Symeonidis, P., & Zioupos, A. (2016). Matrix and tensor factorization techniques for recommender systems. Springer.

Tan, Y. K., Xu, X., & Liu, Y. (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, pp. 17–22. https://doi.org/10.1145/2988450.2988452

Vinagre, J., Jorge, A., & Gama, J. (2014). Evaluation of recommender systems in streaming environments, In Workshop on Recommender Systems Evaluation: Dimensions and Design (REDD 2014), held in conjunction with RecSys 2014, Silicon Valley, USA.

Yu, Z., Lian, J., Mahmoody, A., Liu, G., Xie, X. (2019). Adaptive user modeling with long and short-term preferences for personalized recommendation., In IJCAI, pp. 4213–4219.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.