Modeling Idle Customers to Tackle the Sparsity Problem in Time-dependent Recommendation

Completed Research Paper

Haidong Zhang
Institute of Automation, Chinese
Academy of Sciences
Beijing, China
haidong_zhang14@yahoo.com

Wancheng Ni
Institute of Automation, Chinese
Academy of Sciences
Beijing, China
wancheng.ni@ia.ac.cn

Xin Li
Department of Information Systems,
City University of Hong Kong
Hong Kong
Xin.Li.PhD@gmail.com

Yiping Yang
Institute of Automation, Chinese
Academy of Sciences
Beijing, China
yiping.yang@ia.ac.cn

Abstract

Recommender systems have been widely used to provide personal and convenient services for users. As one of successful recommendation methods, collaborative filtering explores users' interests from item consumptions. However, it suffers from the data sparsity problem that most users have interacted with a small number of items. Particularly, data sparsity causes the discontinuous user activities over time, which limits the time-dependent recommendation methods for tracking users' changing interests. In this paper, we extend existing methods and propose an inhibited hidden Markov model to alleviate the sparsity problem. The model considers the statuses of users' interests at each time unit and allows for capturing users' dynamic interests under idle status. We derive an EM algorithm to estimate the model parameters and predict users' actions. We perform a comprehensive experiment on the datasets of various sparsity levels. The results show our model has been consistently and significantly better than the state-of-the-art algorithms.

Keywords: Recommender systems, data sparsity, collaborative filtering, hidden Markov model

Introduction

Recommender systems have attracted significant interest from academia and industry. They alleviate the information overload problem and provide users with personalized services and product recommendations. Therefore, they have been widely used in many applications, such as in e-commerce Xiao and Benbasat (2007), movie recommendation (Azaria et al. 2013), social network services (Gupta et al. 2013), etc.

Collaborative filtering is one common and successful recommendation technique, in which the recommendation is made based on historical transactions and user-item interactions. Collaborative filtering is generally classified into memory-based and model-based methods (Cacheda et al. 2011). The former explore the similarity between users or items and make recommendations based on similar users' choices or item consumption. The latter use machine learning or data mining algorithms to explore the potential patterns. Both methods have success in real-world applications (Adomavicius and Tuzhilin 2005).

One major challenge for the collaborative filtering methods is the data sparsity problem (Adomavicius and Tuzhilin 2005), where most users only consume a small number of items. The sparsity problem often results in difficulties identifying and aggregating data from similar users or deriving potential consumptions.

Much research has investigated the sparsity problem. A typical approach is to utilize auxiliary information, such as users' demographics (Gogna and Majumdar 2015), items' content (Liu et al. 2013), or social contextual information (Ma et al. 2011) to compensate for the relationships between users or items. But such side information is difficult to acquire or not valid. Another common approach is to reduce the dimensionality of the user-item matrix or cluster users or items to smooth the impact of sparsity (Desrosiers and Karypis 2010). However, this method has limitations in theoretical interpretation.

In this research, we join these efforts to tackle the sparsity problem, particularly for time-dependent recommendation. Time-dependent recommender systems incorporate time factors into modeling of the evolution of user interests (Sahoo et al. 2012; Yin et al. 2015), which lead to observable performance improvements (Koren 2009; Xiong et al. 2010). A milestone work in this area is the hidden Markov model (HMM) (Sahoo et al. 2012), which models users' inherent interests with latent states and captures users' changing interest with the transition between the states. Nevertheless, time-dependent methods also face the data sparsity problem. Due to limited consumption, user-item interactions may be discontinuous over time, which would affect the recommendation performance (Luo et al. 2014).

In this paper, we improve Sahoo et al.'s framework (Sahoo et al. 2012) and build an inhibited hidden Markov model (IHMM) to tackle the sparsity problem. In the model, we allow the latent states of users' interests to be in an active and an inactive state, which controls whether the state can make emissions at the current time period. So, we do not observe the users' activities because the users are idle. The model provides a flexible framework to capture inactivity over time. We derive an expectation maximization (EM) algorithm to estimate the model parameters and make predictions under active states. Experiments on the Netflix dataset show that our model is more effective than the classic methods.

Our paper focuses on the sparsity problem in time-dependent recommendation. Results show that user inactivity may be part of the reason behind the sparsity problem and one aspect that can be modelled in recommender systems.

Related Work

Data sparsity is one major challenge for collaborative filtering (Adomavicius and Tuzhilin 2005). Many researchers have investigated this problem. In this section, we briefly summarize the related works that tackle the sparsity problem and organize them into two groups.

The first approach to deal with data sparsity is to incorporate auxiliary information of users or items into collaborative filtering for imputing missing relationships. A typical way is to integrate additional information, such as social network, item content, or user profiles into the memory-based collaborative filtering for neighbors selection (Jiang et al. 2015; Ozbal et al. 2011). Additional information can also be incorporated into the model-based collaborative filtering framework. For example, Gogna and Majumdar

(2015) investigated groups of similar users having similar demographics into matrix factorization as regularization terms. Ma et al. (2011) conducted latent factors analysis on the consumption and social relationship networks in the Bayesian probabilistic matrix factorization (BPMF) framework. Wang and Li (2015) leveraged items' relationships to link the collaborative topic regression models that fused topic bias generated by latent Dirichlet allocation (LDA) into probabilistic matrix factorization (PMF).

