Just in Time Recommendations - Modeling the Dynamics of Boredom in Activity Streams

Komal Kapoor, Karthik Subbian,
Jaideep Srivastava
Department of Computer Science
University of Minnesota, Minneapolis, MN 55455
{kapoor,karthik,srivasta}@cs.umn.edu

Paul Schrater
Department of Psychology & Computer Science,
University of Minnesota
Minneapolis, MN 55455
schrater@cs.umn.edu

ABSTRACT

Recommendation methods have mainly dealt with the problem of recommending new items to the user while *user visitation behavior* to the familiar items (items which have been consumed before) are little understood. In this paper, we analyze user activity streams and show that user's temporal consumption of familiar items is driven by boredom. Specifically, users move on to a different item when bored and return to the same item when their interest is restored. To model this behavior we include two latent psychological states of preference for items - sensitization and boredom. In the sensitization state the user is highly engaged with the item, while in the boredom state the user is disinterested. We model this behavior using a Hidden Semi-Markov Model for the gaps between user consumption activities. We show that our model performs much better than the state-of-the-art temporal recommendation models at predicting the revisit time to the item. Moreover, we attribute two main reasons for this: (1) recommending items that are not in the bored state for the user, (2) recommending items where user has restored her interests.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; H.2.8 [**Database Management**]: Database Applications - Data Mining

General Terms

Algorithms, Experimentation, Human Factors, Measurement

Keywords

Dynamic Preferences; Boredom; Temporal Recommender Systems; Activity Streams

1. INTRODUCTION

> *"Boring is the right thought at the wrong time"*
> - Jack Gardner, Words Are Not Things

Recommendation systems are portals to the world of information, as they facilitate and control users interactions with content. The success of these recommendation systems directly depends on the quality of user engagement. The existing internet platforms (such as Last.fm, Netflix.com) allow users to engage with two types of items in a session: new and familiar. For instance, in the Last.fm music dataset[1], on average 23% of a user's interactions are with the new items and the rest are with the familiar items. However, most of the existing models [27, 28, 17, 38, 16, 30, 37, 8, 32] deal only with the recommendation of new items to the user, while understanding *user consumption choices* for the familiar items remains mostly unexplored.

Changing preferences cause the user interest in familiar items to be sensitive to time. Existing temporal models [39, 26, 6, 11, 29, 24] have largely focused on predicting future rating value for a user-item pair using time dynamics. A popular approach is to use time decaying functions to characterize the rating behavior of the user over time [11]. Others estimate the temporal interest of a user for a particular item by combining the user, item and time (latent) factors [27, 17]. While these methods are time-sensitive, understanding the temporal dynamics of user behavior is not their main focus. More specifically, they do not answer the question, *"When the user would visit, revisit or engage with an item?"*, rather they answer "What is the rating of the user-item pair in future?". As a result, such methods do not adequately adapt to the temporal patterns in users engagement with items.

In this work, we model the time-gap between successive consumption activities of a user in the activity stream by specifically focusing on the psychological state of boredom. Users often get bored with a particular item they were engaging with before and move on to a different item of interest. This is similar to an user listening to a single song multiple times or watching multiple movies from a single genre and then switching to a different album or movie genre after certain period of engagement. Mostly they return to the original item of interest after a *gap period*. Such temporal patterns in item consumption significantly impact recommendation design for these systems.

The *gap-behavior* in activity streams is governed by two important content consumption characteristics: (1) user is definitely not interested in an item she is bored of (despite its popularity and her own past interest) and (2) user may revisit the item, if her interest is restored. This is an important observation in consumer research in order to understand

WSDM'15, February 2–6, 2015, Shanghai, China.

http://dx.doi.org/10.1145/2684822.2685306.

---

[1] See experiments section for more details

the changing consumer preferences [31, 14]. We extend this idea further using behavioral psychology to represent these characteristics as two important states of user behavior [12]: sensitization and boredom. In the sensitization state the user is highly engaged with the item, while in the boredom state the user is disinterested. The activity gap characterizes these two states in a most natural way. In the sensitized state the activity gaps are quite small as the user actively revisits the item and in the boredom state the gap is relatively large. The duration in each state and gap lengths may vary depending on the user and item characteristics.

Surprisingly, most of the related work assume that the popular and well rated items by the user are good choices for recommendation. These models completely ignore the fact that the *user may get bored* of these recommendations, despite her past interactions. We perform several experiments in this paper to confirm that sensitization and boredom states exist in user activity streams. Moreover, we show that such behavioral models can predict the revisit time more accurately than existing state-of-the-art techniques.

1.1 Contributions and Organization

We explicitly model user latent psychological states, *sensitization* and *boredom*, using a Hidden Semi-Markov Model (HSMM) and use the model to predict the *the gap between user activities*. The model works in an *online manner* which is well-suited for activity streams. Furthermore, our model is flexible enough to compute a preference score for items as a function of time. We use this flexibility to propose a STiC recommender that ranks familiar items based on the dynamic preference score. Our model is found to be better suited for the recommending task than several state-of-the-art baselines [25, 36, 11, 26, 9].

There are three important results shown in this work. Existing time-sensitive recommendation models are good at predicting ratings for the future, but do not perform well in predicting the revisit time of the user. We demonstrate through our model and experiments that activity streams exhibit two important psychological states of user behavior: sensitization and boredom. Moreover, to the best of our knowledge, this is the first work that talks about modeling gap between user activities using latent psychological states to understand the dynamics of user's consumption behavior.

The paper is organized as follows. In the remainder of this section we discuss the related work. In section 2, we discuss the temporal content consumption behavior using the semi-markov model. We describe the the dataset and the details of the model estimation process in section 3. We also validate our model by comparing our approach to several variants in this section. Followed in section 4 we evaluate our approach on a recommendation task and compare it against popular baselines, such as SVD++, TimeSVD++, Tensor-ALS, and Restricted Boltzmann Machine (RBM). We present the conclusion in section 5.

