Models of Dynamic User Preferences and their Applications to Recommendation and Retention

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Komal Kapoor

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Prof. Jaideep Srivastava and Prof. Paul R. Schrater

Dec, 2014

Acknowledgements

The four and half years of my graduate school have been an extremely enriching experience for me. This has been largely due to the good fortune of having interacted with several individuals who have inspired, motivated and stood by me during this wonderful journey and to whom I owe the completion of this work. I have found lifelong mentors, guides and friends and I take this opportunity to thank them.

First and foremost, I want to thank my advisers Prof. Jaideep Srivastava and Prof. Paul R. Schrater. Words cannot express how grateful I am to have their guidance and blessings for all these years. I first talked to Prof. Jaideep shortly before deciding to come to United States and pursue PhD at the University of Minnesota. His unconditional faith and confidence in me, even then, had me pushing myself to exceed the expectations he set for me. In addition to his wise advice on academic matters I admire his patience, compassion and respect for everyone. I hope to ingrain the qualities I have learnt from him in my future endeavors.

I am extremely indebted to Prof. Schrater for the hours and hours of rigorous discussions and brain-storming sessions, where we crafted so many wonderful ideas, some of which we got to complete in this time frame while time will tell the faith of the rest. Our meetings left me completely invigorated and inspired towards my research. I have admired his courage and spirit, and find myself fortunate to be a part of his research group.

I also want to thank Prof. Joseph A. Konstan and Prof. Sudipto Banerjee for taking time out from their busy schedules to serve on my thesis committee. They have been most encouraging and supportive, and have provided invaluable feedback for improving and extending my work. I also want to thank Prof. Arindam Banerjee for serving on my preliminary exam committee.

I would dearly want to thank all the current and past DMR lab and Empowered lab members who have made these years so much more enjoyable. They have been constant pillars of support, and have provided their invaluable feedback, suggestions and advice for improving my work.

I am especially thankful to Nisheeth Srivastava and Kyong Jin Shim for being such wonderful mentors to me. I am grateful to Nisheeth Srivastava for introducing me to several instigating research projects that deeply inspired me and crucially shaped the development of my own research agenda.

Last but not the least, I owe my thesis to the support of my family and friends. My parents, my sister and brother, my fiance and the rest of my family have given me their unconditional support throughout this time. I also want to thank my wonderful friends in India who I wish I could spent more time with and my family away from home- my roommates and friends from Minneapolis in whose company these four and half years slipped away so quickly leaving me wanting for more.

Dedication

To Dr. Ram Kapoor and Dr. Rajni Kapoor, my parents who gave me wings to fly and my dear fiance Saral Jain, who has stood by me through all these years.

Abstract

Computational models of preferences are indispensable in today's era of information overload. They help facilitate access to all types of resources such as videos, songs, images etc. via several means such as content recommendation, site personalization and customization, and promotional targeting and marketing. They further serve as important business intelligence tools providing content providers insights to improving their practices. Vanilla models of preferences such as the static and time decay models commonly used today, albeit powerful, are limited in their abilities to cater to the volatile and shifting tastes and needs of the users. On the other hand, researchers in the domain of behavioral psychology have studied various aspects of the formation and evolution of individual preferences over several decades.

Despite several advances, findings from behavioral research have had little or no impact on the design of computational models for dynamic preferences on the web. This is because, most of these studies have been qualitative and/or have relied on carefully constructed user experiments and surveys for testing their methods. The recent proliferation of online interfaces, however, allows the accumulation and analysis of large quantities of user preference logs, opening new avenues for understanding user dynamic behavior via data driven means. In this thesis, we therefore focus on developing a repertoire of tools and techniques for analyzing, modeling and predicting temporal and history dependent dynamics in preferences of online users.

For this purpose, we adapt techniques from survival analysis, a branch of statistics used for analyzing duration data, to empirically measure changes in user preferences from their activity streams. We specifically use hazard functions which allow us to relate user dynamic preferences to user's dynamic choice probabilities for items, a quantity that can be conveniently measured from temporal logs of user consumption behavior. The dynamics in user preferences is further studied by analyzing their consumption behavior separately with respect to their (a) consumption of known (familiar) items; and (b) consumption of new items.

We show that user consumption of a familiar item over time is driven by boredom. That is, we find that users move on to a new item when they get bored and return to the same item when their interest is restored. To model this behavior, we propose a Hidden Semi-Markov Model (HSMM) which includes two latent psychological preference states of the user 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 find that the gaps between consumption activities characterize these two states in the most natural way. We further find that our two state model for item consumption not only better predicts the revisit time of the user for items, but also, improves how items are recommended to the users, compared to existing state-of-the-art. This is because our model has two advantages over other methods. First, by modeling boredom it can avoid devalued items in the user recommendation list and second, by identifying items which the user would want to consume again, it can re-introduce items which have not been consumed for some time.

We further focus on a user's incorporation of new items in their consumption list (novelty seeking). We find that a user's preferences for novelty vary with time and such dynamics can be related to their boredom with familiar items. We then introduce for the first time, a novel approach to selectively incorporate novelty in a user's recommendation list using our prediction of their novelty seeking behavior. We further show that our approach is robust in terms of a new metric for accuracy more suitable to the problem of selective novelty recommendation based on user's novelty seeking preference.

Finally, in the last section of this thesis we use hazard models to estimating the dynamic interest of the user in the content provider. This is achieved by using a Cox Proportional Hazard model to estimate the dynamic rate of a users' return to the service as a function of time since the user's last visit. We use our model to address the problem of retention for web services and show that our model allows better user segmentation based on predicted return time. The model further incorporates several behavioral and temporal features of the users interaction with the service which provides valuable insights to the service's practices.

Based on the experimental findings on various real world datasets, from different sections of the thesis, the benefits of well-grounded dynamics preference models is apparent for improving user experience on the web in several important ways. We hope that the rigorous treatment of the problem of dynamics in user preferences provided in this work, assists and motivates future research in this area.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Decision theory has classically associated a single numerical measure with an item, called its *utility*, to quantify a subject's preferences towards it. This allows one to readily specify the probability P_a^u with which a user u chooses an item a in her choice set O, through formulas like:

$$P_a^u = \frac{U(a)}{\sum_{o \in O} U(o)} \tag{1.1}$$

where, $U(o)$ is the utility associated with item $o \in O$. Such preference models have been extensively used in various applications such as in modeling individual decision making in economics [1], for predicting consumer purchase behavior [2] and in designing personalized assistive agents and recommender systems [3]. However, static utilities fail to explain many kinds of human behaviors observed in practice. Psychological studies have shown that preferences are dynamic, and are affected by the frequency of exposure to a commodity. Moderate exposure is needed to acquire preferences. However, existing preferences spontaneously devalue after repetitive exposure and is associated with the psychological state of boredom or stimulus satiation [4]. At the same time, less frequent repetition can reinstate one's preferences for a commodity, also identified as the mere-exposure effect [5] and is referred to as reinforcing, inertial or sticky behavior [6]. The inherent drive for exploration further constitutes an important element of human behavior which leads individuals towards desiring new and novel content. User's preferences for novelty are known to result from their curiosity for new information [7, 8] or are linked to stimulus satiation responses to familiarity [7]. Such dynamic interactions between user past experiences and their future choices form an important element

of their temporal content consumption process. (Figure 1.1).

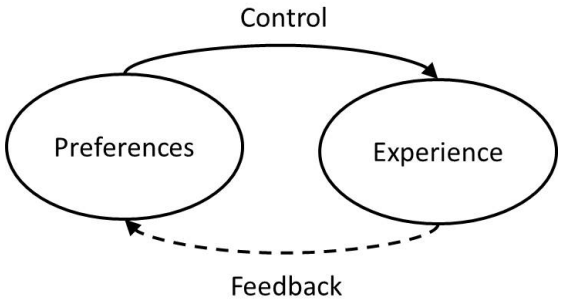


Figure 1.1: The dynamic interaction between user preferences and their choice of content consumed.

Dynamics in preferences have only recently come under the purview of the computer science community due to the increasing need for designing automated agents that can assist humans in their day to day decision making. Choosing the next movie to watch, the next song to listen, the next article to read etc. are ubiquitous daily choices. The recommendation community has been instrumental in advancing research in representations, models and methods for extracting and applying knowledge of user preferences to help users find preferred content. Several early recommendation methods use similarity to past content of choice for finding other content relevant to the user. For example, a user who likes a particular movie A is recommended other movies from the same genre as A (Content-based). Alternatively, the user is recommended other movies liked by a user who also liked A (Collaborative). While methods have been perfected to exploit such similarity structures between users and their preferences for items for making recommendations, these models have accrued criticism for concentrating extensively or entirely on past behavior, resulting in recommendations which are 'too similar' and are often disliked by the users. Furthermore, researches have shown that this problem is further exacerbated when recommendations are made over time [9]. As a result, a major initiative in the recommendation community is to move beyond similarity to produce diverse and novel recommendations [10, 11]. Furthermore, temporal models have been proposed to accommodate changes in user preferences [12, 13]. These methods have however lacked a model of the psychology of preference dynamics of users for predicting changes in their interests ahead of time.

User choices for viewing a movie, for example, depends not only on the types of movies she likes generally, but also, the movie she saw recently allowing psychological factors such as boredom and the need for variety to emerge. The incorporation of such behavioral insights into recommendations allows designs of the next generation of assistive agents that understand user need state better. Models of dynamics user preferences further allows content providers to direct efforts for customer retention. There is tremendous competition among the rapidly increasing number of web services for survival making it vital for them to invest in growth and retention solutions. This directly results in a great deal of emphasis being placed by services on retaining and further engaging their current user base. The highly dynamic nature of user visitation behavior calls for a novel retention metric to track the dynamic user return rate and identify correlates associated with user return behavior.

The subsequent chapters of this thesis describe novel computational approaches and algorithms to model user dynamic preferences. The proposed techniques are applied for making better temporal recommendations of content to online users. The dynamic models developed in this work, are further used to provide improved solutions for retention for web services the findings are discussed. However, before getting into the technical details, in this chapter we provide a brief overview of psychological underpinnings of our work. We also discuss computational models from consumer research and recommendation and identify their shortcomings. Finally, we provide the layout of this thesis discussing the subsequent chapters and their key contributions.

1.1 Background

1.1.1 Dynamics in Preferences

Predicting changes in preferences, based on the past behavior and experiences of the users is a non-trivial problem with no proposed solutions. Although, little studied in the user modeling community, evolution in human preferences has been a subject of much psychological research. Some key insights and findings from the state-of-the-art in behavioral psychology are discussed in this section.

Studies in psychology of preferences have been devoted towards understanding the role of familiarity and novelty in driving an individual's future interests. A correlation between familiarity and preference has been identified as the mere-exposure effect, first formalized by

Zajonc [14], according to which repeated exposure to a stimuli is sufficient for an enhancement in liking. Further, Martindale has shown that prototypicality and mere-exposure are important factors for predicting aesthetic preferences [15]. Alternatively, human behavior has been described to be exploratory or sensation seeking. The inherent drive for exploration leads individuals towards desiring new and novel content. Berylne in numerous of his works has found collative aspects of the stimulus such as novelty, incongruity and complexity to contribute to its arousal potential. Such a preference is hypothesized to result from curiosity for new information [8]. He further proposed that preferences show an inverted-U relationship expressed by the wundt curve with the arousal potential of the stimulus 1.2. Exploratory behavior is also linked to stimulus satiation responses arising on repeated exposure [7].

Laboratory experiments have shown mice to show spontaneous alternation behavior, which describes their tendency for alternating among stimuli without any external incentives. This behavior, is explained to arise due to a decrease in preference for a stimulus from exposure. The preferences are suggested to reinstate on removal of the stimulus due to forgetting. Several modulating effects of memory on the spontaneous alternation phenomenon further support this hypothesis.

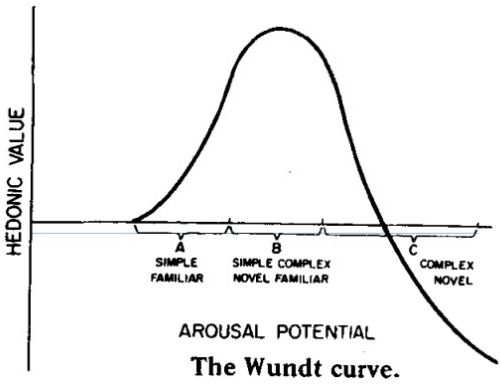


Figure 1.2: The Wundt curve proposed by Berlne for explaining the relationship between the arousal potential of a stimulus and its hedonic value [16].

A crucial question that appears is; what determines future behavior of the organism - the desire for familiarity or a desire for exploration? Several studies have found exploration and

exploitation tendencies to occur in moderation, with an excess in one leading to the other. For example, detailed experiments on the mere-exposure effect have shown that the favorable effects of exposure on preference are limited by boredom [17]. Alternatively, high uncertainty in the environment is found to result in anxiety and displeasure. Hebb [18] and Leuba [19] independently postulated this idea by suggesting that organisms are driven towards maintaining an optimal level of stimulation in their environment. However, most of the presented work deals with preferences of individuals in forced-exposure situations. Experimental research on individuals who are allowed to freely choose stimuli they wish to expose themselves to, have been limited. For example, a study of free choices for music listening have shown that subjects tend to alternate among their preferred alternatives once they have exploited their choice set [20]. However, subject choices in potentially infinite choice spaces have never been explored. Finally, the theory of information foraging provides some information theoretic answers for balancing exploration and exploitation policies in an uncertain environment for maximum reward [21].

1.1.2 Computational Models

The psychological research in preferences discussed so far has primarily been theoretical in nature. There have been efforts in the areas of consumer research and recommendations towards developing computational models for dynamic preferences in real world scenarios. Some of these models are reviewed next.

Consumer Choice Models

Variety seeking behavior has received a lot of attention from the consumer research community. McAlister compiled a taxonomy of factors responsible for varied behavior in consumers [22]. In his work, he segregated true variety seeking behavior resulting from internal motivations from that produced by external factors such as unavailability of a product, launch of new products etc. True variety seeing behavior was further proposed to manifest in two forms; a desire for unfamiliar alternatives and a desire to alternate among familiar alternatives. The former exemplified the desire for novelty while, the latter was seen as a weak form of exploratory behavior where the desire for familiarity coupled with devaluation effects due to satiation produced an alternating behavior. Lancaster [23] proposed that the preference for items can be composed from the preference for its attributes. McAlister further modeled variety seeking behavior in

soft drink preferences using a dynamic attribute satiation model [24]. The model assumed an ideal level of inventory at the attribute level and penalized departures from the ideal level. The inventory was designed to dwindle over time to incorporate the effects of forgetting.

Subsequent efforts in consumer research have advanced towards developing a general consumer choice model which allows consumers to either exhibit a short term loyalty for their last purchased brand (inertia) or devaluation for the last purchased brand (variety seeking) [6, 25]. Bawa et al [26] used a single peaked function, to model the conditional probability of repeat purchase given the number of times the brand was re-purchased since user's last switch (run length). Recent efforts have expanded these models to incorporate heterogeneities between consumers and external environment variables affecting user choices [27].

However, most of the existing efforts in consumer research have not accounted for long term changes in consumer interests. A recent work by Garcia-Torres [28] uses a utility based model of consumer choice making which incorporates the process of preference formation by allowing for integration between old habits and the acquisition of new preferences.

Models for Recommendation

Recommendations is a relatively new field with only a few decades of history. The rise in popularity of the web has significantly pushed this research area forward by producing an unprecedented need for assistive agents. The online environment also allowing the collection of large scale and detailed user data which fuel many of the sophisticated machine learning and data mining models at the heart of some of the most popular recommendation methods. Most of the early models for recommendation assumed a static view of preferences. Such models include the popular nearest neighbors [29] and matrix factorization methods [30] for recommendation and their subsequent probabilistic renditions. However, the lack of temporal awareness in these models was obvious. Ding et al. [12] showed that it was important to include temporal changes in preferences while making recommendations. He proposed using a decay function to emphasize recency of past behavior while predicting future recommendation needs. Koren offered a better solution by modifying the matrix factorization model to incorporate changes in preferences over time [13]. However, Koren's and other similar approaches can be described as a corrective scheme for preferences changes rather than as dynamic model for preferences.

A dynamic model for preferences was proposed recently by Sahoo et.al [31] for predicting blog reading behavior in employees. They used a hidden markov model for modeling dynamic

participation of users in different latent classes based on their changing preferences over time. Sahoo et. al's approach however, assumes that global preferences of a user to remain stable over time.

1.2 Contributions and Organization

The subsequent chapters of this thesis are organized as follows:

1. In Chapter 2, we develop the mathematical tools needed for measuring changes in preferences for items using user activity data. The hazard function, commonly used in statistics for survival analysis, is adapted for this purpose. We further use our proposed hazard functions for item consumption to provide the first evidence of spontaneous devaluation in preferences of online users.

2. Chapter 3 presents a Hidden Semi-Markov Model (HSMM) for a user consumption of familiar items. The HSMM model incorporates two latent psychological states of preference for an item - sensitization and boredom. Each preference state is associated with a state specific hazard function for item consumption estimated from user past activity streams. We show that our model performs much better than the state-of-the-art temporal recommendation models at making temporal recommendation of familiar items.

3. Chapter 4 focuses on user novelty seeking behavior i.e. their consumption of novel, unknwon or new items. A predictive model for user novelty seeking is developed using user history of recent item consumptions including the diversity of their familiar set and their boredom with the familiar set. User novelty seeking predictions are further used to modulate the introduction of novel items to the user. The proposed recommender, called adaNov-R, is shown to be robust in term of the accuracy of its recommendation and its ability to adapt to user's specific desire for novelty compared to existing non-adaptive approaches.

4. Chapter 5 defines the problem of return time prediction for free web services. Our solution is based on the Cox's proportional hazard model from survival analysis. The hazard based approach offers several benefits including the ability to work with censored data, to model the dynamics in user return rates, and to easily incorporate different types of

covariates in the model. We compare the performance of our hazard based model in predicting the user return time and in categorizing users into buckets based on their predicted return time, against several baseline regression and classification methods and find the hazard based approach to be superior.

5. Finally, Chapter 6 summarizes the conclusions of the research presented in this thesis and some future research directions.

Chapter 2

Measuring Spontaneous Devaluations in User Preferences

2.1 Introduction

Recommendation systems have become a popular means of suggesting relevant content to the user. Methods in recommendations have focused on constructing estimates of user preferences based on their history of choices. These preference estimates are then used to suggest new content to the user using content-based or collaborative methods. Content-based methods use a user's preference estimates to find similar content, while collaborative methods use a user's preference estimates to identify similar users (neighborhood) and recommend content popular in the identified neighborhood. But, it's not sufficient for a recommender agent to only estimate a user's past preferences; it's also important to predict their future preferences given past experiences. This makes the task of a recommender even more challenging by requiring it to predict when and how a user's preferences will change in the future. The recommendations community, however, lacks models which can predict changing preferences of users and doing so is generally accepted as a hard problem. On the other hand, user's recent choices have been found to be a good predictor of their future behavior. Efforts in modeling temporal recommendations have exploited this aspect of user choices by designing recommendation systems which systematically emphasize recency with good results. The critical shortcoming of this formulation is that such a system merely reacts to preference *changes* rather than trying to predict them.

While little work has been done on predicting changes in user preferences in the recommendation literature, psychologists and behaviorists have long studied the dynamics of individual preferences. Several theories have been proposed to explain why individuals seek out new content (novelty seeking, exploratory and information seeking behavior) [32]. Other studies talk about individuals making choices to actively seek an optimal level of stimulation in their environment [33]. The theory of flow [34] suggests that an environment which provides an optimal level of challenge for a given level of skill leads to a desirable state of flow. Despite such theoretical developments, it has been difficult to operationalize these aspects of individual choices to solve real world problems. However, modeling properties of individual behavior is critical for advancing designs of automated agents which interact with individuals on a daily basis.

In this work, we study one aspect of dynamic individual preferences. Individuals are often found to develop disinterest and even dislike for their dearly preferred content both temporarily and lastingly. It's common to find that one's clothes, food, entertainment, jobs etc. have grown boring despite being enjoyable in the past. We call this phenomenon a spontaneous devaluation of one's preferences or boredom for a stimulus. Spontaneous devaluation is seen to arise when repeated exposure to a stimulus creates a feeling of satiation towards it leading to a loss in interest [4]. Alternatively, spontaneous devaluation has been linked to lost opportunity for novel experiences when similar experiences are repeated too often [22]. Both theories concur in suggesting that, in contrast to recency-based expectations, *repeated exposure to familiar choices spontaneously devalues one's preference for them.*

Human behavior driven by these dynamics could be modeled as systematically alternating between one's set of choices, assuming that the time spent in experiencing other stimuli is sufficient to mitigate the effects of boredom for a particular stimulus. Several studies on user purchase behavior have found buyers to alternate among their preferred alternatives [35, 22, 6] etc. However, in practice users have a non-uniform liking for different alternatives in their choice space. Furthermore, users have a pronounced tendency to stick to their recent choices [35] which has been responsible for the success of the previously proposed recommender models. We call this behavior the *'sticky'* behavior in users. This phenomenon has also been called reinforcement or inertial behavior. Such behavior can be explained to arise due to an actual increase in liking on exposure [6] or a tendency to avoid switching costs.

The presence of both stickiness and devaluation effects in user preferences make predicting

the temporal choices of a user non-trivial. In this paper, we analyze user music listening behavior to extract signals of stickiness and boredom. Our analysis is limited to the music domain due to availability of public datasets, nevertheless, we expect our results to generalize to other items like movies, videos, books, vacation packages, shopping etc. which are fairly susceptible to boredom effects. We demonstrate the use of hazard functions for measuring these phenomena. Our work provides the first proof of spontaneous devaluation in music listening preferences of users and its impact on user choices. This work can inform design of future methods that incorporate these dynamics, producing agents that can cater to new needs of users suffering from boredom.

The rest of the paper is organized as follows: Section 2 provides a summary of the related work. Section 3 gives an overview of the dataset and pre-processing details. Section 4 lays out terminology relevant to our analysis. Section 5 provides details of our methodology. Our results are summarized in Section 6. We end with a discussion of the contributions of this work and possible future extensions in Section 7.

2.2 Related Work

2.2.1 Dynamic Preferences

Stimulus satiation was initially used by researchers to explain spontaneous alternation in rats [4]. Rats were placed in a T-shaped maze and provided an unlimited supply of food at the left and the right corners of the maze at equal distances. The experiment was set up such that that the rat had to return to the starting point before each trial. It was seen that rats chose to alternate between the left and the right ends on repeated trials. Glanzer [36] suggested that such a behavior arose due to stimulus satiation such that each time the organism was exposed to the stimulus, satiation for the stimulus increased causing the rat to switch directions. Further, satiation for the stimulus diminished when the organism could no longer perceive the stimulus and the rat returned back to the same direction.

Researchers have found individuals to engage in more complex forms of variety seeking behavior while making choices. McAlister proposed a taxonomy of factors responsible for varied behavior in individuals [22]. These were classified into two categories based on whether they arose due to external factors (such as unavailability of a product, launch of new products etc.) or due to internal motivations. When arising out of internal motivations, variety seeing behavior

was suggested to manifest in two forms; a desire for unfamiliar alternatives or a desire to alternate among familiar alternatives. The former was linked to individuals seeking an optimal level of stimulation [32, 33], while, the latter was seen as a weak form of exploratory behavior. It was also linked to devaluation in preferences due to satiation. A single peaked preference function was proposed to characterize the attractiveness of a stimulus on repeated exposure [37]. McAlister also proposed a dynamic attribute satiation model [24] which assumed an ideal level of inventory for different attributes of the items. The inventory was designed to dwindle over time to incorporate the effects of forgetting.