A unique piece of auxiliary information is consumers' ratings of items, which can be used to enhance the similarity computation. Desrosiers and Karypis (2010) fused all ratings into kernel functions to extend the SimRank algorithm's similarity computation that only used the common users and items. Ghazarian and Nematbakhsh (2015) used the Pearson VII Universal Kernel function to consider all ratings and enhance user-based collaborative filtering. Huang et al. (2004) and Chen et al. (2011) made use of association retrieval technology to explore extensional users' or items' similarities on user-item bipartite graphs.

The second method is to conduct clustering or dimensionality reduction methods on a user-item matrix for smoothing. Kim and Cho (2003) performed singular value decomposition (SVD) on ratings and calculated the similarities of decomposed vectors in user-based collaborative filtering. Koren et al. (2009) regularized the latent factors to handle the overfitting caused by sparsity in matrix factorization. Gong (2010) clustered users and items with k-means and predicted ratings with item-based collaborative filtering.

Although existing methods provide us ways to alleviate the data sparsity problem, they have their limitations. The additional information may be difficult to acquire. Dimension reduction addresses the problem but does not provide any theoretical insights on the reasons behind the sparse data.

Time-dependent recommendation methods explore the evolution of users' interests over time. One typical method is to employ a time-decay factor to reduce the weight of historical consumption of items over time. It assumes recent consumptions contribute more weight to users' current interests (Ding and Li 2005; Liu et al. 2010; Yu and Li 2010). Another method is to develop time-related variables to explore users' changing patterns. Xiang and Yang (2009) and Koren (2009) considered the changes of user bias and item bias as time variables in the matrix factorization framework. A recent prominent method is to model users' drifting interests with transitive latent variables. For example, Xiong et al. (2010) proposed a Bayesian Probabilistic Tensor Factorization that incorporated the time factor depending on the preceding time step into BPMF. Sahoo et al. (2012) combined the aspect model with HMM for tracking users' changing interests with the Markov chain.

Existing time-dependent methods seldom consider the sparsity problem. Jiang et al. (2015) studied users' dynamics at both rating and review levels in high-involvement product recommendation. They inferred topic ratings of the products over time and imputed the sparse user-item matrix by integrating the dynamics into the similarity between items. Luo et al. (2014) assumed the tendency of user factors and item factors depends on the recent corresponding factors in the probabilistic matrix factorization framework and imputed unobserved observations in the current user-item matrix with a sampling method.

In this research, we take a look at the data sparsity problem from the perspective of discontinuous user activities over time. We aim to model such idle users in a time-dependent recommendation method for tracking users' dynamic interests over time.

An Inhibited Hidden Markov Model

In this paper, we extend Sahoo et al. (2012) HMM model and propose an inhibited hidden Markov model (IHMM) that models time periods without user activities in the emission of latent states.

Model Formulation

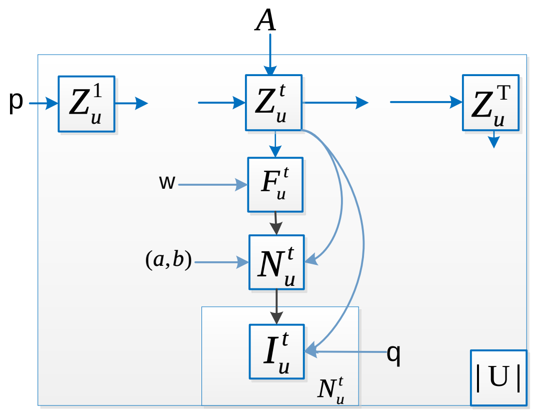

Figure 1. An Idle Hidden Markov Model for Collaborative Filtering

The HMM proposed by Sahoo et al. (2012) tracks the evolution of users' interests with a Markov chain. When a user is at a state, the state will emit observations, which are the users' consumptions of items. In our model, we follow the idea of user interest transition. Different from HMM, we design a state inhabitation mechanism that controls whether a state can emit observations at a time point. For the states that cannot emit, we consider the users are idle and not in a mood to consume. The model associates the latent state with the user's mood: to consume (emit) or not to consume (emit).

Figure 1 shows the graphical representation of our proposed model. In this model, we assume there are $|U|$ users in the dataset, and each user has the same HMM structure. We employ latent states $S = \{1, \dots, K\}$ to denote users' interests, and their initial probability is π. The discrete latent variable $Z_u^t \in S$ represents the possible interest of user u at time unit t, and the transition of states is used to capture each user's changing interest over time, whose probability is denoted as A. Each state can emit some observations that are users' consumptions of items. The set of items preferred by user u at time unit t and its number are denoted as I_u^t and N_u^t, following multinomial distributions with parameter θ and negative binomial distributions (NBD) with parameters a and b, respectively. Here we assume the entire dataset has $|I|$ items and T time units, $t \in [1, T]$.

In our model, we use a binary variable F_u^t to determine the status (active, or not) of latent states. When $F_u^t = 0$, the latent state of user u is inactive at time unit t, which means lack of observations at the corresponding time point. When $F_u^t = 1$, the latent state of user u is active at time unit t, which means the existence of emissions at this time point. We assume this variable follows the Bernoulli distributions with parameter ω, as in formula (1).

$$\begin{cases} P(F_u^t = 1 \mid Z_u^t = k) = w_k \\ P(F_u^t = 0 \mid Z_u^t = k) = 1 - w_k \end{cases} \tag{1}$$

When $F_u^t = 0$, we assume it is impossible to have observations. When $F_u^t = 1$, we assume the state must emit some observations. Thus:

$$\begin{cases} P(N_u^t = 0 \mid Z_u^t = k, F_u^t = 0) = 1 \\ P(N_u^t \neq 0 \mid Z_u^t = k, F_u^t = 0) = 0 \end{cases} \tag{2}$$

$$\begin{cases} P(N_u^t = 0 \mid Z_u^t = k, F_u^t = 1) = 0 \\ P(N_u^t \neq 0 \mid Z_u^t = k, F_u^t = 1) = 1 \end{cases} \tag{3}$$

Overall, our model has the following parameters:

- π: The initial probability of latent states associated with users' interests is a K-dimensional vector.
- A: The transitive probability of latent states representing the evolution of users' interests over time is a K*K matrix.
- ω : The parameters of binomial distributions followed by F_u^t are represented with a K-dimensional vector.
- a, b: The parameters of negative binomial distributions followed by N_u^t are represented with a K-dimensional vector respectively.
- θ: The parameters of multinomial distributions followed by I_u^t, the set of items preferred, are represented with a K*$|I|$ matrix.