1.2 Related Work

The problem of recommending interesting items to users based on their history of past ratings and user profile has been well-studied for a few decades now. Some of these approaches take advantage of historical ratings and are referred to as "Collaborative filtering" methods [38, 16, 30, 37]. While the other that make use of the user-profile attributes are called "Content-based filtering" techniques [35, 32]. There are several approaches that combine these techniques and are referred to as "Hybrid" [8, 32]. There are many survey articles [33, 1] that discuss a variety of these approaches. The recommendation problem can be mapped to a standard classification setting, hence latent factor models [27, 17] and dimensionality reduction techniques are also applied. As these problems can be treated as matrix completion problems, matrix factorization [27] based models are also quite widely used.

There are several recent related works that discuss the importance of understanding the changing user interests over time [39, 26, 6, 11, 29, 24]. Most of these penalize the objective or use a corrective scheme for accommodating the changing preferences, rather than explicitly modeling them. Many temporal models for recommendation were designed to detect drifts in users interests and altered their algorithms accordingly [6, 26]. Other methods, have used seasonality and trends [3] as additional context for segmenting the user ratings. There are also tensor factorization [9] approaches, that extend the matrix factorization [41] techniques to include the temporal component.

There has also been some research on implicit feedback data sets [18, 3]. However, most of these works do not explicitly model the user behavioral states which is essential, as shown in this work, to estimate the user revisit time for an item. Understanding future preferences is not specific to recommender systems, and have received much interest in several other fields, such as consumer research. The relationship between repetition of a stimulus (such as food, drinks, commodity items etc.) and its attractiveness has been modeled using an inverted-U shaped function. This relationship was used by McAlister to propose the dynamics attribute satiation model of consumer choice applied to soft drink consumption behavior [31]. More general consumer choice models were later introduced which accommodated either a short term loyalty for the last purchased brand or a devaluation of the last purchased brand [4, 14, 19]. There are also some recent progress on dynamic content consumption analysis [2]. However, most of these approaches do not model the explicit user behavioral states in estimating the time-sensitive future preferences. Furthermore, several of these consumer research approaches are based on questionnaires and surveys.

2. TEMPORAL CONTENT CONSUMPTION

We identify two types of temporal dependencies in the consumption of items:

1. *Reinforcing response*: Systematic exploitation of our recent choices aids our future decision making. As a result, we find ourselves sticking to items such as listening to the same music bands again and again, watching the same kinds of movies and frequenting the same types of restaurant etc. Consumer research scientists have identified this effect as inertia or a short term loyalty for the last purchased brand [20].

2. *Devaluing response*: Psychologists have associated repetitive exposures to stimuli with satiation and repulsion [15]. Stimulus satiation often produce shifts in interests and other variety and novelty seeking behavior [20]. Satiation is identified as a temporary phenomenon which diminishes with time due to forgetting [15].

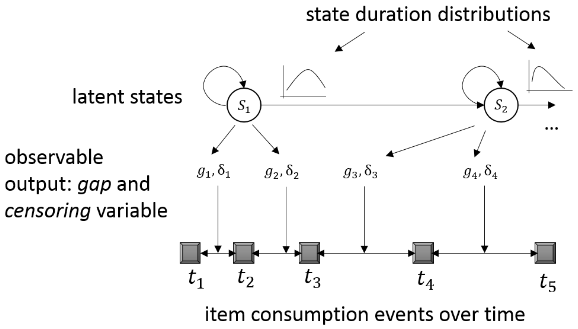

Figure 1: Using observed gap sequence and censoring variable for training a latent state model for item consumption.

The reinforcing and devaluing response is closely associated to a user's content consumption behavior in activity streams [21]. In this work, we model these two response characteristics with psychological preference states of *sensitization* and *boredom*. An item in the sensitization state is consumed rapidly with *small* gaps between its successive consumptions. A *longer* time gap characterizes temporary boredom with the item followed by forgetting. In other words, these states characterize an overall *likeness* for each item. An item with high likeness score takes longer to devalue and recur earlier than an item with relatively low likeness score.

We *explicitly model* these psychological states in this work. We also characterize user's preference for an item as a function of these psychological states using hazard functions which we will discuss a bit later.

2.1 A Semi-Markov Model

The gaps between successive consumptions of an item help us characterize the psychological preference states of the users. We propose a latent state dynamic model for item consumption to infer user preference states. We specifically use a hidden semi-Markov model (HSMM) because of it's ability to model both the consumption gaps (emission distribution) and the time spend by an item in a particular state (state duration distribution).

Let us consider an item i consumed by the user u at times t_1^{ui}, $t_2^{ui} \dots t_n^{ui}$, where t_n^{ui} is the last consumption event for the item in the observation period. The gap observations g_1^{ui}, $g_2^{ui} \dots g_n^{ui}$ denote the time gap between the consumption events, such that $g_x^{ui} = t_{x+1}^{ui} - t_x^{ui}$, for $x = 1 \dots (n-1)$ and $g_n^{ui} = T - t_n^{ui}$, where T is time of the end of the observation period. The last gap length observation is incomplete as we haven't observed the next return for that item yet. Such observations whose values are only known to be larger than a certain value are said to be right censored and are handled using a special status variable (δ_t^{ui}). The status variable is set to 0 for censored observations and and is set to 1 otherwise. It is important to handle censored observations while modeling duration data to prevent a bias towards smaller durations [22]. The $\{g, \delta\}_{1\dots n}^{ui}$ constitute the observable output from the model. This is shown in Figure 1. For simplicity, we drop the superscript ui and it is assumed, unless otherwise stated, that variables are always defined with respect to a particular user and item.

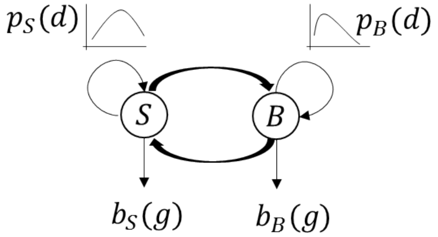

Figure 2: The hidden semi-markov model for gap length sequence.