Researchers have subsequently focused on modeling the choice probabilities of consumers directly given their past choices. Consumers were found to exhibit either a short term loyalty for their last purchased brand (inertia) or devaluation for the last purchased brand (variety seeking) [35, 6]. Kahn [25] compared seven models for user choice behavior with similar results. Bawa et al [26] used a single peaked function, to model the conditional probability of repeat purchase given the number of times the brand was re-purchased since user's last switch (run length). Chintagunta [38] used hazard rates to model the level of inertia and variety seeking as a function of time between purchases. Recent efforts have expanded these models to incorporate heterogeneities between consumers and external environment variables affecting user choices [27].

Most of the research in this area, however, has been limited to panel datasets and analysis of user surveys and questionnaires. In this work we have adopted a data driven approach to elicit changes in user preferences towards a stimulus as a function of their past exposure to it. Our efforts do not look at variety seeking or inertial behavior in users in general, but at changes in choice probabilities with respect to particular stimulus, grounding ourselves in psychological theories of boredom and novelty seeking, which provides a causal explanation for the existence of these patterns.

2.2.2 Recommender Systems

State-of-the-art methods in recommender systems have assumed a static view of human preferences. Ding et al. [12] showed that the static view of user preferences used while generating recommendations was flawed as it did not take changing user interests into account. They used a decay function to gradually devalue the impact of a user's past history while making prediction of his future likings. Recently, a temporal model of recommendation was developed [13, 39]

which was an important part of the solution to the KDD Cup on Yahoo Music dataset and the Netflix challenge. The model incorporated several time-sensitive user and item biases in the standard factor model. Gradual changes in user preferences over time were captured using a linear function. Their model showed that modeling temporal dynamics in user choices was essential for improving the performance of the recommender. Sahoo [31] has proposed a dynamic model of blog reading behavior in employees. He used a Hidden Markov Model to predict future interests of employees based on their previous choices. However, user transitions are assumed to be driven by a static transition matrix. At present, the recommendation community lacks models that predict *changes* in user preferences.

Also related to our work are methods to introduce diversity and novelty in recommendations. Lathia et al. [9] showed that popular recommendations methods such as kNN and SVD produced recommendations which were very similar (low in temporal diversity) on iterated train-test experiments on temporally ordered data. Many methods that systematically introduce diversity in the recommendations have been proposed [40, 10, 41, 42]. However, these methods focus on jointly optimizing both similarity and diversity indices described on the space of items being recommended rather than predicting changes in user preferences.

2.3 Data

Our analysis is based on complete temporal music listening histories of users provided by Last.fm. Last.fm is a popular music website with millions of active users. It allows users to purchase tracks, listen to online radios and playlists etc. and has additional social networking features as well. Recently, Last.fm made available a dataset of complete music listening histories of around 1000 users as recorded till May 2009 [43]. This is the only publicly available dataset, to our knowledge, to provide complete temporal records of user choices. Because Last.fm hosts several online radios, it is quite probable that parts of the user histories capture radios, and playlists rather than active user choices. We filtered these effects by using the time gap between two consecutive tracks played by the user. Last.fm has a generous list of API's available to developers. The API, track.getInfo, was used to retrieve the duration of most of the songs in our dataset. We compared the time gap between song 1 and song 2 in that temporal order in the user history with the length of song 1. If the time gap was found to be more than the length of song 1 by less than 5 seconds, song 2 was identified to belong to an automated

play list. All tracks 'not on *auto-play*' were assumed to be *active* user choices. We could not remove auto-play effects for the songs whose lengths were unavailable through the API. This corresponded to 0.05% of the songs. We only considered the first 1 year of each user history in our analysis. All the users which had less than 30 records of activity were eliminated from the dataset. Also, we only kept those artists in the user history which the user had listened to 15 or more times in that period of 1 year. We summarize some important statistics about the dataset in Table 1.

Property	**Value**
# unique tracks	1,084,872
# unique artists	174,091
# Users	957
Mean history length - # songs heard	6716
Mean history length - # active days	177
Mean # unique artists heard	37

Table 2.1: Statistics from the Last.fm dataset

2.4 Terminology

Based on both the novelty-seeking and stimulus satiation theories of devaluation of preferences, repeated exposure to a stimulus causes devaluation in one's preferences towards it. Additionally, devalued preferences can get reinstated after a period of reduced or no exposure. A music piece can stimulate the listeners because of the combined effect of its multiple features (artist, genre, tempo, strong female vocals, etc.). For simplicity and ease of access, we use the artist of the songs as our basic stimulus. More sophisticated stimulus definitions that model the interaction between multiple features of a song can enhance our method.

Preferences have been linked to choice probabilities in the past. It is only a logical extension to relate changes in preferences to changes in choice probabilities, and in our case conditional choice probabilities. We suspect that the phenomenon of devaluation produces two different patterns in the choice probabilities of users for an artist.

Hypothesis 1: The probability that a user will listen to an artist again will decrease after he has listened to the artist some number of times. When this happens, we say that the user's preferences for the artist have devalued.

Hypothesis 2: Devalued preferences can get reinstated after a sufficient period of non/reduced exposure to the artist.

Through our experiments, we look for signals suggestive of spontaneous devaluation in choices probabilities of Last.fm users. By doing so, we establish a methodology for detecting this phenomenon and analyzing its properties.

We consider the state of the user at some time t to be defined by the artist of the song the user was listening to at that time. The temporal history of the user comprises the sequence of states visited by him as a function of time; i.e. $H^u(t) = s_a$ if user u was listening to artist a at time t. User u is said to *enter* a state a at time t if $H^u(t) = s_a$ and $H^u(t-1) \neq s_a$. A user u is said to *exit* a state a at time t if $H^u(t) \neq s_a$ and $H^u(t-1) = s_a$. We can now define the following conditional choice probabilities:

1. **Conditional probability of exit**: This is the conditional probability of a user u exiting state a at time t given that he last entered state a at time $t-r$ and has not exited state a yet. Formally, the probability is equal to $P(H^u(t) = s_a | H^u(t-1) = s_a, \ldots, H^u(t-r) = s_a, H^u(t-r-1) \neq s_a)$. Here, r is the time spent listening to the artist and corresponds to the idea of a run length in Bawa's model [26]. We make the simplifying assumption that this probability depends only on r. Hence, we can also represent the conditional probability of exiting state a by user u when time spent in state is r as $P^{ua}(\text{exit}|\text{time spent in state } a = r)$.

2. **Conditional probability of entry**: This is the conditional probability of user u entering a state a at time t given that the user last exited state a at time $t-(o+1)$. Formally, this corresponds to $P(H^u(t) = s_a | H^u(t-1) \neq s_a, \ldots, H^u(t-o) \neq s_a, H^u(t-o-1) = s_a)$. Here, o is the time spent not listening to the artist a. Again, for simplicity, we assume that this probability depends only on o. We later relax this assumption with interesting effects, described in Section 6.3. Thus, this probability can also be represented as the conditional probability of entering state a after having exited it o units of time ago or $P^{ua}(\text{entry}|\text{time spent out of state } a = o)$.

The definition of time has been kept ambiguous in the definitions above. We now define it more formally. Time can be defined in terms of the order in which songs are heard by the user such

that $H^u(t)$ refers to the t-th song heard by user u. Such a definition, however, does not take the actual time gap between consecutive listenings into account. It is important to consider the actual time gap between user choices. This is because a user satiated with an artist can get unsatiated both by listening to other artists or due to forgetting if he returns to the system after a long time. To analyze the impact of actual clock time on the satiation level, we define time in terms of days since the first historical record of the user. Accordingly, $H^u(t)$ refers to the state of the user on t-th day since day 1. For simplicity, the state of the user on a day is defined by the artist listened to most frequently by him on that day.

2.5 Methodology

Survival Analysis is a statistical method commonly used for modeling time-to-event data. The purpose of this kind of analysis is to model the probability of survival (where the occurrence of the event corresponds to death) beyond a certain point in time. For simplicity, we use a discrete measures of time $t \in \mathbb{N}$. The survivor function at time t is defined as:

$$S(t) = P(T > t) \tag{2.1}$$

Where, T is a random variable denoting the time of death. The instantaneous rate of occurrence of the event at time t, conditioned on having survived up to time t, is captured using the hazard function. The hazard function is also called the conditional failure rate and is defined as:

$$\lambda(t) = \lim_{\Delta t \to 0|} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t} = -S'(t)/S(t) \tag{2.2}$$

We use the hazard rate function to compute the exit and entry conditional probabilities defined in the previous section. We set $\Delta t = 1$. This allows us to use the terms hazard rate and conditional probability of death interchangeably. We can construct the two different hazard curves based on how we define our events.

1. **Exit Hazard Rate:** Here, we measure time from the point when a user u entered a state a. The event corresponds to his 'exit' from the state. The random variable T^{ua}_{exit} denotes the time of exit or death. This hazard rate captures the conditional probability of exiting the state at time $t+1$ having survived in the state for time t or greater; $\lambda^{ua}_{\text{exit}}(t) = P^{ua}(T^{ua}_{\text{exit}} = t | T^{ua}_{\text{exit}} \geq t)$.

2. **Entry Hazard Rate:** Here, we measure time from the point when a user u exited a state a. The event corresponds to his 'entry' back into the state. The random variable T^{ua}_{entry} denotes the time of entry or death. This hazard rate captures the conditional probability of entering a state at time t having survived outside the state for time t or greater; $\lambda^{ua}_{\text{entry}}(t) \ = \ P^{ua}(T^{ua}_{\text{entry}} = t | T^{ua}_{\text{entry}} \geq t)$.

An exit and entry hazard rate can be defined for each artist a user listens to. For our analysis, we pool across the different users and the artist choices to compute an average exit and entry hazard rate for the entire dataset. We normalize the time of entry and exit variables to mitigate the effects of differences in a user's preferences for different artists and differences across users. The time of event variable is log transformed as well as it becomes harder to exactly predict the time of an event as time for which the event has not happened increases. In other words, this means that if a user has not returned to an artist in a month, its more difficult to predict the exact day of his return, than, when he has has not returned to the artist for a day. The log transform accommodates this non-linearity in the predictability of return time.

$$T^{\text{N}}_i = \frac{\log_2(T^{ua}_i)}{\log_2(\frac{1}{P^u(a))}} \tag{2.3}$$

for a user u and artist a and $i \in \{`entry', `exit'\}$. $P^u(a)$ is the prior probability of user u being in state a.

$$P^u(a) = \frac{N^u(a)}{L^u} \tag{2.4}$$

where, $N^u(a)$ is the number of times user u was in state a and L^u is the length of user u's history. The average hazard rates for the normalized time of event variable can then be computed across users and artists:

$$\lambda_i(t) \ = \ P(T^{\text{N}}_i = t / T^{\text{N}}_i \geq t) \tag{2.5}$$

We discretize t into intervals $(0, 0.1], (0.1, 0.2]$ and so on. The hypothesis presented by us in section 2.4 can now be represented using the hazard rates.

1. **Hypothesis 1** The exit hazard rate for an artist should be an increasing function of time. This indicates that a user's preferences for an artist decrease with increased exposure to the artist.

2. **Hypothesis 2** The entry hazard rate for an artist should be an increasing function of time. This indicates that user preferences for the artist are reinstated after sufficient time gap.

The *sticky* or inertial view of user choices, on the other hand, suggest that a user's probability of visiting a state would increase on having visited it. Contrary to the devaluation hypothesis, the conditional probability of visiting a state again would increase as time spent in the state increases. This implies that the exit hazard rate for an artist is a decreasing function of time for sticky users. The entry hazard rate, would also be a decreasing function of time as a user would be less likely to visit a state which they has not visited for long periods of time.

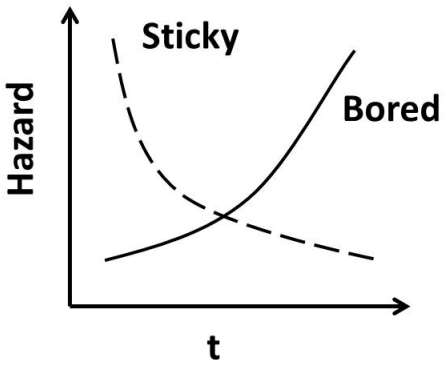
A common analysis methodology is to compare the hazard rate of interest in an analysis with that generated from a control experiment. This is done to remove the effects of covariates not being considered in the analysis. We define four baseline models to serve as controls. We constructed listening sequences by simulating user histories using each of the baseline models for every user. The user histories were simulated by sampling randomly from the temporal preference vector (Pref) generated by each of the model. In order to make the baseline models as close to the real data as possible, the parameters of the models were fitted to the actual user histories.

1. **Random (R)** The user is assumed to sample states randomly from his average preference vector (P^u).
   $Pref^u(t) = P^u$

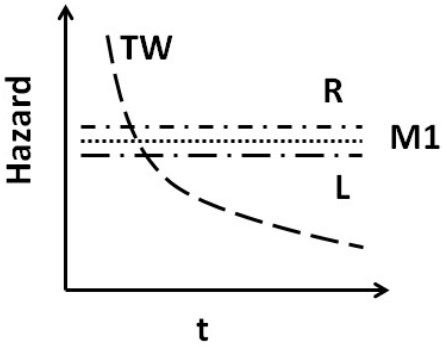
2. **1st order Markov (M1)** A user's switching probability from one state to the other is assumed to be controlled by a 1st order Markov model. The dynamics of the Markov model are controlled by a static transition matrix (T^u) which is learnt for each user u's history using maximum likelihood estimation. $Pref^u(t) = Pref^u(t-1) * T^u$

3. **Time weighted (TW)** We use a recency based model for generating user histories. $Pref^u(t) = \alpha^u * Pref^u(t-1) + c^u(t-1)$, where, $c^u(t-1)$ is $1 * |A|$ choice vector, which is set to 1 at index i if $H^u(t-1) = s_i$, and is 0 otherwise. The parameter α^u is a —A—*1 vector which was fit to the user u's history using stochastic gradient descent. We introduced a small exploratory component to this model to prevent extremely long lengths of continuous listening of the same artist. Therefore, our modified preference vector is computed as $Pref'^u(t) = 0.95 * Pref^u(t) + 0.05 * P^u$

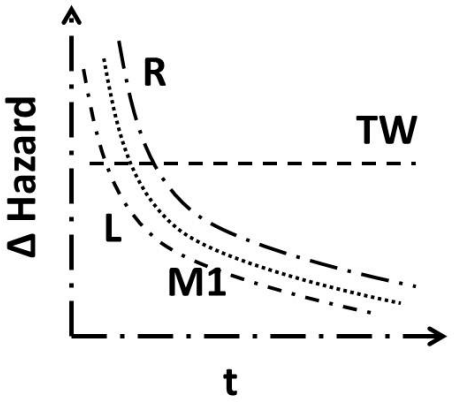
4. **Linearly increasing or decreasing (L)** We used the temporal model of user preference used by Koren [39]. $Pref^u(t) = P^u + \text{sign}(t - L^u/2) * (t - L/2)^{\beta^u}$. The parameter



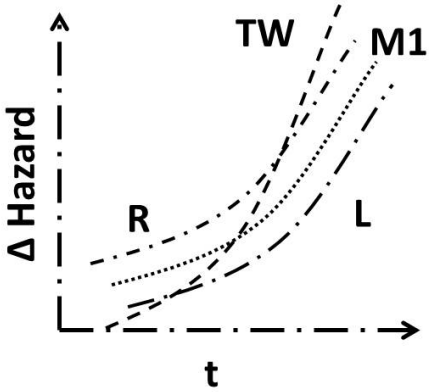
(a) Expected hazard rate for a sticky and boredom-prone user



(b) Expected hazard rates for the baseline models



(c) Expected Δ Hazard Rates for sticky users



(d) Expected Δ Hazard Rates for boredom-prone users

Figure 2.1: Figure (a) and (b) depicts the expected hazard rates for sticky and boredom-prone users and the baseline models. Both the entry and exit hazard rates should decrease with time for sticky users and increase with time for uses susceptible to boredom. Figure (c) and (d) shows the expected Δ hazard rates computed against each baseline model for sticky and boredom-prone users.

β^u is a —A—*1 vector and was fitted to the user u's history using stochastic gradient descent.

The Log-Rank test can be used to test whether the survival distributions generated by the simulated models are sufficiently different from that of the real data. The hypothesis test is defined as:

H_o**:** The real data and the simulated data have different survivor function

H_a**:** The real data and the simulated data have the same survivor function

The Log-Rank test on the real and the simulated survival functions rejects the null hypothesis with a *p*-value $< 10^-6$. The discrepancy between the real data and the baseline model predictions can be quantified using a Δ hazard rate obtained by subtracting the simulated hazard rates from the hazard rates computed on real data.

$$\lambda^{\Delta}(t) \;=\; -\frac{S'^{\text{real}}(t)}{S^{\text{real}}(t)} \;-\; -\frac{S'^{(\text{simulated})}(t)}{S^{(\text{simulated})}(t)} \tag{2.6}$$

We generate four Δ hazard rates for both the entry and exit time events for our analysis, namely real vs. random (λ_i^{A-R}) , real vs. Markov (λ_i^{A-M1}), real vs. time weighted (λ_i^{A-TW}) and real vs. linear (λ_i^{A-L}), where $i \in \{`entry', `exit'\}$.

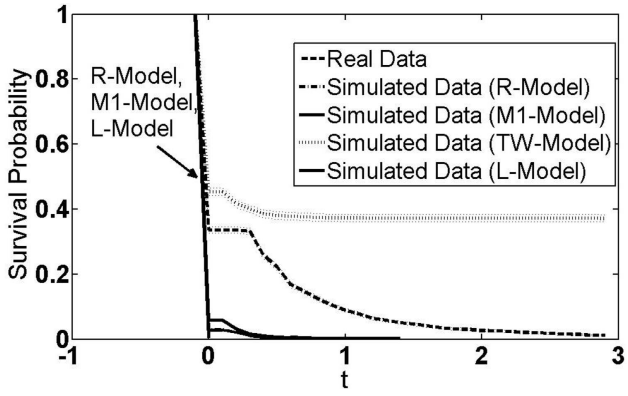
We display the entry and exit hazard rates expected for the event times obtained from the 'sticky' and 'boredom-prone' models and those expected from the baseline models in Figure 1. The entry and the exit hazard rates for a random, markovian and linear model should be independent of time spent in the state. A TW model on the other hand, is essentially a sticky model. Hence, the exit and entry hazard rates for TW model would decrease with time. The objective of this study is to understand the form of the exit and entry hazard rates for the real data. Figure 1 displays the expected Δ hazard rates if the real data follows the sticky and the boredom-prone model, respectively.

2.6 Results

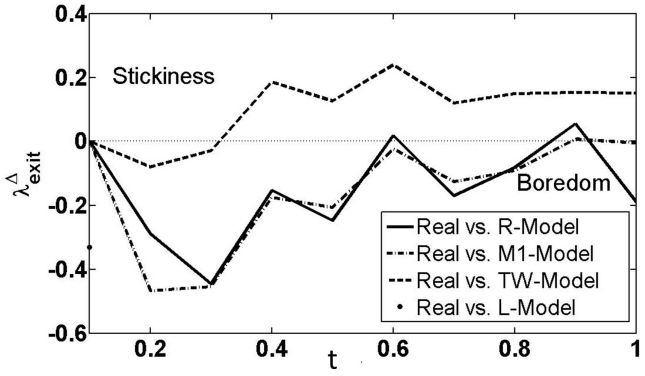
In this section we examine the obtained Δ exit and Δ entry hazard rates in close detail.

2.6.1 Δ Exit Hazard Rates

Figure 2 displays the survivor functions for the exit time for the real data and data generated by each simulated model. It also depicts the obtained Δ exit hazard rates. The changes in $\lambda_{\text{exit}}^{A-R}$, $\lambda_{\text{exit}}^{A-M1}$ and $\lambda_{\text{exit}}^{A-L}$, directly represent changes in the λ_{exit} for the real data. Changes in $\lambda_{\text{exit}}^{A-TW}$ would depict changes in the exit hazard rate for real data against a decreasing baseline.



(a) Kaplan-Meier survival functions and 95% confidence interval



(b) Nelson-Aalen Δ exit hazard functions

Figure 2.2: The figure illustrates the survival and the hazard functions computed for the exit time variable. The negative Δ exit rates for low values of t are indicative of sticky behavior, while the increase in Δ exit hazard rate indicate a devaluation in preference.

1. Real Vs.Random, Markov and Linear models: The $\lambda_{\text{exit}}^{A-R}$ and $\lambda_{\text{exit}}^{A-M1}$ are negative throughout suggesting that the exit rate for the real data is lower than that expected for the baseline models. This supports the sticky view of user preferences suggesting that a user has a lower rate of exiting a state after having visited it. However, contrary to what is expected for the sticky model, the Δ exit hazard rate increases with time after a point. We expect the Δ hazard rate to eventually flatten out, becoming uninformative. The survival function for R, M1 and L models drops sharply indicating a lower probability for large sequences than those observed in the real data. The L model has the sharpest drop in

survival probability, such that we did not enough samples of exit times greater than 0.1.

2. Real vs. Time-Weighted model: $\lambda_{\text{exit}}^{A-TW}$ is negative for low values of t, suggesting larger stickiness in users than generated by the TW model. However, the Δ exit rate increases thereafter, becoming positive after some time. Since, the exit hazard rate for the TW model is expected to decrease with time, this suggests that the exit hazard rate for real data increases more than the decrease observed in the TW model.

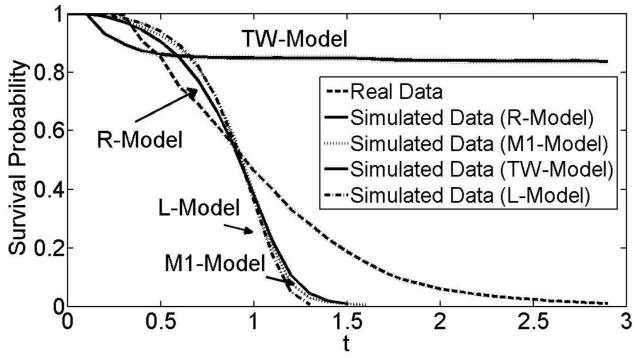
From these observations we can conclude that users have high stickiness towards the state on entering the state. However, the stickiness for a state reduces with time and the dynamics driven by boredom start dominating as time spent in the state increases. A user is thus likely to stick to his previous state at a higher rate initially and a decreased rate as time in the state increases.

2.6.2 Δ Entry Hazard Rates

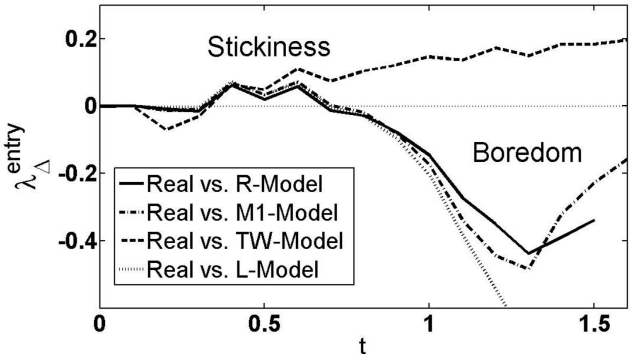
Figure 3 displays the survivor functions computed for the entry time variable for real and simulated data and the obtained Δ entry hazard rates. Similar to the Δ exit hazard rates, the changes in $\lambda_{\text{entry}}^{A-R}$, $\lambda_{\text{entry}}^{A-M1}$ and $\lambda_{\text{entry}}^{A-L}$ functions would depict changes in the entry hazard rate for the actual data. The TW model is expected to have a declining entry hazard rate, being a sticky model. The changes in $\lambda_{\text{entry}}^{A-TW}$ should reflect changes in the entry hazard rate for the real data against a decreasing baseline.

