Modeling Sequential Preferences with Dynamic User and Context Factors

Duc-Trong Le[1](✉), Yuan Fang[2], and Hady W. Lauw[1]

[1] School of Information Systems, Singapore Management University, Singapore
ductrong.le.2014@phdis.smu.edu.sg,hadywlauw@smu.edu.sg
[2] Institute for Infocomm Research, Singapore
yfang@i2r.a-star.edu.sg

Abstract. Users express their preferences for items in diverse forms, through their liking for items, as well as through the sequence in which they consume items. The latter, referred to as "sequential preference", manifests itself in scenarios such as song or video playlists, topics one reads or writes about in social media, etc. The current approach to modeling sequential preferences relies primarily on the sequence information, i.e., which item follows another item. However, there are other important factors, due to either the user or the context, which may dynamically affect the way a sequence unfolds. In this work, we develop generative modeling of sequences, incorporating dynamic user-biased emission and context-biased transition for sequential preference. Experiments on publicly-available real-life datasets as well as synthetic data show significant improvements in accuracy at predicting the next item in a sequence.

Keywords: sequential preference, generative model, user-biased emission, context-biased transition

1 Introduction

Users express their preferences in their consumption behaviors, through the products they purchase, the social media postings they like, the songs they listen to, the online videos they watch, etc. These behaviors are leaving increasingly greater traces of data that could be analyzed to model user preferences. Modeling these preferences has important applications, such as estimating consumer demand, profiling customer segments, or supporting product recommendation.

There are diverse forms of expression of preferences yielding different types of observations. Most of the previous works deal with *ordinal preference*, where the objective is to model the observed interactions between users and items [1]. In this scenario, a user's preference for an item is commonly expressed along some ordinal scale, e.g., higher rating indicating greater liking or preference.

In this work, we are interested in another category, namely: *sequential preference*, where the objective is to model the sequential effect between adjacent items in a sequence. In this scenario, preference is expressed in terms of which other items may be preferred after consuming an item. For instance, a user's stream of

tweets may reveal which topics tend to follow a topic, e.g., commenting on politics upon reading morning news followed by more professional postings during working hours. The sequence of songs one listens to may express a preference for which genre follows another, e.g., more upbeat tempo during a workout followed by slower music while cooling down. Similarly, sequential preferences may also manifest in the books one reads, the movies one watches, etc.

Problem. Given a set of item sequences, we seek a probabilistic model for *sequential preferences*, so as to estimate the likelihood of future items in any particular sequence. Each sequence (e.g., a playlist, a stream of tweets) is assumed to have been generated by a single user.

To achieve this goal, we turn to probabilistic models for *general* sequences. While there are several such models studied in the literature (see Section 2), here we build on the foundation of the well-accepted Hidden Markov Model (HMM) [16], which has been shown to be effective in various applications, including speech- and handwriting-recognition, etc. We review HMM in Section 3. Briefly, it models a number of hidden states. To generate each sequence, we move from one state to another based on *transition* probability. Each item in the sequence is sampled from the corresponding state's *emission* probability.

While HMM is fundamentally sound as a basic model for sequences, we identify two significant factors, yet unexploited, which would contribute towards greater effectiveness for modeling sequential preferences. *First*, the generation of an item from a state's emission in HMM is only dependent on the state. However, as we are concerned with *user-generated* sequences, the selection of items may be affected by the user's preferences. However, due to the sparsity of information on individual users, we stop short of modeling individual emissions. Rather, we model latent groups, whereby users in the same group share similar preferences over items, i.e., emissions. *Second*, the transition to the next state in HMM is only dependent on the previous state. We posit that *context* in which a transition is about to take place also plays a role. For example, in the scenario of musical playlists, let us suppose that a particular state represents the genre of soft rock. There are different songs in this genre. If a user likes the artist of the current song, she may wish to listen to more songs by the same artist. Otherwise, she may wish to change to a different genre altogether. In this case, the artist is an observed *feature* of the context that may influence the transition dynamically.

Contributions. In this work, we make the following contributions. *First*, we develop a probabilistic model for sequences, whereby transitions from one state to another state may be *dynamically* influenced by the context features, and emissions are influenced by latent groups of users. We develop this model systematically in Section 4, and describe how to learn the model parameters, as well as to generate item predictions in Section 5. *Second*, we evaluate these models comprehensively in Section 6 over varied datasets. Experiments on a synthetic dataset investigate the contributions of our innovations on a dataset with known parameters. Experiments on publicly available real-life sequence datasets (song playlists from Yes.com and hashtag sequences from Twitter.com) further showcase accuracy improvements in predicting the next item in sequences.

2 Related Work

Here, we survey the literature on modeling various types of user preferences.

Ordinal Preferences. First, we look at *ordinal preferences*, which models a user's preference for an item in terms of rating or ranking. The most common framework is matrix factorization [11, 17, 20], where the observed user-by-item rating matrix is factorized into a number of latent factors, so as to enable prediction of missing values. Another framework is restricted Boltzmann machines [21] based on neural networks. Meanwhile, latent semantic analysis [8, 9] models the association among users, items, and ratings via multinomial probabilities. These works stand orthogonally to ours, as the main interactions they seek to model are user-to-item ratings/rankings, rather than item-to-item sequences.

Sequential Preferences. Our work falls into *sequential preferences*, which models sequences of items, so as to enable prediction of future items. As mentioned in Section 1, our contribution is in factoring dynamic context-biased transition and user-biased emission. To make the effects of these dynamic factors clear, we build on the foundation of HMM [16], and focus our comparisons against this base platform. Aside from HMM, there could potentially be different ways to tackle this problem such as probabilistic automata [7] and recurrent neural networks [14], which are beyond the scope of this paper. Other works deal with sequences, but with different objectives. Markov decision processes [2, 22, 23] are concerned with how to make use of the transitions to arrive at an "optimal policy": a plan of actions to maximize some utility function. Sequential pattern mining [15] finds frequent sequential patterns, but these require exact matches of items in sequences. [4, 13] model sequences in terms of Euclidean distances in metric embedding space. Aside from different objectives, these works also model explicit transitions among items, in contrast to our modeling of latent states.