We include two latent psychological states in the model to capture the states of *Sensitization* (S) and *Boredom* (B). Each state is further associated with an emission density distribution b_m (and a cumulative distribution B_m) for the next gap length g for $m \in \{S, B\}$. More formally, $b_m(g) = P(G = g|m)$, where $P(.)$ is a state-conditioned distribution on gap-length random variable (G). The likelihood of an observed output $\{g, \delta\}$ for a state m can be computed as: $P(\{g, \delta\}|m) = (1 - \delta) * b_m(g) + \delta * (1 - B_m(g))$. Here, the likelihood for a data which is not censored is the probability density function $b_m(g)$, while the likelihood of a censored data is equal to the probability of $P(G > g) = (1 - B_m(g))$.

The semi-markov model allows us to explicitly model state durations, which is the time an item spends in a particular state before transitioning to another state. We denote the duration density distribution by $p_m(D = d)$, where D is the duration random variable. Figure 2 displays the discussed parameters of our model. We model $\log(G)$ (rather than G) to include non-linearity in the perception of time [40]. A parametric form is assumed for the emission and the state duration distributions: $b_m(\log(g)) = \text{Log-logistic}(\mu_m, \sigma_m))$ and $p_m(d) = \text{Gamma}(\alpha_m, \beta_m)$. Our choice of parametric form allows us to capture time dependence characteristics of our data discussed further in section 3.3. The complete set of model parameters include $\lambda = (A, \pi, b_m(g), p_m(d))$, where π denotes the initial state probability distribution over m latent states and A denotes the transition probability matrix between those states. For our model with two latent states $A(m, n) = 1$ for $m \neq n$ and 0 otherwise.

2.2 Prediction

Given the model parameters, and the observed gap sequence, we an use the HSMM model to track the past preference states of the user and make predictions about her future behavior. A good reference for the estimation and inference methodologies for HSMM can be found here [43, 42]. In this subsection, we briefly describe the prediction procedures relevant for our discussion.

At any point let $t_1 \dots t_n$ denote the observed consumption events for an item, $g_{1\dots(n-1)}$ denote the corresponding gap length observations. The model parameters (λ) are estimated via maximum likelihood estimation using the forward-backward algorithm [43]. Using inference, we can compute a distribution for the latent states variables $s_{1\dots(n-1)}$, corresponding to the gap observations, using the entire observed gap sequence, i.e. $P(s_i|g_{1\dots(n-1)}, \lambda)$ for $i = 1 \dots (n-1)$ using the forward-backward algorithm. A one-step lookahead using the forward algorithm allows us to also predict the distribution for the next latent state s_n of the item.

For brevity, we denote this distribution as $\mathfrak{s}_\mathfrak{n}$ such that $\mathfrak{s}_\mathfrak{n}(m) = P(s_n = m|g_{1...(n-1)}, \lambda)$. Since we have only two states, $\mathfrak{s}_\mathfrak{n}(S) = 1 - \mathfrak{s}_\mathfrak{n}(B)$.

We compute the expected gap till the next consumption of the item ($\mathbb{E}(G_n|g_{1...(n-1)}, \lambda)$) using the state conditioned emission distributions as follows. The expectation of the state emission distribution provides us the expected gap length conditioned on the item state and model parameters ($\mathbb{E}(G|m, \lambda)$). We then marginalize out the future state variable using the next state distribution ($\mathfrak{s}_\mathfrak{n}$) to compute the expectation for the next gap length;

$$\mathbb{E}(G_n|g_{1...(n-1)}, \lambda) = \mathfrak{s}_\mathfrak{n}(S) * \mathbb{E}(G|S, \lambda) + \mathfrak{s}_\mathfrak{n}(B) * \mathbb{E}(G|B, \lambda) \ . \tag{1}$$

We further obtain a dynamic measure of item consumption rate using techniques from survival analysis. Survival analysis [23, 13] is a field of statistics which deals with duration data, such as the time of occurrence of an event, referred to as *death*. A hazard function is used to compute a temporal measurement of the event rate conditioned on survival until or beyond a certain time computed as follows:

$$h(t) = P(T = t|T >= t) = \frac{f(t)}{1 - F(t)}, \tag{2}$$

where, f and F are the probability density and cumulative distributions. We use the hazard function for the gap length variable to capture the instantaneous rate of an item's consumption given the time since it's last consumption $(t - t_n)$; i.e. $P(G_n = (t - t_n)|G_n >= (t - t_n)|g_{1...(n-1)}, \lambda)$. The hazard function can be directly associated with a user's preference for the item, which provides us a unique mechanism for quantifying user's dynamic preference (DP).

However, here again, we have direct access to the state condition gap distribution, rather than the gap distribution. Hence, the state conditioned dynamic preference score for some time $t > t_n$ is computed as,

$$DP(t|m, \lambda) \ = \ \frac{b_m(t - t_n)}{1 - B_m(t - t_n)}, \tag{3}$$

Furthermore, marginalizing over the predicted state distribution for the future state ($\mathfrak{s}_\mathfrak{n}$) provides us the dynamic preference score for time t given model parameters and the observed gap sequence as follows:

$$DP(t|g_{1...(n-1)}, \lambda) = \frac{\mathfrak{s}_\mathfrak{n}(S) * b_S(t - t_n) + \mathfrak{s}_\mathfrak{n}(B) * b_B(t - t_n)}{\mathfrak{s}_\mathfrak{n}(S) * (1 - B_S(t - t_n)) + \mathfrak{s}_\mathfrak{n}(B) * (1 - B_B(t - t_n))} \ . \tag{4}$$

3. EXPERIMENT SETUP

We apply our HSMM model of item consumption to music listening data. The domain of music is particularly well suited for our analysis, with repetition naturally occurring even at the song level. For other types of domains (e.g. movies, books, clothes, holiday destinations), repetitive behavior emerges at a higher level of abstraction such as by defining similarity clusters on the attributes of the items (genre, trend, categories etc.).

3.1 Data

We use a public dataset from the popular music service Last.fm [7] that contains the complete music listening histories of around 1000 users as recorded until May, 2009. This is also the only publically available dataset, to our knowledge, that provides the comprehensive listing of users choices during a period of time. The dataset contains the song name, the artist name and the timestamp for the different songs the user listened to during this period.