1. Real Vs.Random, Markov and Linear models: The $\lambda_{\text{entry}}^{A-R}$, $\lambda_{\text{entry}}^{A-M1}$ and $\lambda_{\text{entry}}^{A-L}$ functions are positive initially suggesting that the users have a higher rate of entry than that expected from the baseline models. This again can be attributed to the sticky nature of user choices, such that users have a high rate of returning to the artists they had listened to recently. The Δ hazard rates decrease for intermediate values of t suggesting a prominent devaluation in preferences. The Δ hazard rates eventually increase for larger values of t. However, they do not cross the 0-line again suggesting that a user always has a lower rate of return than that generated by the baseline models. This can be attributed to phasing out of an artist who is not being actively sampled.

2. Real vs. Time-Weighted model: The $\lambda_{\text{entry}}^{A-TW}$ function is slightly negative at the beginning suggesting that the actual entry hazard rate is lower than that of a TW model. Our



(a) Kaplan-Meier survival functions and 95% confidence interval



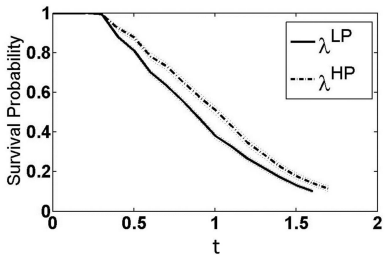
b) Nelson-Aalen Δ exit hazard functions

Figure 2.3: This figure illustrates the survival and the hazard functions computed for the entry time variable. The Δ hazard rates are positive for all the model for low values of t which is indicative of sticky behavior. A decline in the Δ entry hazard rates corresponding to the R, M1 and L models for intermediate values of t indicate that the preferences were temporally devalued. The increase in the Δ entry hazard rates corresponding to all the models for larger values of t suggest that preferences were reinstated

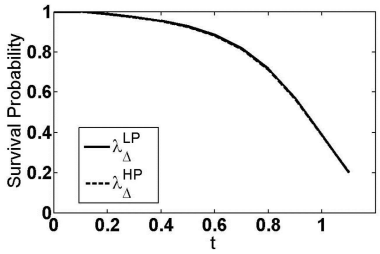
TW model is seen to pull back users which have just left an artist at a higher rate than observed in real data. The hazard rate increases thereafter indicating the actual data seems to have a larger rate of return than that of the TW model.

The analysis on the Δ entry hazard rates reveals aspects of sticky behavior in users which produces quick switches in and out of the artist. Also, we find indicators of devalued preference for intermediate values of time spent out of the state. Preferences are reinstated after longer periods of time spent away from the artist, however, the rate of return eventually flattens out

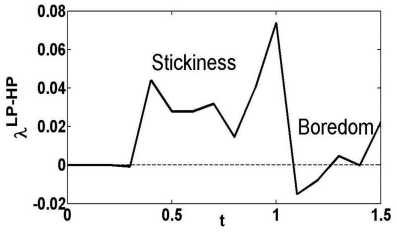
becoming uninformative.



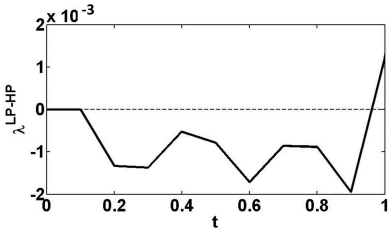
(a) Kaplan-Meier survival functions (real data)



(b) Kaplan-Meier survival functions (M1-Model)



(c) Nelson-Aalen Δ exit hazard functions (real data)



(d) Nelson-Aalen Δ exit hazard functions (M1-Model)

Figure 2.4: This figure illustrates the survival and the hazard functions computed for the entry time variable conditioned on the PRT. The conditioned survival function for the simulated data are coincident but vary significantly for the real data. The positive values of the Δ hazard function $\lambda_{\text{entry}}^{LP-HP}$ for low values of t indicate an increased stickiness when conditioned on lower values of PRT. The negative values of the Δ hazard function for larger values of t are indicative of increased boredom effects when conditioned on lower values of PRT.

2.6.3 Previous Return Time

In our previous analyses, we found evidence suggesting that users quickly switch in and out of an artist in a short span of time. Such a characteristic of user temporal choices suggest that a user's level of exposure to an artist is not completely defined by the 'in time'. A user who has just switched out of the artist and has switched back in almost immediately after, somewhat continues to be in state a. Therefore, we suspect that the previous return time (PRT) $T_{\text{entry}}^{N,P}$ also indicates how much a user has been exposed to the artist recently. A low PRT indicates higher exposure to the artist than a larger PRT. A corollary to hypothesis 1 in terms of the $T_{entry}^{N,P}$ for the artist follows:

Corollary 1' The probability that a user listens to an artist again will depend on his PRT to the artist. We suspect that of if the user has returned to the artist quite quickly previously, he will have a lower rate of returning quickly to the artist in the future.

In order to test this hypothesis we generate two conditional entry hazard rates.

1. $\lambda^{LP}_{\text{entry}}$ Entry Hazard Rate given a low PRT, $T^{N,P}_{\text{entry}} < 1$
2. $\lambda^{HP}_{\text{entry}}$ Entry Hazard Rate given a high PRT, $1 < T^{N,P}_{\text{entry}} < 1.5$

We compute the Δ hazard rate for the two conditional entry hazard rates.

$$\lambda^{\text{LP-HP}}_{\text{entry}} = \lambda^{LP}_{entry} - \lambda^{HP}_{entry} \tag{2.7}$$

$\lambda^{\text{LP-HP}}_{\text{entry}}$ function is computed for the real data and data simulated using a Markov model. The simulated data serves as a comparison. Figure 4 displays the obtained $\lambda^{\text{LP-HP}}_{\text{entry}}$ functions and the survival functions for $\lambda^{\text{LP}}_{\text{entry}}$ and $\lambda^{\text{LP}}_{\text{entry}}$ for the real data and simulated data. The log rank test is rejected with a *p*-value of less than $10^{-}4$ on the conditional survival functions of the simulated and the real data. However $\lambda^{\text{LP-HP}}_{\text{entry}}$ varies by very small amounts. On the contrary, $\lambda^{\text{LP-HP}}_{\text{entry}}$ on the real data varies in an interesting way. We see that $\lambda^{\text{LP-HP}}_{\text{entry}}$ is highly positive initially, which indicates increased stickiness when PRT is low. However, $\lambda^{\text{LP-HP}}_{\text{entry}}$ decreases and becomes negative eventually which indicates a lower rate of return for larger values of t when PRT is low than when PRT is high. Hence, once a user is out of the state he has a lower rate of returning back to the state when previous return time is low than rate of return for a user-artist pair for whom previous return time was high.

2.7 Discussion

In this work we have outlined a methodology for analyzing music listening histories of Last.fm users for studying the phenomenon of spontaneous devaluation in user preferences or boredom. We constructed hypothesis about boredom-prone behavior in Last.fm users and tested them through experiments on real and simulated data. Exploratory analysis of dynamic hazard rates computed on both the real and simulated data suggest that real data has strong evidence of spontaneous devaluation of preferences, as hypothesized. We also found strong evidence suggesting stickiness or reinforcement nature of past choices in users. Crucially, stickiness and

boredom effects on user choices were found to be spaced out in time suggesting that methods can be designed to systematically appease the two driving forces effecting user temporal needs. The results obtained from this analysis motivate the design of sophisticated dynamic models of user choices impacting recommendation methods, product design and advertising.

Our findings suggest that methods which only focus on maximizing similarity, or focus on maximizing both similarity and diversity at all times, accommodate only some aspects of user behavior, leaving useful *temporal* information on the table. Sophisticated temporal models of individual preferences, well grounded in cognitive and psychological analysis of the dynamics of their choices, are required for the design of automated methods that can predict user temporal needs well.

Being able to say *when* a user is likely to be bored should yield considerably more responsive and accurate product recommendations. However, the gap between this exploratory analysis and usable applications, while bridgeable, is non-trivial. We suspect heterogeneities to exist among users and their behavior towards different items, which this analysis has not considered. This is principally because extricating good estimates of dynamic hazard rates for different user-item pairs requires large amounts of historical data, while we were limited in our analysis to the Last.fm publicly release dataset. Unavailability of datasets providing complete temporal histories of users makes procurement of data a challenge. While gaining access to more data would be the best solution, clustering methods can reduce the data scarcity problem in the interim. Additionally, for simplicity, we have assumed the user behavior for an item is independent of the other items experienced by him. However, one can expect similar/dissimilar items to increase/decrease one's level of satiation with an item. Extending our approach into a full-fledged recommendation system would require us to address user and item level heterogeneities and similarities between items in a single framework. Potential solutions can benefit from hierarchical approaches to cluster items using multiple features allowing estimation of the impact of history on the hazard rates for similar items.

Our work constitutes the first study on dynamics of preferences of online music listeners, and demonstrates that there is significant value in trying to study the temporal browsing history of users along the lines we have suggested. We hope our work will motivate further studies on this topic in the future. Also, larger datasets would be made accessible for studying aspects of user choices, allowing advancement in the design of predictive agents of temporal user choices.

Chapter 3

Modeling the Dynamics of Boredom in User Activity Streams

3.1 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 [30, 44, 45, 46, 47, 48, 49, 50, 51] 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 [52, 53, 54, 55, 56, 39] 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 [55]. Others estimate the

[1] See experiments section for more details

temporal interest of a user for a particular item by combining the user, item and time (latent) factors [30, 45]. 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 the changing consumer preferences [24, 57]. We extend this idea further using behavioral psychology to represent these characteristics as two important states of user behavior [58]: 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.

3.1.1 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 [59, 60, 55, 53, 61].

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 3.2, 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 3.3. We also validate our model by comparing our approach to several variants in this section. Followed in section 3.4 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 3.5.

3.1.2 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 [46, 47, 48, 49]. While the other that make use of the user-profile attributes are called "Content-based filtering" techniques [62, 51]. There are several approaches that combine these techniques and are referred to as "Hybrid" [50, 51]. There are many survey articles [63, 64] that discuss a

variety of these approaches. The recommendation problem can be mapped to a standard classification setting, hence latent factor models [30, 45] and dimensionality reduction techniques are also applied. As these problems can be treated as matrix completion problems, matrix factorization [30] 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 [52, 53, 54, 55, 56, 39]. 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 [54, 53]. Other methods, have used seasonality and trends [65] as additional context for segmenting the user ratings. There are also tensor factorization [61] approaches, that extend the matrix factorization [66] techniques to include the temporal component.

There has also been some research on implicit feedback data sets [67, 65]. 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 [24]. 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 [68, 57, 69]. There are also some recent progress on dynamic content consumption analysis [70]. 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.

3.2 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 [25].

2. *Devaluing response*: Psychologists have associated repetitive exposures to stimuli with satiation and repulsion [7]. Stimulus satiation often produce shifts in interests and other variety and novelty seeking behavior [25]. Satiation is identified as a temporary phenomenon which diminishes with time due to forgetting [7].

The reinforcing and devaluing response is closely associated to a user's content consumption behavior in activity streams [71]. 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.

3.2.1 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 [72]. The $\{g, \delta\}_{1...n}^{ui}$ constitute the observable output from the model. This is shown in Figure 3.1. 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.

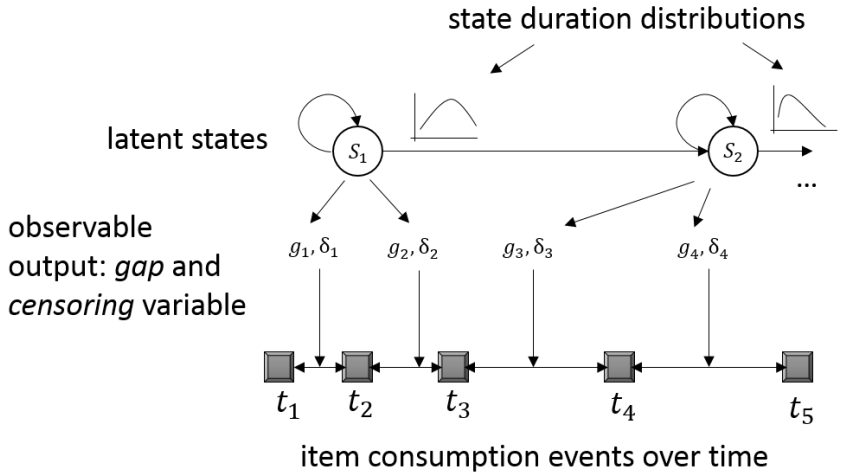


Figure 3.1: Using observed gap sequence and censoring variable for training a latent state model for item consumption.

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 3.2 displays the discussed parameters of our model. We model $\log(G)$ (rather than G)

to include non-linearity in the perception of time [73]. A parametric form is assumed for the emission and the state duration distributions: $b_m(\log(g)) =$ Log-logistic(μ_m, σ_m)) and $p_m(d) =$ Gamma(α_m, β_m). Our choice of parametric form allows us to capture time dependence characteristics of our data discussed further in section 3.3.3. 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.

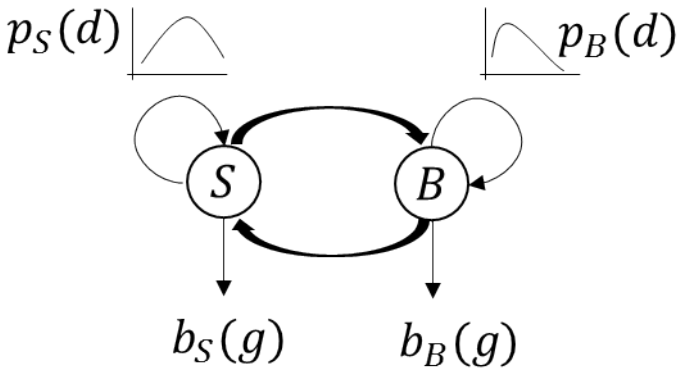


Figure 3.2: The hidden semi-markov model for gap length sequence.

3.2.2 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 [74, 75]. 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 [74]. 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;

$$\mathbb{E}(G_n|g_{1...(n-1)}, \lambda) = \mathfrak{s}_\mathfrak{n}(S) * \mathbb{E}(G|S, \lambda) + \mathfrak{s}_\mathfrak{n}(B) * \mathbb{E}(G|B, \lambda) \ . \tag{3.1}$$

We further obtain a dynamic measure of item consumption rate using techniques from survival analysis. Survival analysis [76, 77] 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:

$$h(t) = P(T = t|T >= t) = \frac{f(t)}{1 - F(t)}, \tag{3.2}$$

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,

$$DP(t|m, \lambda) \ = \ \frac{b_m(t - t_n)}{1 - B_m(t - t_n)}, \tag{3.3}$$

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:

$$DP(t|g_{1...(n-1)}, \lambda) = \frac{\mathfrak{s}_\mathfrak{n}(S) * b_S(t - t_n) + \mathfrak{s}_\mathfrak{n}(B) * b_B(t - t_n)}{\mathfrak{s}_\mathfrak{n}(S) * (1 - B_S(t - t_n)) + \mathfrak{s}_\mathfrak{n}(B) * (1 - B_B(t - t_n))} \ . \tag{3.4}$$

3.3 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.).

3.3.1 Data

We use a public dataset from the popular music service Last.fm [43] 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 3.1 summarizes the basic statistics for the final training and testing dataset used for our experiments.

3.3.2 Clustering

In Figure 3.3, 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

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 3.1: Last.fm dataset statistics (modeling boredom experiments).

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.

$$f = \frac{n^t}{t_{n^t} - t_1 + \epsilon}, \tag{3.5}$$

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.

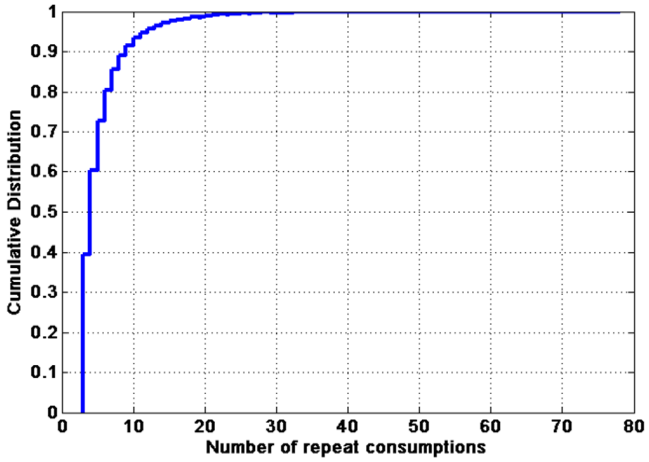


Figure 3.3: Cumulative distribution of the number of times users repeated their items.

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 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.

3.3.3 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 3.4 (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 3.4 (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.

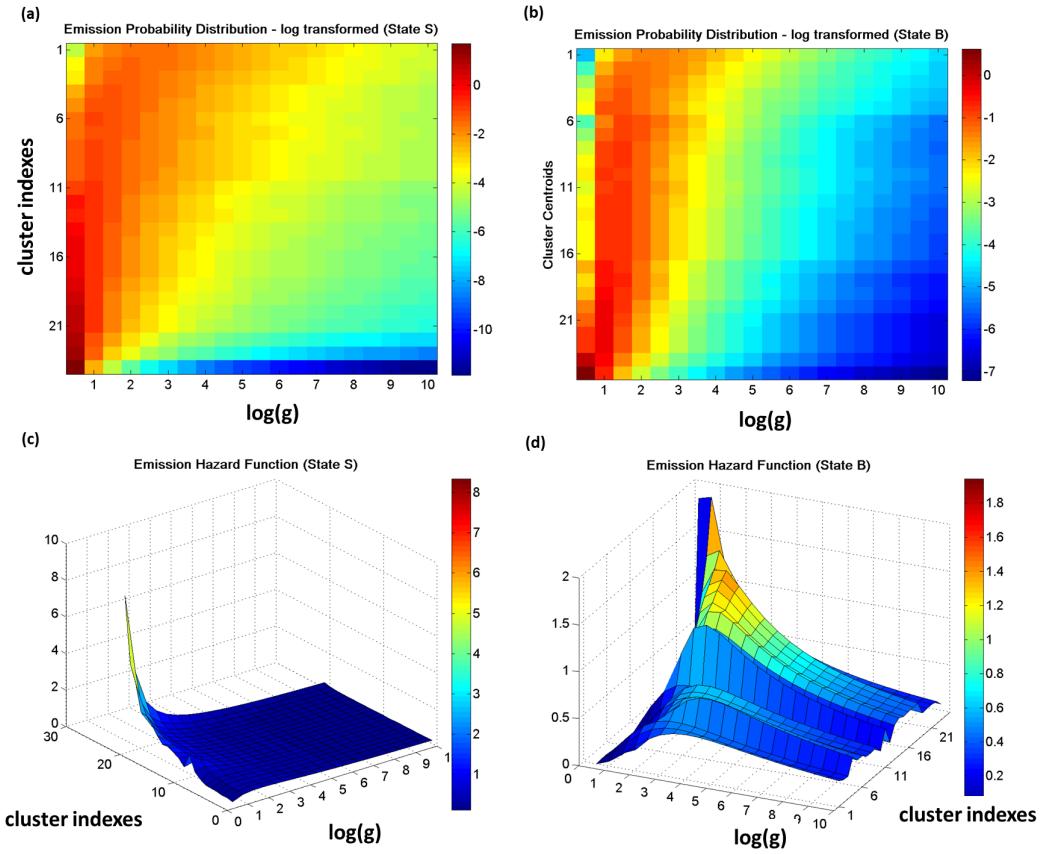


Figure 3.4: Emission probability and hazard function distributions. The cluster indxes are labeled in increasing order of likeness score. (To be viewed in color)

State duration distributions Figure 3.5 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.

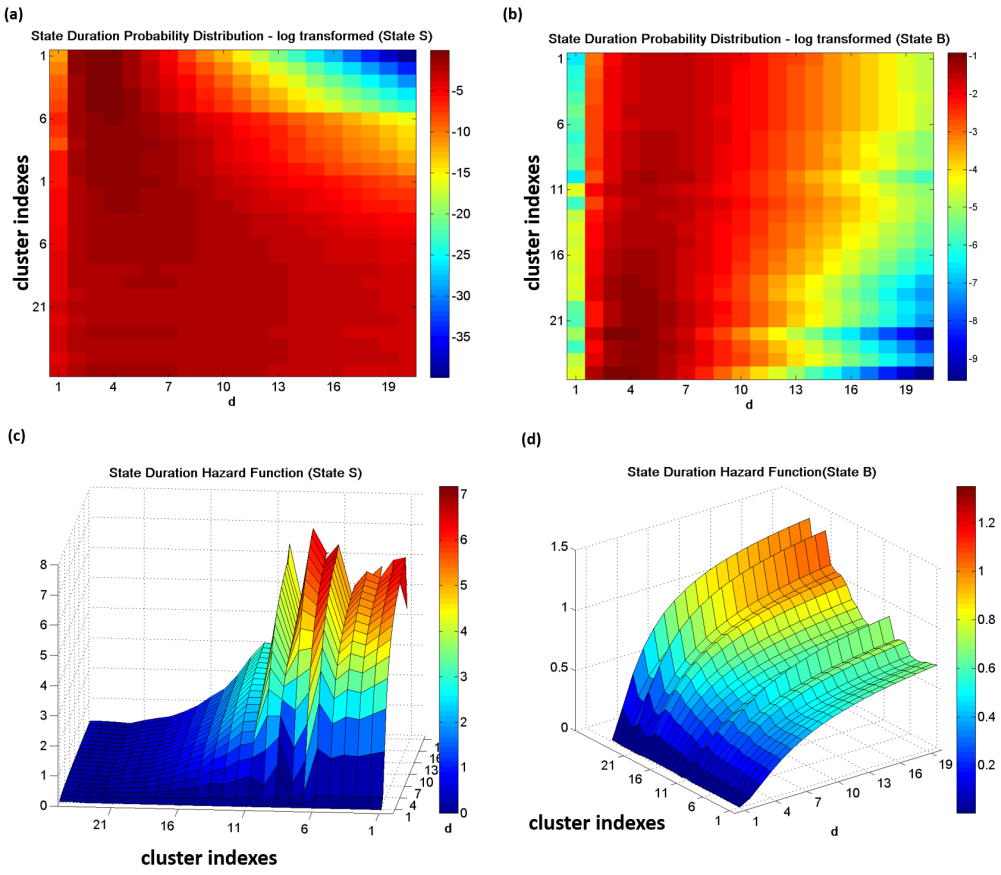


Figure 3.5: State duration probability and hazard function distributions. The cluster indxes are labeled in increasing order of likeness score. (To be viewed in color)

3.3.4 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 3.2. 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.

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

Table 3.2: RMSE scores on the log-transformed gap length sequence.

3.4 STiC Recommender

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

3.4.1 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 (3.4). This 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).

