IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 25, 2019, accepted December 20, 2019, date of publication December 26, 2019, date of current version January 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962539

A Hybrid Generative Model for Online User Behavior Prediction

MINH-DUC NGUYEN[1], (Member, IEEE), AND YOON-SIK CHO[2], (Member, IEEE)
[1]Department of Software Convergence, Sejong University, Seoul 05006, South Korea
[2]Department of Data Science, Sejong University, Seoul 05006, South Korea

Corresponding author: Yoon-Sik Cho (yscho@sejong.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant 2018R1C1B5045931.

ABSTRACT With the increase of rich datasets from various online platforms, predicting user behavior has been one of the most active research topics. The user behavior on these online platforms includes listening to music, watching videos, purchasing products, checking-in to places, and joining online sub-communities. Predicting online user behavior is an important challenge for various applications. Personalization, recommendation systems, target advertisements are based on user behavior prediction, where user's next purchases or actions need to be predicted. In this paper, we propose a hybrid generative model that can predict user behavior considering multiple factors. While previous work has been focused on two aspects individually: predicting repeat behavior or predicting new behavior, our model considers both aspects simultaneously during the learning process. The user-specific preference component is used to capture repeat behavior patterns, while the latent group preference component is used to discover new behavior. Besides these two components, we also consider the exogenous effect, which is not captured in the former two. Our experimental results on real-world datasets show how our proposed model outperforms the state-of-the-art model.

INDEX TERMS Online user behavior prediction, topic modeling, latent Dirichlet allocation, mixture model, generative model.

I. INTRODUCTION

Predicting online user behavior is crucial in various online applications, including recommendation systems, online target advertising, and personalization systems, to name a few. These applications are generally associated with predictive mechanisms that predict the future behavior of users, such as purchasing products, checking-in to place, or selecting songs and videos. In this sense, predicting behavior is similar to predicting or recommending consumed items for users. Thus, we use both "behavior" and "item" terms in this paper without any difference in the meaning.

With the growth and intense competition in e-commerce and online services, online user behavior prediction has become an active research field. Various techniques have been applied to tackle this problem, such as Matrix factorization [1], [2], Latent Dirichlet Allocation [3]–[5], Markov model [6]–[9], Recurrent neural network [10]–[12] and so on. Those techniques usually focus on predicting behavior which user have not performed (new behavior) or behavior which user has done in the past (repeat behavior). The importance of predicting repeat behavior has been discussed in several past research [13], [14]. Other authors [15] focused on new behavior and indicated that the quality and usefulness of recommendations depend on the novelty of suggested items. For instance, recommending unseen movies for users is more useful than suggesting movies that users have watched before. As such, we believe that it is crucial to consider both types of behavior to understand online user behavior fully.

In this paper, we propose a hybrid generative model that considers both aspects of user behavior based on Latent Dirichlet Allocation (LDA). We named our model as "generative" model because the prediction process in our model is similar to the generative process. LDA is a generative statistical model which is originally used in text analysis to discover the hidden topic in the collection of documents [16]. Although LDA is introduced for text analysis, several researchers [3], [5], [17], [18] have employed this technique in recommendation tasks and achieved considerable results. Besides, this method was chosen as a basis for our model

The associate editor coordinating the review of this manuscript and approving it for publication was Manik Sharma.

development because of two features that it demonstrates: (1) By using LDA, our model can explore new behavior, and (2) Combining LDA with other components is feasible. Throughout this paper, we treat the inferred ‘‘topics’’ in LDA as ‘‘latent group preference’’.

Apart from LDA, we use another component to exploit the behavior history pattern by considering the specific distribution of item consumption of each user, which we name as ‘‘user-specific preference’’. Finally, we add one component to capture the exogenous factor, or, in this case, ‘‘popularity’’, which can affect the user behavior that is unexplained by the two.

Our primary contributions can be summarized as follows:

- We propose a generative mixture model for online user behavior prediction.
- We consider three major factors of user behavior: 1) latent group preference, 2) user-specific preference, 3) exogenous effect.
- We test our model on real-world datasets and show compelling results compared to the state-of-the-art model.

The remainder of this article is structured as follows. In Section II, we provide a summary of related work. Section III introduces our model, followed by Section IV with experimental evaluations. Finally, Section V presents our conclusions.

II. RELATED WORK

In this section, we briefly present some of the research literature related to predicting online user behavior problems. These studies can be divided into two different groups: studies focusing on new behavior and studies focusing on repeat behavior. Both types of behavior should be considered to build an effective recommendation system.

A. NEW BEHAVIOR PREDICTION

Researches on predicting new behavior concentrate on behavior that users have not performed in the past. Authors in [15] show that suggesting useful novel items for users will decide the success of the recommendation system. Several approaches have been proposed to tackle this problem. Content-based recommendation is a conventional and traditional approach for suggesting new items [19]. Systems based on this approach examine the properties of items or preferences of users to make suitable predictions, such as text content of an article, lyric of a song, characteristic of a user, etc. However, this information is not fully available on every platform, which challenges the application of this approach. Thus, collaborative filtering is a more suitable technique for discovering new items without detailed content.

Collaborative filtering [20] approach requires the value of rating or co-occurrence between users and items, which can be conveniently obtained in different platforms. When applying this approach, user behavior is summarized into a user-item matrix, where the item represents various online behaviors, including songs listened to, places visited, purchased products, or even selected tags when expressing themselves. The user-item matrix represents the collection of users' past histories, and are fed into the model for inference. Lower dimensional representation [21] is a widely-used technique for this approach, which enables two important advantages: This technique (1) works with sparse data, which is common in online user behavior datasets, and (2) allows us to discover the *latent factor* in the data. The term *latent factor* means that this factor cannot be observed directly, but can be inferred from the datasets. Applying lower-dimensional representation to the user-item matrix, we obtain two low-rank matrices of user and item in terms of this latent factor. From these matrices, the hidden relationship between users and items can be extracted and be used to predict the new consumption.

Matrix Factorization (MF) is a successful method using lower-dimensional representation for recommendation system [1]. This method became well-known during the Netflix Prize,[1] a competition organized by Netflix to improve the movies recommendation system. Probabilistic Matrix Factorization [22] and Non-negative matrix factorization [23] are two MF-based model which have been employed in several recommendation studies in different applications, such as location-based recommendation [24], [25], user interests prediction [26] or online rating prediction [2].

Topic modeling is a type of statistical model which employs the idea of lower-dimensional matrices to capture the latent relationships inside the dataset. Originally, topic modeling is used in text analysis to discover the hidden topics in the collection of documents. Topic modeling techniques are based on the assumption that each document is a mixture of topics, and each topic is a mixture of words. pLSA [27] and LDA [16] are two popular methods for topic modeling. In general, pLSA is resembled LDA, except that LDA uses the Dirichlet prior to its distributions, leading to better distributions. Several studies show that topic modeling can be applied in user behavior prediction problem. Authors in [4] used LDA to model the web browsing/application activities of mobile device users, which allows telecommunication providers to understand the interests of users for future advertisements or content recommendation. Authors in [28] applied LDA on the purchase history of users on an online shopping service to suggest products for users. Authors in [29] proposed a Twitter followee recommendation algorithm based on LDA. A recommendation model for different types of dataset employing LDA has been developed by [5].