We construct our dataset using a subset of the data comprising the first four months of listening activities for each user. Of this dataset, the first 3 months is used for training and the fourth month is used for testing purposes. During this period a user is seen to listen to multiple songs over time. Her listening activity is further broken down into sessions where a session is defined as a continuous stream of listening activity interrupted by only *small* pauses. Based on visual examination and with the intention of accommodating most of the listening activity of a day in one session, we use 6 hours as the threshold on the gap between two songs for terminating the session. We use these sessions as the unit of time throughout our discussion. Hence, an item consumption at time t for a user corresponds to her listening to the corresponding song in the t-th session.

For each user a set of familiar (I^u) is identified and includes those which have been consumed at least three times during the training period. The training and test data is filtered to remove all users which have less than 10 familiar items. Table 1 summarizes the basic statistics for the final training and testing dataset used for our experiments.

Training Data	No of users	687
	Mean no of familiar items per user	224
	Mean number of sessions per user	68
Test Data	No of users	593
	Mean number of sessions per user	25
	Mean number of familiar items consumed per session	14

Table 1: Dataset statistics.

3.2 Clustering

In Figure 3, we show the cumulative distribution of the number of repeat consumptions of an item in the training period. More than 90% of user-items have fewer than 10 repetitions making it difficult to obtain a statistical estimate of a separate HSMM model for each user-item pair. Instead, we cluster the user-item pairs and train a separate HSMM model for each cluster. The average rate of consumption or *likeness* score f, as defined below, is used for clustering.

$$f = \frac{n^t}{t_{n^t} - t_1 + \epsilon}, \tag{5}$$

where, n^t is the total number of item consumptions during the training period, t_1 and t_{n^t} is the time of the first and last item consumption during the training period. The constant ϵ is the minimum time period over which the average consumption rate is computed.

We consider two approaches for clustering user-item pairs based on the *likeness* score - equal interval binning and k-means clustering. We further consider different number of clusters for partitioning the data. A large number of clusters result in noisy and sparse clusters. On the other hand, too few clusters overgeneralize the model. We set

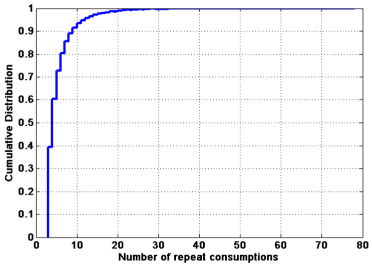

Figure 3: Cumulative distribution of the number of repeatitions of an item for the training data.

aside a validation dataset by removing a 30% random sample from the training data, and use this to evaluate the clustering schemes and the number of clusters. The models are trained on the remaining training data. The performance of a clustering scheme is measured using Root Mean Squared Error (RMSE) between the predicted and observed log-transformed gap length sequences in the validation dataset. The k-means clustering algorithm with 25 clusters is found to perform the best. Our analysis going forward is based on these user-item clusters, and the corresponding estimates of model parameters λ_c.

3.3 Model Parameters

We now analyze the model parameters trained on our dataset and discuss their relationships to the latent psychological states. We also show the existence of sensitization and boredom states through our analysis.

Emission probability distributions Figures 4 (a) and (b) show the emission probability distributions $b_m(\log(g))$ for the two latent states S and B, respectively. The probability distributions are plotted corresponding to each clusters. The log-gap lengths are marked along the x-axis while the y-axis indicates the index for the clusters which are organized in increasing order of likeness scores. The value of the probability distributions (log transformed to highlight the differences between the clusters) for a particular clusters and gap length is indicated by a color. First we note that the for the same cluster, the emission distribution is spread across longer gap lengths for state B than state S. This justifies the nomenclature for the states as we expected items in the sensitization states to be consumed faster than items in the boredom state. Secondly, we find that items which have a higher likeness score have shorter return cycles than items with lower likeness score.

The hazard functions for the two states show significant differences (Figure 4 (c) and (d)). As before, the hazard functions are plotted for log-gap lengths along the x-axis for each cluster, and the different clusters are organized along the y-axis in increasing order of likeness score. For the state S, items have declining hazard function which indicates that the event rate decreases with log-time. On the other hand, the hazard function for the state B gradually increases and then declines. Such a uni-modal shape of the hazard function indicates a peak rate of occurrence at a particular log-time and fits well with our boredom hypothesis. This is the main reason for our choice of log-logistic distribution that fits well both a declining and a uni-modal hazard function.

State duration distributions Figure 5 shows the state duration distributions and the hazard functions for the state duration for the latent states S and B. The state duration length is marked on the x-axis, while the y-axis indicates the clusters. The color is used to denote the magnitude of the log-transformed probability distribution and the hazard functions. First, we find that clusters with lower likeness scores have a shorter dwell time in the sensitization state and longer dwell time in the boredom state than clusters with higher likeness scores. Secondly, the hazard has an increasing shape for both the states which indicates that the rate of moving out of the state increases with time spent in the state. This indicates that items in the sensitization state eventually devalue while those in the boredom state eventually return to the sensitization states when user preferences recover. The gamma distribution allows an increasing/declining hazard function and provides an adequate fit for the temporal dynamics of the state transitions.

3.4 Relaxing Modeling Assumptions

We now consider several relaxations of our model (HSMM) for item consumption and evaluate them at predicting the gap sequences for the items in the dataset. The following relaxations are considered:

1. **HMM** We use a Hidden Markov Model (HMM) to model the timing of item consumption. As before, we consider two latent states S and B and model the emission distributions for each state using a Log-logistic distribution on the log-transformed gap length. The HMM model assumes that the state durations are geometrically distributed and are independent of the time spent in the state. A transition matrix captures the probability of transitioning between states. The complete set of model parameters include $\lambda = (A, \pi, b_m(g))$.
2. **Loglogistic** We do not model the temporal order in the gap sequence. Instead gap lengths between item consumptions are assumed to follow a Log-logistic distribution. Such a model picks up the predominant recency based dynamics in the data producing a declining hazard function for the consumption event. The complete set of model parameters include $\lambda = (\mu, \sigma)$.
3. **Exponential** Consumption events are modeled to occur at a constant rate using an exponential distribution. The model parameters include $\lambda = (\mu)$.