Hybrid Models. Efforts to integrate ordinal and sequential preferences combine the "long-term" (items a user generally likes) and "short-term" preferences (items frequently consumed within a session). [27] models the problem as random walks in a session-based temporal graph. [26] designs a two-layer representation model for items: the first layer models interaction with previous item and the second layer models interaction with the user. [6, 18] conduct joint factorization of user-by-item rating matrix and item-by-item transition matrix. It is not the focus of our current work to incorporate ordinal preferences directly, or to rely on full personalization by associating each user with an individual parameter.

Temporal Models. Aside from the notion of sequence, there are other *temporal factors* affecting recommendation. [19] assumes that users may change their ordinal preferences over time. [3] models the scenario where users "lose interest" over time. [10] takes into account the life stage of a consumer, e.g., products for babies of different ages, while [28] intends to model evolutions that advance "forward" in event sequences without going "backward". [25] seeks to predict not what, but rather when to recommend an item. [5] considers how changes in social relationships over time may affect a user's receptiveness or interest to change. In these and other cases, the key relationship being modeled is that between user and time, which is orthogonal to our focus in modeling item sequences.

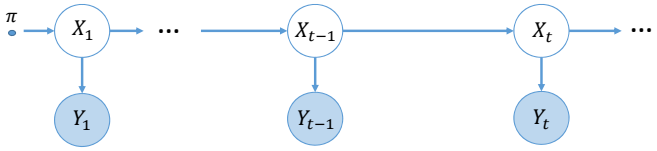

Fig. 1. A standard HMM for sequential preferences

3 Preliminaries

Towards capturing sequential preferences, our model builds upon HMM. The standard HMM assumes a series of discrete time steps $t = 1, 2, \ldots$, where an item Y_t can be observed at step t. To model the sequential effect in this series of observed items, HMM employs a Markov chain over a *latent* finite state space across the time steps. As illustrated in Fig. 1, at each time step t a latent state X_t is transitioned from the previous state X_{t-1} in a Markovian manner, i.e., $P(X_t|X_{t-1}, X_{t-2}, \ldots, X_1) \equiv P(X_t|X_{t-1})$, known as the *transition* probability.

Formally, consider an HMM with a set of observable items $\mathcal{Y}$ and a set of latent states $\mathcal{X}$. It can be fully specified by a triplet of parameters $\theta = (\pi, A, B)$, such that $\forall x, u \in \mathcal{X}, y \in \mathcal{Y}, t \in \{1, 2, \ldots\}$,

- π is the initial state distribution with $\pi_x \triangleq P(X_1 = x)$;
- A is the transition matrix with $A_{xu} = P(X_t = u|X_{t-1} = x)$;
- B is the emission matrix with $B_{xy} = P(Y_t = y|X_t = x)$.

Given a sequence of items $Y_1, \ldots, Y_t$, the optimal parameters θ^* can be learned by maximum likelihood (Eq. 1). Note that we can easily extend the likelihood function to accommodate multiple sequences, but for simplicity we only demonstrate with a single sequence throughout the technical discussion. Moreover, given θ^* and a sequence of items $Y_1, \ldots, Y_t$, the next item y^* can be predicted by maximum a posteriori probability (Eq. 2). Both learning and prediction can be efficiently solved using the forward-backward algorithm [16].

$$\theta^* = \arg\max_\theta P(Y_1, ..., Y_t; \theta) \tag{1}$$

$$y^* = \arg\max_y P(Y_{t+1} = y|Y_1, \ldots, Y_t; \theta^*) \tag{2}$$

4 Proposed Models

In a standard HMM, item emission probabilities are invariant across users, and state transition probabilities are independent of contexts at different times. However, these assumptions often deviate from real-world scenarios, in which different users and contexts may have important bearing on emissions and transitions. In this section, we model dynamic emissions and transitions respectively, and ultimately jointly, to better capture sequential preferences.

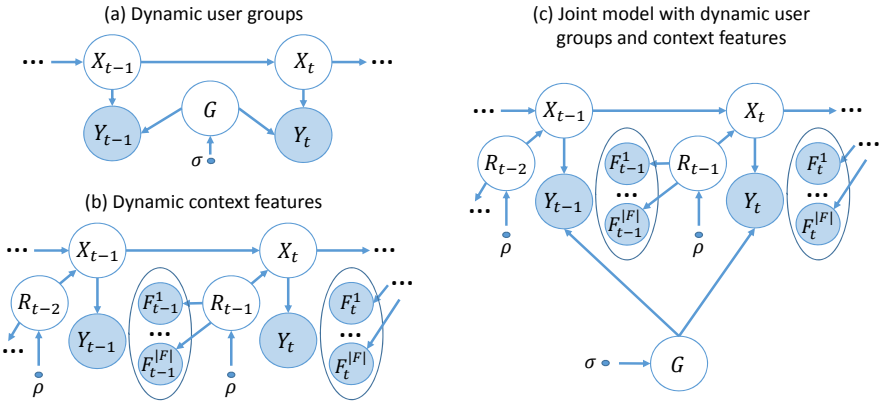

Fig. 2. Sequential models with dynamic user groups and contexts

4.1 Modeling Dynamic User-Biased Emissions (SEQ-E)

It is often attractive to consider personalized preferences [18], where different user sequences may exhibit different emissions even though they share a similar transition. For instance, while two users both transit from soft rock to hard rock in their respective playlist, they might still choose songs of different artists in each genre. As another example, two users both transit from spring to summer in their apparel purchases, but still prefer different brands in each season. However, a fully personalized model catered to every individual user is often impractical due to inadequate training data for each user. We hypothesize that there exist different groups such that users across groups manifest different emission probabilities, whereas users in the same group share the same emission probabilities.

In Fig. 2(a), we introduce a variable G_u to represent the group assignment of each user u. For simplicity, our technical formulation presents a single sequence and hence only one user. Thus, we omit the user notation u when no ambiguity arises. Assuming a set of groups $\mathcal{G}$, the new model can be formally specified by the parameters (π, σ, A, B), such that $\forall x \in \mathcal{X}, y \in \mathcal{Y}, g \in \mathcal{G}, t \in \{1, 2, \ldots\}$,

- π and A are the same as in a standard HMM;
- σ is the group distribution with $\sigma_g = P(G = g)$;
- B is the new emission tensor with $B_{gxy} = P(Y_t = y | X_t = x, G = g)$.