EM Algorithm

To estimate the parameters $\Theta = \{\pi, A, \omega, a, b, \theta\}$ of our proposed model, we develop an EM algorithm. We follow the forward-backward approach in the Expectation Maximization (EM) framework to iteratively find the maximum likelihood of the parameters given a series of observations. In the Expectation step, we compute the forward variables and backward variables and derive the posterior probabilities of latent states given the observations. In the Maximization step, we employ maximum a posterior (MAP) to update the parameters. Then we predict the status (active, or not) of latent states associated with users' interests and infer the probabilities of preferences for items by each user under the active situation.

Given a sequence of users' consumptions $\{I_u^{1:T}\}_{|U|}$, we can get the log-likelihood function of the parameters as follows:

$$\begin{aligned}Q(\Theta,\Theta^{old}) = {}& \sum_u \sum_k P(\mathrm{Z}_\mathrm{u}^1 = \mathrm{k} \mid \mathrm{I}_\mathrm{u}^{1:\mathrm{T}};\Theta^{old}) \log \pi_\mathrm{k} + \sum_u \sum_{t=2}^{T} \sum_j \sum_k P(\mathrm{Z}_\mathrm{u}^{t-1} = \mathrm{j}, \mathrm{Z}_\mathrm{u}^t = k \mid \mathrm{I}_\mathrm{u}^{1:\mathrm{T}};\Theta^{old}) \log \mathrm{A}_{\mathrm{jk}} \\ & + \sum_u \sum_{t=1}^{T} \sum_k [P(\mathrm{Z}_\mathrm{u}^t = k, F_\mathrm{u}^t = 0 \mid \mathrm{I}_\mathrm{u}^{1:\mathrm{T}};\Theta^{old}) \log(1-w_k) + P(\mathrm{Z}_\mathrm{u}^t = k, F_\mathrm{u}^t = 1 \mid \mathrm{I}_\mathrm{u}^{1:\mathrm{T}};\Theta^{old}) \log(w_k)] \\ & + \sum_u \sum_{t=1}^{T} \sum_k P(\mathrm{Z}_\mathrm{u}^t = k, F_\mathrm{u}^t = 1 \mid \mathrm{I}_\mathrm{u}^{1:\mathrm{T}};\Theta^{old}) \log p(N_u^t \mid a_k, b_k) + \sum_u \sum_{t=1}^{T} \sum_k P(\mathrm{Z}_\mathrm{u}^t = k, F_\mathrm{u}^t = 1 \mid \mathrm{I}_\mathrm{u}^{1:\mathrm{T}};\Theta^{old}) \log p(\{I_{uj}^t\} \mid N_u^t, \theta_k)\end{aligned} \tag{4}$$

where $I_u^{1:T}$ represents the set of items preferred by user u.

In the Expectation step, we follow the forward-backward algorithm, and firstly define the forward variables and backward variables for inferring the posterior distributions in formula (4). Because there are two latent state statuses, i.e., active or inactive, we define two kinds of forward variables to present the posterior probabilities of observing $I_u^{1:t}$ given state Z_u^t :

$$\hat{\alpha}(\mathrm{Z}_\mathrm{u}^t) = P(\mathrm{Z}_\mathrm{u}^t, F_\mathrm{u}^t = 0 \mid \mathrm{I}_\mathrm{u}^{1:\mathrm{t}}) \tag{5}$$

$$\tilde{\alpha}(\mathrm{Z}_\mathrm{u}^t) = P(\mathrm{Z}_\mathrm{u}^t, F_\mathrm{u}^t = 1 \mid \mathrm{I}_\mathrm{u}^{1:\mathrm{t}}) \tag{6}$$

where $\hat{\alpha}(*)$ is the active forward variable, and $\tilde{\alpha}(*)$ is inactive. The forward variable represents the probability of states given the observations from the start time unit to the current time unit. It is equal to the sum of these active and inactive variables, formula (7).

$$\alpha(\mathrm{Z}_\mathrm{u}^t) = P(\mathrm{Z}_\mathrm{u}^t \mid \mathrm{I}_\mathrm{u}^{1:\mathrm{t}}) = \hat{\alpha}(\mathrm{Z}_\mathrm{u}^t) + \tilde{\alpha}(\mathrm{Z}_\mathrm{u}^t) \tag{7}$$

Here, we suppose F_u^t only depends on the current latent state and is independent of the previous states. Hence, the transitive probability of successive states with inactive or active status can be simplified as:

$$P(\mathrm{Z}_\mathrm{u}^t, F_\mathrm{u}^t \mid \mathrm{Z}_\mathrm{u}^{t-1}, F_\mathrm{u}^{t-1}) = P(\mathrm{Z}_\mathrm{u}^t, F_\mathrm{u}^t \mid \mathrm{Z}_\mathrm{u}^{t-1}) = P(F_\mathrm{u}^t \mid \mathrm{Z}_\mathrm{u}^\mathrm{t}) P(\mathrm{Z}_\mathrm{u}^t \mid \mathrm{Z}_\mathrm{u}^{t-1}) \tag{8}$$