All models are learnt using the training data for the same frequency-based user-item clusters as described earlier, and evaluated on the test data. The performance is measured using the prediction error on the log-transformed gap length sequences in the test data using RMSE. The results are summarized in Table 2. Our HSMM model performs significantly better (p-value$<10^{-5}$) than all the other models which illustrates the value achieved by the different components of our model.

4. STiC RECOMMENDER

A temporal recommendation algorithm based on our item consumption model is proposed and evaluated.

4.1 Design

Our HSMM model, as mentioned earlier, predicts the time when an item would be consumed next based on the psychological state of the user. Further state and time based preference score for the item can be computed using (4). This

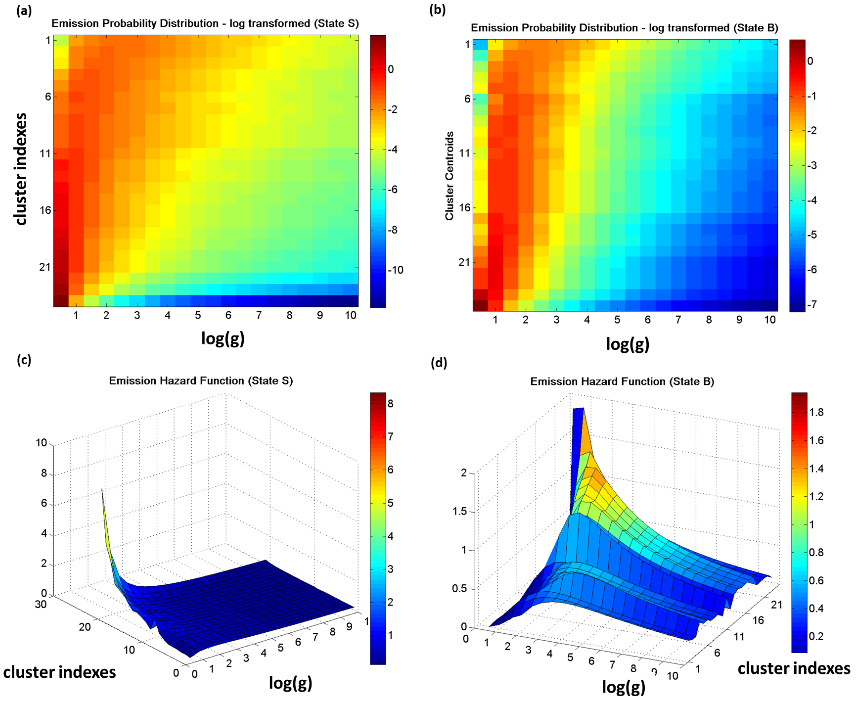

Figure 4: Emission probability and hazard function distributions. The cluster indxes are labeled in increasing order of likeness score. (To be viewed in color)

Model	**RMSE on the Test Data**
HSMM	0.9791
HMM	1.0691
Loglogistic	1.1943
Exponential	1.1860

Table 2: RMSE scores on the log-transformed gap length sequence

provides us valuable information for making time sensitive recommendations to the users. We now propose the (**STiC**) recommender which uses **S**tate and **Ti**me **C**onditioned preference scores for dynamically ranking items. The scores are computed in an online manner for the next user session using her past consumption history and the cluster level model parameters learnt from the training period (λ_c).

4.2 Evaluation

There are certain challenges in evaluating a time-sensitive recommendation based on the dynamic preferences of users. Firstly, a direct assessment of an user's temporal preference is hard to obtain. For example, even when abundant explicit feedback in terms of ratings for items are available, a user rarely rates the same item repeatedly nor does the rating correspond to the consumption preference at that time (as the user may rate the item after arbitrary long time). As a result, we base our model evaluation on actual consumption choices resulting from an activity stream, as it reflects the real-time interests of a user.

We compare our model against various popular static and temporal recommendation methods. Both the training and the test data is transformed into a per user choice matrix (C^u) such that $C^u(i,t) = 1$, if the item i is consumed during the session t, 0 for all the items that are not consumed during that session.

4.2.1 Metrics:

The standard RMSE metric meant for explicit rating data is not applicable to our setup. We consider the following metrics, well suited to implicit datasets [3, 18, 34], for evaluating our model and the comparison baselines. The metrics have been modified to make the evaluation sensitive to time.

1. **T-Precision, T-Recall and T-F_1 measures** Improvements in RMSE scores provide little information on the impact on user experience. Furthermore, since users are generally only recommended a list of top K items, more recently evaluation based on precision, recall and F_1 have become popular [10]. We compare the top-10 recommendation list generated by the model for a user sessions against the actual items consumed by the user in the sessions and compute the precision, recall and F_1. These scores are then averaged across all user sessions in the test period.

2. **T-AUC** The AUC scores measure the likelihood of the recommender to rank preferred items over the not-preferred items. We compute the average AUC score across user sessions in the test period.

3. **T-Rank** The rank metric was recently proposed to evaluate recommenders in the presence of implicit feedback [18]. The metric computes the expected percentile rank of an item selected during the test period