4.2 Modeling Dynamic Context-Biased Transitions (SEQ-T)

In standard HMM, the transition matrix is invariant over time. In real-world applications, this assumption may not hold. The transition probability may change depending on contexts that vary with time. Consider modeling a playlist of songs, where the transitions between genres are captured. The transition probabilities could be influenced by characteristics of the current song (e.g., artist, lyrics and sentiment). A fan of the current artist may break her usual pattern of genre transition and stick to genres by the same artist for the next few songs. As another

example, a user purchasing apparels throughout the year may follow seasonal transitions. If satisfied with certain qualities (e.g., material and style) of past purchases, she may buy more such apparels out of season to secure discounts, breaking the usual seasonal pattern. We call such characteristics *context features*.

It is infeasible to differentiate transition probabilities by individual context features directly, which would blow up the parameter space and thus pose serious computational and data sparsity obstacles. Instead, we propose to model a single context factor that directly influences the next transition. The context factor, being latent, manifests itself through the observable context features.

As illustrated in Fig. 2(b), consider a set of context features $F = \{F^1, F^2, \ldots\}$. As feature values vary over time, let $F_t = \left(F_t^1, F_t^2, \ldots\right)$ denote the feature vector at time t. Each feature F^i takes a set of values $\mathcal{F}^i$, i.e., $F_t^i \in \mathcal{F}^i, \forall i \in \{1, ..., |F|\}, t \in \{1, 2, \ldots\}$. Similarly, let R_t denote the latent context factor at time t, and $\mathcal{R}$ denote the set of context factor levels, i.e., $R_t \in \mathcal{R}, \forall t \in \{1, 2, \ldots\}$. Finally, the model can be specified by the parameters (π, ρ, A, B, C), such that $\forall x, u \in \mathcal{X}, i \in \{1, \ldots, |F|\}, f \in \mathcal{F}_i, t \in \{1, 2, \ldots\}$,

- π and B are the same as in a standard HMM;
- ρ is the distribution of the latent context factor with $\rho_r = P(R_t = r)$;
- C is the feature probability matrix with $C_{rif} = P(F_t^i = f | R_t = r)$;
- A is the new transition tensor with $A_{rxu} = P(X_t = u | X_{t-1} = x, R_{t-1} = r)$.

4.3 Joint Model (SEQ*)

As discussed, user groups and context features can dynamically bias the emission and transition probabilities, respectively. Here, we consider both users and contexts in a joint model, as shown in Fig. 2(c). Accounting for all the parameters defined earlier, the joint model is specified by a six-tuple $\theta = (\pi, \sigma, \rho, A, B, C)$. The algorithm for learning and inference will be discussed in the next section.

5 Learning and Prediction

We now present efficient learning and prediction algorithms for the joint model. Note that the user and context-biased models are only degenerate cases of the joint model— the former assumes one context factor level (i.e., $|\mathcal{R}| = 1$) and no features (i.e., $F = \emptyset$), whereas the latter assumes one user group (i.e., $|\mathcal{G}| = 1$).

5.1 Parameter Learning

The goal of learning is to optimize the parameters $\theta = (\pi, \sigma, \rho, A, B, C)$ through maximum likelihood, given the observed items and features. Consider a sequence of $T > 1$ time steps. Let $\underline{Y} \triangleq (Y_1, \ldots, Y_T)$ as a shorthand; and similarly for $\underline{F}, \underline{X}, \underline{R}$. Subsequently, the optimal parameters can be obtained as follows.

$$\theta^* = \arg\max_\theta \log P(\underline{Y}, \underline{F}; \theta) \tag{3}$$

We demonstrate with one sequence for simpler notations. The algorithm can be trivially extended to enable multiple sequences as briefly described later.

Expectation Maximization (EM). We apply the EM algorithm to solve the above optimization problem. Each iteration consists of two steps below.

– **E-step**. Given parameters θ' from the last iteration (or random ones in the first iteration), calculate the expectation of the log likelihood function:

$$Q(\theta|\theta') = \sum_{\underline{X},G,\underline{R}} P(\underline{X}, G, \underline{R}|\underline{Y}, \underline{F}; \theta') \log P(\underline{Y}, \underline{F}, \underline{X}, G, \underline{R}; \theta') \quad (4)$$

– **M-step**. Update the parametes $\theta = \arg\max_\theta Q(\theta|\theta')$.

Given the graphical model in Fig. 2(c), the joint probability $P(\underline{Y}, \underline{F}, \underline{X}, G, \underline{R})$ can be factorized as

$$P(G)P(X_1) \cdot \prod_{t=1}^{T} \left(P(Y_t|G, X_t)P(R_t) \prod_{i=1}^{|F|} P(F_t^i|R_t) \right) \cdot \prod_{t=1}^{T-1} P(X_{t+1}|X_t, R_t). \quad (5)$$

Maximizing the expectation $Q(\theta|\theta')$ is equivalent to maximize the following, assuming that $Y_t = y_t$ and $F_t^i = f_t^i$ are observed, $\forall t \in \{1, \ldots, T\}, i \in \{1, \ldots, |F|\}$.

$$\begin{aligned}
&\sum_{x\in\mathcal{X}} P(X_1 = x|\underline{Y}, \underline{F}; \theta') \log \pi_x + \sum_{g\in\mathcal{G}} P(G = g|\underline{Y}, \underline{F}; \theta') \log \sigma_g \\
&+ \sum_{t=1}^{T} \sum_{r\in\mathcal{R}} P(R_t = r|\underline{Y}, \underline{F}; \theta') \log \rho_r \\
&+ \sum_{t=1}^{T-1} \sum_{x\in\mathcal{X}} \sum_{u\in\mathcal{X}} \sum_{r\in\mathcal{R}} P(R_t = r, X_t = x, X_{t+1} = u|\underline{Y}, \underline{F}; \theta') \log A_{rxu} \\
&+ \sum_{t=1}^{T} \sum_{x\in\mathcal{X}} \sum_{g\in\mathcal{G}} P(X_t = x, G = g|\underline{Y}, \underline{F}; \theta') \log B_{gxy_t} \\
&+ \sum_{t=1}^{T} \sum_{i=1}^{|F|} \sum_{r\in\mathcal{R}} P(R_t = r|\underline{Y}, \underline{F}; \theta') \log C_{rif_t^i} \quad (6)
\end{aligned}$$