3.4.2 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.

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 [65, 67, 78], 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 [79]. 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 [67]. The metric computes the expected percentile rank of an item selected during the test period 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:
$$\text{T-Rank} = \frac{\sum_{u,i,t} C^u(i,t) * \text{rank}^{ui}(t)}{\sum_{u,i,t} C^u(i,t)}, \tag{3.6}$$
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.

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 [80]. 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 [65]. 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 [60]. 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 [55] 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 [53]. Changes in preference factors with time are captured using a linear function. The TSVD++ model is trained using the user choice matrices.

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 [61].

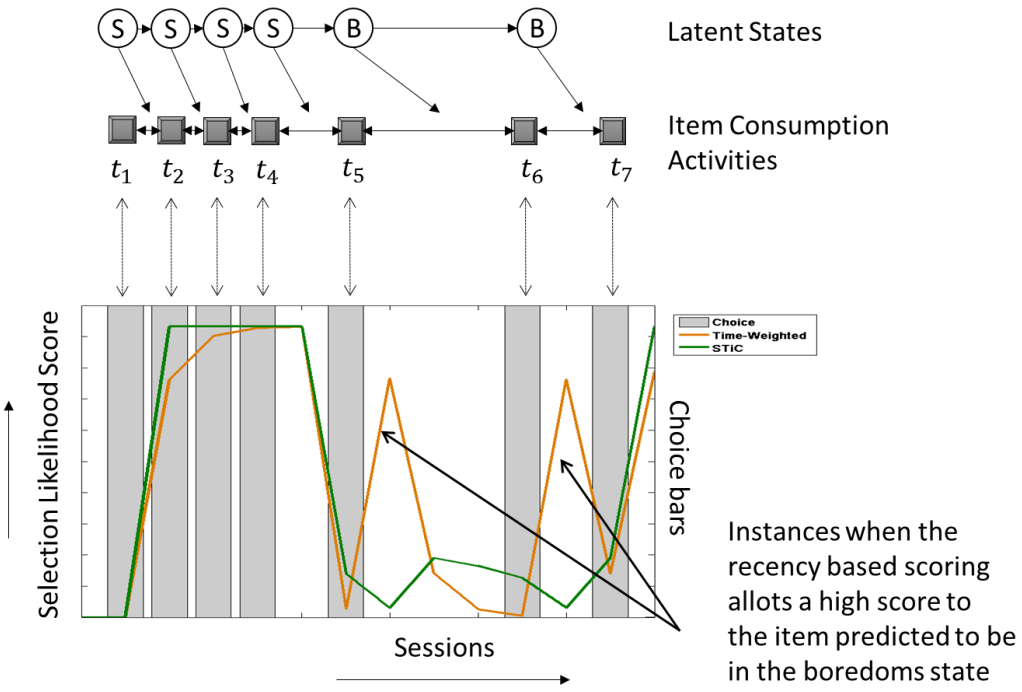


Figure 3.6: STiC model predictions compared against that of time-weighted model. The top part of the figure shows the STiC model's state predictions for the item. The bottom part displays how the same item is scored by the two models. The item is scored high by the time-weighted model even after the user state has changed (the item has become boring). Instead, the STiC model gives a low score to the item at those instances.

3.4.3 Results

The evaluation results are summarized in Table 3.3. Incorporating time is generally found to improve the performance over non-temporal counterparts [30, 53, 61], 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.

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 3.3: 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.

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 ($\mathfrak{s}_\mathfrak{n}$) for an item and corresponding observed gap lengths (Figure 3.6) (a)). For the same item Figure 3.6 (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;

$$\text{Boredom-Score}(t) = \text{Time till next consumption at time 't'}. \tag{3.7}$$

We borrow the concept of *future lifetime* [77] 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:

$$E(T|T > t_0) = \frac{1}{1 - F(t_0)} \sum_{t=t_0}^{t_s} 1 - F(t) \tag{3.8}$$

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:

$$\mathbb{E}(G_f|G > (t - t_n), m, \lambda) = \frac{1}{1 - B_m(t - t_n)} \sum_{s=(t-t_n)}^{t_s} 1 - B_m(s)\ . \tag{3.9}$$

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

score:

$$\begin{aligned}\text{Boredom-Score}(t) =& \mathbb{E}(G_{fn}|G_n > (t - t_n), g_{1\ldots(n-1)}, \lambda) \\ =& \mathfrak{s}_\mathfrak{n}(S) * \mathbb{E}(G_f|G > (t - t_n), S, \lambda) \\ &+ \mathfrak{s}_\mathfrak{n}(B) * \mathbb{E}(G_f|G > (t - t_n), B, \lambda) \ .\end{aligned} \tag{3.10}$$

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 3.7). 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.

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 3.4: 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.

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 3.4 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.

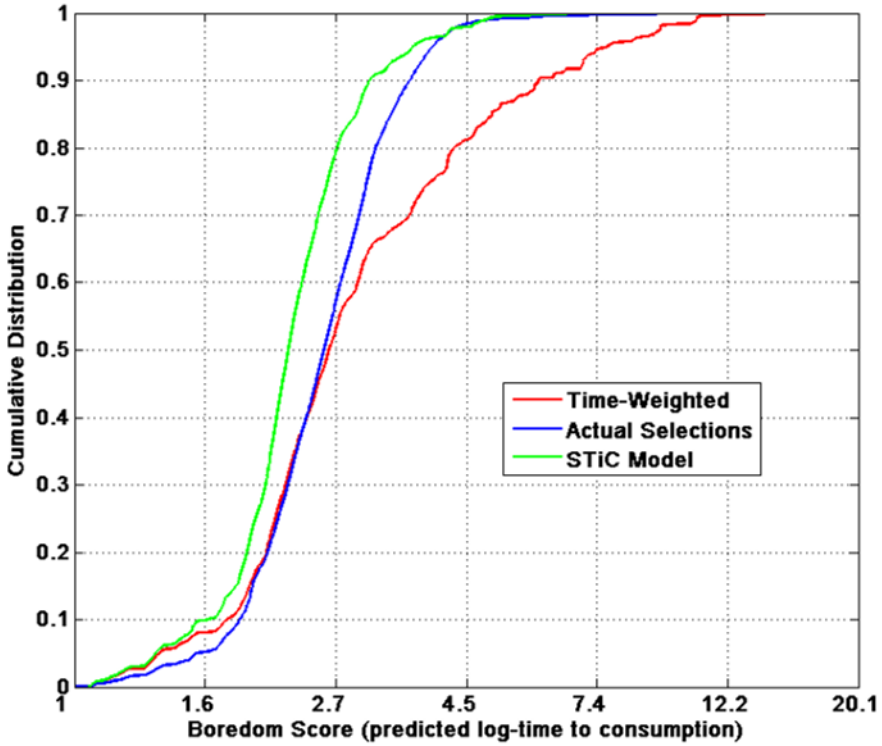


Figure 3.7: The cumulative distribution for the recommendation likelihood for the time-weighted and the STiC model given boredom scores. Time-weighted model recommends more item with higher boredom scores, while STiC is more conservative than the actual user.

3.5 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.

Chapter 4

Adapting Novelty Recommendation Using Predictions of User Novelty Seeking Behavior

4.1 Introduction

Recommender systems are indispensable in today's information age, solving the problem of information overload by providing users easy access to relevant content. Since their inception, the recommendation community has made major advances to the notion of user relevance and personalization. A wide variety of sophisticated algorithms have been developed that exploit similar items and similar users [30, 81, 82, 83], and several contextual cues such as social [84], temporal [85, 53] and mood [86, 87] information to identify interesting content for their users. Although standard recommendation designs focus primarily on providing accurate recommendations to the user based on item preferences, experts have highlighted the shortcoming of such formulations in meeting the diverse needs and goals of the user from the system [88]. Furthermore, familiar and accurate recommendation have been found to inhibit the range of the user experiences and are identified to be detrimental to user satisfaction in some cases [9, 89]. This is because such methods fail to understand the dynamics of user preferences that constantly evolve towards new, diverse or familiar items.

User's preferences in variety and novelty per se have been identified across several domains [90, 91]. Furthermore, studies in recommender systems have shown that users sometime prefer novel recommendation even if they are less accurate [88, 92]. As a result, a prominent direction for research in recommendation systems today is the recommendation of novel items to the user. The existing solutions [87, 93] have addressed this problem by introducing novelty at a constant rate to the user determined using a system level tunable parameter. However, such methods do not consider the specific novelty needs of the users which cannot be satisfied by static methods for novelty recommendation.

The individualized and variable preferences of uses for novelty and variety, as revealed in several studies [94, 95, 96], require the design of new recommendation methodologies. In this work, we therefore focus on the problem of adaptive novelty recommendation to modulate the introduction of novel items to the user based on a model for monitoring and predicting the novelty needs for the user.

To address this problem we first formalize the concept of item novelty to incorporate both new and forgotten items. Based on our formulation we then quantify user novelty needs using their novelty seeking behavior observed from their activities. We further propose a model to predict user future novelty seeking behavior, using various behavioral features derived from the past consumption of items. Our predictive model allows us to propose a novel Adaptive Novelty Recommender or *adaNov-R*, that considers both static item preferences as well as the dynamic novelty preferences for novel item recommendation. We show that our proposed recommender allows us to meet both the variable novelty requirements of the users and system level design need for making novel recommendation. Furthermore we achieve such improvements in performance using only off-the-shelf techniques and with minimal effort for implementation from existing systems.

The rest of the paper is organized as follows. We discuss the related psychology and recommendation literature in the following *Conceptual Background* section. We then define relevant terminologies used in our subsequent discussion in the *Terminologies* section. We discuss the characteristics of our datasets and analyze user novelty seeking behavior in our datasets in the *Dataset* and *User Novelty Seeking Behavior* sections, respectively. We then propose our predictive model for novelty seeking the in *Novelty Seeking Prediction* section. We propose our adaNov Recommender and evaluate it's performance in the *Adaptive Recommendation* and *Results* section. We finally conclude with the *Conclusion and Future Work* section.

4.2 Conceptual Background

In this section we provide a brief overview of related research from the field of psychology and recommender systems, relevant to our work.

4.2.1 Psychological Bases for Novelty Seeking

Although the importance of familiarity for preference formation has been documented in several psychological studies (the mere exposure effect [97, 14], preference-for-prototypes [15] etc.), Berlyne [16] identified the impact of new stimuli as some of the most obscure and complex motivations for human behavior. Several common day like experiences, reveal a prominent desire in individuals for novel, varied and complex stimuli even at physical, social and monetary costs to oneself [8]. For example, a user's curiosity for other songs from a preferred artist or genre makes him explore specific new songs. On the other hand, a user may explore varied types of new songs to identify his/her taste in music or to overcome boredom.

Studies have shown that users have individualized preferences for novelty and variety that themselves vary over time [94, 91, 95, 96]. For example, users variety seeking behaviors have been found to vary based on the product category level [95] and display format [96] and user current satiation level with the product attributes [98]. However, most of these studies have been conducted using controlled experiments and user surveys. User online multimedia consumptions provides a new lens for studying user novelty and variety seeking behaviors using unobtrusive empirical analysis and modeling methods. Furthermore, multimedia consumption such as music listening have been identified as a key domain where static and utilitarian models of user preferences fail and the individualized variety and novelty seeking preferences are critical to accommodate [99]. Such findings motivates the analysis of individual differences in novelty seeking using user activity logs from the music domain.

To the best of our knowledge, the only study of this kind has been the recent work by Zhang et al. [100] which measures novelty seeking in terms of self novelty (which is used to exploit an individual's desire for diversity) and crowd novelty (which is used asses a user's degree of anti-conformity) from user checkins and online shopping traces. Our definition of novelty seeking is closely related to the former, however we concentrate specifically on the new and forgotten items of the users identified using a time window. Furthermore, we specifically focus on how user novelty seeking impacts recommendation design further discussed in the

subsequent sections.

4.2.2 Novelty in Recommendation Systems

The idea of novelty has found emphasis in recommendation research for evaluating the performance of a recommender in terms of how different its recommendations are from other items previously seen by the user. An item can be novel to a users in three ways (a) *new to system*: item is new (in the system) and as a result novel (or unfamiliar) to user (b) *new to user*: item is known to the system but novel to the user (c) *oblivious/forgotten*: item is known to the system as well as familiar to the user but the user has become unaware of its existence due to the length of time elapsed since its last consumption [101]. The repetition of such items in a user's future consumptions has been shown to produce increased diversity [101, 102], emotional excitement due to nostalgia [103] etc. *This* differs from an alternative definition of novelty also commonly used in recommender research, which looks at a user's consumption of niche and long tail items [104, 105]. In this work, we define novelty in terms of items which are either new to user or have otherwise become oblivious. We do not specifically discuss items which are new to the system as a discussion of the such better belongs to the sub area of recommender research involving cold start problems [106].

Introduction of novelty and diversity in user recommendations has been shown to lead to a better overall user satisfaction with the system [107]. It is also found imperative for preventing user recommendations from becoming narrow and concentrated over a few set of items over a period of time, also known as the "filter bubble" effect [89]. Eli Pariser provides a detailed discussion of this effect emerging in systems which rely heavily on system generated recommended content. Subsequent research have found the effect to vary based on the consumption level of the users [108].

It has been shown that standard recommendation algorithms perform poorly in terms of recommending items which are new from the one's recommended before [9]. Instead approaches such as topic diversification [92], use of item taxonomy [109], bubble declustering [110] etc. have been used to improve the novelty and diversity of recommendation lists. Others have looked at both the accuracy and the diversity of recommendations and suggested approaches for dealing with the apparent trade-off between the two such as multi-criterion optimization [**?**], item re-ranking and re-weighting [93] and heat spreading algorithms [10]. However, none of these approaches measure their performance in terms of being able to provide the right amount

of novelty as actually desired by the user. Different novelty requirements of users make the existing *one-size-fits-all* approaches insufficient. Our *exploratory analysis* in a later section corroborates these findings. Instead, we require recommendation system to adapt the novelty provided by the system to the user's need for novelty. In order to realize such a system we also develop a model for predicting user novelty seeking based on behavioral features identified to be closely related to novelty seeking in prior studies including the diversity of user's past experiences and his/her boredom with the environment.

4.3 Terminologies

Despite much discussion in psychological and the recommendation literature, novelty seeking remains an abstract concept with few known quantitative measures applicable to user activity streams. In this section we therefore propose some intuitive measures for novelty seeking to study such behavior in online users based on activity logs.

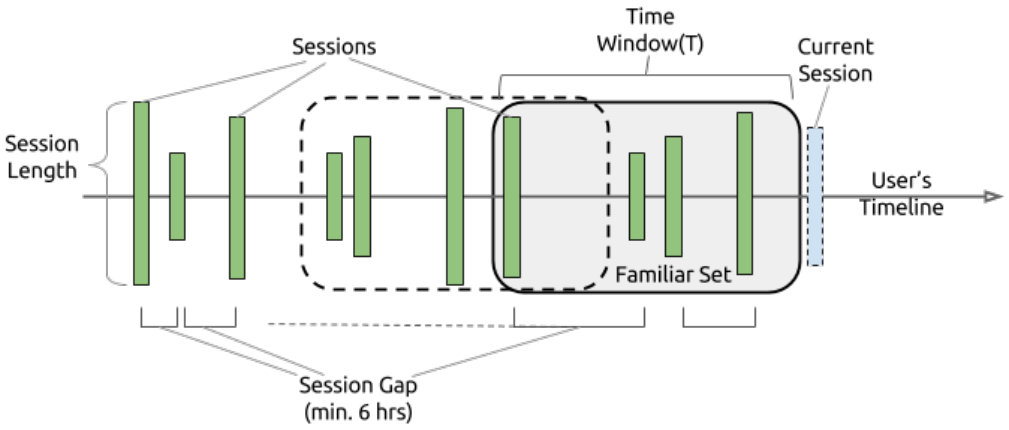


Figure 4.1: A user timeline showing how the familiar set ($famSet$) is set up to predict novelty seeking of user in the current session.

4.3.1 Session (S)

A session is defined a continuous period of user activity where a user consumes one or more items within a small time gap (identified using a minimum duration threshold). The history of a user u is represented as a sequential list of sessions — $H^u = \{S^u_{t_1}, S^u_{t_2}, \dots\}$, where $S^u_{t_x}$ denotes the x-th session of the user starting at time t_x.

4.3.2 Familiar Set (*famSet*) and Novel set (*novSet*)

A user's familiar set is defined as the set of items repeatedly consumed by the user in the recent past. Items which were consumed in the past but have not been consumed since a long time are identified as forgotten items [101] and not included in the familiar set. We use a time window T to determine recent user sessions to constitute the familiar set. A user u's familiar set $famSet^u_t$ at some time t constitute the set of unique items consumed over concurrent sessions within the time period $[t - T, t)$. To eliminate noise, items which are consumed less than a threshold number of times within the time window are eliminated from the familiar set. A user's novel set at time t, $novSet^u_t$ thus constitutes the set of items that do not belong to the user's familiar set at time t; $novSet^u_t = I - famSet^u_t$, where I is the set of all items in the system (see Figure 4.1).

4.3.3 Session Novelty Seeking Score (*nvSeek*)

Based on the definition of $famSet^u_t$ and $novSet^u_t$ for a user, we now define user novelty seeking score in a session starting at time t as the fraction of number of novel items (from set $novSet^u_t$) consumed by the user in that session over the total number of items consumed by the user in that session.

$$\text{nvSeek}^u_t = \frac{|S^u_t \cap novSet^u_t| + c}{|S^u_t| + 2 * c} \tag{4.1}$$

Here, an item corresponds to the unit of resource for the system. The allows the definition to be generic enough to accommodate both individual commodities like songs, books, movies etc. and their categories such as artist, genre and style. For example, a music recommender may consider items to correspond to individual songs or song categories such as artist and genre. Similarly, a movie recommender may consider genre and director as category level items for analyzing user novelty seeking behavior. The parameter c is Laplacian correction to avoid bias due to small sessions.

4.4 Dataset

We use music listening activity logs for our subsequent analysis of user novelty seeking behavior. The domain of music has enjoyed considerable attention in the field of psychology [111] as a medium for understanding various forms of human emotions and behaviors. Additionally, music provides a fertile ground for novelty seeking research and system development for reasons outlined below:

1. *Low risk/cost of consumption:* The cost of consuming a song is significantly lower (in terms of time and resources used) than other multimedia entities (books, movies etc) or online purchases. As a result, music listeners are more likely to experiment with novel items compared to other domains such as movies and books.

2. *Ease of availability*: As a result of proliferation of low cost (or free) online music streaming services, internet music has become ubiquitous and accompanies various user activities such as work, exercise and relaxation [111]. Hence, users spend a substantial amount of time listening to music making them easily susceptible to boredom and subsequent novelty seeking.

For our analysis we use two datasets which include the music consumption logs from two popular online music streaming websites. The first is a publicly available dataset from the online music service Last.fm[1] . The other is a more recent dataset (proprietary) from another online music service (name is withheld due to privacy reasons).

For our experiments we consider user novelty seeking behavior at both the song and the artist level. Our findings for both song and artist listening were similar and therefore we only report artist level results due to space constraints. The plots and results for song level results can be found online[2] . In each dataset, we filter users with less than 20 distinct item streams in their recorded history to eliminate inactive users in our analysis. The threshold for the gap between user activities for determining sessions is set to 6 hours based on a visual examination of the gap distribution. We set the time window for Last.fm dataset as 1 month and that for the proprietary data as 3 weeks. Small time window lengths are used as music listeners are expected to change their music preferences frequently and the system require to quickly adapt

[1] www.last.fm

[2] http://www.cs.umn.edu/∼vikas/nseeking

to the user needs. For both the dataset we include only those items in a user's *famSet* that was repeated atleast 2 times during the time window T. Finally, a Laplace correction $c = 3$ is used throughout. Both datasets are divided into burn-in, training and test periods. An initial burn-in period equal to the length of the time window T is maintained to accumulate enough data about the user for identifying the familiar set and subsequent novelty seeking behavior. We summarize the partitions and basic statistics on the datasets in the Table 4.1.

Table 4.1: Last.fm and proprietary dataset statistics (novelty seeking prediction experiments).

Name	**Last.fm**				**Proprietary Data**			
	Burn	Train	Test	**Total**	Burn	Train	Test	**Total**
Duration	1st month	2nd month	3rd month	3 months	1st-3rd week	4th week	5th week	5 weeks
Number of Users	882	758	733	882	1,642	1,209	933	1,642
Avg. Session/User	21	20	20	56	7	3	3	11
Avg. Session Length (# items)	40	39	38	39	24	23	23	24

4.5 User Novelty Seeking Behavior

We study the novelty seeking behavior of the users in our two datasets using our proposed measures discussed earlier. Via our empirical analysis we aim to verify the following two hypotheses about the user novelty seeking behavior:

1. **H1:** Users have different novelty seeking behaviors
2. **H2:** Users have dynamic novelty seeking behaviors

To understand how users differ in their novelty seeking behavior we compute the distribution of the session novelty seeking scores of the users in the first session in training period. The distributions are shown in Figures 4.2 (a,b). As apparent from the plot plots, users show substantial novelty seeking behavior in both our datasets. Users consume on an average around 43% and 46% new artists per session in the Last.fm and the proprietary data respectively ($\mu(nvSeek_t^u)_{Last.fm} = 0.4599$, $\mu(nvSeek_t^u)_{Proprietary} = 0.434$). Furthermore, the standard deviations of the session novelty seeking scores for the users of Last.fm and the proprietary datasets is found to be 0.1718 and 0.176 respectively. Both these standard deviations are found

to be significantly higher than 0 (p-value~ 0 obtained using one-sided chi-squared test for variance). *Hence, we find that users vary in their novelty seeking behaviors*, i.e. **empirical evidence supports H1**.

We then study how novelty seeking behavior varies for each user. For our analysis we consider the session novelty scores of each user as a separate random variable and estimate it's standard deviation across sessions in training period. To obtain a reliable estimate of the standard deviation of the novelty seeking score of a user, we focus on the Last.fm dataset and those users who have more than 10 sessions during the training period. Figure 4.2(3) shows the distribution of the standard deviations of users' novelty seeking scores. The mean of the distribution (0.1206) is found to be significantly greater than 0 (p-value $\sim$ 0 obtained using one-sided t-test), thus signifying differences in novelty seeking behaviors of the same user, i.e. **empirical evidence supports H2**.

Our empirical analysis reveals that the user requirements for novelty tend to be varied and dynamic in nature which makes existing novelty recommendation approaches insufficient. It is therefore critical that the novelty recommendation strategies are adaptive to the novelty needs of the users allowing them to accommodate the dynamic user preferences along with static ratings.