B. REPEAT BEHAVIOR PREDICTION

Predicting what behavior/consumption will be repeated by users in the future is an important task in recommendation problem, which is discussed in several studies [13], [14]. Research on repeat behavior aim to find behavior patterns from the history of each individual to predict the future. An early work that analyzes the Web re-visitation patterns is conducted by [30] to improve the navigation experiences.

[1] https://www.netflixprize.com/

Inspired by this research, repeat consumption analysis has been applied to many other domains, including music listening [31], video watching [32], location visiting [33].

Techniques for this approach are various. Authors in [34], [35] exploited behavior patterns by calculating the probability of each item in the consumption history of users, and then use the probabilities distribution to predict the future. Other studies [6]–[8], [11], [12], [14], [35]–[37] consider temporal information as a significant feature to capture the pattern. Markov models [] and recurrent neural network [38] are two wide-used methods for finding behavior patterns based on temporal features.

C. COMBINATION OF NEW AND REPEAT BEHAVIOR PREDICTION

In common, most users tend to perform both new and repeat behavior, which means recommendation systems must suggest both types of behavior for users. A combination model that can predict both new and repeat behavior is demanded to improve the user experience in online services. In fact, there are some works trying to integrate different models into one to solve the problem. Multinomial mixture model [34] is one of the recent works on user behavior prediction in which the personal history of item consumption is combined with broader global population preference. The personal consumption history component accounts for predicting repeat consumption, while the other component affects the prediction of novel consumption. Personalized Location Models [33] is another mixture model that combines the location history of users and additional information such as population pattern, geographic constraints, and social context to predict the future location of users.

We extend this line of work by proposing a hybrid generative model based on LDA, a topic modeling technique. The two other components we extend on LDA in our model are *history consumption of each user* and *global population preference/items popularity*. Extending the LDA allows us to overcome the limitation of Multinomial mixture model [34] in predicting novel items. Multinomial mixture model only uses popularity to predict new items, leading to the same suggestions across all users, which is not practical in real-world applications. By integrating LDA, our model can discover the hidden preference of users, which is specific for each user. Then, our model uses this hidden preference to recommend suitable items for users with a high personalization. The combination of three components helps our model to achieve better performance compared to Multinomial mixture model.

III. PROBLEM FORMULATION

Consider a user-item matrix, where a row corresponds to a user, and a column corresponds to each. $A(u, v)$ reflects the occurrence frequency of behavior taken by user $u \in \{1, \ldots, U\}$ on item $v \in \{1, \ldots, V\}$ within the period-of-interest. The item could be the songs the users have listened to, the places users have visited, or the sub-communities users are following. The occurrence matrix has been actively used in text mining literature as well, where each row and column corresponds to document and word, respectively. LSA decomposes the occurrence matrix (or the document-term matrix) into three matrices, consisting of mapping of documents to a lower-dimensional vector space of the latent semantic space. Similarly, LDA uses the document-term matrix to infers the topic distribution of each document. When LDA is applied to the user-behavior matrix, the "topic" distribution of users can be inferred, allowing lower-dimensional representations for each user. Throughout the paper, we use the "topic" terminology from topic modeling [16] and refer to the user's latent profiles that affect user behavior.

LDA has been widely used in various areas with only small changes to the corresponding inference algorithms [39]. One of the many other disciplines besides text analysis is the probabilistic recommendation models [40], where the items can be predicted through inferred "topics". The lower-dimensional representation of the occurrence matrix can further improve the performance of predictions of unseen items. Our model is based on this idea, where we assume each user has its topic distribution, which we name as latent group preference. We further consider two other factors other than topic-preference: 1. user-specific preference, and 2. popularity. By following the same experimental set-up [34], the dataset is divided into training set and test set based on the given time point. The details of the experimental set up are provided in Section IV.

A. TOPIC MODELING

In this section, we briefly describe the fundamentals of LDA and how it can be applied in predicting future behavior problem.

LDA is a generative probabilistic model for text corpora and other collections of discrete data. The basic idea of LDA is that each document of a collection is represented as a finite mixture over underlying topics, while each topic is modeled by a distribution over words. For each topic, words with the highest probabilities can give us the idea of what the topic is.

In LDA, a corpus is a collection of M documents, denoted by D = $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$. A document contains a sequence of N words, denoted by $\mathbf{w} = (w_1, w_2, \ldots, w_N)$. A word is an item from a vocabulary indexed by $\{1, \ldots, V\}$. The number of topics K is identified and fixed in the corpus. For each document, LDA assumes that it can be modeled according to the following generative process:

1. Choose the number of word N from a Poisson distribution

2. Choose a multinomial distribution $\boldsymbol{\theta}$ from a Dirichlet distribution with parameter $\boldsymbol{\alpha}$

3. For each of the N words w_n:

(a) Choose a topic indicator $\mathbf{z}_n$ from Multinomial distribution with parameter $\boldsymbol{\theta}$

(b) Choose a word w_n from $p(w_n|\mathbf{z}_n, \boldsymbol{\beta})$, a multinomial probability conditioned on the topic $\mathbf{z}_n$

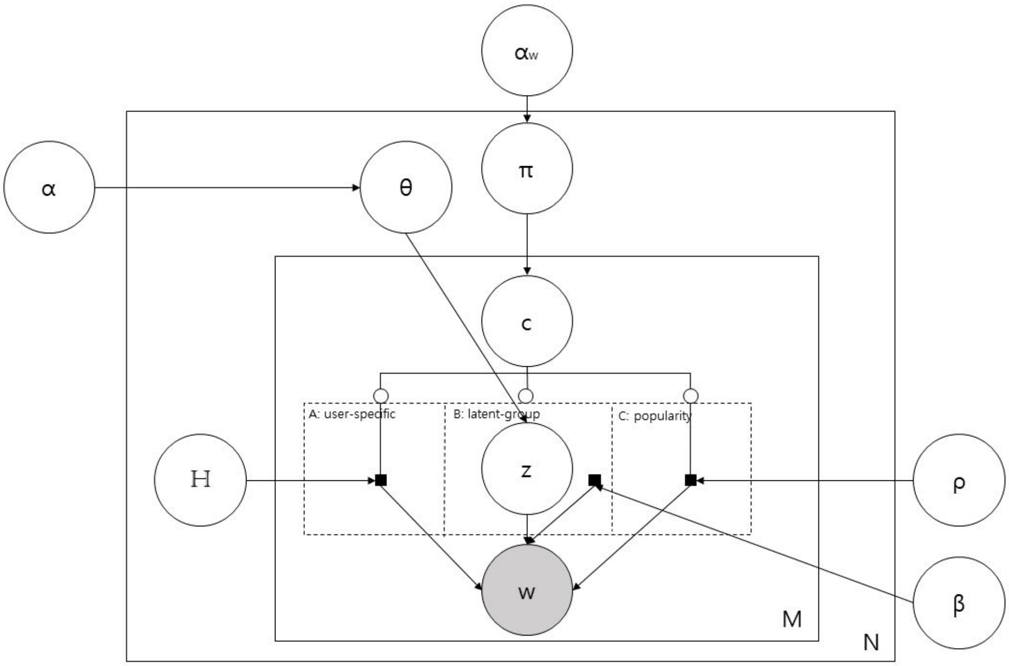


FIGURE 1. The generative process of our model. *c* is the selector variable which determines one of the gates of A, B, C expressed in the dotted rectangle. The observed variable is expressed in the shaded circle.