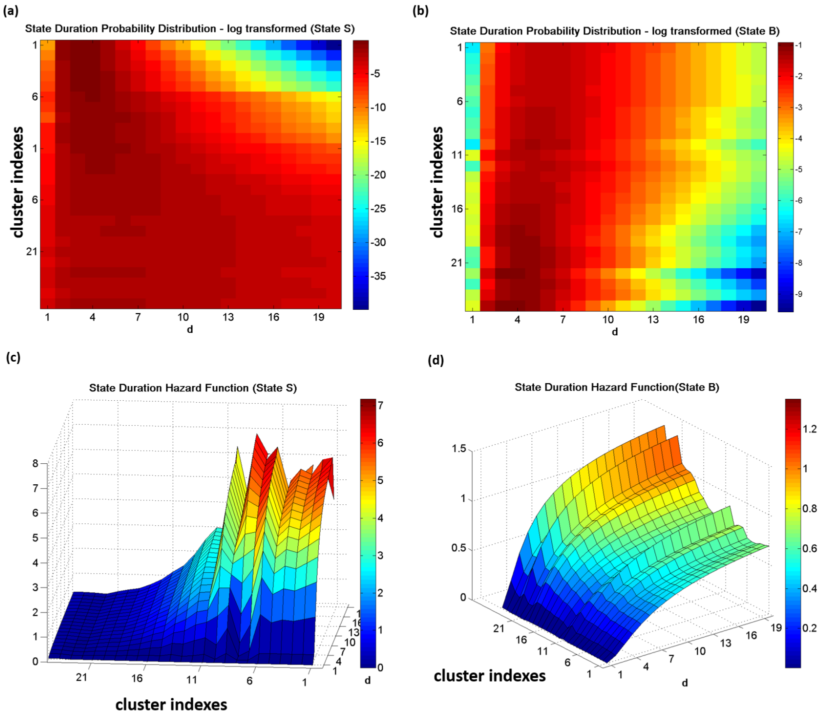

Figure 5: State duration probability and hazard function distributions. The cluster indxes are labeled in increasing order of likeness score. (To be viewed in color)

in the recommender's ranking list. For a temporal setting, session specific rank scores are computed and averaged across all users and session in the test period:

$$\text{T-Rank} = \frac{\sum_{u,i,t} C^u(i,t) * \text{rank}^{ui}(t)}{\sum_{u,i,t} C^u(i,t)}, \qquad (6)$$

where $\text{rank}^{ui}(t)$ denotes the percentile rank of item i in the ranked list of items generated for the user u for the session t.

It should be noted that for a recommender, higher values of T-Precision, T-Recall, T-F_1, and T-AUC scores and low values of T-Rank scores are preferred.

4.2.2 Baselines:

We compare the STiC recommender against several state-of-the-art static and temporal recommendation approaches. Some of the approaches have been modified to work with implicit activity data. We further use the validation dataset to obtain the optimal parameters for the baselines.

1. **Static** The model computes a preference score vector by computing the average number of time each item was consumed per user session during the training period. By definition this model is time-insensitive.
2. **SVD++** Matrix factorization based approaches such as Singular Value Decomposition (SVD) are known to perform well when an explicit user-item ratings matrix is known and prediction accuracy is evaluated using RMSE on the user ratings [5]. The SVD++ model is shown to perform better at top-K recommendations than basic SVD and is used for comparison. The implicit data is converted into an explicit rating using the complementary cumulative distribution of a user's item consumptions [3]. Items in the top 80-100% of the distribution are given a rating of 5, those in the 60-80% are given a rating of 4 and so on.
3. **Restricted Boltzmann machines (RBM)** Another time-insensitive baseline includes RBM's, a two-layer undirected graphical models used for collaborative filtering process [36]. In this approach, a conditional multinomial is used to model the columns of the observed rating matrix and a conditional Bernoulli distribution is used for hidden user features. The rating matrix used was same as the SVD++ baseline.
4. **Time-Weighted** Previous research [11] have found that incorporating time by time weighting user ratings (usually using an exponential decay) such that recent ratings are weighted more than old ratings leads to performance improvements. Hence, we compare our model against a time-weighted recommender that computes a temporal preference score vector over the items using an exponential moving average: $P^u(t) = \lambda^u * P^u(t-1) + (1-\lambda^u) * C^u(t-1)$; $P(1) = C(1)$. Here λ^u is the decay weight vector which is learnt from the training dataset using stochastic gradient descent.
5. **TSVD++** The TSVD++ model extends matrix factorization models to incorporates temporal drifts in user interests [26]. Changes in preference factors with time are captured using a linear function. The TSVD++ model is trained using the user choice matrices.

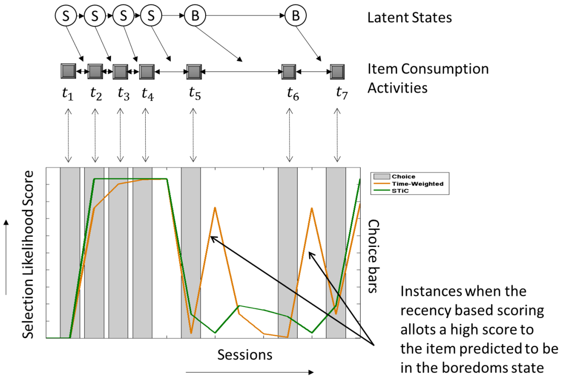

Figure 6: STiC model predictions compared against that of time-weighted model. The top part of the figure shows the STiC model's state predictions for the item. The bottom part displays how the same item is scored by the two models. The item is scored high by the time-weighted model even after the user state has changed (the item has become boring). Instead, the STiC model gives a low score to the item at those instances.

6. **Tensor Factorization (Tensor)** Tensor factorization allows us to further generalize matrix factorization to include time. The binary rating (or activity) matrix along with the time dimension is considered as a three dimensional tensor. A low rank factorization is performed on the tensor by minimizing the total squared error on the observed ratings. Alternating least squares is used to approximate the user, item and time factors. The factors are then combined to reconstruct the complete rating matrix. The implementation details are described in [9].

4.3 Results

The evaluation results are summarized in Table 3. Incorporating time is generally found to improve the performance over non-temporal counterparts [27, 26, 9], as seen from the better performance of temporal models with latent factors, such as *TSVD++* and *Tensor* over *SVD++* and *RBM*. Similarly the *time-weighted* (non-latent temporal) model, performs better compared to its *static* counterpart. Our approach *STiC* outperforms all the baselines, including the latent factor temporal models, as it explicitly models the user behavioral states.

While all of our baseline latent factor models perform well in terms of low RMSE scores (shown in brackets) on the training choice matrix (RBM (0.788227), SVD++ (1.28967), TSVD++ (0.198687), Tensor(0.176084)), they did not perform well in the temporal choice prediction task. Instead, the static and time-weighted models which are trained per user fair much better. Such findings can be explained on two grounds. Firstly, the latent factor model are optimized to minimize the squared error of the predicted to the observed values rather than their ability to rank items based on users preferences. Secondly, they are primarily intended to identify similarities between users and items to discover new items for them. Instead, for our task, we are more interested in predicting the temporal characteristics of user choices for a restricted set of familiar items. This further demonstrates the importance of using gap measurements in predicting the next expected visit of the user to a item. Our STiC model is a hybrid approach that combines the individual likeness scores with a cluster based model for preference dynamics, and is superior to the rest of the models.