The optimization problem is further constrained by laws of probability, such that $\sum_{x\in\mathcal{X}} \pi_x = 1, \sum_{g\in\mathcal{G}} \sigma_g = 1, \sum_{r\in\mathcal{R}} \rho_r = 1, \sum_{u\in\mathcal{X}} A_{rxu} = 1, \sum_{y\in\mathcal{Y}} B_{gxy} = 1$ and $\sum_{f\in\mathcal{F}^i} C_{rif} = 1$. Applying Lagrange multipliers, we can derive the following updating rules.

$$\begin{aligned}
\pi_x &= \frac{P(X_1 = x|\underline{Y}, \underline{F}; \theta')}{1} = \frac{\sum_{g\in\mathcal{G}} \sum_{r\in\mathcal{R}} \gamma_{gxr}(1)}{1}, \quad (7)\\
\sigma_g &= \frac{P(G = g|\underline{Y}, \underline{F}; \theta')}{1} = \frac{\sum_{x\in\mathcal{X}} \sum_{r\in\mathcal{R}} \gamma_{gxr}(1)}{1}, \\
\rho_r &= \frac{\sum_{t=1}^{T} P(R_t = r|\underline{Y}, \underline{F}; \theta')}{\sum_{t=1}^{T} \sum_{k\in\mathcal{R}} P(R_t = k|\underline{Y}, \underline{F}; \theta')} = \frac{\sum_{g\in\mathcal{G}} \sum_{x\in\mathcal{X}} \sum_{t=1}^{T} \gamma_{gxr}(t)}{T}, \\
A_{rxu} &= \frac{\sum_{t=1}^{T-1} P(R_t = r, X_t = x, X_{t+1} = u|\underline{Y}, \underline{F}; \theta')}{\sum_{t=1}^{T-1} P(R_t = r, X_t = x|\underline{Y}, \underline{F}; \theta')} = \frac{\sum_{t=1}^{T-1} \sum_{g\in\mathcal{G}} \xi_{gxur}(t)}{\sum_{t=1}^{T-1} \sum_{g\in\mathcal{G}} \gamma_{gxr}(t)}, \\
B_{gxy} &= \frac{\sum_{t=1}^{T} P(X_t = x, G = g|\underline{Y}, \underline{F}; \theta') I(y_t = y)}{\sum_{t=1}^{T} P(X_t = x, G = g|\underline{Y}, \underline{F}; \theta')} = \frac{\sum_{t=1}^{T} \sum_{r\in\mathcal{R}} \gamma_{gxr}(t) I(y_t = y)}{\sum_{t=1}^{T} \sum_{r\in\mathcal{R}} \gamma_{gxr}(t)}, \\
C_{rif} &= \frac{\sum_{t=1}^{T} P(R_t = r|\underline{Y}, \underline{F}; \theta') I(f_t^i = f)}{\sum_{t=1}^{T} P(R_t = r|\underline{Y}, \underline{F}; \theta')} = \frac{\sum_{t=1}^{T} \sum_{g\in\mathcal{G}} \sum_{x\in\mathcal{X}} \gamma_{gxr}(t) I(f_t^i = f)}{\sum_{t=1}^{T} \sum_{g\in\mathcal{G}} \sum_{x\in\mathcal{X}} \gamma_{gxr}(t)},
\end{aligned}$$

where $I(\cdot)$ is an indicator function and

$$\gamma_{gxr}(t) \triangleq P(G = g, X_t = x, R_t = r|\underline{Y}, \underline{F}; \theta'), \tag{8}$$

$$\xi_{gxur}(t) \triangleq P(G = g, X_t = x, X_{t+1} = u, R_t = r|\underline{Y}, \underline{F}; \theta'). \tag{9}$$

Note that, to account for multiple sequences, in each updating rule we need to respectively sum up the denominator and numerator over all the sequences.

Inference. To efficiently apply the updating rules, we must solve the inference problems for $\gamma_{gxr}(t)$ and $\xi_{gxur}(t)$ in Eq. 8 and 9. Towards these two goals, similar to the forward-backward algorithm [16] for the standard HMM, we first need to support the efficient computation of the below probabilities.

$$\alpha_{gxr}(t) = P(Y_1, \ldots, Y_t, F_1, ..., F_t, X_t = x, G = g, R_t = r; \theta') \tag{10}$$

$$\beta_{gxr}(t) = P(Y_{t+1}, ..., Y_T, F_{t+1}, ..., F_T|X_t = x, G = g, R_t = r; \theta') \tag{11}$$

Letting $\theta' = (\pi', \sigma', \rho', A', B', C')$ and $C'(r,t) = \prod_{i=1}^{|F|} C'_{rif_t^i}$, both probabilities can be computed recursively, as follows.

$$\alpha_{gxr}(t) = \begin{cases} \pi'_x \sigma'_g \rho'_r C'(r,1) B'_{gxy_1}, & t = 1 \\ \rho'_r C'(r,t) B'_{gxy_t} \sum_{u \in \mathcal{X}} \sum_{k \in \mathcal{R}} \alpha_{guk}(t-1) A'_{kux}, & \text{else} \end{cases} \tag{12}$$

$$\beta_{gxr}(t) = \begin{cases} B'_{gxy_T} C'(r,T), & t = T-1 \\ \sum_{k \in \mathcal{R}} \rho'_k C'(k,t+1) \sum_{u \in \mathcal{X}} B'_{guy_{t+1}} A'_{rxu} \beta_{guk}(t+1), & \text{else} \end{cases} \tag{13}$$

Subsequently, $\gamma_{gxr}(t)$ and $\xi_{gxur}(t)$ can be further computed.

$$\xi_{gxur}(t) = \frac{\alpha_{gxr}(t) A'_{xur} B'_{guy_{t+1}} \sum_{k \in \mathcal{R}} \beta_{guk}(t+1) \rho'_k C'(k,t+1)}{\sum_{h \in \mathcal{G}} \sum_{v \in \mathcal{X}} \sum_{k \in \mathcal{R}} \alpha_{hvk}(T)} \tag{14}$$

$$\gamma_{gxr}(t) = \begin{cases} \sum_{x \in \mathcal{X}} \xi_{gxur}(t) & t = T \\ \sum_{u \in \mathcal{X}} \xi_{gxur}(t) & \text{else} \end{cases} \tag{15}$$

5.2 Item Prediction

Once the parameters are learnt, we can predict the next item of a user given her existing sequence of items $\{Y_1, Y_2, ..., Y_t\}$ and context features $\{F_1, F_2, ..., F_t\}$. In particular, her next item y^* can be chosen by maximum a posteriori estimation:

$$\begin{aligned} y^* &= \arg\max_y P(Y_{t+1} = y|Y_1, \ldots, Y_t, F_1, ..., F_t) \\ &= \arg\max_y P(Y_1, \ldots, Y_t, Y_{t+1} = y, F_1, ..., F_t) \\ &= \arg\max_y P(Y_1, \ldots, Y_t, Y_{t+1} = y, F_1, ..., F_t, F_{t+1})/P(F_{t+1}) \\ &= \arg\max_y \sum_{g \in \mathcal{G}} \sum_{x \in \mathcal{X}} \sum_{r \in \mathcal{R}} \alpha_{gxr}(t+1). \end{aligned} \tag{16}$$

While we do not observe features at time $t+1$, in the above we can adopt any value for F_{t+1} which does not affect the prediction. Instead of picking the best candidate item, we can rank all the candidates and suggest the top-K items.