In online user behavior prediction, we consider a word as an item, such as a song, a location, or a product, while a document corresponds to a user. In this aspect, the behavior of a user is represented as a collection of selected items. The dataset of user behavior is a collection of discrete data, which is suitable for applying LDA. LDA, as mention above, allows us to discover the underlying relationships between users and items. By utilizing the underlying relationships, one can even predict the potential unseen behavior for each user, which hasn't been seen in the past.

LDA can predict the unseen words for each document based on the inferred topic distribution and the estimated parameters. The topic-to-word probability $\boldsymbol{\beta}$ represents the probability of each word's occurrence for each topic. By combining the inferred topic mixture distribution with the parameter, the probability of observing word i in document d is computed as follows:

$$p(w_i|\mathbf{w}_d) = \sum_k p(w_i|\mathbf{z}_k)p(\mathbf{z}_k|\mathbf{w}_d) \quad (1)$$

This occurrence probability of words in a document corresponds to the probability of items that a user takes, which can be used to predict the potential behaviors in the future. This show that LDA can be applied in behaviors prediction problem. However, LDA underperforms when users behave far from each other, where LDA tries to find similar patterns from users within a group. The following section addresses the current limitation of LDA for behavior modeling and introduces a model that well combines the user-specific preference and the community preference from similar users.

B. HYBRID GENERATIVE MODEL FOR ONLINE USER BEHAVIOR PREDICTION

The hybrid generative model we propose in this article considers three major factors that affect user behavior. 1. User-specific preference 2. Latent group preference 3. popularity, or the exogenous effect. Our generative process consists of two-stages, where the first stage determines the underlying component of the three, and the second stage is deterministic to the component from the first stage. Figure 1 illustrates the generative process of our model. The generative process can be summarized as follows:

- **Initialization**
  1) For each user $u \in \mathcal{U}$, sample the 3×1 weight vector, and sample the $K \times 1$ topic distribution.
     $\boldsymbol{\pi}_u \sim \text{Dirichlet}(\boldsymbol{\alpha_w})$,
     $\boldsymbol{\theta}_u \sim \text{Dirichlet}(\boldsymbol{\alpha})$.
  2) For each user u, initialize an empty multiset of indicator variables, $\mathbb{H}_u = \emptyset$, which will be used for user-specific preference in Stage 2.
  3) For each topic k, sample the behavior distribution $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\boldsymbol{\kappa})$, where κ is a hyperparameter.
  4) Sample the popularity distribution
     $\boldsymbol{\rho} \sim \text{Dirichlet}(\boldsymbol{\kappa_p})$.
- **Generating Behavior**
  Let M_u be the total number of selections[2] of user u from a (behavior) set $\mathcal{V} = \{1, \ldots, V\}$.
  1) **Stage 1:** For each m-th selection of user u, sample

[2]For the purposes of data generation M_u can be sampled from, say, a Poisson distribution. This is not relevant for inference, however, where M_u is specified in the data.

the underlying component from the mixture distribution $\boldsymbol{\pi}_u$,
$\mathbf{c}_u^m \sim \text{Multinomial}(\boldsymbol{\pi}_u)$, where $m \in \{1, \ldots, M_u\}$, and the $\mathbf{c}_u^m$ determines one of the three cases in Stage 2. $\mathbf{c}_u^m$ is an indicator vector of size 3, where only one selected component is 1, and the others are 0.

$$\text{if} \quad \mathbf{c}_u^{m\top} = \begin{cases} [1, 0, 0]^\top, & \text{then goto Stage 2-a} \\ [0, 1, 0]^\top, & \text{then goto Stage 2-b} \\ [0, 0, 1]^\top, & \text{then goto Stage 2-c} \end{cases}$$

2) **Stage 2**: For each selection from Stage 1, choose the behavior item following respect to $\mathbf{c}_u^m$
   a) Stage 2-a (user-specific preference)
      Choose a behavior item from the user-specific preference set.[3]

$$\mathsf{w}_u^m \sim \mathbb{H}_u$$

   b) Stage 2-b (latent-group preference)
      Choose a behavior item following the generative process of LDA

$$\mathsf{w}_u^m \sim \mathrm{p}(\mathsf{w}_u^m | \boldsymbol{\theta}, \boldsymbol{\beta}).$$

   c) Stage 2-c (popularity)
      Sample a selection from the popularity distribution,

$$\mathsf{w}_u^m \sim \mathrm{p}(\mathsf{w}_u^m | \boldsymbol{\rho}).$$

3) For each selection from Stage 2, augment the corresponding indicator multisets per each user (allowing repetitions)

$$\mathbb{H}_u \leftarrow \mathbb{H}_u \cup \{\mathsf{w}_u^m\}.$$

The joint distribution of a component mixture, topic mixture a set of M components, M topics, and a set of M selected behaviors is given by:

$$p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z}, \mathbf{w} | \boldsymbol{\alpha}_w, \boldsymbol{\alpha}, \mathbb{H}, \boldsymbol{\beta}, \boldsymbol{\rho}) = p(\boldsymbol{\pi} | \boldsymbol{\alpha}_w) p(\boldsymbol{\theta} | \boldsymbol{\alpha}) \prod_{m=1}^{M} p(\mathbf{c} | \boldsymbol{\pi}) p(\mathbf{z} | \boldsymbol{\theta}) p(\mathbf{w} | \mathbf{c}, \mathbf{z}, \mathbb{H}, \boldsymbol{\beta}, \boldsymbol{\rho}), \quad (2)$$

where $\boldsymbol{\pi}$ is the component mixture, $\boldsymbol{\theta}$ is the topic mixture, $\mathbf{c}$ is the indicator for the selected component, and $\mathbf{z}$ is the topic indicator. The set of M observed variable $\mathbf{w}$ is sampled from the multinomial distribution of parameter $\mathbb{H}$, $\boldsymbol{\beta}$, $\boldsymbol{\rho}$ respect to the selected components.

[3] If $\mathbb{H}_u$ is empty, we repeat stage 1 until $\mathbb{H}_u$ can contain an item.

C. VARIATIONAL INFERENCE

To infer the hidden topic and the component, we need to compute the posterior distribution of the hidden variables. This requires normalizing the distribution by marginalizing over the hidden variables in Equation 2. Due to the coupling, summing over all the latent variable is computationally intractable, and we rely on variational inference. Variational inference posits computationally tractable distribution with variational parameters. By making the use of Jensen's inequality, variational distribution with variational parameters can be obtained through optimization procedure that attempts to find the tightest possible lower bounds.

A factorized variational distribution over the latent variables $q(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z})$ is suggested as below:

$$q(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z}) = q_{\text{dir}}(\boldsymbol{\pi} | \boldsymbol{\tau}) q_{\text{dir}}(\boldsymbol{\theta} | \boldsymbol{\gamma}) \prod_{m=1}^{M} q_{\text{mul}}(\mathbf{c}^m | \boldsymbol{\lambda}^m) q_{\text{mul}}(\mathbf{z}^m | \boldsymbol{\phi}^m), \quad (3)$$

where $\{\boldsymbol{\tau}\}$, $\{\boldsymbol{\gamma}\}$ are the set of variational parameters for each user, and $\{\boldsymbol{\lambda}\}$, and $\{\boldsymbol{\phi}\}$ are the set of variational parameters for each selected behavior for corresponding user.