We investigate the differences between our STiC Model and the popular time-weighted model (our best performing baseline) in further detail. Our other baselines (which perform significantly worse) are not further considered due to space limitations. A major difference between the time-weighted and the STiC models stems from the fact that the time-weighted model assumes user preferences to be predominantly recency based, while the STiC model captures different user states of sensitization and boredom and allows for both recency and diversity driven behaviors based on the user state. As we discussed in Section 1, this impacts quality of user experience in two important ways: (1) *not recommending* the items that are *boring* or user has lost interest and (2) *recommending* items where the user has restored recent interest. We illustrate below the importance of these two factors through more detailed experiments.

(A) Not recommending items which are boring: We examine the one-step lookahead state predictions made by the STiC model ($\mathfrak{s}_n$) for an item and corresponding observed gap lengths (Figure 6) (a)). For the same item Figure 6 (b) displays the selection likelihood scores, scaled to the same range, as generated by both the models. We find that the time-weighted model continues to score the item based on recency even when the user's preference state for the item, as predicted by the STiC model, has changed. Hence, items which a user is bored of, are scored high by the time-weighted model but not by the STiC model.

In order to generalize our findings across users we allot a time-sensitive boredom score to items;

$$\text{Boredom-Score}(t) = \text{Time till next consumption at time `t'}. \tag{7}$$

We borrow the concept of *future lifetime* [13] from survival analysis to compute the boredom score using our STiC model. The future lifetime is defined for an event as the remaining time till death given survival until a specified time. Given the cumulative distribution (F) over the time of the occurrence of the event and some maximum threshold for time (t_s), the expected future lifetime at t_0 can be computed as:

$$E(T|T > t_0) = \frac{1}{1 - F(t_0)} \sum_{t=t_0}^{t_s} 1 - F(t) \tag{8}$$

For our scenario, the boredom score directly maps to the expected future lifetime for item consumptions. We denote the future gap as random variable G_f and the next future gap as random variable G_{fn}. At some time 't', the gap since the last consumption of the item is $t - t_n$, and the expected next future gap is defined as $\mathbb{E}(G_{fn}|G_n > (t - t_n), g_{1\ldots(n-1)}, \lambda)$. We first compute the state conditioned expected future gap using the state emission distributions:

$$\mathbb{E}(G_f|G > (t - t_n), m, \lambda) = \frac{1}{1 - B_m(t - t_n)} \sum_{s=(t-t_n)}^{t_s} 1 - B_m(s)\,. \tag{9}$$

Model	**T-Precision**	**T-Recall**	**T-F_1**	**T-AUC**	**T-Rank**
Static	0.108	0.1229	0.115	0.5986	0.3827
Time-Weighted	0.133	0.1842	0.1545	0.6542	0.3682
SVD++	0.072	0.1312	0.093	0.5175	0.4766
RBM	0.0862	0.1298	0.1036	0.5436	0.4276
TSVD++	0.0772	0.1001	0.0872	0.571	0.4212
Tensor	0.1031	0.1195	0.1107	0.545	0.3982
STiC	**0.1641**	**0.2148**	**0.1861**	**0.692**	**0.3254**

Table 3: Comparing the STiC model with popular static and temporal recommendation models on a variety of temporal evaluation metrics. The STiC model is found superior to all baselines on all evaluation metrics.

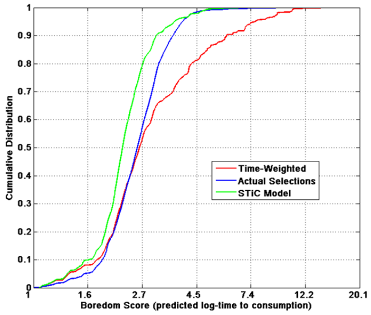

Figure 7: The cumulative distribution for the recommendation likelihood for the time-weighted and the STiC model given boredom scores. Time-weighted model recommends more item with higher boredom scores, while STiC is more conservative than the actual user.

We then marginalizing over the the future state predictions ($\mathfrak{s}_\mathfrak{n}$) to compute the boredom score:

$$\begin{aligned}\text{Boredom-Score}(t) =&\mathbb{E}(G_{fn}|G_n > (t - t_n), g_{1\ldots(n-1)}, \lambda)\\ =&\mathfrak{s}_\mathfrak{n}(S) * \mathbb{E}(G_f|G > (t - t_n), S, \lambda)\\ &+ \mathfrak{s}_\mathfrak{n}(B) * \mathbb{E}(G_f|G > (t - t_n), B, \lambda)\ .\end{aligned} \tag{10}$$

We now map the cumulative distribution of the likelihood to occur in the top-10 recommendation list for the two models; Time-weighted and STiC, against the boredom scores predicted by the STiC model (Figure 7). The threshold t_s is set to 60 (a reasonable high value) sessions. For reference, the actual consumption likelihood of the user is also plotted in the same figure. We find that the time-weighted model recommends more item with higher boredom scores than the STiC model and those actually consumed by the user. The STiC model on the other hand is found to be slightly more conservative than the actual user.

(B) Recommending restored items in addition to sensitized items: The STiC model further allows partitioning the items consumed in a future sessions into two sets: Sensitized and Restored items. If $P(S|g_{1\ldots(n-1)}, \lambda) > P(B|g_{1\ldots(n-1)}, \lambda)$, then the item is allocated to sensitized set. Otherwise the item is added to the restored set.

We use our classification scheme to further compare the recommendation performance on specifically the restored items. Empirically, users were found to consume sensitized items only around 23% of the times. For the rest of the times they consumed items from the restored sets. This suggests that the ability to recommend the restored items is crucial for improving recommendation performance. The Table 4 summarizes the performance scores for the models separated based on the item set. We find that both the time-weighted and the STiC model are extremely good at recommending sensitized items. The time-weighted model, is particular bad at recommending restored items while the STiC model continues to work well.