5.3 Complexity Analysis

We conduct a complexity analysis for learning the joint model SEQ*. Consider one sequence of length T with $|\mathcal{X}|$ states, $|\mathcal{Y}|$ items, $|\mathcal{G}|$ user groups, $|\mathcal{R}|$ context factor levels, $|F|$ features and $|\mathcal{F}|$ values for each feature. For this one sequence, the complexity of one iteration of the EM is contributed by three main steps:

- *Step 1:* Calculate α, β: $O\left(T|\mathcal{G}||\mathcal{X}||\mathcal{R}|^2(|\mathcal{X}|+|F|)\right)$. Because $\rho'_r, C'(r,t)$ in Eq. 12 are independent of g, x, u, k while $\rho'_k, C'(k, t+1)$ in Eq. 13 are independent of g, x, u, r, we can further simplify this to: $O\left(T|\mathcal{R}|(|\mathcal{G}||\mathcal{X}|^2|\mathcal{R}|+|F|)\right)$.
- *Step 2:* Calculate ξ, γ using α, β: $O\left(T|\mathcal{G}||\mathcal{X}|^2|\mathcal{R}|^2|F|\right)$. As $\rho'_k C'(k, t+1)$ in Eq. 14 is independent of g, x, u, r, we reduce it to: $O\left(T|\mathcal{R}|(|\mathcal{G}||\mathcal{X}|^2|\mathcal{R}|+|F|)\right)$.
- *Step 3:* Update θ using γ, ξ: $O\left(T|\mathcal{G}||\mathcal{X}||\mathcal{R}|(|\mathcal{X}|+|F|)\right)$. As y in B_{gxy} of Eq. 7 is independent of g, x, r, we first compute the denominator, and update a normalized score to y in the B_{gxy} while computing the numerator. Likewise, i, f in C_{rif} are independent of g, x, r. Thus, we have: $O\left(T|\mathcal{R}|(|\mathcal{G}||\mathcal{X}|^2+|F|)\right)$.

The overall complexity of SEQ* is $O\left(T|\mathcal{R}|(|\mathcal{G}||\mathcal{X}|^2|\mathcal{R}|+|F|)\right)$ for one sequence, one iteration. The complexities of lesser models are (by substitution):

- HMM with $|\mathcal{G}|=|\mathcal{R}|=1, |F|=|\mathcal{F}|=0$: $O\left(T|\mathcal{X}|^2\right)$
- SEQ-E with $|\mathcal{R}|=1, |F|=|\mathcal{F}|=0$: $O\left(T|\mathcal{G}||\mathcal{X}|^2\right)$
- SEQ-T with $|\mathcal{G}|=1$: $O\left(T|\mathcal{R}|(|\mathcal{X}|^2|\mathcal{R}|+|F|)\right)$

The result implies that the running times of our proposed models are quadratic in the number of states and context factor levels, while linear in all the other variables. HMM is also quadratic in the number of states. Comparing to HMM with the same number of states, our joint model incurs a quadratic increase in complexity only in the number of context factor levels (which is typically small), and merely a linear increase in the number of groups and context features.

6 Experiments

The objective of experiments is to evaluate *effectiveness*. We first look into a synthetic dataset to investigate whether context-biased transition and user-biased emission could have been simulated by increasing the number of HMM's states. Next, we experiment with two real-life, publicly available datasets, to investigate whether the models result in significant improvements over the baseline.

6.1 Setup

We elaborate on the general setup here, and describe the specifics of each dataset later in the appropriate sections. Each dataset has of a set of sequences. We create random splits of 80:20 ratio of training versus testing. In this sequential preference setting, a sequence (a user) is in either training or testing, but not necessarily in both. This is different from a fully personalized ordinal preference

setting (a different framework altogether), where a user would be represented in both sets.

Task. For each sequence in the testing set, given the sequence save the last item, we seek to predict the last item. Each method generates a top-K recommendation, which is evaluated against the held-out ground-truth last item.

Comparative Methods. Since we build our dynamic context and user factors upon HMM, it is the most appropriate baseline. To investigate the contribution of user-biased emission and context-based transition *separately*, we compare the two models SEQ-E and SEQ-T respectively against the baseline. To see their contributions *jointly*, we further compare SEQ* against the baseline. In addition, we include the result of the frequency-based method FREQ as a reference, which simply choose the most popular item in the training data.

Metrics. We rely on two conventional metrics for top-K recommendation. Inspired by a similar evaluation task in [24], the first metric we use is *Recall@K*.

$$Recall@K = \frac{\text{number of sequences with the ground truth item in the top } K}{\text{total number of sequences in the testing set}}$$

If we assume the ground truth item to be the only true answer, average precision can be measured similarly (dividing by K) and would show the same trend as recall. In the experiments, we primarily study top 1% recommendation, i.e., *Recall@1%*, but will present results for several other K's as well. Actually, it is not clear that the other items in the top-K would really be rejected by a user [24]. Instead of precision, we rely on another metric.

The second metric is Mean Reciprocal Rank or *MRR*, defined as follows.

$$MRR = \frac{1}{|S_{\text{test}}|} \times \sum_{s \in S_{\text{test}}} \frac{1}{\text{rank of target item for sequence } s}$$

We prefer a method that places the ground-truth item higher in the top-K recommendation list. Because the contribution of a very low rank is vanishingly small, we cut the list off at 200, i.e., ranks ≥ 200 contribute zero to *MRR*. Realistically, a recommendation list longer than 200 is unlikely in realistic scenarios.