Given the factorized variational distribution $q(\cdot)$, we next bound the log likelihood of the observed data using Jensen's inequality. Specifically, we consider the so called evidence lower bound (ELBO) defined as follows:

$$\begin{aligned} \log p(\mathbf{w}^m | \boldsymbol{\alpha}_w, \boldsymbol{\alpha}, \mathbb{H}, \boldsymbol{\beta}, \boldsymbol{\rho}) &\geq \mathcal{L}(\boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\phi}) \\ &\triangleq \mathbb{E}_q[\log p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z}, \mathbf{w} | \boldsymbol{\alpha}_w, \boldsymbol{\alpha}, \mathbb{H}, \boldsymbol{\beta}, \boldsymbol{\rho})] \\ &\quad - \mathbb{E}_q[\log q(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z})], \end{aligned} \quad (4)$$

To maximize the lower bound in Equation 4, we optimize the variational parameters by taking the derivatives of $\mathcal{L}(\boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\phi})$ and setting them to zero respectively.

For $\boldsymbol{\tau}$ and $\boldsymbol{\gamma}$ of user 'u', the update equations are as follows:

$$\boldsymbol{\tau}_u \leftarrow \boldsymbol{\alpha}_w + \textstyle\sum_{m=1}^{M_u} \boldsymbol{\lambda}_u^m, \quad (5)$$

$$\boldsymbol{\gamma}_u \leftarrow \boldsymbol{\alpha} + \textstyle\sum_{m=1}^{M_u} \boldsymbol{\phi}_u^m. \quad (6)$$

The update equations of variational parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\phi}$ on 'm'-th behavior of user 'u' are given below, which are obtained from optimizations. The update equation of $\boldsymbol{\phi}$ we obtain is exactly same as the original LDA model in [16]. This is expected as one of the sub-component in our generative model follows the generative process of LDA.

$$\begin{aligned} \boldsymbol{\phi}_{u,k}^m &\propto \boldsymbol{\beta}(\mathbf{w}_u^m, k) \exp(\mathbb{E}_q[\log p(\boldsymbol{\theta}_{u,k})] \\ &= \boldsymbol{\beta}(\mathbf{w}_u^m, k)[\psi(\gamma_{u,k}) - \psi(\textstyle\sum_r \gamma_{u,r})], \end{aligned} \quad (7)$$

where we follow the same notations ψ(di-gamma function) from [16], and denote $\boldsymbol{\beta}(\mathbf{w}_u^m, k)$ as the probability of the selection of m-th behavior of user u at the given topic k.

Similar reasoning allows us to achieve the update equation of $\boldsymbol{\lambda}$.

$$\begin{aligned} \boldsymbol{\lambda}_{u,0}^m &\propto \mathbb{H}_u(\mathbf{w}_u^m) \exp(\mathbb{E}_q[\log p(\boldsymbol{\pi}_{u,0})] \\ \boldsymbol{\lambda}_{u,1}^m &\propto \textstyle\sum_{k=1}^{K} \boldsymbol{\phi}_{u,k}^m \boldsymbol{\beta}(\mathbf{w}_u^m, k) \exp(\mathbb{E}_q[\log p(\boldsymbol{\pi}_{u,1})] \\ \boldsymbol{\lambda}_{u,2}^m &\propto \boldsymbol{\rho}(\mathbf{w}_u^m) \exp(\mathbb{E}_q[\log p(\boldsymbol{\pi}_{u,2})]. \end{aligned} \quad (8)$$

TABLE 1. **General characteristics of datasets matrices: matrix size (total users and items), number of user-item Pairs, total number of events and average repetition per item $\bar{n}$.**

	U x I size	# Pairs	# Events	Avg.Repetition
`redditS`	20k x 17k	391k	562k	8.3
`lastfm`	668 x 2439	277k	10m	28.9
`goSFloc`	1k x 7k	49k	86k	2
`goNYloc`	633 x 7k	24k	40k	1.8
`twOCloc`	13k x 11k	65k	291k	7.3
`twNYloc`	14k x 10k	167k	455k	4.2

The right-subscript {0, 1, 2} of $\boldsymbol{\lambda}$ corresponds to three cases: *user-specific preference, latent group preference*, and *popularity* respectively. Note that for the *global-topic* case, all the possible topics have been considered by summing over all K topics. Just as the $\boldsymbol{\phi}$ is normalized, we normalize $\boldsymbol{\lambda}$ in a way the summation of $\boldsymbol{\lambda}_{u,0}$, $\boldsymbol{\lambda}_{u,1}$, and $\boldsymbol{\lambda}_{u,2}$ becomes unity.

With these update equations, we iteratively update each variational parameters until the ELBO converges. During the iteration, we can update the model parameters $\boldsymbol{\beta}$. The update equation for $\boldsymbol{\beta}$ is straightforward by summing over all $\boldsymbol{\phi}$ and normalizing respect to each topic. We omit the update equation in this paper, which can be found in [16]. The variational Expectation-Maximization(EM) will allow us to achieve the inference of hidden variables and parameter estimations, which leads to predictions of unseen behaviors.

IV. EXPERIMENTS AND RESULTS

A. DATASET

We conduct experiments using real-world datasets.[4] To compare the prediction performance to the state-of-the-art model [34], we use the same datasets in [34]. As we want to focus more on *active* users and *active* items, the datasets have been preprocessed, filtering out users and items below the threshold. The following section provides a detailed description of four datasets and the preprocessing steps.

Table 1 shows the general characteristics of datasets after preprocessing. The first column contains the name of the datasets we use in this experiment. The second column illustrates the size of the user-item matrix. The third column is the number of unique user-item pairs, which is greater than 0 (non-zero pairs). We count the total occurrences of user-item pairs, and have it as the total number of events which is shown in the fourth column. This total number of events reflects the activeness of users for each dataset. The fifth column contains the average repetition rate per item for each dataset. This value is computed by getting the average number of occurrences of each item in all users and then getting the mean value of all items. We only use non-zero user-item pairs to calculate this value. The average repetition rate per item shows how likely users will repeat their behaviors for each dataset.

[4]https://archive.ics.uci.edu/ml/datasets/Repeat+Consumption+Matrices

1) DESCRIPTION

Gowalla is a location-based social networking website where users share their locations by checking-in. The dataset from Gowalla includes 6,442,890 check-ins from 196,591 users throughout Feb. 2009 to Oct. 2010. Each check-in in this dataset contains user id, location id, timestamp, and the coordinate of each check-in. The location id in this dataset is considered as behavior index (or consumed items). As in [34], we consider the check-ins in San Francisco and New York. In our experiments, we use active users who have more than 10 check-ins. The locations which are visited less than five times are also disregarded. The data from September 2009 to August 2010 are used for training, while the following two months are used for testing.

Focusing on the check-in feature in Twitter, `twOCloc` and `twNYloc` datasets can be seen as a location-based dataset. The data were collected from geo-located tweets from Orange and New York between May 2015 and February 2016. Similarly, in our experiments, we consider active users and locations, where the users who have tweet less than 5 different days and locations, which has been tweeted less than 3 different events, are filtered out. The data from May 2015 to January 2016 are used for training, while the last month is used for testing.