According to the forward recursive formula described in Bishop (2006), we can get the recursive expressions of these two kinds of forward variables as follows:

$$\mathring{a}(Z_u^t)=\mathop{\mathring{a}}\limits_{Z_u^{t-1}}[a(Z_u^{t-1})P(Z_u^t \mid Z_u^{t-1})]P(F_u^t=0\mid Z_u^t)\frac{P(I_u^t \mid Z_u^t,F_u^t=0)}{P(I_u^t \mid I_u^{1:t-1})} \tag{9}$$

$$\overset{\text{Ù}}{a}(Z_u^t)=\mathop{\mathring{a}}\limits_{Z_u^{t-1}}[a(Z_u^{t-1})P(Z_u^t \mid Z_u^{t-1})]P(F_u^t=1\mid Z_u^t)\frac{P(I_u^t \mid Z_u^t,F_u^t=1)}{P(I_u^t \mid I_u^{1:t-1})} \tag{10}$$

where $P(I_u^t|I_u^{t-1})$ is used to normalize the variables so that $\sum_i \alpha(Z_u^t = i) = 1$.

Similarly, we can define two kinds of backward variables and calculate the recursive expressions.

$$\overset{\text{Ù}}{b}(Z_u^t) = P(I_u^{t+1:T} \mid Z_u^t, F_u^t=1) / P(I_u^{t+1:T} \mid I_u^{1:t}) \tag{11}$$

$$\mathring{b}(Z_u^t) = P(I_u^{t+1:T} \mid Z_u^t, F_u^t=0) / P(I_u^{t+1:T} \mid I_u^{1:t}) \tag{12}$$

$$\overset{\text{Ù}}{b}(Z_u^t) = \mathop{\mathring{a}}\limits_{Z_u^{t+1}} b(Z_u^{t+1})P(I_u^{t+1} \mid Z_u^{t+1})P(Z_u^{t+1} \mid Z_u^t) / P(I_u^{t+1:T} \mid I_u^{1:t}) = \mathring{b}(Z_u^t) \tag{13}$$

where $\hat{\beta}(*)$ is the active backward variable and $\tilde{\beta}(*)$ is the inactive. The backward variable can be derived as formula (14).

$$\begin{aligned} b(Z_u^t) &= P(I_u^{t+1:T} \mid Z_u^{t+1}, F_u^{t+1}) / P(I_u^{t+1:T} \mid I_u^{1:t}) = \frac{\mathring{a}_{F_u^t} P(I_u^{t+1:T}, F_u^t \mid Z_u^t)}{P(I_u^{t+1:T} \mid I_u^{1:t})} = \frac{\mathring{a}_{F_u^t} P(I_u^{t+1:T} \mid Z_u^t, F_u^t)P(F_u^t \mid Z_u^t)}{P(I_u^{t+1:T} \mid I_u^{1:t})} \\ &= \overset{\text{Ù}}{b}(Z_u^t)P(F_u^t=1 \mid Z_u^t) + \mathring{b}(Z_u^t)P(F_u^t=0 \mid Z_u^t) = \overset{\text{Ù}}{b}(Z_u^t) = \mathring{b}(Z_u^t) \end{aligned} \tag{14}$$

Thereby, we can derive the posterior distributions in formula (4). Firstly, the posterior probability of state Z_u^t with active status or inactive status given a sequence of observations can be derived as the product of corresponding forward and backward variables.

$$P(Z_u^t, F_u^t = 1 \mid I_u^{1:T}) = \overset{\text{Ù}}{a}(Z_u^t)\overset{\text{Ù}}{b}(Z_u^t) \tag{15}$$

$$P(Z_u^t, F_u^t = 0 \mid I_u^{1:T}) = \mathring{a}(Z_u^t)\mathring{b}(Z_u^t) \tag{16}$$

The posterior probability of state Z_u^t given a sequence of observations can be obtained from the above two formulas (15) and (16).

$$P(Z_u^t \mid I_u^{1:T}) = \mathring{a}(Z_u^t)\mathring{b}(Z_u^t) + \overset{\text{Ù}}{a}(Z_u^t)\overset{\text{Ù}}{b}(Z_u^t) \tag{17}$$

Secondly, two successive states consist of these two states with active or inactive status. Hence, the joint probability of two successive states given the observations listed in formula (4) can be inferred as:

$$P(Z_u^{t-1}, Z_u^t \mid I_u^{1:T}) = \mathop{\mathring{a}}\limits_{F_u^t}\mathop{\mathring{a}}\limits_{F_u^{t-1}} P(Z_u^{t-1}, F_u^{t-1}, Z_u^t, F_u^t \mid I_u^{1:T}) \tag{18}$$

where the joint posterior distributions of two successive latent states with active or inactive status given the sequence of observations are derived from the forward and backward variables.

$$P(Z_u^{t-1}, F_u^{t-1}, Z_u^t, F_u^t \mid I_u^{1:T}) = \mathring{a}(Z_u^{t-1})P(I_u^t \mid Z_u^t, F_u^t)P(Z_u^t \mid Z_u^{t-1})P(F_u^t \mid Z_u^t)\mathring{b}(Z_u^t) / P(I_u^t \mid I_u^{1:t-1}) \tag{19}$$