For each dataset, we create five random training/testing splits. For each "fold", we run the models ten times with different random initializations (but with common seeds across comparative methods for parity). For each method, we average the *Recall@K* and *MRR* across the fifty readings. All comparisons are verified by one-sided paired-sample Student's t-test at 0.05 significance level.

6.2 Synthetic Dataset

We begin with experiments on a synthetic dataset, for two reasons. First, one advantage of a synthetic dataset is the knowledge of the actual parameters (e.g., transition and emission probabilities), which allows us to verify our model's ability to recover these parameters. Second, we seek to verify whether the effects

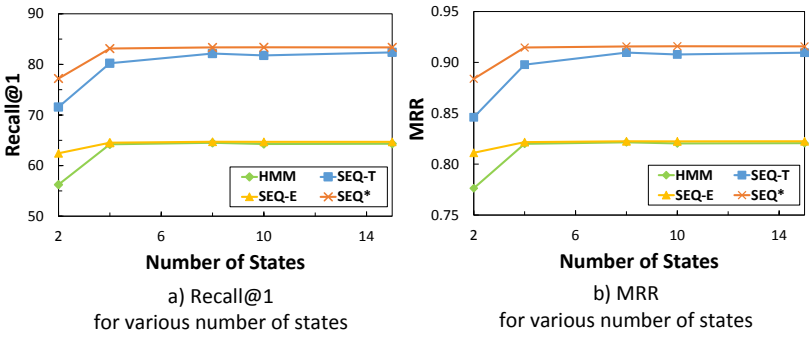

Fig. 3. Performance of comparative methods on Synthetic Data for *Recall@1* and *MRR*

of context-biased transition and user-biased emission could have been simulated by increasing the number of hidden states of traditional sequence model HMM.

Dataset. We define a synthetic dataset with the following configuration: 2 groups ($|\mathcal{G}| = 2$), 2 states ($|\mathcal{X}| = 2$), 2 context factor levels ($|\mathcal{R}| = 2$), 4 items ($|\mathcal{Y}| = 4$), 4 features ($|F| = 4$) each with 2 feature values (present or absent).

Here, we discuss the key ideas. A six-tuple $\theta = (\pi, \sigma, \rho, A, B, C)$ is specified as follows: $\pi = [0.8, 0.2]$, $\sigma = [0.9, 0.1]$, $\rho = [0.3, 0.7]$. The transition tensor A is such that we induce self-transition to the same state for the first context factor level, and switching to the other state for the second context factor level. The emission tensor B is such that the four (state, group) combinations each tend to generate one of the four items. The feature matrix C is such that each context factor level is mainly associated with two of the four features.

We then generate 10 thousand sequences, each of length 10 ($T = 10$). For each sequence, we first draw a group according to σ. At time $t = 1$, we draw the first hidden state X_1 from π, followed by drawing the first item Y_1 from B. We also draw a context factor level from ρ and generate features via C. For time $t = 2, \ldots, 10$, we follow the same process, but each hidden state is now drawn from A according to the previous state and context factor level at time $t - 1$.

Results. We run the four comparative methods on this synthetic dataset, fixing the context factor levels and groups to 2 for the relevant methods, while varying the number of states. Fig. 3(a) shows the results in terms of *Recall@1*, i.e., the ability of each method in recommending the ground truth item as the top prediction. There are several crucial observations. *First*, the proposed model SEQ* outperforms the rest, attaining recall close to 85%, while the baseline HMM hovers around 65%. SEQ* also outperforms SEQ-T and SEQ-E.

Second, as we increase the number of states, most models initially increase in performance and then converge. Evidently, increasing the number of states alone does not lift the baseline HMM to the same level of performance as SEQ* or SEQ-T, indicating the effect of context-biased transition. Meanwhile, though SEQ-E and HMM are similar (due to inability to model context factor), SEQ* is

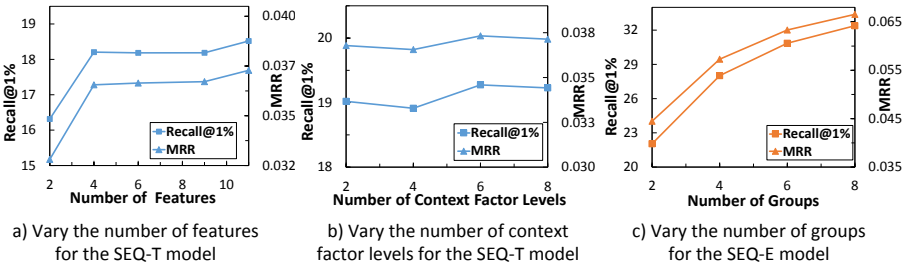

Fig. 4. Effects of features, context factor on SEQ-T & groups on SEQ-E on Yes.com

slightly better than SEQ-T, indicating the contribution of user-biased emission. Fig. 3(b) shows the results for *MRR*, showing similar trends and observations.

6.3 Real-Life Datasets

We now investigate the performance of the comparative methods on real-life, publicly available datasets covering two different domains: song playlists from online radio station Yes.com, and hashtag sequences from users' Twitter streams.

Playlists from Yes.com. We utilize the *yes_small* dataset[3] collected by [4]. The dataset includes about 430 thousand playlists, involving 3168 songs. Noticeably, the majority of playlits has length which is shorter than 30. To keep the playlist lengths relatively balanced, we filter out playlists with fewer than two songs and retain up to the first thirty songs in each playlist. Finally, we have 250 thousand playlists (sequences) consisting of 3168 unique songs (items).

Features. We study the effect of features on the context-biased transition model SEQ-T. Each song may have tags. There are 250 unique tags. We group tags with similar meanings (e.g.,"male vocals" and "male vocalist"). As the first feature, we use a binary feature of whether the current song and the previous song shares at least one tag. For additional features, we use the most popular tags. Note that we never assume knowledge of the tags of the song to be predicted. Fig. 4(a) shows the performance of SEQ-T, with two context factor levels, for various number of features. Fig. 4(a) has dual vertical axes for *Recall@1%* (left) and *MRR* (right) respectively. The trends for both metrics are similar: performance initially goes up and then stabilizes. In subsequent experiments, we use eleven features (similarity feature and ten most popular tags).