5. CONCLUSIONS

Understanding the changing user preferences is very important in the context of recommendation. Most of the changing user interests are available in the form of activity streams, where each activity (such as listening to a song or viewing an shopping item) represents the user's interest to a specific item. In this paper, we proposed a behavior-based model for understanding changing user's interests using a hidden semi-Markov model. We used latent psychological states, sensitization and boredom, to represent the user's behavior in this model. We showed that existing state-of-the-art temporal models fails to predict the time of next expected visit of an user to an item as compared to our model. We attribute two main causes for this: (1) not recommending the bored items and (2) recommending the items where an user has restored her interests. In our experiments, we performed several analysis to justify these two reasons, in addition to overall superior performance of our model.

6. ACKNOWLEDGMENTS

We gratefully acknowledge the University of Minnesota Doctoral Dissertation Fellowship, SOBACO grant, the NSF EAGER grant number 00032116 and the DARPA grant number W911NF-12-C-0028 for sponsoring this work.

7. REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[2] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. The dynamics of repeat consumption. In *Proceedings of the 23rd international conference on World wide web*, pages 419–430. International World Wide Web Conferences Steering Committee, 2014.

[3] Linas Baltrunas and Xavier Amatriain. Towards time-dependant recommendation based on implicit feedback. In *Workshop on context-aware recommender systems (CARSŠ09)*, 2009.

[4] Kapil Bawa. Modeling inertia and variety seeking tendencies in brand choice behavior. *Marketing Science*, 9(3):263–278, 1990.

[5] James Bennett and Stan Lanning. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35, 2007.

[6] Huanhuan Cao, Enhong Chen, Jie Yang, and Hui Xiong. Enhancing recommender systems under volatile userinterest

Item set	Model	T-Precision	T-Recall	T-FMeasure	T-Rank
Sensitized items	Time-Weighted	0.1853	0.4752	0.2666	0.0245
	STiC	**0.1956**	**0.4785**	**0.2777**	**0.0189**
Restored items	Time-Weighted	0.0223	0.0634	0.033	0.4428
	STiC	**0.0511**	**0.109**	**0.0696**	**0.3847**

Table 4: Recommendation performance of the STiC and the time-weighted recommender for different item sets. Both the time-weighted and STiC model perform well on sensitized items while, time-weighted is particular bad at recommending restored items compared to STiC.

drifts. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1257–1266. ACM, 2009.

[7] O. Celma. Music recommendation datasets for research. 2010.

[8] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR workshop on recommender systems*, volume 60. Citeseer, 1999.

[9] Pierre Comon, Xavier Luciani, and André LF De Almeida. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23(7-8):393–405, 2009.

[10] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM, 2010.

[11] Yi Ding and Xue Li. Time weight collaborative filtering. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 485–492. ACM, 2005.

[12] John D Eastwood, Alexandra Frischen, Mark J Fenske, and Daniel Smilek. The unengaged mind defining boredom in terms of attention. *Perspectives on Psychological Science*, 7(5):482–495, 2012.

[13] Regina C Elandt-Johnson. *Survival models and data analysis*, volume 110. John Wiley & Sons, 1980.

[14] Moshe Givon. Variety seeking through brand switching. *Marketing Science*, 3(1):1–22, 1984.

[15] Murray Glanzer. Curiosity, exploratory drive, and stimulus satiation. *Psychological Bulletin*, 55(5):302, 1958.

[16] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.

[17] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):89–115, 2004.

[18] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE, 2008.

[19] Barbara E Kahn, Manohar U Kalwani, and Donald G Morrison. Measuring variety-seeking and reinforcement behaviors using panel data. *Journal of Marketing Research*, pages 89–100, 1986.

[20] B.E. Kahn, M.U. Kalwani, and D.G. Morrison. Measuring variety-seeking and reinforcement behaviors using panel data. *Journal of Marketing Research*, pages 89–100, 1986.

[21] Komal Kapoor, Nisheeth Srivastava, Jaideep Srivastava, and Paul Schrater. Measuring spontaneous devaluations in user preferences. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1061–1069. ACM, 2013.

[22] Nicholas M Kiefer. Economic duration data and hazard functions. *Journal of economic literature*, pages 646–679, 1988.

[23] David G Kleinbaum and Mitchel Klein. *Survival analysis.* Springer, 1996.

[24] Noam Koenigstein, Gideon Dror, and Yehuda Koren. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 165–172. ACM, 2011.

[25] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[26] Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[27] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[28] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Recommendation systems: A probabilistic analysis. In *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pages 664–673. IEEE, 1998.

[29] Tong Queue Lee, Young Park, and Yong-Tae Park. A time-based approach to effective recommender systems using implicit feedback. *Expert systems with applications*, 34(4):3055–3062, 2008.

[30] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[31] Leigh McAlister. A dynamic attribute satiation model of variety-seeking behavior. *Journal of Consumer Research*, pages 141–150, 1982.

[32] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, pages 187–192, 2002.

[33] Prem Melville and Vikas Sindhwani. Recommender systems. In *Encyclopedia of machine learning*, pages 829–838. Springer, 2010.

[34] Douglas W Oard, Jinmook Kim, et al. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, pages 81–83. Wollongong, 1998.

[35] Michael J Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.

[36] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.

[37] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[38] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, page 4, 2009.

[39] John Z Sun, Kush R Varshney, and Karthik Subbian. Dynamic matrix factorization: A state space approach. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1897–1900. IEEE, 2012.

[40] Taiki Takahashi. Loss of self-control in intertemporal choice may be attributable to logarithmic time-perception. *Medical hypotheses*, 65(4):691–693, 2005.

[41] Hsiang-Fu Yu, Cho-Jui Hsieh, Si Si, and Inderjit S Dhillon. Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. In *ICDM*, pages 765–774, 2012.

[42] Shun-Zheng Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243, 2010.

[43] Shun-Zheng Yu and Hisashi Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden markov model. *Signal Processing Letters, IEEE*, 10(1):11–14, 2003.