In the Maximization step, we estimate the parameter Θ based on maximum a posterior (MAP). As Wang and Blunsom (2013) and Sahoo et al. (2012) pointed out, Dirichlet priors can be placed on the initial probability, each row of transitive probability, and each row of emitting observation probability given a certain latent state, i.e.:

$$p \sim \mathit{Dirichlet}(x \mid a_1, ¼, a_K) \text{ where } a_i = a / K \tag{20}$$

$$A_{j,:} \sim \mathit{Dirichlet}(x \mid a_1, ¼, a_K) \text{ , where } a_i = a / K \tag{21}$$

$$\theta_{k,:} \sim Dirichlet(x|\alpha_1, \alpha_2, \dots, \alpha_{|I|}),\ where\ \alpha_i = \alpha/|I| \tag{22}$$

In our experiments, we set α to 100. The MAP of the parameters can be inferred as follows:

$$\hat{\pi}_k = \frac{\sum_u P(Z_u^1 = k \mid I_u^{1:T}; \Theta^{old}) + \alpha_k - 1}{\sum_u \sum_k P(Z_u^1 = k \mid I_u^{1:T}; \Theta^{old}) + \alpha - K} \tag{23}$$

$$\hat{A}_{jk} = \frac{\sum_u \sum_t^T P(Z_u^{t-1} = j, Z_u^t = k \mid I_u^{1:T}; \Theta^{old}) + \alpha_k - 1}{\sum_u \sum_t^T \sum_l P(Z_u^{t-1} = j, Z_u^t = l \mid I_u^{1:T}; \Theta^{old}) + \alpha - K} \tag{24}$$

$$\hat{\theta}_k(I_j) = \frac{\sum_u \sum_{t=1}^T P(Z_u^t = k \mid I_u^{1:T}; \Theta^{old}) \sum_j 1_i(I_{uj}^t) + \alpha_j - 1}{\sum_u \sum_{t=1}^T P(Z_u^t = k \mid I_u^{1:T}; \Theta^{old}) N_u^t + \alpha - |I|} \tag{25}$$

The variable ω follows Bernoulli distributions, and their conjugate priors are also Dirichlet distributions. We can derive the parameters as follows:

$$\hat{\omega}_k = \frac{\sum_u \sum_{t=1}^T P(Z_u^t = k, F_u^t = 1 \mid I_u^{1:T}; \Theta^{old}) + \kappa - 1}{\sum_u \sum_{t=1}^T P(Z_u^t = k \mid I_u^{1:T}; \Theta^{old}) + \kappa + \eta - 2} \tag{26}$$

where κ and η are the parameters of Dirichlet distribution, and we set them to 1.001. Because the parameters of NBD, i.e., a and b, cannot be solved in closed form, we employ the Newton method to get their numerical values.

Prediction

With the estimated model parameters, we can derive the latent states of users' interests and whether they are active or not in the next time period. Based on this information, we can predict users' consumptions of items in the next period.

The probability of each item that may be viewed can be derived from the active latent states as follows:

$$P(i \in I_u^{t+1}) = \sum_k P(i \in I_u^{t+1} \mid Z_u^{t+1} = k, F_u^{t+1} = 1) P(Z_u^{t+1} = k, F_u^{t+1} = 1) \tag{27}$$

With various numbers of items, N_u^t, preferred by user u, item i is included in this set. The probability of item i given the active latent state in formula (27) can be calculated as:

$$P(i \in I_u^{t+1} \mid Z_u^{t+1} = k, F_u^{t+1} = 1) = \sum_{N_u^t = 1}^{\infty} P(i \in I_u^{t+1} \mid N_u^{t+1}; \theta_k) P(N_u^{t+1} | a_k, b_k) = 1 - (1 + b_k \theta_{ki})^{-a_k} \tag{28}$$

In formula (27), the probability of each active state can be seen as the sum of the products of the last states and the transitive probability.

$$\begin{aligned} P(Z_u^{t+1} = k, F_u^{t+1} = 1) &= \sum_l \sum_{F_u^t} P(Z_u^{t+1} = k, F_u^{t+1} = 1, Z_u^t = l, F_u^t) \\ &= \sum_l \sum_{F_u^t} P(Z_u^{t+1} = k, F_u^{t+1} = 1 \mid Z_u^t = l, F_u^t) P(Z_u^t = l, F_u^t) \\ &= \sum_l \sum_{F_u^t} P(Z_u^{t+1} = k \mid Z_u^t = l) P(F_u^{t+1} = 1 \mid Z_u^{t+1} = k) P(Z_u^t = l, F_u^t) \end{aligned} \tag{29}$$

Experiment

Experiment Setup

We evaluate the proposed method using the Netflix dataset[1] that comes from Netflix contest. This dataset was collected from 1999 to 2005. It includes more than 100 million ratings given by approximately 480,000 users on over 17,000 movies.

In the experiment, we manipulate the dataset to get datasets with different sparsity levels and also keep the experiment manageable. First, we filter out items without enough users' consumption. We only keep the movies which are consumed by at least 2,000 users, which provides us 5,264 movies. Second, we filter out users without enough activity. We vary the required number of consumptions for each user to be from 100 to 1,500. By setting the smaller threshold, more users (with less consumption) are kept in the dataset, thus leading to a higher sparsity level. Finally, we randomly select 2,000 users from the filtered dataset to keep the data size manageable.

We create 5 datasets by varying the required users for each item condition. Table 1 reports the statistics of the datasets. In the table, we adopt the sparsity level defined by Sarwar et al. (2000):

$$S_g = 1 - \frac{N_T}{N_U \times N_I} \tag{30}$$

where S_g is the level of data sparsity, N_T is the number of user-item pair transactions, N_U and N_I are the number of users and items respectively. A larger value means a higher data sparsity.