Context Factor. We then vary the number of context factor levels of SEQ-T (with eleven features). Fig. 4(b) shows that for this dataset, there is not much gain from increasing the number of context factor levels beyond two. Therefore, for greater efficiency, subsequently we experiment with two context factor levels.

Latent Groups. We turn to the effect of latent groups on the user-biased emission model SEQ-E. Fig. 4(c) shows the effect of increasing latent groups. More

[3] `http://www.cs.cornell.edu/~shuochen/lme/data_page.html`

Table 1. Performance of comparative methods on Yes.com for *Recall@K*

		FREQ	HMM	SEQ-T	SEQ-E	SEQ*	Imp.
5 States	Recall@1%	6.8	13.8	$18.4^{\dagger}$	$22.0^{\S}$	$24.1^{\dagger\S}$	+10.3
	Recall@50	9.6	19.2	$25.1^{\dagger}$	$29.5^{\S}$	$32.1^{\dagger\S}$	+13.0
	Recall@100	16.2	29.3	$37.0^{\dagger}$	$42.6^{\S}$	$46.1^{\dagger\S}$	+16.8
10 States	Recall@1%	6.8	22.3	$23.2^{\dagger}$	$27.8^{\S}$	$28.6^{\dagger\S}$	+6.3
	Recall@50	9.6	30.0	$31.1^{\dagger}$	$36.9^{\S}$	$38.1^{\dagger\S}$	+8.1
	Recall@100	16.2	43.4	$44.9^{\dagger}$	$52.1^{\S}$	$53.5^{\dagger\S}$	+10.2
15 States	Recall@1%	6.8	26.1	$26.5^{\dagger}$	$30.1^{\S}$	$30.6^{\dagger\S}$	+4.5
	Recall@50	9.6	34.7	$35.5^{\dagger}$	$39.4^{\S}$	$40.2^{\dagger\S}$	+5.5
	Recall@100	16.2	49.3	$50.8^{\dagger}$	$55.1^{\S}$	$56.3^{\dagger\S}$	+7.0

Table 2. Performance of comparative methods on Yes.com for *MRR*

	FREQ	HMM	SEQ-T	SEQ-E	SEQ*	Imp.
5 States	0.014	0.028	$0.037^{\dagger}$	$0.044^{\S}$	$0.049^{\dagger\S}$	+0.021
10 States	0.014	0.045	$0.047^{\dagger}$	$0.057^{\S}$	$0.059^{\dagger\S}$	+0.014
15 States	0.014	0.053	$0.054^{\dagger}$	$0.062^{\S}$	$0.063^{\S}$	+0.009

groups lead to better performance. Because of the diversity among sequences, having more groups increases the flexibility in modeling emissions while still sharing transitions. For the subsequent comparison to the baseline, we will experiment with two latent groups, as the earlier comparison has shown that the results with higher number of groups would be even higher.

Comparison to Baseline. We now compare the proposed models SEQ-T, SEQ-E, and SEQ* to the baseline HMM. Table 1 shows a comparison in terms of *Recall@K* for 5, 10, and 15 states. In addition to *Recall@1%* (corresponding to top 31), we also show results for *Recall@50* and *Recall@100*. The symbol † denotes statistical significance due to the effect of context-biased transition. In other words, the outperformance of SEQ-T over HMM, and that of SEQ* over SEQ-E, are significant. The symbol § denotes statistical significance due to the effect of user-biased emission, i.e., the outperformance of SEQ-E over HMM, and that of SEQ* over SEQ-T, are significant. Finally, our overall model SEQ* is significantly better than the baseline HMM in all cases. The absolute improvement of the former over the latter in additional percentage terms is shown in the *Imp.* column. For all models, more states generally translate to better performance, and the improvements are somewhat smaller but still significant. Table 2 shows a comparison in terms of *MRR*, where similar observations hold.

Hashtag Sequences from Twitter.com. We conduct similar experiments on the Twitter dataset[4] [12]. There are 130 thousand users. In our scenario, each sequence corresponds to the hashtags of a user. The average length of our dataset

[4] `https://wiki.cites.illinois.edu/wiki/display/forward/Dataset-UDI-TwitterCrawl-Aug2012`

Table 3. Performance of comparative methods on Twitter.com for *Recall@K*

		FREQ	HMM	SEQ-T	SEQ-E	SEQ*	Imp.
5 States	Recall@1%	8.4	16.9	$17.1^{\dagger}$	$20.6^{\S}$	$21.0^{\dagger\S}$	+4.1
	Recall@50	16.1	28.3	$28.6^{\dagger}$	$33.2^{\S}$	$33.7^{\dagger\S}$	+5.4
	Recall@100	25.5	40.6	$40.9^{\dagger}$	$46.0^{\S}$	$46.5^{\dagger\S}$	+5.9
10 States	Recall@1%	8.4	21.8	$22.0^{\dagger}$	$26.5^{\S}$	$26.9^{\dagger\S}$	+5.1
	Recall@50	16.1	34.2	$34.4^{\dagger}$	$39.4^{\S}$	$39.8^{\dagger\S}$	+5.7
	Recall@100	25.5	47.2	$47.4^{\dagger}$	$52.0^{\S}$	$52.4^{\S}$	+5.2
15 States	Recall@1%	8.4	25.2	$25.3^{\dagger}$	$29.9^{\S}$	$30.0^{\dagger\S}$	+4.8
	Recall@50	16.1	38.1	$38.2^{\dagger}$	$43.1^{\S}$	$43.3^{\dagger\S}$	+5.1
	Recall@100	25.5	51.2	$51.3^{\dagger}$	$55.2^{\S}$	$55.3^{\dagger\S}$	+4.1

Table 4. Performance of comparative methods on Twitter.com for *MRR*

	FREQ	HMM	SEQ-T	SEQ-E	SEQ*	Imp.
5 States	0.019	0.045	$0.046^{\dagger}$	$0.062^{\S}$	$0.063^{\dagger\S}$	+0.0183
10 States	0.019	0.063	0.064	$0.084^{\S}$	$0.086^{\dagger\S}$	+0.0227
15 States	0.019	0.076	$0.078^{\dagger}$	$0.100^{\S}$	$0.101^{\dagger\S}$	+0.0246

is 19. If a tweet has multiple hashtags, we retain the most popular one, so as to maintain the sequence among tweets. Similarly to the treatment of stop words and infrequent words in document modeling, we filter out hashtags that are too popular (frequency ≥ 25000) or relatively infrequent (frequency ≤ 1000). Finally, we obtain 114 thousand sequences involving 2121 unique hashtags. Similarly to Yes.com, we run the models for two levels of context factor and two latent groups, but with seven features extracted from the tweet of the current hashtag (not the one to be predicted): number of retweets, number of hashtags, time intervals to the previous one and two tweets, time interval to the next tweet, and edit distances with the previous one and two observations.