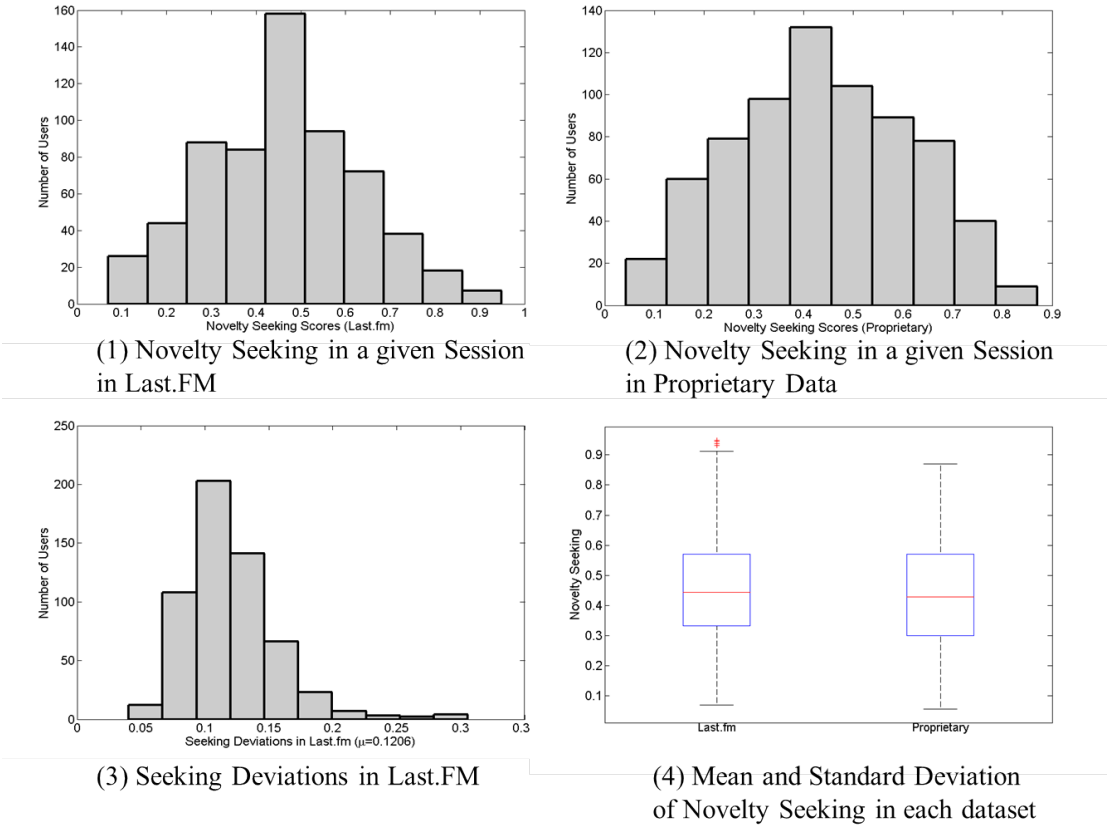
4.6 Novelty Seeking Prediction

Unlike the knowledge of data we have from the users in their current session, an online recommendation system would rather prefer to predict the individualized dynamic preferences of the users for next session. In this section we develop a model capable to predict user's future novelty needs based on the *recent* consumption behavior.

4.6.1 Features

Several behavioral measures can be employed for the purpose. These measures can be broadly grouped as explicit or implicit indicators:

1. *Explicit Indicators*: These include feedback explicitly provided by the user to indicate their need for novelty such as by answering a set of questions, clicking a button called 'Surprise Me!' or by having access to a tuner (similar to volume tuner on tapes) to adjust the amount of novelty in their recommendations. Such explicit indicators can provide a



(1) Novelty Seeking in a given Session in Last.FM

(2) Novelty Seeking in a given Session in Proprietary Data

(3) Seeking Deviations in Last.FM

(4) Mean and Standard Deviation of Novelty Seeking in each dataset

Figure 4.2: Variations in novelty seeking, across and within users, for the datasets.

fairly accurate measurement of the user's desired novelty but depend on user feedback which requires extra user effort.

2. *Implicit Indicators*: The system may infer a user's desire for novelty using various behavioral and personality cues from the user. For example, a user who has recently queried, browsed or liked new items [112] might be seeking novelty. Alternatively, some users may be more novelty seeking in general than others and such users can be identified from their past behavior. Methods which rely on implicit indicators can be completely unobtrusive to the users and can be applied even when user interaction is limited.

In this work, we concentrate on implicit indicators of user novelty seeking behavior. We leave the design of systems that incorporate explicit feedback of the users for future work. We

incorporate in our model, features based on the following two aspects of their item consumption behavior:

Diversity of their familiar set: As discussed earlier, users vary in their preferences for novelty and variety in their consumptions. This distinguishes users who repeat favorites on a loop from those who listen to top 10 radio. The individualized preferences of users for novelty and variety are identified using a measure of diversity of their currently preferred items computed using the time window approach used earlier. For a user, we define the diversity at time t (div_t) as the number of unique items in the familiar set at time t normalized by the volume of their consumptions of their familiar items in the last time window as follows:

- *Feature 1*: Diversity of their familiar set;

$$div_t^u = \frac{|\text{famSet}_t^u|}{\text{Number of consumptions of famSet}_t^u \text{ in } [t-T, t]} \tag{4.2}$$

Past works have computed user diversity via various other means such as the dissimilarity among items consumed [92], temporal diversity[9], item unpopularity [113, 114] etc. This definition can be further extended to include diversity at different granularity levels such as *genres*, *music directors* etc. in the items consumed.

Boredom with their familiar set: Users further seek more novelty when they are bored with their current selection of items (familiar set). However, in contrast to diversity, there are no easy measures for user boredom with items. A recent work by Kapoor et al. [115] addresses this problem by proposing a latent state model which estimates the boredom state of the user for an item, in addition to another sensitization state using the frequency and gaps between the consumptions of that item in the past. The sensitization state is identified as the one in which the user consumes the item frequently i.e. with small gaps between consumptions whereas the boredom state is identified as the one in which the item is consumed after *longer* gaps. The model further predicts when the user state for an item transitions between the sensitization and boredom states. Using the model, Kapoor et al. propose an approach to track and predict (a) *time gap* till the item's next consumption at time t ($G_t^{(i,u)}$) , and (b) *dynamic preference* for the item at time t ($dpref_t^{(i,u)}$).

Based on the model by Kapoor et al. we define following two features related to boredom with the familiar set:

- *Feature 2*: Cumulative **negative** preference for items in the familiar set; $negCumPref_t^u = \sum_{i \in famSet_t^u} dpref_t^{(i,u)}$. The lower is the dynamic preference for an item, the higher is the user's boredom with that item at that time. We therefore measure a user's boredom with an item at time t as the negative transformation of his/her dynamic preference for that item. The overall boredom of the user with the familiar set at time t is obtained by summing over the negative dynamic preference scores for each item in the familiar set.

- *Feature 3*: Cumulative gap till the next consumption of items in the familiar set; $CumGap_t^u = \sum_{i \in famSet_t^u} G_t^{(i,u)}$.
  The predicted gap is the likely period of time in the future in which the user would want to consume the item again given the boredom accumulated after the last consumption. Hence, larger the predicted gap for an item higher is the user's boredom with that item. We therefore measure the user's boredom with familiar set by summing over the predicted gap till the next consumption of each item in the familiar set.

4.6.2 Regression & Evaluation

We now apply a logistic regression model to the three features defined above (div_t^u, $negCumPref_t^u$ and $CumGap_t^u$) to predict $nvSeek_t^u$. The logistic regression model further ensures that our estimate $nvSeek_t^u$ falls within $[0, 1]$.

$$\widehat{nvSeek_t^u} = Logistic_\theta(D_t^u, CumPref_t^u, CumGap_t^u) \tag{4.3}$$

The novelty seeking prediction model is evaluated using Root Mean Squared Error (or RMSE) between the predicted and actual novelty seeking score for user sessions in the test period. Since to the best of our knowledge, this is the first model for predicting user future novelty seeking, we don't have any baselines against which we can compare our model. We instead, evaluate our model against a constant baseline $\overline{nvSeek}$ to show the benefits of individualized and dynamic novelty seeking prediction. The results are summarized in Table 4.2. The logistic models (*diversity* and *diversity* + *boredom* model) perform better than the constant model for both datasets in terms of RMSE. The improvement in performance are significant (p-value ~ 0 computed using the deviance test chi-squared statistic).

We further analyze our model parameters. Table 4.3 shows the regression coefficient and the significance level for each feature. Both the div_t^u and the $negCumPref_t^u$ features are found to be significant for the prediction task with p-values less than 10^{-4}. A positive value of the

Table 4.2: Novelty-seeking prediction performance evaluated using the RMSE metric.

Model	Last.fm	Proprietary Dataset
Constant	0.1686	0.1809
Logistic Model(diversity feature)	0.1574	0.1620
Logistic Model (*diversity+boredom* feature)	0.1420	0.1549

coefficient for diversity indicates that novelty seeking increases with diversity in the familiar set. This is in agreement with our hypothesis that users who prefer more diverse items are also more novelty seeking. The boredom related feature $negCumPref_t^u$ has a positive co-efficient which supports our boredom hypothesis i.e. Users display higher novelty seeking behavior when their preferences for the familiar set are low and vice-versa. However, $CumGap_t^u$ feature was not found to be statistically significant for both the datasets. This is quite likely because of the correlations between the $negCumPref_t^u$ and $CumGap_t^u$ features due to the fact that they are derived from the same boredom prediction model, which we haven't explored.

Table 4.3: The feature coefficients and their significance for the logistic regression model for novelty-seeking. Significance indicators- $0.00001 **, 0.0001*$

	Last.fm	Proprietary Dataset
Feature	**Coefficient**	**Coefficient**
div_t^u	4.7886 **	3.0963 **
$negCumPref_t^u$	0.0124 **	0.0071 *
$CumGap_t^u$	0.0006	-0.0034

Finally, the standard deviation computed for the predicted novelty seeking score for the test data ($\sigma_{Last.fm} = 0.0874$, $\sigma_{Proprietary} = 0.1034$) further provides evidence of the ability of our model to identify differences in the novelty needs of the users of the system. Both these standard deviations are found to be significantly higher than 0 (p-value~ 0 obtained using one-sided chi-squared test for variance).

In summary, the results imply that our model can provide a reliable estimate of user future novelty seeking needs using how diverse the user's preferred items were and how bored is s/he with her/his preferred items.

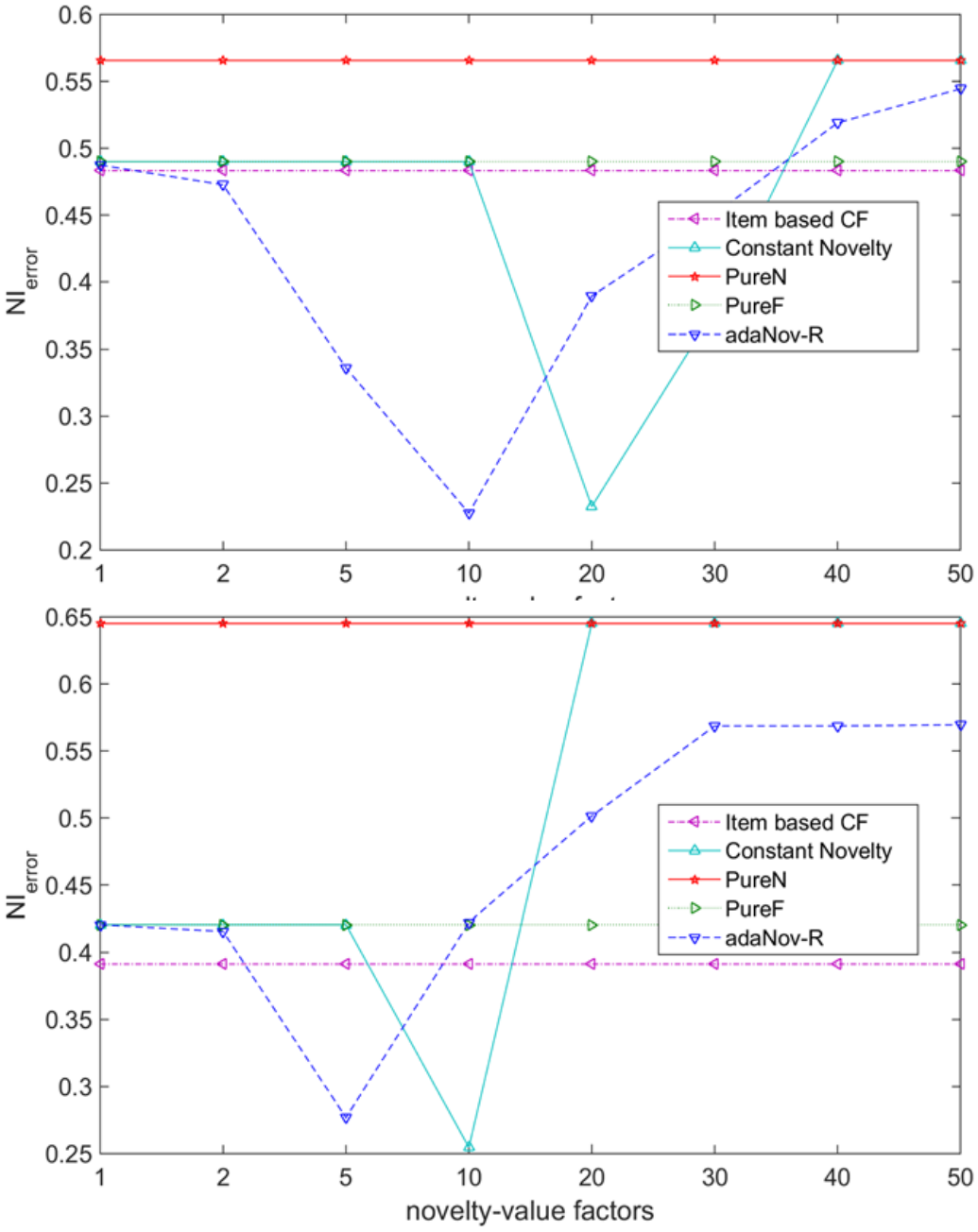
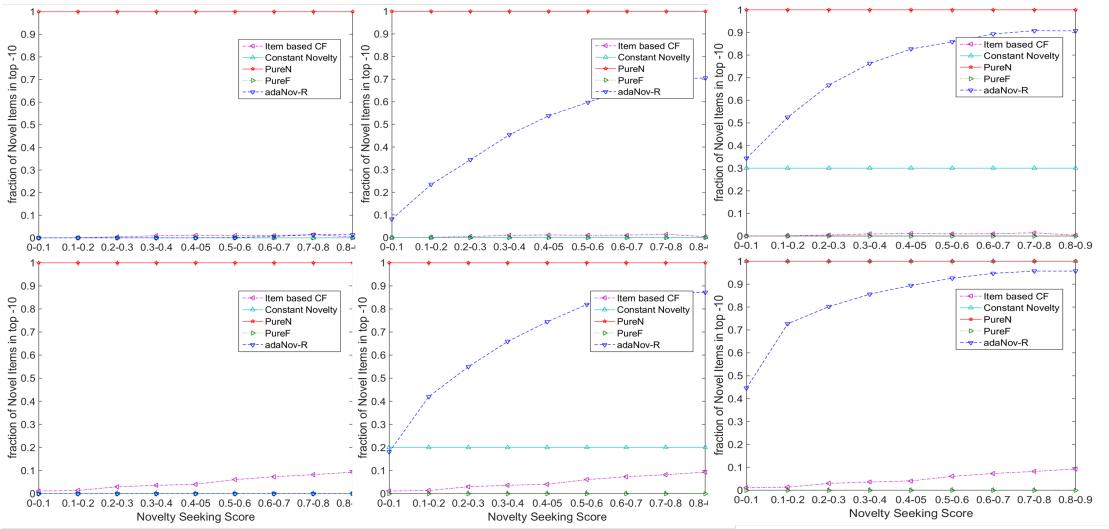


Figure 4.3: Novelty introduction error broken down by novelty-value factors for the Last.fm (top) and the Proprietary (bottom) datasets.



(a) Novelty-value factor = 1 (b) Novelty-value factor = 10 (c) Novelty-value factor = 20

Figure 4.4: Novelty introduction error broken down by *novelty seeking score* and *novelty value factor* for the Last.fm (top) and the Proprietary (bottom) datasets.

4.7 Adaptive Recommendation

Dynamic novelty preferences of users over time, as identified by previous analysis, make existing recommenders' item preference based novelty approach [87, 93] insufficient. They consider item ratings and consumption to have static preference thus recommending a set of novel items irrespective of amount of novelty actually desired from the system by user in a session. We aim to bridge this specific gap between static and dynamic item preferences of users by using their dynamic novelty seeking desires.

However, the problem of novelty in recommendation poses several unique challenges for the recommendation community. We focus on three specific aspects of the problem:

1. *Skew in the recommendation performance of familiar vs. novel items*: Recommendation of novel items is inherently a different problem from the recommendation of familiar items. The later involves a smaller subset items which a user is already aware of and the system has a strong indication from the user that s/he has preferred those items in the past. The former, on the other hand, involves the recommendation of unknown items from a large multitude of items in the system's inventory which user might like. As a

result, recommendation of novel items is more challenging with higher uncertainties in their accuracy than the recommendation of familiar items.

2. *Incentives for novel recommendation*: Despite loss in accuracy, novel recommendation may be more valuable to the system than familiar items. Since users of certain systems (such as music and movie streaming websites) may find novel recommendations more useful and satisfying than familiar items. Such systems may chose to incentivize novel recommendations over familiar recommendations in their system design.

3. *User Need for Novelty*: A user's acceptance of novel recommendation varies based on their novelty seeking behavior. A user who is seeking more novelty desires more novel items from the recommender than a user who is familiarity seeking at the moment.

In this work, we propose a novel approach for novelty recommendation which is based on modulating the introduction of novel items in a user's recommendation list given his/her current novelty seeking needs and the system's incentives for novel recommendations. We consider a user session s_t^u for which the system generates a top-N recommendation list. Our adaptive novelty recommender (*adaNov-R*) consists of three modules, namely: (1) Novelty Seeking Prediction module, (2) Item Ranking module, and (3) Adaptive Recommendation module as summarized in Figure 4.5. We discuss these components in detail in the following subsections.

4.7.1 Novelty-seeking Prediction Module

This module generates a prediction of the novelty seeking score of the user for the session in which the recommendations have to be generated. As explained earlier, the module can either use explicit or implicit indicators to learn the novelty seeking preference in next session. In this case, we use the logistic regression model described in the previous section to predict user's session novelty seeking score ($\widehat{nvSeek_t^u}$).

4.7.2 Item Ranking Module

This module is responsible for producing a preference ranking list for novel and familiar items for each user based on their history of past consumption (or rating). Several existing recommendation algorithms can be adopted for this purpose. The standard recommendation techniques such as item-based or user-based collaborative filtering or matrix factorization can be used to

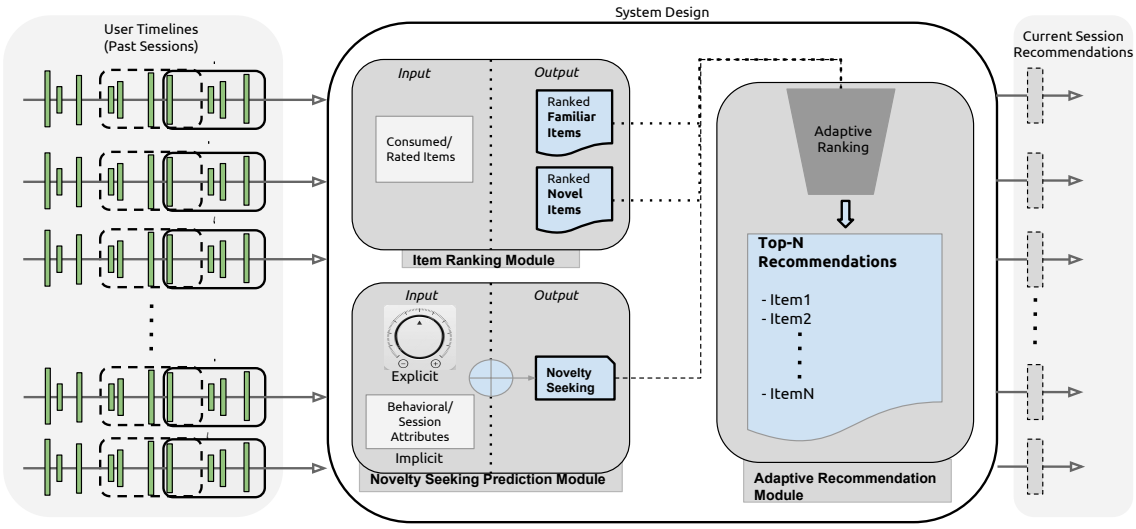


Figure 4.5: Adaptive novelty seeking recommender system design.

generate a ranking (based on predicted scores) for the preferred items of the user denoted as R_t^u. The ranking list R_t^u is then post processed to generate two new ranking lists F_t^u (list of familiar items - items that exist in $famSet_t^u$) and N_t^u (list of novel items - items that exist in $novSet_t^u$) such that items in each list maintain their relative order as in the original list. Alternatively, we can use specialized algorithms for ranking novel [93, 110] and familiar items [115] respectively to generate N_t^u and F_t^u.

We use the off-the-shelf *item-based* collaborative filtering technique for our implementation. The item-based recommender computes preference scores for items based on their similarity to the items already rated by the user. This approach is easy to train and has been shown to achieve good performance in several real life systems [82]. Since, we do not have explicit ratings in our dataset, we modify the item-based recommendation algorithm to include the dynamic preference score $dpref_t^(i, u)$ (defined in last sub-section) instead of ratings for generating preference scores for items. This allows our ranking list to be sensitive to the temporal dynamics in a user's preferences for past preferred items with good recommendation performance. The modified item scoring function is then formulated as:

$$dpref_t(j,u) = \frac{\sum_{i \in Neighbors(j)} sim(i,j) * dpref_t^{(i,u)}}{\sum_{i \in Neighbors(j)} sim(i,j)} \tag{4.4}$$

where, j is an item for which the dynamic preference score is predicted, $sim(i,j)$ is the similarity between i and j computed using cosine similarity and $Neighbors(j)$ are the nearest neighbors (top 50) of item j.

We use this scoring function to generate the ranking list R_t^u and then extract the ranking lists N_t^u and F_t^u from R_t^u.

4.7.3 Adaptive Recommendation Module

The Adaptive Recommendation Module uses the novelty seeking prediction score from Novelty Seeking Prediction module and the item ranking lists N_t^u and F_t^u generated by the Item Ranking module to generate the final top-N recommendation list for the user session. Our approach for generating the final ranking list involves incorporating the top novel and familiar items in the list such that the fraction of the list occupied by novel items (and familiar items) is based on optimizing a new metric of recommendation performance for different novelty seeking users proposed by us.

F-measure have been used as a standard metric for evaluating recommendation performance in top-N recommendation task. The f-measure metric ($F1$) determines how well a system can recommend preferred items to the users and is defined as the harmonic mean of the precision (p) and recall (r) scores computed as below:

$$p_t^u = \frac{|R_t^u \cap S_t^u|}{|R_t^u|} \quad ; \quad r_t^u = \frac{|R_t^u \cap S_t^u|}{|S_t^u|} \tag{4.5}$$

$$p = Avg_{u,t}\, p_t^u \quad ; \; r = Avg_{u,t}\, r_t^u; \; F1 = \frac{2 * p * r}{p + r} \tag{4.6}$$

$$\tag{4.7}$$

However, vanilla $F1$ cannot reflect the design principles for our work, which is to consider both novel and familiar items, user novelty needs and the system's incentives for novel recommendation in the system design. We therefore propose a new metric, the weighted F-measure, for evaluating our recommendation performance and optimize this metric for different novelty seeking users. The weighted $F1$ metric is derived from multi-class cost sensitive learning literature [116] such that the novel and familiar items are considered as two different classes of

items for measuring recommendation performance. The class specific precision and recall can then be computed as follows:

$$p_t^u(Z) = \frac{|R_t^u \cap S_t^u \cap ZSet_t^u|}{|R_t^u|} \quad ; \; p(Z) = Avg_{u,t}\, p_t^u(Z) \tag{4.8}$$

$$r_t^u(Z) = \frac{|R_t^u \cap S_t^u \cap ZSet_t^u|}{|S_t^u \cap ZSet_t^u|} \quad ; \; r(Z) = Avg_{u,t}\, r_t^u(Z) \tag{4.9}$$

Here, $Z = \{fam, nov\}$.