Table 1. The Filtered Datasets

	Data I	Data II	Data III	Data IV	Data V
Filter: # items for each user	>100	>500	>800	>1000	>1500
# randomly selected users	2,000	2,000	2,000	2,000	2,000
# items	5,264	5,264	5,264	5,264	5,264
# user-item interactions	698,519	1,581,031	2,142,706	2,525,057	3,389,121
sparsity	0.934	0.850	0.796	0.760	0.678

In our experiments, we used a sliding window approach that divides the dataset into training/testing sets at the monthly level. We partition the whole dataset into time slices according to calendar month. The sliding window can hold the first *n* time slices as the training set for estimating the model parameters and the next time slice, i.e., the (*n+1*)-th month, as the test set for evaluation. Then we shift the time window of training data by one month and consider the next n time slices from the 2nd month as the training set and the (n+2)-th month as the evaluation test set. We set the size of sliding window as n=48 months, and conduct 24 rounds of experiments to the end of the dataset. We perform top-5 and top-10 predictions.

Evaluation Metrics

To evaluate the performance of the proposed model, we apply the commonly used metrics, precision, recall, and F-measure, on the top-N recommendations. In our experiments, we compare with the benchmark recommendation methods on N=5 and N=10. The precision measure refers to the proportion of recommended relevant items for users retrieved in the predictions. The recall measure refers to the proportion of recommended relevant items for users retrieved in the test data. When we increase or decrease the number of recommendations, one of them strengthens while the other weakens. To balance the precision and recall, the F-measure combines the precision and recall into a single metric for comparison: F1=2´ P´ R/(P+R).

[1] http://www.netflix.com

Baseline Algorithms

In this research, we compare our prosed model with the following three state-of-the-art methods.

1. **HMM Model** (Sahoo et al. 2012), which is the ancestor of our model and employs latent interest state transition to model users' dynamic interests.

2. **User-based Collaborative Filtering** (UCF) (Jannach et al. 2010), one classic recommendation method. It assumes similar users have common interests. This method employs the Pearson Correlation (Wang et al. 2006; Ziegler and Lausen 2004) to measure the similarity between users and make recommendations.

3. **Singular Vector Decomposition** (SVD) (Kim and Cho 2003), which conducts dimensionality reduction to alleviate the sparsity problem. This method performed SVD on user-item ratings as fomula (31) and calculated the similarities of decomposed vectors $U_k\sqrt{S_k}$ in the user-based collaborative filtering.

$$R = U_k S_k V_k^T \tag{31}$$

We repeated the experiments in each experiment setup 10 times and conducted pairwise t-tests to compare our proposed approach with the baseline methods.

Results and Discussion

Table 2. Recommendation Performance

	Top-5 Recommendation			Top-10 Recommendation		
Alg.	P	R	F1	P	R	F1
Data V				Sparsity: 0.678		
IHMM (K=40)	**0.0902**	**0.0229**	**0.0362**	**0.0817**	**0.0412**	**0.0539**
HMM(K=60)	**0.0863**	**0.0217**	**0.0343**	**0.0766**	**0.0390**	**0.0509**
SVDCF	0.0271	0.0066	0.0105	0.0264	0.0129	0.0171
UBCF	0.0239	0.0057	0.0092	0.0229	0.0110	0.0147
Data IV				Sparsity: 0.760		
IHMM (K=40)	**0.0688*****	**0.0211*****	**0.0320*****	**0.0622****	**0.0378*****	**0.0465*****
HMM(K=30)	0.0572	0.0179	0.0270	0.0520	0.0323	0.0394
SVDCF	0.0225	0.0068	0.0103	0.0229	0.0139	0.0171
UBCF	0.0210	0.0063	0.0096	0.0207	0.0126	0.0155
Data III				Sparsity: 0. 796		
IHMM (K=30)	**0.0674*****	**0.0228*****	**0.0337*****	**0.0612*****	**0.0412*****	**0.0485*****
HMM(K=50)	0.0557	0.0190	0.0280	0.0504	0.0341	0.0401
SVDCF	0.0245	0.0081	0.0120	0.0236	0.0156	0.0185
UBCF	0.0213	0.0069	0.0102	0.0212	0.0138	0.0165
Data II				Sparsity: 0.850		
IHMM (K=40)	**0.0446*****	**0.0196*****	**0.0269*****	**0.0397*****	**0.0342*****	**0.0363*****
HMM(K=50)	0.0339	0.0145	0.0201	0.0308	0.0263	0.0279
SVDCF	0.0152	0.0065	0.0090	0.0152	0.0130	0.0139
UBCF	0.0155	0.0063	0.0089	0.0146	0.0121	0.0131
Data I				Sparsity: 0.934		
IHMM (K=40)	**0.0148****	**0.0108****	**0.0121****	**0.0134*****	**0.0197*****	**0.0154*****

HMM(K=40)	0.0116	0.0087	0.0096	0.0102	0.0151	0.0116
SVDCF	0.0099	0.0068	0.0077	0.0097	0.0137	0.0107
UBCF	0.0076	0.0056	0.0061	0.0098	0.0137	0.0108

The significance level between the largest and the second largest value in each column: p<0.1 *; p<0.05 **; p< 0.001 ***;
The numbers in bold are not significantly different from the largest number in each column.