`lastfm` dataset consists of the music listening records of 992 users from lastfm.com from 2006 to 2009. In this dataset, each record contains user id, timestamp, artist id, artist name, track id, and track name. For this dataset, the behaviors (or items) are artists whom users listen to. Artists having less than 100 songs and listened by under 50 different users are excluded from the item set. The records of each user from 2006 to March 2009 are used as training data, and the records from the next three months are used as test data.

`redditS` dataset contains data from Reddit, a popular social network with approximately 1.5 billion visits per month. Reddit consists of many subreddits with different topics. Users can subscribe to several subreddits and also give comments to articles in subreddits. The behavior of a user corresponds to a comment of that user on any articles in a subreddit, where the subreddit becomes the behavior index. `redditS` contains subreddits which have more than 1,000 subscribers. It was randomly sampled from users who gave comments more than 1,000 times in Reddit since 2015, which lead to a dataset of over 20,000 users. Data from Jan 2015 to Feb 2016 are used for training, and the following two months are used for testing.

For all datasets, we remove users and items whose number of occurrence in training data equals to 0. If a user does not have any behaviors before, it is impossible to predict his/her future behaviors, and we can only guess randomly.

2) CHARACTERISTICS

Table 2 provides a summary of each dataset. The average number of unique items per user is computed by counting all the selected items for each user. The average unique items

TABLE 2. Statistics in training data across all datasets: average unique items per user and User-item pairs repetition ratio.

	Avg. Number of unique items per user	User-item pairs repetition
`redditS`	18.9	62%
`lastfm`	378	73%
`goSFloc`	34.9	17%
`goNYloc`	26.9	19%
`twOCloc`	8.8	43%
`twNYloc`	9.6	32%

TABLE 3. Statistics in test data across all datasets: average unique items per user and new item ratio.

	Avg. Number of unique items per user	New items
`redditS`	7.9	30%
`lastfm`	72.8	19%
`goSFloc`	9.3	63%
`goNYloc`	8.1	68%
`twOCloc`	1.8	33%
`twNYloc`	1.6	50%

per user of `lastfm` is far higher than other datasets. This is expected as other types of behaviors in our datasets than lastfm requires more energy; checking in or writing requires more energy than listening to music. The average unique items per user values of two `go` datasets are much higher than `tw` while the user-item pairs repetition ratios are lower despite that both are location-based dataset. This indicates that users in Gowalla tend to explore many places but are not likely to repeat their visits. Gowalla promoted users to explore new places through social gaming setting [41], and this might have contributed to higher rates of unique items in `go` datasets.

The last column in Table 2 shows the chances of items being repeated. The repetition ratio of items in digital item datasets (`redditS` and `lastfm`) is considerably higher than physical item datasets (`go` and `tw`). It is unsurprising that the repetition ratio in digital environments is high when we consider the nature of the items in each dataset. For `redditS`/`lastfm`, an item is a subreddit/artist. Each item can contain several smaller sub-items, which are articles/songs. An interaction between a user and a sub-item will be counted as an interaction between that user and item, which contains that sub-items. People tend to listen to more than one song of their artists or comment in various articles in their subreddits, which creates the repetition ratio on items in these datasets high. In contrast, an item in `go` or `tw` is a unique place which does not contain any sub-items. Visiting a place also requires more energy than consuming virtual contents, which make the ratio of repetition in `go` and `tw` comparably less than `redditS` and `lastfm`.

Table 3 shows the average number of unique items per user and the average proportion of new items from each user in the test dataset. An item in a test dataset is considered as a new item for a user if it does not appear in the training set of that user. If not, that item is a repeat item. The higher new item proportion of location-based datasets record is projected. Users have to spend more resources to visit a place than listening to a song, so they are intended to explore new locations.

These statistics reveal the diverse nature of the datasets in our experiments, where these datasets have different new items ratio and the average number of items for each user. This diversity allows us to validate the performance and also the adaptation ability of models with various scenarios.

B. EVALUATION METHODS

In our experiments, we predict each user's behavior and rank the candidates from most likely to least likely per user. Once we obtain the probability of each item for each user, we sort the items and rank them for each user. Recall@k is one of our evaluation metrics, which is used to evaluates the capability of models to assign a high rank to items for each user in the test data set. Recall@k is calculated as:

$$\text{Recall}@k = \frac{1}{N_{test}} \sum_{u} \sum_{j} \frac{n_{uj}\boldsymbol{I}((rank(u,j) <= k)}{\sum_{j'} n_{uj'}} \tag{9}$$

This metric measures what fraction of items in test dataset were ranked in the top k by our model for user u. For user u, given the rank of all items, if the rank $rank(u, j)$ of item j in the test dataset is in the top k predicted behavior, the accuracy is 1; otherwise, it is 0. We denote the function for assigning accuracy by $\boldsymbol{I}((rank(u,j) <= k)$. n_{uj} corresponds to the number of times the behavior between user u and item j happens. N_{test} is the number of users in the test dataset.

Average rank is another metric for the evaluation of models. This metric is calculated as below:

$$AverageRank = \frac{1}{\sum_u \sum_{j'} n_{uj'}} \sum_{u} \sum_{j} n_{uj} rank(u,j) \tag{10}$$

We use this metric along with recall@k as it allows us to see the overall performance of models. While recall@k only focuses on high-rank items, average rank considers every candidate, that is useful if we want to evaluate the predictive performance of novel items which do not appear in the history of the user.

We also compute the Area Under Curve (AUC) of Receiver Operating Characteristic (ROC) curve tao provide a holistic view. This metric helps compare the overall prediction results. The ROC illustrates the performance of models with different thresholds and allows us to see the difference between models conveniently.

Finally, we divide items in the test dataset into two different types: "repeat" and "new" and then apply the same evaluation method for each type of data. The purpose of this experiment is to evaluate the capability of models in predicting new items, which may be more significant and attractive than predicting the repeat items.

C. PREDICTION RESULTS

We compare our hybrid generative model (HGM)[5] with the predictive multinomial mixture model (MMM) [34], which is the state-of-the-art work in item-consumption prediction. Other approaches such as Non-negative matrix factorization [23], Hierarchical Bayes Poisson factorization [42] or Latent Dirichlet Allocation [16] have also been compared in [34], and MMM outperformed previous baselines. We omit other baselines and set MMM as our only baseline. MMM and our model have different approaches but share some similarities. History consumption of user and global population preference are two main components in MMM, which reflect an individual's past consumption and population patterns.

In our model, we consider history consumption as a user-specific preference, which is unique for each user. The global population corresponds to the exogenous effect, which also can be extended with features like the distance between locations. As we want to conduct a fair comparison, we fix user-specific preference as history consumption and exogenous effect as the global population. However, we want to highlight that the user-specific preference in our model can be extended to other types (i.e., behavior triggered by friendship), and our exogenous effect can be extended to other types besides the global population (i.e., trends, outbreak). The main difference is the way how we consider the LDA based topic component, which the Mixture model misses. We performed experiments on all dataset with different topic numbers K and choose the best topic number for each dataset.[6]

The results of this experiment are organized as follows. Firstly, we show the overall performance across all datasets, then display the results on predicting repeat behavior. Finally, we discuss the new behavior prediction results.