The overall performance score metrics, weighted precision (wp) and weighted recall (wr), are then measured as a weighted average of the class specific performance scores, weighted by (a) fraction of the items of the two classes consumed by the users (user novelty seeking score) and (b) class specific cost factor (novelty value factor - *NVF*). The weighted F-measure ($wF1$) is further computed as a harmonic mean of the weighted precision and recall scores.

$$wp_t^u = \frac{nvSeek_t^u * NVF * p_t^u(nov) + (1 - nvSeek_t^u) * p_t^u(fam)}{nvSeek_t^u * NVF + (1 - nvSeek_t^u)} \tag{4.10}$$

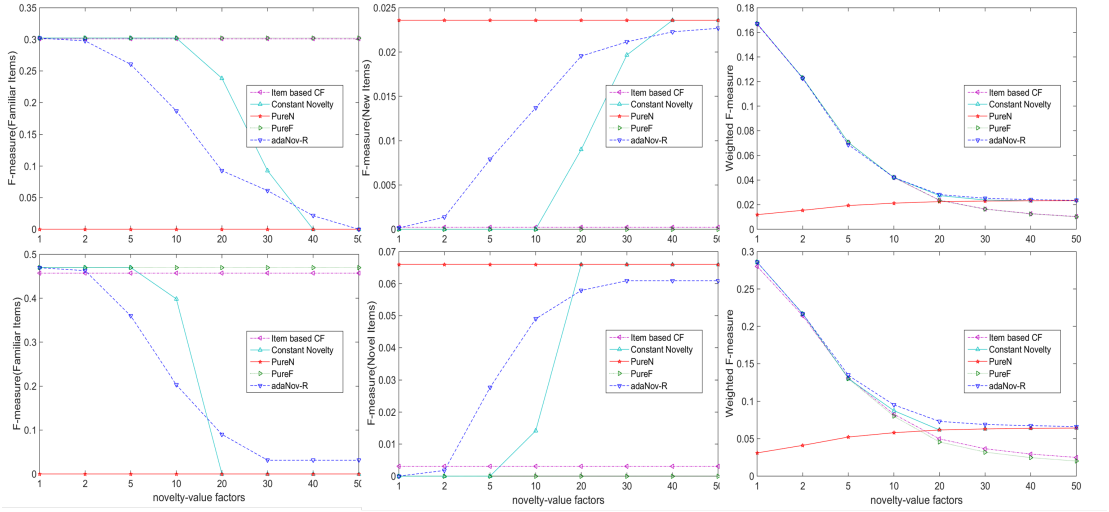
$$wr_t^u = \frac{nvSeek_t^u * NVF * r_t^u(nov) + (1 - nvSeek_t^u) * r_t^u(fam)}{nvSeek_t^u * NVF + (1 - nvSeek_t^u)} \tag{4.11}$$

$$wp = Avg_{u,t}\, wp_t^u \; ; \; wr = Avg_{u,t}\, wr_t^u \; ; \; wF1 = \frac{2 * wp * wr}{wp + wr} \tag{4.12}$$

Our choice of weighting scheme is further explained. Weighting by novelty seeking score allows the recommender to weight the recommendation performance for the two classes of items using the class wights which is the fraction of items of the two classes actually consumed by the user. This allows us to place more emphasis on the system's recommendation accuracy for novel items (vs. familiar items) for sessions in which the user is more novelty seeking and similarly place more emphasis on it's recommendation of familiar items for sessions in which s/he is less novelty seeking. Furthermore, we consider different values (negated costs) associated with recommendation of novel and familiar items by adapting methods from cost sensitive evaluation. The novelty-value factor (NVF) denotes the value (negated cost) of 1 correct novel recommendation versus the value of 1 correct familiar recommendation, and hence denotes the value of novelty, serendipity etc. for the system. The value of the novelty-value factor may be set for particular application using domain knowledge or via user studies and live experiments.

Now to learn the right fraction of novel items we propose a greedy approach to optimize the weighted f-measure of a recommender for different novelty seeking users. We segregate

users into ten different partitions based on their novelty seeking score such that $[[0-0.1) \rightarrow \textit{partition}_1, \ldots, [0.9-1] \rightarrow \textit{partition}_{10}]$. We then learn a rule-based novelty introduction function $\textit{partition}_i \rightarrow x_i$ for partitions 1-10 on training data where x_i is fraction of novel items to include in the recommendation list that maximizes the weighted f-measure for partition i. For example, $\textit{partition}_i \rightarrow 0.4$ suggests that inclusion of 4 novel items in the top 10 recommendation list provides the best weighted f-measure for $\textit{partition}_i$ over all other fractions of novel items. Having once trained our novelty introduction function, the Adaptive Recommendation Module first uses the predicted novelty seeking score to identify the expected partition in which the user falls in that session and then uses the learned fraction for that partition to incorporate an appropriate amount of top novel and familiar items in his/her final recommendation list.



(a) F-measure (Familiar Items) (b) F-measure (New Items) (c) Weighted F-measure

Figure 4.6: Weighted F-measure broken down by *novelty value factor* for Last.fm(top) and Proprietary data (bottom).

4.8 Results

We evaluate our adaptive novelty recommender on two aspects (a) ability to meet the user's need for novelty; and (b) accuracy of the *top-N* recommendations. Both these aspects are evaluated for users with different novelty seeking scores and different novelty-value factors;

$NVF = \{1, 2, 5, 10, 20, 30, 40, 50\}$. The wide range of novelty value factors allows us to test the general robustness of the approach for different system design consideration.

We further compare our model against alternative strategies for novelty recommendation. We first consider the two extreme models *PureN* - which recommends only novel items and *PureF* - which recommends only familiar items. We then compare our model against a constant novelty recommender (*CN*) that recommends a constant number of novel items to the users. The constant is optimized for a given novelty value factor of the system using the training data. The *CN* allows us to show the value of an approach in adapting to the user time-specific novelty needs against a constant factor. Finally we compare ourselves against the standard *item-based* recommendation approach which only considers users' static item preferences.

4.8.1 Novelty Introduction Error(NI_{error})

We evaluate various recommendation strategies on their ability to meet the novelty needs of the user by incorporating enough novel items in the recommendation list as are needed by the user. This achieved by computing the root mean square error (RMSE) between the fraction of novel items recommended and the fraction of novel items actually consumed by the user for all user sessions in the test period. We call this error the *Novelty Introduction Error*. The novelty introduction metric ignores the *quality* of those novel recommendations which is the focus of the next subsection.

$$NI_{error} = \sqrt{\sum_{u,t} (|novel^{u}_{recommended,t}| - |novel^{u}_{consumed,t}|)^2} \tag{4.13}$$

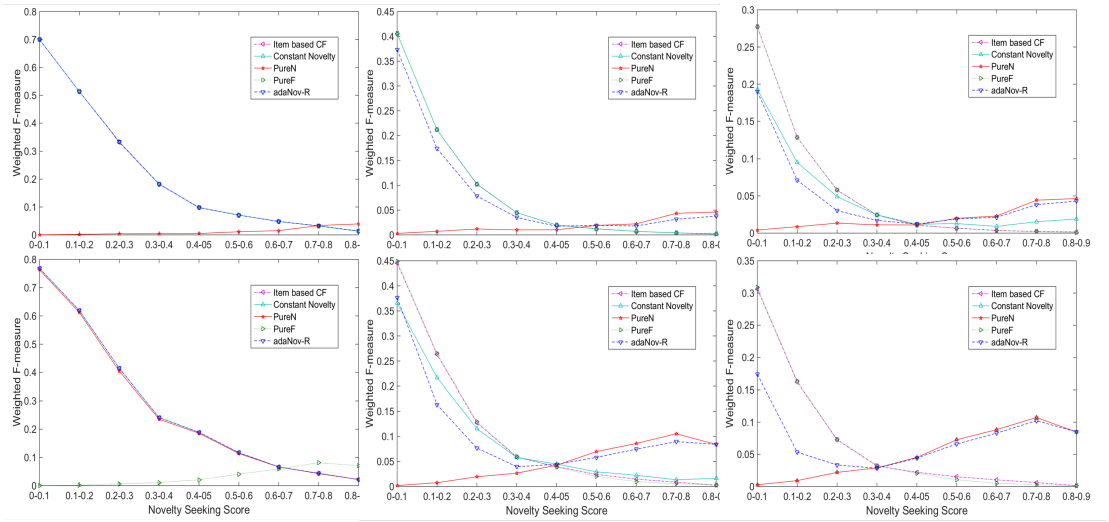
We analyze the novelty introduction error for our model and the baselines for different novelty-value factors and users with different novelty seeking scores.

Robustness to the novelty value factor: Figure 4.3 displays the novelty introduction error for the various models for different novelty value factors. We find that the novelty value factor impacts the NI_{error} for the adaNov-R and constant novelty model. For both the models, error decreases and then increases. This is because, for low values of novelty value factor, a system is more inclined to make accurate familiar recommendations than possibly inaccurate novel recommendations, resulting in lower introduction of novelty than required. On the other hand, for high novelty value factors, system provides incentives to make more novel recommendations resulting in the introduction of more novel items than required. The PureF, item based and

PureN are independent to the choice of novelty value factor. Furthermore, PureF and item-based have comparable novelty introduction errors than other models, their behavior is not adaptive to different user novelty seeking scores, in addition to different design requirements of the system, discussed further in the next subsection.

Impact of the novelty seeking score: We further look at the number of novel items recommended for different user novelty seeking scores. We find that as expected, PureN, PureF and constant recommend the same number of novel items for all ranges of user novelty seeking behavior (as shown in Figure 4.4) . However, adaNov-R recommends more novel items when a user is more novelty seeking (for NVF ¿ 2). The recommendation of novel items increases with novelty seeking score for items based model as well but the increase is very small. Furthermore, the actual number of novel items provided to users with different novelty seeking scores varies based on the novelty value factor.

In summary, the adaNov-R allows the system to vary the number of novel items recommended based on user novelty seeking behavior. However the error metric for adaNov-R is sensitive to the novelty value factor, as the model allows the introduction of more (or less) novel items when the value of novel recommendation is higher (or lower).



(a) Novelty-value factor= 1 (b) Novelty-value factor$_c$= 10 (c) Novelty-value factor= 20

Figure 4.7: F-measure (familiar), f-measure (novel) and weighted f-measure broken down by *novelty seeking score* of the user (Last.fm(top), Proprietary data (bottom)).

4.8.2 Recommendation Accuracy

We evaluate the ability of the recommendation strategies to make accurate recommendations to the user using the weighted f-measure metric.

Robustness to the novelty value factor: We look at the overall weighted f-measure metric (shown in Figure 4.6) for different models for different novelty value factors. We find that the performance of adaNov-R is either superior to or comparable to the best performing baseline based on the novelty value factor. As expected, PureF and item-based favor lower novelty-value factors and PureN favors higher novelty value factors. The *adaNov-R* on the other hand can adapt to the choice of novelty value factor, and has a behavior that is comparable to that of PureF (PureN) for the lower (higher) values of novelty-value factor. It further has a superior weighted f-measure than all the other models for values of novelty value factor = 10 and 20. The results are significant for a $p - value < 10^{-5}$ (Wilcoxon rank sum test). The f-measure scores for adaNov-R for familiar and novel items further show that our model can balance it's recommendation performance on both classes of items quite well. The constant novelty model has the second best overall performance and this is because although it is adaptive to the choice of novelty value factor for the system, it cannot adapt to differences in user novelty needs as further investigated in the next subsection.

Impact of the novelty seeking score: We find (Figure 4.7) that the performance of PureF and item-based is high for users with low novelty seeking behavior but consistently declines as user novelty seeking increases. The performance of PureN on the other hand is low for lower values of novelty seeking but increases as novelty seeking score increases. The adaNov-R, is found to perform reasonably well for all values of novelty seeking which suggest that our model can modify it's behavior to suit the accuracy needs of users with different novelty seeking behaviors. Finally, the constant novelty model is not adaptive to user novelty seeking needs and achieves the performance to PureF (PureN) for low (high) values of novelty value factor.

In summary, the adaNov-R has a overall better performance as it can adapt to different novelty value factors and different user novelty seeking behaviors whereas other baselines favor some system designs and user needs over others.

4.9 Conclusion & Future Work

Our contributions in this work are summarized as follows:

- We defined an intuitive measure for the variable novelty seeking preference of the users.
- Using our measure we empirically verified individual and temporal variations in user novelty seeking behavior.
- We developed a model for predicting the variable novelty seeking preferences of the users with good predictive performance.
- We proposed an adaptive novelty recommender *adaNov-R* to adapt for recommendation of novel items to the predicted novelty seeking need of the user, using off-the-shelf techniques.
- We exhaustively validated our model for different system design criteria for novelty as well as for differing novelty seeking needs of users.

Our work is the first of it's kind to consider dynamics in user needs for novelty. Although, we have tested our methods only on the music domain, our techniques can be easily applied to the recommendation of other multimedia content like movies, videos and blogs. We can further extend our work to incorporate other item categories such as genres and director for movies and topics for blogs. Other directions for future work include:

- Extending our measures for novelty and user novelty seeking to incorporate other notions commonly used in existing literature such as item dissimilarity [114].
- User evaluation of the adaptation scheme to meet the novelty seeking needs of the users.
- Identification of other behavioral traits, such as personality characteristics like (a) Exploratory excitability (b) Impulsiveness (c) Extravagance, and (4) Disorderliness for estimating the novelty needs of the users.

Chapter 5

Predicting User return Time Using Hazard Models

5.1 Introduction

User attention is perceived as the most important resource in the internet era [117]. The web is described [118] as a *'virtual theme park where most rides are free such that revenue is generated through "selling eyeballs" to advertisers'*. The ad-supported economy of the web has the web-services vying for users' time rather than their money. Having a large loyal and dedicated user base has several indirect benefits as well. Many services grow with their users, improving themselves based on their feedback and through the power of big data analytics on their activities logs. A common example is the Google search engine, which has perfected its query auto-correct feature primarily using user click-through data, as well as improved its search performance regularly using user search histories. Furthermore, an active community can be tapped to create new content that benefits the other users of the service and the service as a whole as seen for popular social networks such as Facebook and Twitter.

There is tremendous competition among the rapidly increasing number of web services for the finite and limited resource corresponding to user attention. Although, attracting new customers is crucial for any business, it is generally much easier and cheaper to retain existing customers [119, 120]. This directly results in a great deal of emphasis being placed on engaging one's current user base. Customer retention efforts have been heavily researched in sectors such as telecommunication [120], financial services [121], internet services [122] and other utilities

etc. which follow the subscription based model. The methods in these domains have focused on identifying potential churners in the user population, where churners are defined as those current subscribers who are not likely to renew their subscription in the coming months. Once detected, the churner population is targeted with retention strategies like offers, customer solutions and recommendations to win them back.

However, such methods cannot be directly applied to solving the user retention problem for web services due to the following reasons:

1. **Difficult to define churn for a non-contractual setting.** A non-contractual business such as a free web service, does not receive a definitive indicator of termination from the user. To counteract this problem, some alternative definitions of churn have been proposed. Churn is defined as a significant drop in a user activity levels [123]. Another work addresses this problem by first providing a definition for a loyal user of a service and then defining churners as those users who were loyal to the service but are no longer so [124]. However, such methods remain sensitive to changes in their proposed definition of churn.

2. **Highly dynamic user visitation behavior.** Web services offer none or negligible switching costs to users. With no financial commitments towards a service, users switch quite frequently between different services. The highly dynamic nature of user visitation behavior makes it difficult to define typical activity volumes for a user and to segregate users as active and inactive with respect to the service.

To adapt to the unique incentive structures and dynamic user base, in this work we propose a novel retention metric which tracks the user return rate for addressing growth and retention in web services. The user return rate is defined as the fraction of the existing users returning to the service on a particular day. It is beneficial for a web service to improve its user return rate in order to increase its revenue. Predictive analysis of user return times can direct such improvements efforts. Return time prediction allows a service to identify indicators of earlier (longer) return times for their users. Identifying such indicators and quantifying their impact on user return times offers a service insights into its practices. It also enables a service to employ corrective measures and improve the experience to its users. Secondly, a service can identify sections of its user base that are not likely to return soon. Studies have shown that the longer the users stay away from a service, the less likely are they to return in the future [125]. Early

identification of users who are not likely to return soon to the service allows the deployment of suitable marketing strategies to encourage those users to engage with the service again.

We propose a hazard model [126] from survival analysis to predict user return times. The hazard based models are preferred over the standard regression based methods for this problem due to their ability to model particular aspects of duration data such as censoring. More importantly, the Cox's proportional hazard regression model is used as it can incorporate the effects of covariates [1] . We apply the model for return time prediction on real-world datasets from two popular online music services.

We now summarize the key contributions made by us in this paper:

(a) We formally define an approach for targeting retention solutions in free web services via user return time prediction.

(b) We propose the Cox's proportional hazard to model dynamic return events and incorporate the effects of covariates for return time prediction. We develop useful return time predictors and conclude correlations between user usage patterns and their return times.

(c) The Cox's proportional hazard model outperforms state-of-the-art baselines in both return time prediction and user classification based on predicted return time.

The rest of the paper is organized as follows. In **Section 2** we provide a brief overview of the related research in the area of churn prediction and the use of hazard based methods. We then formally define our problem and lay out our contributions in **Section 3**. In **Section 4** we describe our hazard based predictive model and provide details of the covariates used and the model estimation procedure. In **Section 5** and **Section 6** we discuss the experimentation setup and the results. We summarize the conclusions from our experimental analysis and provide future directions for our work in **Section 7**.

5.2 Related Work

In this work, we propose a new approach for directing growth and retention solutions for web services through return time prediction. Traditionally studies have focused on the problem of

[1] We use the term covariates to describe features or predictors in our model. The choice of terminology is based on that used in the survival analysis literature from where we adopt our model.

churn prediction defined as a binary classification problem where users are categorized based on several behavioral and demographic features into two categories: future churners or non-churners. The popular data mining techniques used for building classifiers for churn prediction include decision trees such as CART and C4.5 etc. [120], logistic regression [119], support vector machines [127] and neural networks [128], though random forests [127, 129] are found to be better in performance. Ensemble methods have been used to combine multiple classifiers to construct powerful meta-classifiers and to handle the class imbalance problem typical to churn prediction [130]. Alternatively, approaches from survival analysis have been used to model the dynamics in the churn event rate with user tenure [131]. The churn event rate for users is found to decline with tenure such that new users are more likely to churn than tenured users.

A major hurdle to applying these methods to free-to-use services discussed in this paper is to provide an appropriate definition of churn. Studies on user lifetime modeling and retention for online environments have used different criteria for defining the loss of a customer, such as the period of inactivity [132], decrease in activity [123] or indirectly, via a definition of loyal users [124], discussed earlier. Yang et al. [132] have further studied how user participation patterns affect the length of their lifetimes on online knowledge sharing communities. However, most of these methods focus on the length of user participation rather than the volume of their activities. Instead, online businesses are increasingly paying attention to their returning users rather than the count of their registered users. Further, the research community has started concentrating on analyzing and modeling users activities on different types of websites [133, 134]. A major focus of these methods have been to understand how websites memberships, specifically measured in the number of active users, evolves with time and correlate such factors to the success or failure of the website [135]. Also, many studies on the measurement and improvement of intra-site [136, 137, 138] and inter-site engagement have emerged [139]. Many of these studies identify return time as a robust metric for user engagement. All these factors suggest that continuous tracking and improvement of user engagement, measured in terms of their return time, is crucial for the performance goals of web services. Hence, in this work we directly focus on the return time metric for organizing retention efforts for web services. We use a Cox's proportional hazard regression model [126] from survival analysis for this problem as the model can quantify the impact of covariates on the target event rate. This unique property results in the Cox's Model being a popular choice for several online user behavior studies [138,

132]. Additionally, we also define different types of return time predictors suitable for this problem.

Several types of covariates have been used for churn prediction. RFM models [140] propose the use of three variables, Recency, Frequency and Monetary value of their previous interaction for identifying potential churners. Other covariates based on demographics, contractual details, service logs, use patterns, complaints and customer service responses [141, 120] have been found useful. We use some of these covariates in our model. In addition, we also incorporate user behavior related covariates in order to understand how user interactions while engaging with the service affect the rate of their return in the future. A special feature of our model is that it can handle the recency variable implicitly by computing the expected future time of return for the users given their length of absence from the service.

5.3 Return Time Prediction for Web Services

A user's visitation behavior on a free web service tends to be quite flexible and arbitrary post registration partially due to the lack of financial investments and constraints. Instead, the length of the tenure of users of web service displays a power law distribution with most of the users never returning back to the service [142]. In this work, we adopt a unique methodology for analyzing the dynamic user visitation data by directly modeling the user return time.

5.3.1 Problem Statement

We define users as belonging to either of the two activity states - the *in* and the *out* states. When users visit the service, they are said to be in the *in* state; while, when they do not visit the service, they are said to be in the *out* state.

We focus on the problem of predicting the return time of the users which is the time the user spends in the *out* state. The return time for a user can potentially extend to infinity (for users who never return back to the service). Therefore, a threshold, t_d is defined on the return time and we predict the return time for the users up to time t_d. The return time prediction problem may be formally defined as follows:

Definition 1 Given that the last time the user was in the *in* state was at time t_0, the return time prediction problem is to predict the quantity $min(t_r, t_d)$, also called the truncated return

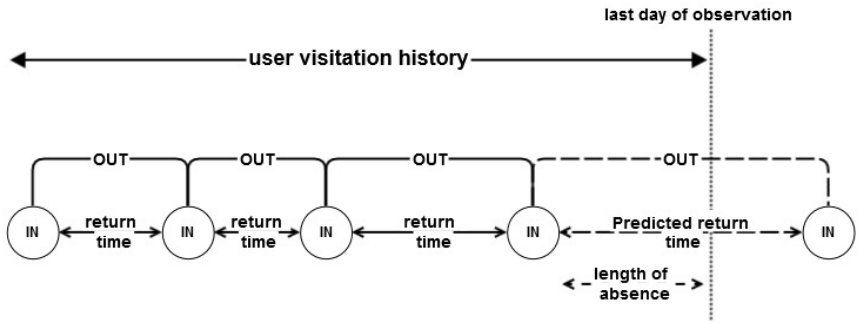


Figure 5.1: State Space Diagram

time (T_{rd}), where t_r is the total time the user spends in the *out* state and ranges from 0 to ∞, t_d is a finite threshold on the return time and either of the following holds:

(a) the user is expected to return to the *in* state at time $t_0 + t_r$, if $T_{rd} = t_r$, or

(b) the user is expected to stay in the *out* state for at least t_d units of time, if $T_{rd} = t_d$

Figure 5.1 provides a diagrammatic representation of the user return time prediction problem.