Table 2 presents the different algorithms' performances on the datasets, in which we highlight the highest value and show the significance level compared with the other values in each column. As we can see, the recommendation performance generally increases with reduction of data sparsity, which is consistent with prior knowledge. The experiment results also demonstrate the advantage of our proposed IHMM approach over existing methods. It is always in the group of best models and has significantly better performance than all other models except in Data V. The HMM approach is consistently the second best model. In Data V (sparsity=0.678), it does not have a significant difference from the IHMM. In other datasets, the IHMM significantly outperforms the HMM (at 95% to 99.9% confidence level), and the IHMM achieves about 15%~25% performance improvement over the HMM model.

The other three algorithms have lower performance as compared with the two HMM models; UCF and SVD have a similar level of performance. The SVD model generally has a better performance than UCF due to its dimensionality reduction design. The IHMM and HMM methods outperform the SVDCF and UBCF methods, because the evolution of users' interests plays a major role in the datasets.

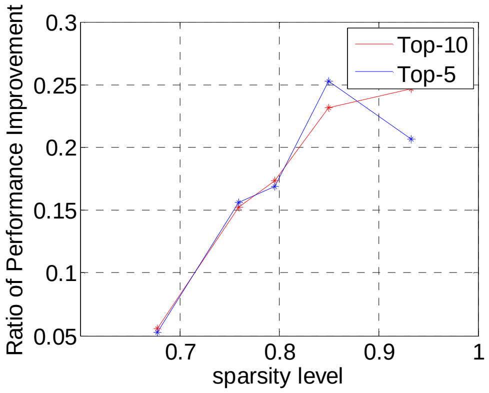

Figure 2. F-measure Improvement over HMM

In Figure 2, we report the F-measure improvement of IHMM over HMM in Top-10 and Top-5 recommendations. As we can observe, the performance improvement of the IHMM generally increases with the level of sparsity. We believe that the advantage of our proposed model comes from its ability to deal with the discontinuous user-item transactions over time.

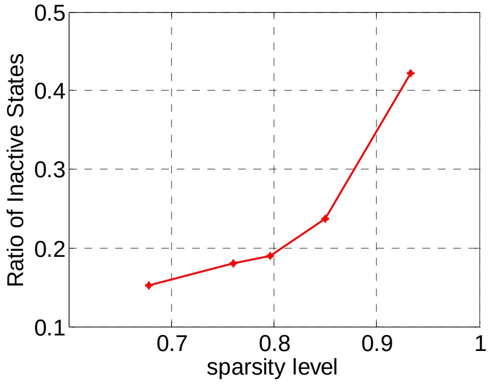

Figure 3. The Ratio of Inactive States in IHMM

In Figure 3, we report the percentage of inactive states of users as identified by our proposed IHMM model. As we can see, when the level of data sparsity increases, the percentage of users in the inactive state also increases. Our proposed IHMM approach does react to the percentage of idle users in the data.

The experiment results clearly demonstrate the advantage of our proposed IHMM approach over existing methods. It can model the discontinuous transactions over time, which brings a large recommendation performance improvement.

Conclusion

In our research, we develop an inhibited hidden Markov model to alleviate the data sparsity problem. This model allows each latent state to be active or inactive at one time unit, which can model time periods without user activities in time-dependent recommendation. We derive an EM algorithm to estimate the model parameters and make predictions under the active states. We compare the model with the state-of-the-art recommendation methods on a real-world dataset. The experiments show that our model has a significant improvement over the benchmark algorithms.

Our proposed approach to deal with the sparsity problem has significant implications for practice. For example, in web or document recommendation, users may not log in the platform at some time for various reasons (e.g., they have found target items, or they turn to other platforms), so that the platform fails to capture users' current interests. In e-commerce services, the transactions would be sparse over time because users would not always buy products. In entertainment recommendation (e.g., movies, music), users may be too busy to view the items for some time. In such applications, considering the discontinuous activities in time-dependent recommendation will bring even more benefits.

In the future, we will continue to study the time-dependent recommendation methods. First, we will continue relaxing the assumptions/restrictions on the HMM framework so that the model can be more generic and fit the diverse requirements of real-world applications. Second, we further model the temporal interdependencies on user interest states in HMM models. Thus the state transition may not be pure Markov and can model more complicated user behaviors. Third, we will also consider the scalability of the model for larger datasets. Our target is to build a unified and effective framework for time-dependent recommendation.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China under Grant 71572169, in part by Guang Dong Natural Science Foundation under Grant 2015A030313876, and was partially supported by the Shenzhen Research Institute, City University of Hong Kong.

References

Adomavicius, G., and Tuzhilin, A. 2005. "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.

Azaria, A., Hassidim, A., Kraus, S., Eshkol, A., Weintraub, O., and Netanely, I. 2013. "Movie Recommender System for Profit Maximization," in: *Proceedings of the 7th ACM conference on Recommender systems*. Hong Kong, China: ACM, pp. 121-128.

Bishop, C. M. 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc.

Cacheda, F., Carneiro, V., Fernandez, D., and Formoso, V. 2011. "Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-Performance Recommender Systems," *Acm Transactions on the Web* (5:1).

Chen, Y., Wu, C., Xie, M., and Guo, X. 2011. "Solving the Sparsity Problem in Recommender Systems Using Association Retrieval," *Journal of Computers* (6:9).

Desrosiers, C., and Karypis, G. 2010. "A Novel Approach to Compute Similarities and Its Application to Item Recommendation," in *Pricai 2010: Trends in Artificial Intelligence: 11th Pacific Rim International Conference on Artificial Intelligence,* B.-T. Zhang and M.A. Orgun (eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 39-51.

Ding, Y., and Li, X. 2005. "Time Weight Collaborative Filtering," in: *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. Bremen, Germany: ACM, pp. 485-492.