1) OVERALL PERFORMANCE

Table 4 displays average Recall@100, average rank, and AUC for all datasets. Our proposed model, which is extended from original LDA, shows an improvement in all metrics, compared to the MMM. The differences in Recall@100 and AUC are minor, but they are more obvious in average rank. As mention above, the average rank is a more overall metric, compared to Recall@k. These results point out that our model may have a better performance with items that do not have a high rank. In the next two subsections, we conduct experiments with repeat and new items separately to examine the ability of the model in predicting two types of items.

2) REPEAT BEHAVIOR PREDICTION PERFORMANCE

Table 5 illustrates Recall@100, average rank, and AUC for repeat items of all datasets. The results of the two models are similar across all datasets, which may be due to the user's history and popularity. Both models tend to assign high ranks for repeat items, showing that user-specific preference (user's history) contributes significantly to the prediction result. However, our model performs slightly better in some datasets, which may come from latent group preference (LDA). We believe that LDA also contributes to the predicting of repeat items.

The effect levels of repeat behavior on overall results are various across datasets. For example, Recall@100 and AUC of repeat items of `redditS` are not far from overall results. In contrast, for `goSFloc` and `goNYloc`, results on repeat items and overall results are disparate. This can be explained when we look at the table 3, where new items ratio of `goSFloc` and `goNYloc` are much higher than `redditS` and `lastfm`. The diversity of datasets shows that considering only overall results is not enough, where analyzing both types of behavior is required.

TABLE 4. Recall@100, average rank and AUC across different data sets. Higher scores are better for Recall@100 and AUC. Lower scores are better for average rank. Best-performing methods indicated in bold font.

		Recall@100	Avg.Rank	AUC
`redditS`	MMM	0.82	202	**0.99**
	HGM	**0.83**	**188**	**0.99**
`lastfm`	MMM	0.67	170	0.93
	HGM	**0.69**	**151**	**0.94**
`goSFloc`	MMM	0.50	903	0.87
	HGM	**0.51**	**857**	**0.88**
`goNYloc`	MMM	**0.47**	1196	0.81
	HGM	**0.47**	**1181**	**0.82**
`twOCloc`	MMM	0.78	310	0.96
	HGM	**0.79**	**240**	**0.97**
`twNYloc`	MMM	0.64	528	**0.95**
	HGM	**0.65**	**495**	**0.95**

TABLE 5. Recall@100, average rank and AUC on the repeat items across different data sets. Higher scores are better for Recall@100 and AUC. Lower scores are better for average rank. Best-performing methods indicated in bold font.

		Recall@100	Avg.Rank	AUC
`redditS`	MMM	0.99	7	0.99
	HGM	**0.99**	**7**	**0.99**
`lastfm`	MMM	0.77	90	0.96
	HGM	**0.78**	**85**	**0.97**
`goSFloc`	MMM	0.97	30	0.99
	HGM	**0.97**	**29**	**0.99**
`goNYloc`	MMM	**0.98**	30	0.99
	HGM	0.97	**29**	**0.99**
`twOCloc`	MMM	0.86	7	0.99
	HGM	**0.87**	**7**	**0.99**
`twNYloc`	MMM	0.75	6	0.99
	HGM	**0.76**	**6**	**0.99**

[5] https://github.com/hellpoethero/Hybrid-Generative-Model

[6] We set K equals to 3 for `twNYloc`, 4 for `goNYloc`, `goSFloc` and `twOCloc`, 7 for `redditS` and 10 for `lastfm`

TABLE 6. Recall@100, average rank and AUC on the new items across different data sets. Higher scores are better for Recall@100 and AUC. Lower scores are better for average rank. Best-performing methods indicated in bold font.

		Recall@100	Avg.Rank	AUC
`redditS`	MMM	0.23	1721	0.88
	HGM	**0.28**	**1599**	**0.88**
`lastfm`	MMM	**0.013**	1000	0.59
	HGM	0.011	**909**	**0.65**
`goSFloc`	MMM	0.15	1871	0.73
	HGM	**0.16**	**1776**	**0.75**
`goNYloc`	MMM	**0.17**	2268	0.64
	HGM	**0.17**	**2241**	**0.67**
`twOCloc`	MMM	0.31	1657	0.79
	HGM	**0.33**	**1278**	**0.85**
`twNYloc`	MMM	0.26	1595	0.84
	HGM	**0.28**	**1496**	**0.86**

3) NEW BEHAVIOR PREDICTION PERFORMANCE

Table 6 displays the Recall@100, average rank, and AUC for new items of all test datasets. Our model outperforms in most of the datasets with different margins. This improvement can be explained by the integration of the latent group preference component in our model, that is the difference between our model and MMM. The latent group preference component enables the ability to find new items by considering the underlying relationship between items.

Specifically, our model performs slightly better than MMM in all three metrics for (`goNYloc` and `goSFloc`). For `tw` datasets, result shows a minor improvement for `twNYloc`, but records a much higher difference for `twOCloc`, especially in average rank and AUC. Although Recall@100 for `twOCloc` of our model is marginally better than MMM, the huge difference in average rank and AUC shows that our model performs better in overall. When it comes to `redditS`, the improvement of Recall@100 is far significant than other datasets. Result for `lastfm` is interesting. Recall@100 value of our model for `lastfm` is slightly worse than MMM; however, the average rank and AUC are considerably higher. We can see that Recall@100 values of both models for `lastfm` are significantly smaller than other datasets. To examine this problem, we plot Recall@k with k varied from 200 to 1000 and ROC curves for each dataset.

Figure 2 shows the Recall@k with k varied from 100 to 1000 for new items of all test datasets. Our model outperforms for most of the datasets with different k. The only exception is `lastfm` dataset. When k equals to 100 and 200, MMM has better results with a little difference. However, our model outperforms significantly when k increases to 500 and 1000. We believe this is due to the characteristic of `lastfm` dataset. When comparing results in table 6 and table 5, we can see that new items have a smaller chance to be assigned to high rank than repeat items. The average number of unique new items per user in `lastfm`,

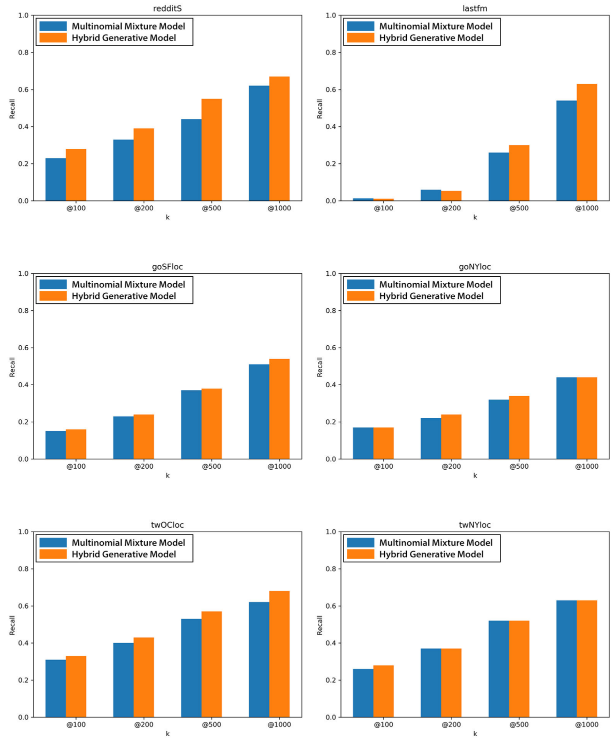


FIGURE 2. Recall@k of multinomial mixture model (MMM) and hybrid generative model (HGM) with k varied from 100 to 1000 for new items across all datasets. The results for MMM are in blue color (left) while results for HGM are in orange (right).

according to table 3, is considerably smaller than other datasets while the average number of unique items per user is high, leading to a result that if we set k to 100, most of the items are repeat items. Once the number of new items is small, it is difficult to compare the accuracy between the two models in predicting new items. ROCs of `lastfm` in Figure 3 indicate that both our model and MMM perform badly at the first stage but then increase remarkably. The gap between the two curves is considerably huge, showing that our model is much better. For other datasets, despite that the gap is smaller, our model always performs better. These results prove that latent group preference has a significant impact on predicting new items.