The task is essentially predicting the next hashtag in a sequence. In brief, Tables 3 and 4 support that the improvements due to context-biased transition (†) and user-biased emission (§) are mostly significant. Importantly, the overall improvements by SEQ* over the baseline HMM (*Imp.* column) are consistent and hold up across 5, 10, and 15 states for both *Recall@K* and *MRR*.

Computational efficiency is not the main focus of experiments. We comment briefly on the running times. For the Twitter dataset, the average learning time per iteration on Intel Xeon CPU X5460 3.16GHz with 32GB RAM for our models with 15 states, 2 groups, 2 context factor levels are 2, 3, and 6 minutes for SEQ-E, SEQ-T and SEQ* respectively. HMM requires less than a minute.

7 Conclusion

In this work, we develop a generative model for sequences, which models two types of dynamic factors. First, transition from one state to the next may be affected by context factor. This results in SEQ-T model, with context-biased transition. Second, we seek to incorporate how different latent user groups may have preferences for certain items. This results in SEQ-E model, with user-biased emission. Finally, we unify these two factors into a joint model SEQ*. Experiments on both synthetic and real-life datasets support the case that these dynamic factors contribute towards better performance than the baseline HMM (statistically significant) in terms of top-K recommendation for sequences.

Acknowledgments. This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its NRF Fellowship Programme (Award No. NRF-NRFF2016-07).

References

1. Adomavicius, G. and Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), pp.734-749 (2005)
2. Brafman, R.I., Heckerman, D. and Shani, G.: Recommendation as a Stochastic Sequential Decision Problem, Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS), pp. 164-173 (2003)
3. Chen, J., Wang, C. and Wang, J.: A Personalized Interest-Forgetting Markov Model for Recommendations. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp. 16-22 (2015)
4. Chen, S., Moore, J.L., Turnbull, D. and Joachims, T.: Playlist prediction via metric embedding. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 714-722 (2012)
5. Chen, W., Hsu, W. and Lee, M. L.: Modeling user's receptiveness over time for recommendation. In Proceedings of the ACM SIGIR Conference (SIGIR), pp. 373-382 (2013)
6. Cheng, C., Yang, H., Lyu, M.R. and King, I.: Where You Like to Go Next: Successive Point-of-Interest Recommendation. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) (2013)
7. Dupont, P., Denis, F. and Esposito, Y.: Links between probabilistic automata and hidden Markov models: probability distributions, learning models and induction algorithms. Pattern Recognition, 38(9), pp.1349-1371 (2005)
8. Hofmann, T.: Probabilistic latent semantic analysis. In Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), pp. 289-296 (1999)
9. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Transactions on Information Systems (TOIS), pp.89-115 (2004)
10. Jiang, P., Zhu, Y., Zhang, Y. and Yuan, Q.: Life-stage Prediction for Product Recommendation in E-commerce. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 1879-1888 (2015)
11. Koren, Y., Bell, R. and Volinsky, C.: Matrix factorization techniques for recommender systems. Computer, (8), pp.30-37 (2009)

12. Li, R., Wang, S., Deng, H., Wang, R. and Chang, K.C.C.: Towards social user profiling: unified and discriminative influence model for inferring home locations. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 1023-1031. (2012).
13. Liu, X., Liu, Y., Aberer, K. and Miao, C.: Personalized point-of-interest recommendation by mining users' preference transition. In Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), pp. 733-738 (2013)
14. Mikolov, T., Karafit, M., Burget, L., Cernock, J. and Khudanpur, S.: Recurrent neural network based language model. In INTERSPEECH, (2), p. 3 (2010)
15. Parameswaran, A.G., Koutrika, G., Bercovitz, B. and Garcia-Molina, H.: Recsplorer: recommendation algorithms based on precedence mining. In Proceedings of the International Conference on Management of Data (SIGMOD), pp. 87-98 (2010)
16. Rabiner, L. R. and Juang, B. H.: An introduction to hidden Markov models. IEEE ASSP Magazine, 3(1), pp. 4-16 (1986)
17. Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), pp. 452-461 (2009)
18. Rendle, S., Freudenthaler, C. and Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In Proceedings of the International World Wide Web Conference (WWW), pp. 811-820 (2010)
19. Sahoo N., Singh P.V. and Mukhopadhyay T.: A hidden Markov model for collaborative filtering. MIS Quarterly, 36(4), (2012)
20. Salakhutdinov, R., and Andriy, M. : Probabilistic matrix factorization. In Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS), 21 (2008)
21. Salakhutdinov, R., Andriy, M. and Hinton, G.: Restricted Boltzmann machines for collaborative filtering. In Proceedings of the International Conference on Machine Learning (ICML), pp. 791–798 (2007)
22. Shani, G., Brafman, R.I. and Heckerman, D.: An MDP-based recommender system. In Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), pp. 453-460 (2002)
23. Tavakol, M. and Brefeld, U.: Factored MDPs for detecting topics of user sessions. In Proceedings of the ACM Conference on Recommender Systems (RecSys), pp. 33-40 (2014)
24. Wang, C. and Blei, D. M.: Collaborative topic modeling for recommending scientific articles. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 448-456 (2011)
25. Wang, J., Zhang, Y., Posse, C. and Bhasin, A.: Is it time for a career switch?. In Proceedings of the International World Wide Web Conference (WWW), pp. 1377-1388 (2013)
26. Wang, P., Guo, J., Lan, Y., Xu, J., Wan, S. and Cheng, X.: Learning Hierarchical Representation Model for Next Basket Recommendation. In Proceedings of the ACM SIGIR Conference (SIGIR), pp. 403-412 (2015)
27. Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q. and Sun, J.: Temporal recommendation on graphs via long-and short-term preference fusion. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 723-732 (2010)
28. Yang, J., McAuley, J., Leskovec, J., LePendu, P. and Shah, N.: Finding progression stages in time-evolving event sequences. In Proceedings of the International World Wide Web Conference (WWW), pp. 783-794 (2014)