Ghazarian, S., and Nematbakhsh, M. A. 2015. "Enhancing Memory-Based Collaborative Filtering for Group Recommender System," *Expert Systems With Applications* (42:7), pp. 3801-3812.

Gogna, A., and Majumdar, A. 2015. "Matrix Completion Incorporating Auxiliary Information for Recommender System Design," *Expert Systems With Applications* (42:14), pp. 5789-5799.

Gong, S. 2010. "A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering," *Journal of Software* (5:7).

Gupta, P., Goel, A., Lin, J., Sharma, A., Wang, D., and Zadeh, R. 2013. "Wtf: The Who to Follow Service at Twitter," in: *Proceedings of the 22nd International Conference on World Wide Web*. Rio de Janeiro, Brazil: ACM, pp. 505-514.

Huang, Z., Chen, H., and Zeng, D. 2004. "Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering," *Acm Transactions on Information Systems* (22:1), pp. 116-142.

Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. 2010. *Recommender Systems: An Introduction*. Cambridge University Press.

Jiang, C., Duan, R., Jain, H. K., Liu, S., and Liang, K. 2015. "Hybrid Collaborative Filtering for High-Involvement Products: A Solution to Opinion Sparsity and Dynamics," *Decision Support Systems* (79), pp. 195-208.

Kim, J. K., and Cho, Y. H. 2003. "Using Web Usage Mining and Svd to Improve E-Commerce Recommendation Quality," in *Proceedings of Intelligent Agents and Multi-Agent Systems: 6th Pacific Rim International Workshop on Multi-Agents, Prima 2003,* J. Lee and M. Barley (eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 86-97.

Koren, Y. 2009. "Collaborative Filtering with Temporal Dynamics," in: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Paris, France: ACM, pp. 447-456.

Koren, Y., Bell, R., and Volinsky, C. 2009. "Matrix Factorization Techniques for Recommender Systems," *Computer* (42:8), pp. 30-37.

Liu, J., Wu, C., and Liu, W. 2013. "Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents for Recommendation," *Decision Support Systems* (55:3), pp. 838-850.

Liu, N. N., Zhao, M., Xiang, E., and Yang, Q. 2010. "Online Evolutionary Collaborative Filtering," in: *Proceedings of the Fourth ACM Conference on Recommender Systems*. Barcelona, Spain: ACM, pp. 95-102.

Luo, C., Cai, X., and Chowdhury, N. 2014. "Self-Training Temporal Dynamic Collaborative Filtering," in *Advances in Knowledge Discovery and Data Mining: 18th Pacific-Asia Conference, Pakdd 2014,* V.S.

Tseng, T.B. Ho, Z.-H. Zhou, A.L.P. Chen and H.-Y. Kao (eds.). Cham: Springer International Publishing, pp. 461-472.

Ma, H., Zhou, T. C., Lyu, M. R., and King, I. 2011. "Improving Recommender Systems by Incorporating Social Contextual Information," *Acm Transactions on Information Systems* (29:2).

Ozbal, G., Karaman, H., and Alpaslan, F. N. 2011. "A Content-Boosted Collaborative Filtering Approach for Movie Recommendation Based on Local and Global Similarity and Missing Data Prediction," *The Computer Journal* (54:9), pp. 1535-1546.

Sahoo, N., Singh, P. V., and Mukhopadhyay, T. 2012. "A Hidden Markov Model for Collaborative Filtering," *Mis Quarterly* (36:4), pp. 1329-1356.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2000. "Analysis of Recommendation Algorithms for E-Commerce," in: *Proceedings of the 2nd ACM conference on Electronic commerce*. Minneapolis, Minnesota, USA: ACM, pp. 158-167.

Wang, H., and Li, W.-J. 2015. "Relational Collaborative Topic Regression for Recommender Systems," *Ieee Transactions on Knowledge and Data Engineering* (27:5), pp. 1343-1355.

Wang, J., Vries, A. P. d., and Reinders, M. J. T. 2006. "Unifying User-Based and Item-Based Collaborative Filtering Approaches by Similarity Fusion," in: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. Seattle, Washington, USA: ACM, pp. 501-508.

Wang, P., and Blunsom, P. 2013. "Collapsed Variational Bayesian Inference for Hidden Markov Models," *Proceedings of the sisteenth International Conference on Articial Intelligence and Statistics*, pp. 599-607.

Xiang, L., and Yang, Q. 2009. "Time-Dependent Models in Collaborative Filtering Based Recommender System," in: *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology* IEEE Computer Society, pp. 450-457.

Xiao, B., and Benbasat, I. 2007. "E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact," *Mis Quarterly* (31:1), pp. 137-209.

Xiong, L., Chen, X., Huang, T.-K., Schneider, J., and Carbonell, J. G. 2010. "Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization," in *Proceedings of the 2010 Siam International Conference on Data Mining*. pp. 211-222.

Yin, H., Chen, B. C. L., Hu, Z., and Zhou, X. 2015. "Dynamic User Modeling in Social Media Systems," *Acm Transactions on Information Systems* (33:3).

Yu, H., and Li, Z. 2010. "A Collaborative Filtering Method Based on the Forgetting Curve," in: *Proceedings of the 2010 International Conference on Web Information Systems and Mining - Volume 01*. IEEE Computer Society, pp. 183-187.

Ziegler, C.-N., and Lausen, G. 2004. "Analyzing Correlation between Trust and User Similarity in Online Communities," in *Trust Management*, C. Jensen, S. Poslad and T. Dimitrakos (eds.). Springer Berlin Heidelberg, pp. 251-265.