We can also see that the improvements in predicting new items for `redditS` and `lastfm` are higher than other location-based datasets. For `redditS`, each item in this dataset corresponds to a subreddit, which contains several articles with the same topic(s). The topics of articles are considered as topics of subreddit containing them. In this aspect, an item in `redditS` is a mixture of several topics, which is suitable for applying topic modeling. Items in `lastfm`, which are artists having several songs, are also considered in this way. Items in `go` and `tw`, which are physical locations, may belong to some specific categories; however, it is difficult to establish a relationship between these items. For online services like Reddit, users can easily access different subreddits with similar topics, but the behavior in the real-world is different. The decision to visit a place is influenced

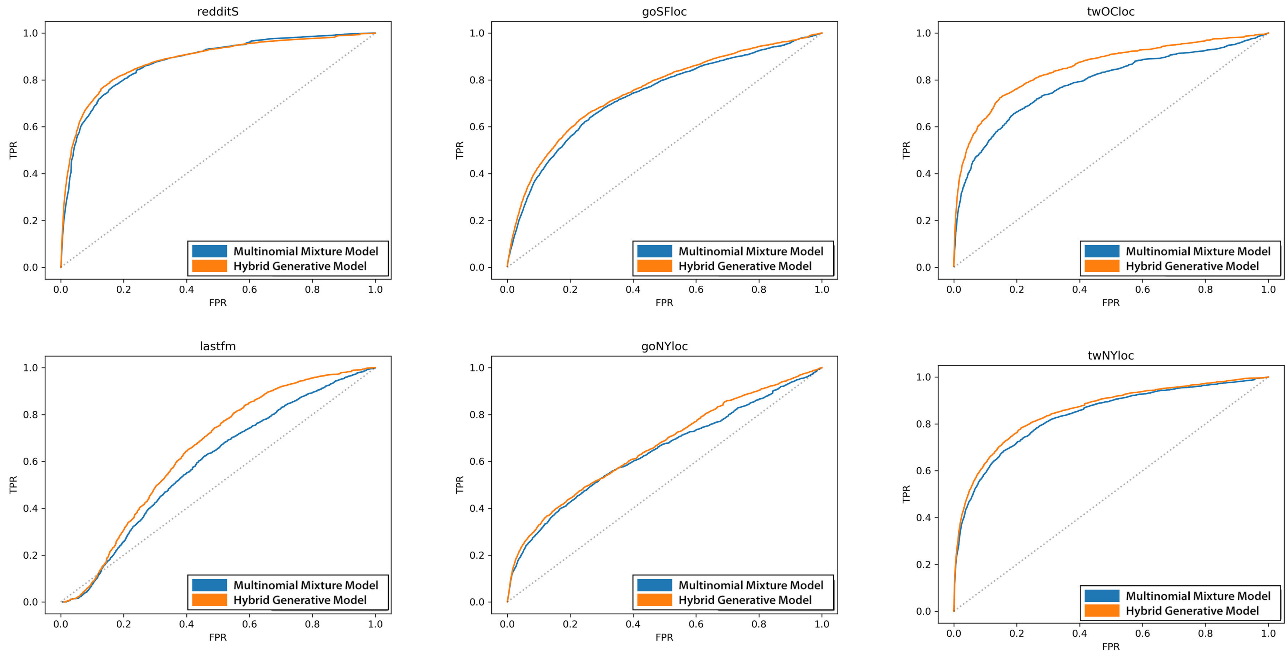


FIGURE 3. ROC curves of multinomial mixture model and hybrid generative model for new items across all datasets.

by many other factors, like distance, time, and so on. People may like to visit many museums, but if the distance between a new museum and their current location is far, they are more likely to give up. Hence, we can conclude that the impact of latent group preference depends on the characteristics of each dataset, for example, location-based service datasets are less affected by latent group preference than music service datasets.

V. CONCLUSION AND DISCUSSION

In this paper, we propose a hybrid generative model based on LDA for investigating the behavior prediction problem. This model contains three mains components: (1) the user-specific preference, which is unique for each user, (2) the latent group preference generated through LDA, and (3) the exogenous effects obtained from the characteristics of each dataset. Our model outperforms the recent mixtures model in most of the experiments with different margins. Our model achieves reasonable results in predicting future behavior, especially predicting new items, which is critical in real consumption services. This improvement comes from the latent group preference component, which is the difference between our model and the recent one. Although our model only considers the personal history of users as user-specific preference and popularity of items as an external factor, other features, such as distance between places, genre of songs, topics of subreddits, can also be included. Our model can be extended comfortably due to the flexibility of LDA. One future direction is integrating temporal features and dealing with the changing of items consumption over time to improve the performance.

REFERENCES

[1] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[2] B. Zhou and R. Wong, "Effective matrix factorization for online rating prediction," in *Proc. 50th Hawaii Int. Conf. Syst. Sci.*, 2017.

[3] G. Xu, Y. Zhang, and X. Yi, "Modelling user behaviour for Web recommendation using LDA model," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Agent Technol.* Washington, DC, USA: IEEE Computer Society, 2008, pp. 529–532, doi: 10.1109/WIIAT.2008.313.

[4] R. Giri, H. Choi, K. S. Hoo, and B. D. Rao, "User behavior modeling in a cellular network using latent Dirichlet allocation," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.*, Sep. 2014, pp. 36–44.

[5] W. Xie, Q. Dong, and H. Gao, "A probabilistic recommendation method inspired by latent Dirichlet allocation model," *Math. Problems Eng.*, vol. 2014, pp. 1–10, Sep. 2014.

[6] M. A. Awad and I. Khalil, "Prediction of user's Web–browsing behavior: Application of Markov model," *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 4, pp. 1131–1142, Aug. 2012.

[7] W. Mathew, R. Raposo, and B. Martins, "Predicting future locations with hidden Markov models," in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*. New York, NY, USA: ACM, 2012, pp. 911–918.

[8] S. Gambs, M.-O. Killijian, and M. N. Del Prado Cortez, "Next place prediction using mobility Markov chains," in *Proc. 1st Workshop Meas., Privacy, Mobility (MPM)*. New York, NY, USA: ACM, 2012, pp. 3:1–3:6.