5.3.2 Time Dependence in User Return Time

The time duration between events has been studied extensively in queuing theory, for example to study the waiting time between customer arrival and customer service events. Such events are commonly modeled to generate from a Poisson process such that the waiting times follow the exponential distribution. An attractive property of the exponential distribution is the memoryless property which entails that the future rate of occurrence of the event is independent of the elapsed time. For a random variable T denoting the time of occurrence of the event, the following equation is said to hold if the memoryless property is satisfied:

$$P(T > t + s|T > s) = P(T > t) \tag{5.1}$$

However, several phenomena are seen to defy the simple memoryless property in interesting ways. For example, the rate of adoption of new products is found to increase with the elapsed

time [143]. Alternatively, the rate of events like responses to surveys, promotions [144] etc. is seen to decline with the elapsed time. The decline in future event rate with the elapsed time, has been referred to as *'inertia'*. We suspect similar type of inertia in user return behavior. For duration data showing time dependence, it becomes meaningful to compute the expected future time of the event given the elapsed time, $E(T|T > s)$. We, now define the problem of predicting the expected future time of return of the users given their length of absence (LOA) from the service.

Definition 2 Given that the last time the user was in the *in* state was at time t_0, and he has already been in the *out* state for time t_s, the future return time prediction problem is to predict the quantity $min(t_{fr}, (t_d - t_s))$, also called the truncated future return time T_{frd}, where t_{fr} is the additional time the user spends in the *out* state and ranges from 0 to ∞, t_d is a finite threshold on the return time and either of the following holds:

(a) the user is expected to return to the *in* state at time $t_0 + t_s + t_{fr}$, if $T_{frd} = t_{fr}$, or

(b) the user is expected to stay in the *out* state for atleast $t_d - t_s$ more units of time, if $T_{frd} = t_d - t_s$

5.4 Method

We consider a time window over which user return time observations are collected. Each return time observation can be associated with a set of covariates influencing its magnitude. Hence, the data can be represented as a set of tuples: $< X, T >$ where, T is the return time observation and X is the vector of covariates associated with that observation. Since a user can return to the service multiple times during the considered time window, we can have multiple tuples corresponding to a single user.

There are two aspects of the collected data that need special attention.

1. Censoring: Duration data which is collected over a fixed time period tends to have incomplete observations corresponding to events which were yet to happen at the end of the study period. Such observations are said to be censored and this particular type of censoring is called right censoring. In order to capture censored observations as well, a special variable *status* is added to the representation of duration times. The *status* variable is set to 0 when the time variable represents the actual observation of return time

whereas it is set to 1 when the time variable represents a censored observation. In the latter case the time duration represents the time gap between the user's last visit and the end of the study period. Ignoring censored observations biases one's analysis towards earlier returns. A major advantage of the hazard based methods is that they can handle censored observations quite well.

2. Recurrent observations: The collected data may contain more than one return time events, also called recurrent events, per user during the study period. The active users have many more return time observations than inactive users. Retaining these observations can bias our analysis towards the active users which is detrimental to our objective of targeting losing customers. However, we lose information if we throw away these observations. Instead, we use a simple weighting scheme for handling recurrent events. We weight each observation corresponding to a user with the inverse of the number of observations made for that user. Hence, each user has a unit weight in the data but we incorporate all observations made for him/her. Later in the paper, we discuss the sensitivity of our results to this weighting scheme. Some care needs to be taken while testing models when working with recurrent data and we discuss that in our *Experiments* section.

5.4.1 Hazard Based Prediction Model

Survival analysis is a branch of statistics which deals with the time of occurrence of events, also called duration modeling. It offers a rich set of methods which allow us to easily address questions like what is the probability that an event is going to happen after t units of time or what is the future rate of occurrence of the event given it has not happened in t units of time. In this work we deal with discrete measures of time. Two functions are useful for analyzing duration information:

The survival function at time t is defined as:

$$S(t) \;=\; P(T > t) \tag{5.2}$$

where T is a random variable denoting the time of occurrence of the event. The hazard function measures the instantaneous rate of occurrence of the event at time t, conditioned on the elapsed time t.

$$\lambda(t) \;=\; P(T \;=\; t \mid T \geq t) = -S'(t)/S(t-1) \tag{5.3}$$

The Cox's proportional hazard model is commonly used to incorporate the effect of covariates on the hazard rate. The model is based on the simple assumption that the covariates affect the magnitude of individual hazard rates but not the shape of the hazard function. Expressed mathematically,

$$\lambda(t) = \lambda_0(t) * \exp(\beta_1 * X_1(t) + \beta_2 * X_2(t) + ...) \tag{5.4}$$

where, λ_0 is the baseline hazard function, $X_1(t)$, $X_2(t)$, etc. are the covariates which may be static or may vary with time and β_1, β_2 etc. are the regression coefficients. The ability of the Cox's model to handle time-varying covariates is an important feature of the model.

One can obtain the survival function from the hazard function using the following equations:

$$\Lambda(t) = \sum_{0}^{t} \lambda(u), \tag{5.5}$$

$$S(t) = \exp(-\Lambda(t)). \tag{5.6}$$

where Λ is defined as the cumulative hazard function. The expected time of return can then be computed using the equation below:

$$E(T) = \sum_{0}^{\infty} S(t). \tag{5.7}$$

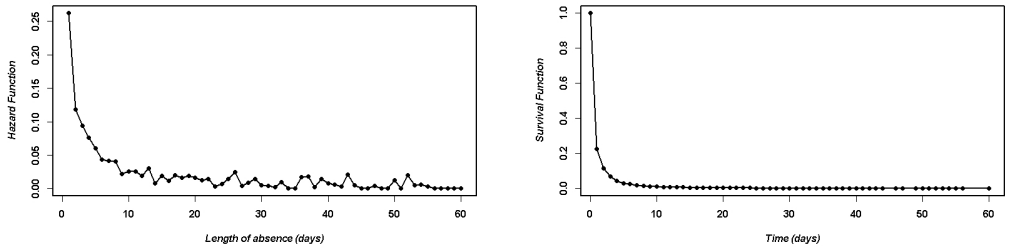
Furthermore, the expected future time of return given the time not returned for (t_s) can be computed as follows:

$$E(T|T > t_s) = \frac{1}{t_s} \sum_{t_s}^{\infty} S(t). \tag{5.8}$$

The survival function is truncated beyond a point of time or when the probability of survival drops below a threshold in order to prevent the return time estimate from diverging. For our prediction problem, we impose t_d as an upper bound on the return time estimate. Hence, the expected return time and the expected future return time computations can be re-defined as:

$$E(T) = \sum_{0}^{t_d} S(t), \tag{5.9}$$

$$E(T|T > t_s) = \frac{1}{t_s} \sum_{t_s}^{t_d} S(t). \tag{5.10}$$



(a) Baseline Hazard Function (b) Survival Function

Figure 5.2: The baseline hazard function and the survival function computed on the Last.fm training dataset.

5.4.2 Model Estimation

The Cox's proportional hazard model is a semi-parametric model as it does not assume a mathematical form for the baseline hazard function. Instead, the model can be broken down into two factors. The first factor represents the effect of the covariates on the hazard rate. The effect parameters (regression coefficients) are learnt by maximizing the partial likelihood which is independent of the baseline hazard function. Once the regression coefficients have been learnt, the non-parametric form of the baseline hazard function is estimated using the Breslow's method. Cox's seminal paper [126] is a good reference for the details of the estimation procedure.

We use the standard survival package in R for estimating the Cox's model. The survival package can handle weighted data instances. We use days as the unit of time for our analysis as most of the users in our datasets are found to return within the first week. A user is considered to have visited the service on a day if he performed at least one activity on the service on that day. One may define more stringent criteria on user activity for this purpose, such as minimum time spent, number of interactions etc. The threshold (t_d) for the return time prediction problem is set to 60 days, which is a reasonably large value and beyond which users are already the focus of retention efforts. Return time observations larger than 60 days are hence assumed to be censored. In our experiments, we also study the performance of the model for different choices of the threshold.

5.5 Experimental Setup

We now evaluate the performance of the Cox's proportional hazard model for solving our proposed return time prediction problem. We consider both the performance of the model at predicting the return time of the user and at classifying users based on their expected return times. Such a categorization procedure is the logical next step for a service looking to target marketing strategies to users not likely to return soon. For both the problems, we also evaluate how well the Cox's model can incorporate the LOA information by re-estimating the expected future return time given the LOA.

5.5.1 Data Collection

For our experiments we use a small public and a larger proprietary dataset. We briefly describe these two datasets:

- The Last.fm dataset. Last.fm, is an online music website catering to millions of active users. Recently, the service made available the complete music listening histories of around a 1000 of its users as recorded until May, 2009 [**?**, 145]. For every song the user listened to, the dataset includes the song title, the artist name and the timestamp at which the song was heard. We use two separate time windows for creating the training and the testing datasets. All user visits observed during Oct, 2008 - Dec, 2008 were used to test the model through cross-validation. We also tested our model on future user visits observed from Jan, 2009 - Mar, 2009.

- Large-scale dataset. Our proposed approach was applied as a part of the growth and retention efforts for a large ad-supported music service. A dataset of around 73,465 users, collected over 3 months from May, 2012 - July, 2012, was used for training and testing our model via cross-validation.

5.5.2 Covariates

We constructed the following covariates for the return time prediction problem.

- **Covariates related to the typical visitation patterns of a user**. Such covariates seek to predict the future return behavior of the users based on how their visitation behavior has

been historically. For example, users who have been highly frequent in the past (loyal to the service) are likely to remain frequent in the future and similarly users who have been infrequent in the past (casual visitors) are likely to visit infrequently in the future.

  - Active Weeks: This covariate is defined as the ratio of the number of weeks since registration when the user visited the service at least once to the total number of weeks elapsed since registration.
  - Density of Visitation: This covariate captures the volume of user activity on the service for the weeks the user is active on the service. It is defined as the average number of days the user visited the service during the weeks the user visited the service at least once.
  - Visit Number: This covariate is used to measure how tenured the user is with the service.
  - Previous Gap: This covariate represents the most recent return time observation (which is the gap between the user's last and prior to the last visit) for the user. For first time users this covariate is set to -1.
  - Time weighted average return time (TWRT): This covariate measures the average return time for a user. The return times are further weighted by the inverse of the length of time elapsed since they were observed under the premise that the more recent return times are more informative about the user's current visitation behavior.

- **Covariates related to user satisfaction and engagement with the service**. Satisfaction and engagement related covariates are more difficult to construct as they attempt to capture latent user emotions about the service. Such can be extracted from any explicit (likes, dislikes, complaints etc.) or implicit (time spend, unique activities etc.) feedback indicators using user past interactions. In this work, we constructed these covariates based on user activities recorded on the last visit to the service (last *In* state)

  - Duration: This covariate captures the time spend at the service measured by the number of songs heard by the user.
  - $\%$ Distinct Songs: This covariate measures the fraction of the number of distinct songs listened by the users over the total number of songs listened by them.

- $\%$ Distinct Artists: This covariate measures the fraction of the number of distinct artists listened by the users over the total number of songs listened by them.
- $\%$ Skips: This covariate measures the fraction of the number of songs skipped by the users of the total number of songs listened by them. The skip information is not directly available for the Last.fm dataset. Instead, we indirectly identified skips by comparing the gap between two consecutive songs (s_1 and s_2) in the data with the length of the song s_1. If the time gap was found to be less than the length of song s_1 by more than 30 seconds, then song s_1 was identified to have been skipped. The API, track.getInfo made freely available by Last.fm was used to retrieve the duration for the songs in the dataset.
- Explicit feedback indicators: These covariates include information obtained directly from the users such as ratings, comments, complaints etc. Explicit feedback measures tend to be highly accurate and are an important source of information about user's satisfaction with the service. However, they are hard to acquire as providing explicit feedback requires user effort. We did not have any explicit feedback indicators for the Last.fm dataset. We had such ratings for our proprietary dataset which were included in the model.

- **Covariates used for abstracting the effects of external factors**. External factors include public holidays and weekends, marketing campaigns and promotions or personal factors which impact the rate of user return. The ability to model external factors is very useful as by modeling these covariates, we can both quantify the impact of these factors and control for these effects to improve our analysis on the other covariates. For simplicity, we have not considered any external covariates in our experiments. However, in our *Conclusion* section, we discuss how the Cox's model can be used to model the day of the month covariate which allows us to incorporate weekly effects and holiday effects in our predictions.

5.5.3 Evaluation Metrics and Baselines

Different baselines are used for evaluating the performance of the Cox's model at the regression and the classification tasks. All baselines are implemented using the same covariates as used in the Cox's model. For the regression problem we compared the Cox's model against simple

average (trivial baseline), linear regression, decision tree regression (RepTree), Support Vector Machine (with linear kernel) and neural networks (multilayer perceptron). Support Vector Machine Regression took too long to run (more than a day) on our large scale dataset and was omitted in those results. The performance of the models were evaluated using Weighted Root Mean Square Error(WRMSE). The WRMSE is computed by weighting the error between the true return time and predicted return time with the weight of the test instance as follows:

$$WRMSE = \sqrt{\frac{\sum_{i=0}^{N} w(i) * (T^{p}_{rd}(i) - T_{rd}(i))^2}{\sum_{i=0}^{N} w(i)}} \tag{5.11}$$

where, N is the number of test instances, $T^{p}_{rd}(i)$ denotes the truncated return time predicted for the i-th observation and $T_{rd}(i)$ denotes the true truncated return time the i-th observation. We can replace $T^{p}_{rd}(i)$ with $T^{p}_{frd}(i)$ and $T_{rd}(i)$ with $T_{frd}(i)$ for computing the WRMSE for the expected future return time predictions.

Our classification baselines included logistic regression, random forest, support vector machine (with a linear kernel) and neural networks (multilayer perceptron). We used weighted F-measure for the minority class for measuring performance at the classification task. The weighted f-measure is defined as the harmonic mean of the weighted precision and weighted recall scores which are defined as follows. The set A denotes the instances actually belonging to the minority class and set P denotes the instances which were predicted to belong to the minority class.

$$\text{Weighted Precision} = \frac{\text{sum of weights of instances in } A \cap P}{\text{sum of weights of instances in } P} \tag{5.12}$$

$$\text{Weighted Recall} = \frac{\text{sum of weights of instances in } A \cap P}{\text{sum of weights of instances in } A} \tag{5.13}$$

The experiments for the baselines were conducted using Weka, the open source data mining software available under the GNU General Public License. The baselines were suitably tuned using a hold out set. Also, Weka provides support for handling weighted data instances allowing us to easily incorporate the weight vector while training the models. Since a direct application of cross-validation would not maintain temporal ordering between observations of the same user, for our evaluation we took special care to ensure that all recurrent data corresponding to a user belonged to the same fold. This was done by first randomly dividing users into different folds and then placing all observation associated with the user in that fold. As a result, the training and testing folds had observations from different users.

5.6 Results

In this section we analyze the results of the experimental evaluation of the Cox's model.

5.6.1 Model Parameters

We only discuss the parameters of model trained on the Last.fm dataset.

The importance of the covariates for the prediction problem can be assessed using different importance indicators (Table 5.1). The regression coefficients and the significance score for the covariates can be obtained directly from the output of the R function for fitting the Cox's model. The regression coefficient tells us how much a unit change in the value of the covariate impacts the user's rate of return. For example, with every song the users listened during their last visit, their hazard rate was found to multiply by $\exp(1.315e-03) = 1.0013$, decreasing their return times estimates. The value of the coefficient was statistically significant at a significance level of 0.01. We found most of the covariates associated with the typical patterns of visitation (Active Weeks, Density, Previous gap) to be highly significant for predicting the return time variable. Also, some of the engagement/satisfaction related covariates, namely duration and $\%$ artists had significant effects on the hazard rate. We also computed the mean of the product of the covariate and its coefficient (MEAN$(X * \beta)$) measured for all instances in the training set. This provided an average score for how much the covariate impacted the magnitude of the baseline hazard function. The density covariate impacted the rate of return the most on an average for our dataset.

Covariates	**Coefficient**	**Significance**	**MEAN**$(X * \beta)$
Active Weeks	9.313e-02	2.140e-02*	4.370e-01
Density	2.366e-01	1.050e-13***	1.244e00
Visit Number	4.941e-05	7.318e-01	2.336e-02
Previous Gap	-5.175e-03	1.470e-03**	-1.222e-02
TWRT	-1.484e-02	2.817e-01	-2.492e-02
Duration	1.315e-03	2.538e-02 *	6.171e-02
$\%$ **Distinct Songs**	6.849e-02	7.653e-01	6.040e-02
$\%$ **Distinct Artists**	-2.251e-01	8.553e-02 .	-1.064e-01
$\%$ **Skips**	3.740e-01	2.322e-01	4.873e-02

Table 5.1: Importance indicators for model covariates for the Last.fm dataset. Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'

Figure 5.2 displays the baseline hazard function and the survival function computed for

the training dataset from Last.fm. The baseline hazard function has a sharply declining shape typical of processes exhibiting inertia. Hence, the longer users stay away from the service the lesser likely they are of returning within sometime in the future. As a result, it is all the more important for a web service to ensure that its user are motivated to return to the service soon. The survival function has a value of 0.0009 at 60 days. This suggests that 0.09$\%$ of users for this dataset did not return within 60 days.

5.6.2 Return Time Prediction

Table 5.2 and Table 5.3 display the weighted root mean square error scores obtained using the hazard based approach and the standard regression based approaches for the large-scale proprietary and the Last.fm datasets, respectively. We find that the hazard based approach is superior in predictive performance than the other baselines and the improvements are highly significant (p-value$< 10^{-10}$, using two-tailed paired t-test). The hazard based approach also fares well in terms of run time. On a Intel(R) Xeon(R) CPU X5650 @ 2.67GHz 24GHz, the hazard based approach takes $\sim$ 8 minutes as compared to neural networks which take $\sim$ 16 minutes to finish one run of cross-validation. Decision tree regression ($\sim$ 4 minutes), linear regression ($\sim$ 26 seconds) and average ($\sim$ 20 seconds) are faster however, the lower run times come at the cost of performance.

As discussed earlier, the hazard based approach allows us to compute the expected future return time for a user given their length of absence (LOA) by incorporating the dynamics in the hazard function. We evaluate the performance of the hazard-based approach in updating its prediction given the LOA values. Since the standard regression approaches do not provide similar functionality, we re-learn those models by incorporating the LOA values as a separate feature. The values for this feature is generated by replicating each return time observation T, T times for all values of LOA ranging from $(0) - (T-1)$. The future return time is appropriately reassigned to range from $(T) - (1)$. Doing so can significantly increase the size of the dataset. The data instances are re-weighted to ensure that each user still holds a unit weight in the test and the training sets. Due to space limitations we only show the comparisons between two of our baselines: decision tree regression (best performing baseline) and linear regression (because of its ease of use), with the hazard based approach for the large-scale proprietary dataset. We find that the hazard based approach is superior than both these models in estimating the expected future return time (Fig.5.3).

	Training Data (10-fold Cross Validation)
Average	19.41
Linear Regression	18.54
Decision Tree Regression	18.14
Neural Networks	18.26
Hazard Based Approach	16.58

Table 5.2: WRMSE for user return time prediction using the proprietary dataset.

	Training Data (10-fold Cross Validation)	Test Data
Average	10.55	10.40
Linear Regression	9.61	9.37
Decision Tree Regression	9.45	9.15
Support Vector Machine	10.76	10.33
Neural Networks	9.58	9.36
Hazard Based Approach	8.76	8.45

Table 5.3: WRMSE for user return time prediction using the Last.fm dataset.

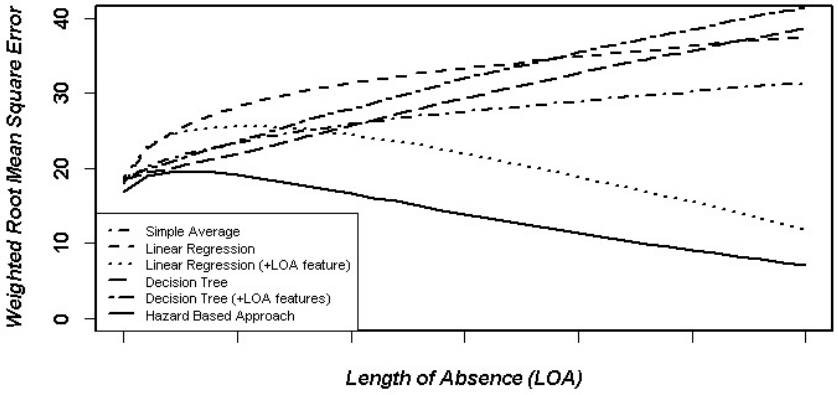


Figure 5.3: WRMSE for different values of LOA for the proprietary dataset. The units on the X-axis have been omitted.

5.6.3 Classification into User Buckets

The users are classified into different categories based on their predicted return times. For the Last.fm dataset we bucketed users based on their predicted return times being larger or within 7 days, while for the larger proprietary dataset we classified them based on their predicted return times being larger or within 30 days. The shorter time period was used for the Last.fm dataset due to scarcity of users in the test set that returned after 7 days. Table 5.4 and Table 5.5 provide the performance scores for the hazard based approach and the other baselines for classifying instances into the minority class for the proprietary and the Last.fm datasets. The proprietary dataset had 15.4% samples and the last.fm dataset had 12.2% samples belonging to the minority class. A naive classifier would have a precision of 0.154 and 0.122, respectively for these datasets. All the models perform better than a naive classifier. Although, the hazard based model is not learnt as a classification model, it still performs superior to the state-of-the-art baselines for our proprietary dataset (p-value$< 10^{-8}$, using two-tailed paired t-test) and is comparable in performance to the best performing baselines for our Last.fm dataset. The hazard based approach has the highest recall of all the models which seems to be at the cost of precision. However, for a rare class problem like ours, recall at identifying most of the at-risk users is far more important to a business and the overheads from the lower precision are low. In terms of run time, on a Intel(R) Xeon(R) CPU X5650 @ 2.67GHz 24GHz, the hazard based approach takes $\sim$ 8 minutes to finish one run of cross-validation, which is lower as compared to the other baselines: neural network classifier ($\sim$ 15 minutes), logistic regression ($\sim$ 11 minutes) and support vector machine ($\sim$ 24 minutes). Random forest has the lowest run time of all the models ($\sim$ 6 minutes).

	Training Data (10-fold Cross Validation)		
	Precision	Recall	F-Measure
Random Forest	0.47	0.10	0.18
Logistic Regression	0.52	0.08	0.15
Support Vector Machine	0	0	0
Neural Networks	0.48	0.17	0.25
Hazard Based Approach	0.41	0.23	0.29

Table 5.4: Weighted precision, recall and f-measure scores for the minority class (expected return time > 30) for the large-scale proprietary dataset.

We also evaluate the performance of the hazard based approach in classifying users into

buckets given the LOA values. Again, the classification baselines do not offer similar capabilities for updating their prediction scores given LOA values. Hence, we incorporate LOA values as an additional feature for classification and replicate instances to populate the values for the feature as done for the standard regression methods earlier. We provide comparison results against the best performing baseline classification approaches - logistic regression and neural networks. We find that the hazard-based approach can incorporate the LOA information and update its prediction much effectively as compared to both logistic regression and neural networks (Fig. 5.4).