[9] J. Kiseleva, H. T. Lam, M. Pechenizkiy, and T. Calders, "Predicting current user intent with contextual Markov models," in *Proc. IEEE 13th Int. Conf. Data Mining Workshops*, Dec. 2013, pp. 391–398.

[10] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016.

[11] T. Donkers, B. Loepp, and J. Ziegler, "Sequential user-based recurrent neural network recommendations," in *Proc. 11th ACM Conf. Recommender Syst. (RecSys)*, 2017.

[12] G. Yang, Y. Cai, and C. K. Reddy, "Spatio–temporal check-in time prediction with recurrent neural network based survival analysis," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, AAAI Press, Jul. 2018, pp. 2976–2983.

[13] A. Anderson, R. Kumar, A. Tomkins, and S. Vassilvitskii, "The dynamics of repeat consumption," in *Proc. 23rd Int. Conf. World Wide Web (WWW)*, New York, NY, USA: ACM, 2014, pp. 419–430. [Online]. Available: http://doi.acm.org/10.1145/2566486.2568018

[14] A. R. Benson, R. Kumar, and A. Tomkins, "Modeling user consumption sequences," in *Proc. 25th Int. Conf. World Wide Web (WWW)*, Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2016, pp. 519–529, doi: 10.1145/2872427.2883024.

[15] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *TOISACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004, doi: 10.1145/963770.963772.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[17] R. Guzman, E. Cervantes, L. Cruz, and J. Luna, "Online courses recommendation based on LDA," in *Proc. 1st Symp. Inf. Manage. Big Data*, vol. 1318, Jan. 2014, pp. 42–48.

[18] T.-M. Chang and W.-F. Hsiao, "LDA-based personalized document recommendation," in *Proc. PACIS*, Jan. 2013.

[19] M. J. Pazzani and D. Billsus, *Content-Based Recommendation System*. Berlin, Germany: Springer, 2007, pp. 325–341.

[20] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," Tech. Rep., Jan. 2007.

[21] A. Bernstein and A. Kuleshov, "Low-dimensional data representation in data analysis," in *Artificial Neural Networks in Pattern Recognition*, N. El Gayar, F. Schwenker, and C. Suen, Eds. Cham, Switzerland: Springer, 2014, pp. 47–58.

[22] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2007, pp. 1257–1264.

[23] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. Cambridge, MA, USA: MIT Press, 2001, pp. 556–562.

[24] P. Bhargava, T. Phan, J. Zhou, and J. Lee, "Who, what, when, and where: Multi–dimensional collaborative recommendations using tensor factorization on sparse user–generated data," in *Proc. 24th Int. Conf. World Wide Web (WWW)*. Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2015, pp. 130–140.

[25] Y. Zhong, N. J. Yuan, W. Zhong, F. Zhang, and X. Xie, "You are where you go: Inferring demographic attributes from location check-ins," in *Proc. 8th ACM Int. Conf. Web Search Data Mining (WSDM)*. New York, NY, USA: ACM, 2015, pp. 295–304.

[26] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi, "Improving user topic interest profiles by behavior factorization," in *Proc. 24th Int. Conf. World Wide Web (WWW)*. Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2015, pp. 1406–1416

[27] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. 15th Conf. Uncertainty Artif. Intell. (UAI)*. San Francisco, CA, USA: Morgan Kaufmann, 1999, pp. 289–296.

[28] K. Christidis, D. Apostolou, and G. Mentzas, "Exploring customer preferences with probabilistic topics models," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discovery Databases*, 2010.

[29] M. Qiu, F. Zhu, and J. Jiang, "It is not just what we say, but how we say them: LDA-based behavior–topic model," in *Proc. SIAM Int. Conf. Data Mining*, May 2013.

[30] E. Adar, J. Teevan, and S. T. Dumais, "Large scale analysis of Web revisitation patterns," in *Proc. 26th Annu. CHI Conf. Hum. Factors Comput. Syst. (CHI)*, New York, NY, USA: ACM, 2008, pp. 1197–1206, doi: 10.1145/1357054.1357241.

[31] K. Kapoor, K. Subbian, J. Srivastava, and P. Schrater, "Just in time recommendations: Modeling the dynamics of boredom in activity streams," in *Proc. 8th ACM Int. Conf. Web Search Data Mining (WSDM)*, Feb. 2015, pp. 233–242.

[32] W. Trouleau, A. Ashkan, W. Ding, and B. Eriksson, "Just one more: Modeling binge watching behavior," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016.

[33] M. Lichman, D. Kotzias, and P. Smyth, "Personalized location models with adaptive mixtures," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst. (GIS)*, New York, NY, USA: ACM, 2016, pp. 67:1–67:4.

[34] D. Kotzias, M. Lichman, and P. Smyth, "Predicting consumption patterns with repeated and novel events," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 371–384, Feb. 2019.

[35] M. Lichman and P. Smyth, "Prediction of sparse user-item consumption rates with zero-inflated Poisson regression," in *Proc. World Wide Web Conf.* Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2018, pp. 719–728.

[36] S. Park and V. Vasudev, "Predicting Web user's behavior: An absorbing Markov chain approach," in *Proc. Workshop E-Bus.*, Nov. 2017, pp. 170–176.

[37] S. Zhao, X. Chen, I. King, and M. R. Lyu, "Personalized sequential check-in prediction: Beyond geographical and temporal contexts," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.

[38] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. INTERSPEECH*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds. Kolkata, India: ISCA, 2010, pp. 1045–1048. [Online]. Available: http://dblp.unitrier.de/db/conf/interspeech/interspeech2010.html

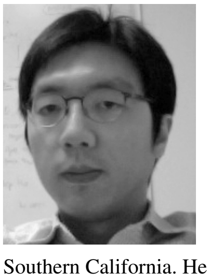
[39] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, p. 77, Apr. 2012.

[40] H. Alharthi, D. Inkpen, and S. Szpakowicz, "A survey of book recommender systems," *J. Intell. Inf. Syst.*, vol. 51, no. 1, pp. 139–160, Aug. 2018.

[41] M. Barker, D. Barker, N. Bormann, and D. Zahay, *Social Media Marketing: A Strategic Approach*. Boston, MA, USA: Cengage, 2016. [Online]. Available: https://books.google.co.kr/books?id=mo56CgAAQBAJ

[42] P. Gopalan, J. M. Hofman, and D. M. Blei, "Scalable recommendation with poisson factorization," 2013, *arXiv:1311.1704*. [Online]. Available: https://arxiv.org/abs/1311.1704

MINH-DUC NGUYEN received the B.Sc. degree in information system from the University of Engineering and Technology, Vietnam National University, Hanoi, in 2016. He is currently pursuing the M.S. degree with Sejong University, South Korea. From 2016 to 2018, he was a Researcher Assistant and a Lecturer with the University of Engineering and Technology, Vietnam National University, respectively. His research areas include data science and social media mining.



YOON-SIK CHO received the B.S. degree in electrical engineering from Seoul National University, South Korea, in 2003, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2014. He was an Academic Mentor for RIPS Program with the Institute for Pure and Applied Mathematics, University of California Los Angeles, Los Angeles, and a Postdoctoral Scholar with the Information Sciences Institute, University of Southern California. He is currently an Assistant Professor with the Department of Data Science, Sejong University, South Korea. His research interests include large-scale data science, social network analysis, and cloud computing.