	Training Data			Test Data		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Random Forest	0.64	0.24	0.35	0.72	0.29	0.41
Logistic Regression	0.68	0.44	0.53	0.66	0.40	0.50
Support Vector Machine	0.61	0.11	0.18	0.82	0.15	0.25
Neural Networks	0.77	0.39	0.52	0.71	0.36	0.48
Hazard Based Approach	0.39	0.79	0.52	0.37	0.81	0.51

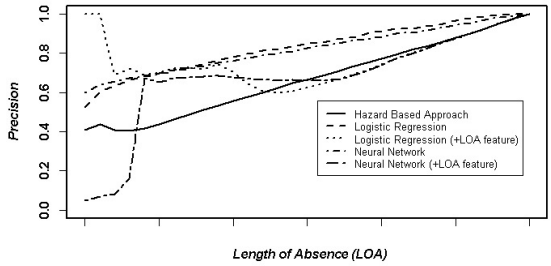
Table 5.5: Weighted precision, recall and f-measure scores for the minority class (expected return time > 7) for the Last.fm dataset.

5.6.4 Sensitivity to the Threshold

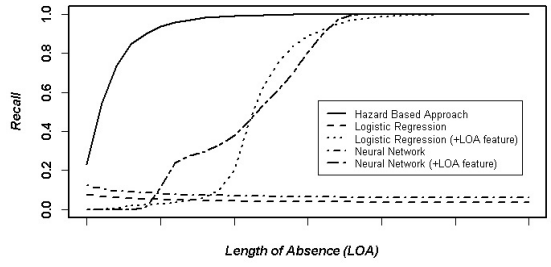
The threshold (t_d) was set to 60 days in our experiments, which was a reasonably large value and beyond which users are already the focus of retention efforts. For completeness, we also evaluate our model for some smaller values of the threshold. Table 5.6 lists the performance of the models at predicting the return time for threshold values of 15, 30 and 45 days. We find that the Cox's model still performs better than the other baselines at the prediction task in these experiments (p-value$< 10^{-8}$, using two-tailed paired t-test).

5.6.5 Alternative Approaches for Handling Recurrent Observations

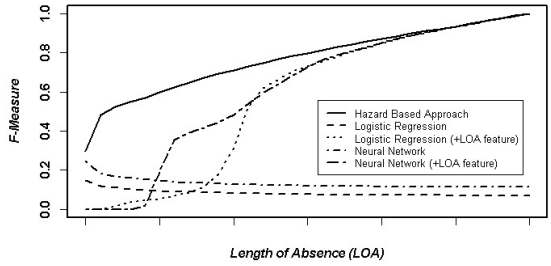
We use a re-weighting scheme for handling recurrent observations which allows us to retain all data instances for a user in the dataset without biasing the models towards active users. However, we now evaluate the sensitivity of our results to our weighting schemes by considering



(a)



(b)



(c)

Figure 5.4: Figures (a), (b) and (c) are the plots of the weighted precision, recall and f-measure scores respectively, for different values of LOA for the large-scale proprietary dataset. The units on the X-axis have been omitted.

alternative approaches for handling recurrent observations. Four such approaches are defined: unweighted, using only the first observation per user, using only the last observation per user and considering only users active on a particular date chosen randomly. The last three approaches eliminate recurrent observations by data selection. We use Root Mean Square Error (RMSE) for evaluation. Due to space constraints we only report RMSE results on our proprietary dataset.

	Training Data (10-fold Cross Validation)		
	$t_d = 15$	$t_d = 30$	$t_d = 45$
Average	6.45	11.77	16.07
Linear Regression	6.11	11.16	15.29
Decision Tree Regression	5.14	10.11	14.61
Neural Networks	5.29	10.36	15.28
Hazard Based Approach	5.04	9.54	13.41

Table 5.6: WRMSE for user return time prediction with different values of t_d using the proprietary dataset.

	Training Data (10-fold Cross Validation)			
	Un-weighted	First Event	Last Event	Single day
Average	7.62	17.35	26.17	7.44
Linear Regression	7.33	16.80	24.96	7.08
Decision Tree Regression	7.37	16.52	24.56	6.99
Neural Networks	7.31	17.42	24.52	7.01
Hazard Based Approach	7.31	15.955	17.76	6.87

Table 5.7: RMSE for user return time prediction with alternative schemes for handling recurrent observations using the proprietary dataset.

We find that the Cox's model outperforms the other baselines when we use only the first or the last observation per user for training and testing the models (p-value$< 10^{-10}$, using two-tailed paired t-test). All the models have comparable performance when we use the un-weighted scheme or work with user observations recorded on a particular day. Both these scheme also record the lowest errors compared to the other schemes for all the models. We suspect this to happen because both these schemes are dominated by the active users and predicting the return time for such users is much easier. In order to investigate this further, we perform a pilot study in which we hold out a small sample of 1000 return time observations selectively chosen to be longer than 30 days from the proprietary dataset. The performance of different versions of the Cox model trained using the various schemes for handling recurrent observations discussed earlier is then tested at predicting these longer return time observations. The RMSE results are reported in table 5.8

	Test data of long return times				
	Weighted	Un-weighted	First Event	Last Event	Single day
RMSE	32.25	40.70	32.34	32.14	41.81

Table 5.8: RMSE for long return time prediction for different versions of the Cox model on the proprietary dataset.

These results further show that both the un-weighted scheme and choosing observations from a single day, perform poorly at predicting longer return times. Since the focus of our methods is to find users which are not likely to return soon, these approaches may not be suitable for our application. Furthermore, the weighted scheme offers a good trade-off between using just the first events or just the last events per user in our model making it more suitable for our problem.

5.7 Conclusion

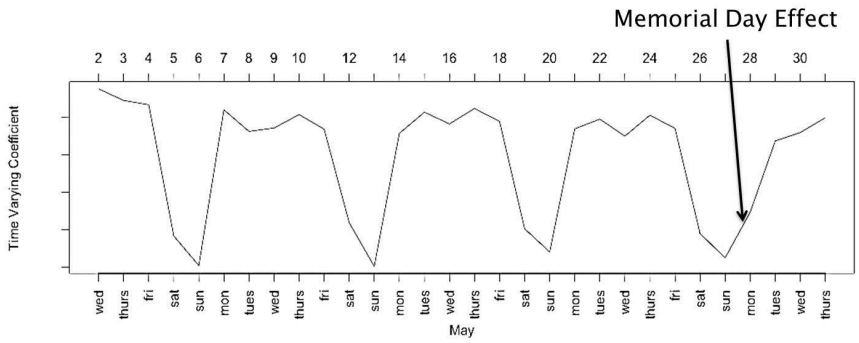


Figure 5.5: The regression coefficient for time-varying covariates corresponding to the different days of the month. The absolute values are omitted here.

In this work, we have focused on the return time performance metric for free web services. We suggest that retention solutions driven by the projected return time of users can directly address the heart of the problem for web services, which is to encourage their users to frequently engage with their service. To facilitate such efforts, we formulate the problem of user return time prediction and define several covariates relevant to the problem. The Cox's proportional hazard model is proposed as the model of choice for this prediction problem due to several reasons including the ability to handle dynamics in user return rate with time and to incorporate the LOA information. A plot of the prediction performance scores against the LOA values allows a service to identify the right amount of gap since the user's last visit needed to start retention efforts. The performance of the hazard based model is found to surpass all the state-of-the-art baselines considered by us. Finally, we find that the ability of the Cox model to quantify the impact of several important covariates, including those related to user usage patterns, on user

return rates to provide important insights that can guide future decision making for the service.

The Cox's model can further accommodate several complexities of the real-world quite well. For example, our analysis till now has been limited to static covariates. However, time-varying covariates including those pertaining to external factors such as holiday and weekend effects can be important for return time prediction and can be easily incorporated in the Cox Model [146]. In our final model for the large scale music service, we incorporated the effect of the day of the month covariate (Fig. 5.5) on the user return rates. Another direction for future research is to account for heterogeneities among users. Several solutions exist for either controlling for such differences between users [147] or for extracting different users segments through clustering [148] can also be applied to the return time prediction problem.

Chapter 6

Conclusion and Discussion

Dynamic preferences of users pose a significant challenge to existing recommendation algorithms. With their growing popularity, recommender systems are used on a daily basis today necessitating the development of systems to address the constantly changing needs of their users with relevant and high quality recommendations. The field of behavioral psychology, on the other hand, has accumulated a huge body of literature on the psychology of preferences dynamics. However, many of the available insights and techniques have remained limited to controlled and restricted settings (laboratory experiments) and hence are not readily applicable to the chaotic environment of the web. This thesis is the first of its kind to combine insights from behavioral psychology with empirical models of user behavior using user online activity logs. While doing so, we bring to the table unique tools, adapted from field of survival analysis traditionally applied to biological and mechanical systems, that allow us to analyze and quantify changes in user preferences using only their past activities. We further develop dynamic user models to model satiation (boredom) and novelty seeking in users with good predictive performance. Finally, we use our dynamic techniques to provide novel solutions for user retention on the web with good results.

We view our study as a first step towards data driven development of psychological models of preference dynamics. Like any first work in an area, this thesis leave more open question than it answers. Here, we summarize some key directions to assist future work in this area.

1. Our work assumes that a user's consumption of an item is independent of the other items. However, items are seldom consumed in isolation. For example, users generally have

multiple playlists of songs each of which fulfills the need for a different genre and style of music, such as pop, rock or country. Similarly, users watch videos and movies from different categories like comedy, drama or suspense. Such categories may again comprise multiple sub-categories forming a natural hierarchy of item sets, which we call *consumption bundles*, with each increasing level of the hierarchy representing smaller and more specialized sets of items. We hypothesize that a user has multiple preference states for an item bundle and changes its preference state for those items with time. For example, a user may be increasingly addicted to a certain set of songs, genre of movies or topics, but having completely saturated those categories, may later seek something new and different. As a result, future preferences models need to be hierarchical in nature to incorporate such dependencies between the items they recommend.

2. Our work has been limited to analyzing and modeling music preferences of users. Extending our work to other types of items introduces new challenges which we discuss briefly. Modeling the hierarchical organization of items becomes all the more important for extending our work to domains such as movies and books. Although users seldom repeat the same movie, users tend to watch movies from the same genre, director, time period etc. Such attributes of items constitute a similarity space which further facilitates the extraction of consumption bundles showing similar dynamics in preferences. Other domains such as clothing introduces cost of an item as another important factor in the process of decision making not addressed in our work.

3. Classical model of recommendation largely utilize similarity to preferred items as a guiding principle for their methods. In this work, we have proposed for the first time a model of item satiation which allows us to identify items a user is bored of at the moment in addition to the items he/she prefers in general. This provides us an unprecedented opportunity to explore user behavior when driven by the need to alleviate boredom. The computational tools developed in this work can be used to address questions like - Which artist would a user transition to when he/she is bored with *Lady Gaga*?

4. Recommendation and retention have classically been two disjoint areas of research. However, recommenders play an important role today in facilitating, directing and controlling user interaction with a content provider, directly impacting their engagement and subsequent loyalty to the system. Furthermore, the ability to now extract user latent preference

states of user, provided by this work, allows future analyzes to study how recommender performance in various preference states of the user impacts user churn. We envisage such analyzes of user retention as an extremely promising area of research with important consequences to future recommender design.

References

[1] Ward Edwards. The theory of decision making. *Psychological bulletin*, 51(4):380, 1954.

[2] S. Senecal and J. Nantel. The influence of online product recommendations on consumers online choices. *Journal of Retailing*, 80(2):159–169, 2004.

[3] G. Adomavicius and A. 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.

[4] William N Dember and Harry Fowler. Spontaneous alternation behavior. *Psychological Bulletin*, 55(6):412, 1958.

[5] Robert F Bornstein and Paul R D'Agostino. Stimulus recognition and the mere exposure effect. *Journal of personality and social psychology*, 63(4):545, 1992.

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

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

[8] DeE BERLYNE. Novelty and curiosity as determinants of exploratory behaviour1. *British Journal of Psychology. General Section*, 41(1-2):68–80, 1950.

[9] Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. Temporal diversity in recommender systems. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217. ACM, 2010.

[10] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. volume 107, pages 4511–4515. National Acad Sciences, 2010.

[11] Mi Zhang and Neil Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130. ACM, 2008.

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

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

[14] Robert B Zajonc. Attitudinal effects of mere exposure. *Journal of personality and social psychology*, 9(2p2):1, 1968.

[15] Colin Martindale and Kathleen Moore. Priming, prototypicality, and preference. *Journal of Experimental Psychology: Human Perception and Performance*, 14(4):661, 1988.

[16] Daniel E Berlyne. Novelty, complexity, and hedonic value. *Perception & Psychophysics*, 8(5):279–286, 1970.

[17] Robert F Bornstein, Amy R Kale, and Karen R Cornell. Boredom as a limiting condition on the mere exposure effect. *Journal of personality and Social Psychology*, 58(5):791, 1990.

[18] Donald Olding Hebb. Drives and the cns (conceptual nervous system). *Psychological review*, 62(4):243, 1955.

[19] Clarence Leuba. Toward some integration of learning theories: The concept of optimal stimulation. *Psychological Reports*, 1(g):27–33, 1955.

[20] Philip Brickman and Barbara D'Amato. Exposure effects in a free-choice situation. *Journal of Personality and Social Psychology*, 32(3):415, 1975.

[21] Peter Pirolli and Stuart Card. Information foraging. *Psychological review*, 106(4):643, 1999.

[22] L. McAlister and E. Pessemier. Variety seeking behavior: An interdisciplinary review. *Journal of Consumer research*, pages 311–322, 1982.

[23] Kelvin Lancaster. Consumer demand: A new approach. 1971.

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

[25] 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.

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

[27] Tomohito Kamai and Yuichiro Kanazawa. The latent class model of brand choice behaviors incorporating variety-seeking and state dependence. 2012.

[28] Abraham Garcia-Torres. Consumer behaviour: evolution of preferences and the search for novelty. *Cornell family papers*, 2009.

[29] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 107–144. Springer, 2011.

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

[31] Nachiketa Sahoo, Param Vir Singh, and Tridas Mukhopadhyay. A dynamic model of employee blog reading behavior.

[32] Daniel E Berlyne. Conflict, arousal, and curiosity. 1960.

[33] Puthankurissi S Raju. Optimum stimulation level: its relationship to personality, demographics, and exploratory behavior. *Journal of consumer research*, pages 272–282, 1980.

[34] Jeanne Nakamura and Mihaly Csikszentmihalyi. The concept of flow. *Handbook of positive psychology*, pages 89–105, 2002.

[35] Abel P Jeuland. Brand choice inertia as one aspect of the notion of brand loyalty. *Management Science*, 25(7):671–682, 1979.

[36] M. Glanzer. The role of stimulus satiation in spontaneous alternation. *Journal of experimental psychology*, 45(6):387, 1953.

[37] Clyde H Coombs and George S Avrunin. Single-peaked functions and the theory of preference. *Psychological Review*, 84(2):216, 1977.

[38] Pradeep K Chintagunta. Inertia and variety seeking in a model of brand-purchase timing. *Marketing Science*, 17(3):253–270, 1998.

[39] 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.

[40] S.M. McNee, J. Riedl, and J.A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 1097–1101. ACM, 2006.

[41] C.N. Ziegler, S.M. McNee, J.A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM, 2005.

[42] O. Celma. *Music Recommendation and Discovery in the Long Tail*. Springer, 2010.

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

[44] 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.

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

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

[47] 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.

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

[49] 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.

[50] 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.

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

[52] 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.

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

[54] Huanhuan Cao, Enhong Chen, Jie Yang, and Hui Xiong. Enhancing recommender systems under volatile userinterest drifts. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1257–1266. ACM, 2009.

[55] 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.

[56] 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.

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

[58] 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.

[59] 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.

[60] 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.

[61] 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.

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

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

[64] 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.

[65] Linas Baltrunas and Xavier Amatriain. Towards time-dependant recommendation based on implicit feedback. In *Workshop on context-aware recommender systems (CARS09)*, 2009.

[66] 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.

[67] 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.

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

[69] 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.

[70] 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.

[71] 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.

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

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

[74] 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.

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

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

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

[78] 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.

[79] 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.

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

[81] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

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

[83] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[84] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

[85] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. Temporal recommendation on graphs via long-and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 723–732. ACM, 2010.

[86] Rong Hu and Pearl Pu. A comparative user study on rating vs. personality quiz based preference elicitation methods. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 367–372. ACM, 2009.

[87] Rong Hu and Pearl Pu. Enhancing collaborative filtering systems with personality information. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 197–204. ACM, 2011.

[88] Sean M. McNee, John Riedl, and Joseph A. Konstan. Making recommendations better: An analytic model for human-recommender interaction. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1103–1108, New York, NY, USA, 2006. ACM.

[89] Eli Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.

[90] Barbara Kahn. Variety: From the consumers perspective. In *Product Variety Management*, pages 19–37. Springer, 1998.

[91] Rebecca K Ratner, Barbara E Kahn, and Daniel Kahneman. Choosing less-preferred experiences for the sake of variety. *Journal of Consumer Research*, 26(1):1–15, 1999.

[92] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM, 2005.

[93] Yoshinori Hijikata, Takuya Shimizu, and Shogo Nishida. Discovery-oriented collaborative filtering for improving user satisfaction. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 67–76. ACM, 2009.

[94] Alexander Chernev. When more is less and less is more: The role of ideal point availability and assortment in consumer choice. *Journal of consumer Research*, 30(2):170–183, 2003.

[95] Hans CM Van Trijp, Wayne D Hoyer, and J Jeffrey Inman. Why switch? product category: level explanations for true variety-seeking behavior. *Journal of Marketing Research*, pages 281–292, 1996.

[96] Itamar Simonson and Russell S Winer. The influence of purchase quantity and display format on consumer preference for variety. *Journal of Consumer Research*, pages 133–138, 1992.

[97] Robert B Zajonc. Mere exposure: A gateway to the subliminal. *Current directions in psychological science*, 10(6):224–228, 2001.

[98] J Jeffrey Inman. The role of sensory-specific satiety in attribute-level variety seeking. *Journal of Consumer Research*, 28(1):105–120, 2001.

[99] Kathleen T Lacher. Hedonic consumption: Music as a product. *Advances in Consumer Research*, 16(1):367–373, 1989.

[100] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, and Xing Xie. Mining novelty-seeking trait across heterogeneous domains. In *Proceedings of the 23rd international conference on World wide web*, pages 373–384. International World Wide Web Conferences Steering Committee, 2014.

[101] Fernando Mouro, Claudiane Fonseca, Camila Arajo, and Wagner Meira Jr. The oblivion problem: Exploiting forgotten items to improve recommendation diversity. In *Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011)*, page 27, 2011.

[102] 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.

[103] Elizabeth Hellmuth Margulis. *On Repeat: How Music Plays the Mind*. Oxford University Press, 2014.

[104] Tamas Jambor and Jun Wang. Optimizing multiple objectives in collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 55–62. ACM, 2010.

[105] Gediminas Adomavicius and YoungOk Kwon. Toward more diverse recommendations: Item re-ranking methods for recommender systems. In *Workshop on Information Technologies and Systems*, 2009.

[106] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

[107] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. volume 22, pages 5–53, New York, NY, USA, January 2004. ACM.

[108] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*, pages 677–686. International World Wide Web Conferences Steering Committee, 2014.

[109] Li-Tung Weng, Yue Xu, Yuefeng Li, and Richi Nayak. Improving recommendation novelty based on topic taxonomy. In *Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM International Conferences on*, pages 115–118. IEEE, 2007.

[110] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. Auralist: introducing serendipity into music recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 13–22. ACM, 2012.

[111] Peter J. Rentfrow. The role of music in everyday life: Current directions in the social psychology of music. *Social and Personality Psychology Compass*, 6(5):402–416, 2012.

[112] Vikas Kumar, Daniel Kluver, Loren Terveen, and John Riedl. More efficient tagging systems with tag seeding. In *Proceedings of 2014 ASE SocialCom Conference, Stanford University, May 27-31, 2014*. Academy of Science and Engineering, Academy of Science and Engineering (ASE, USA), 2014.

[113] Neil Hurley and Mi Zhang. Novelty and diversity in top-n recommendationanalysis and evaluation. *ACM Transactions on Internet Technology (TOIT)*, 10(4):14, 2011.

[114] Saúl Vargas and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 109–116. ACM, 2011.

[115] Komal Kapoor, Karthik Subbian, Jaideep Srivastava, and Paul Schrater. Timing matters! - modeling dynamics of boredom in activity streams.

[116] Xu-Ying Liu and Zhi-Hua Zhou. The influence of class imbalance on cost-sensitive learning: An empirical study. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 970–974. IEEE, 2006.

[117] Rishab Aiyer Ghosh. Cooking pot markets: an economic model for the trade in free goods and services on the internet. *First Monday*, 3(2), 1998.

[118] Jeffrey F Rayport. The truth about internet business models. *Strategy and Business*, pages 5–7, 1999.

[119] Wouter Buckinx and Dirk Van den Poel. Customer base analysis: partial defection of behaviourally loyal clients in a non-contractual fmcg retail setting. *European Journal of Operational Research*, 164(1):252–268, 2005.

[120] Chih-Ping Wei, I Chiu, et al. Turning telecommunications call details to churn prediction: a data mining approach. *Expert systems with applications*, 23(2):103–112, 2002.

[121] Joe Peppard. Customer relationship management (crm) in financial services. *European Management Journal*, 18(3):312–327, 2000.

[122] Bing Quan Huang, M Tahar Kechadi, and Brian Buckley. Customer churn prediction for broadband internet services. In *Data Warehousing and Knowledge Discovery*, pages 229–243. Springer, 2009.

[123] Marcel Karnstedt, Tara Hennessy, Jeffrey Chan, and Conor Hayes. Churn in social networks: A discussion boards case study. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 233–240. IEEE, 2010.

[124] Scott A Neslin, Sunil Gupta, Wagner Kamakura, Junxiang Lu, and Charlotte H Mason. Defection detection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of Marketing Research*, pages 204–211, 2006.

[125] Werner J Reinartz and Vijay Kumar. On the profitability of long-life customers in a noncontractual setting: An empirical investigation and implications for marketing. *The Journal of Marketing*, pages 17–35, 2000.

[126] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 187–220, 1972.

[127] Kristof Coussement and Dirk Van den Poel. Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert systems with applications*, 34(1):313–327, 2008.

[128] Chih-Fong Tsai and Yu-Hsin Lu. Customer churn prediction by hybrid neural networks. *Expert Systems with Applications*, 36(10):12547–12553, 2009.

[129] Yaya Xie, Xiu Li, EWT Ngai, and Weiyun Ying. Customer churn prediction using improved balanced random forests. *Expert Systems with Applications*, 36(3):5445–5449, 2009.

[130] Jonathan Burez and Dirk Van den Poel. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3):4626–4636, 2009.

[131] Zainab Jamal and Randolph E Bucklin. Improving the diagnosis and prediction of customer churn: A heterogeneous hazard modeling approach. *Journal of Interactive Marketing*, 20(3-4):16–29, 2006.

[132] Jiang Yang, Xiao Wei, Mark S Ackerman, and Lada A Adamic. Activity lifespan: An analysis of user survival patterns in online knowledge sharing communities. In *ICWSM*, 2010.

[133] Han Liu, Atif Nazir, Jinoo Joung, and Chen-Nee Chuah. Modeling/predicting the evolution trend of osn-based applications. In *Proceedings of the 22nd international conference on World Wide Web*, pages 771–780. International World Wide Web Conferences Steering Committee, 2013.

[134] Yin Zhu, Erheng Zhong, Sinno Jialin Pan, Xiao Wang, Minzhe Zhou, and Qiang Yang. Predicting user activity level in social networks. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 159–168. ACM, 2013.

[135] Bruno Ribeiro. Modeling and predicting the growth and death of membership-based websites. *Proceedings of the 23rd international conference on World Wide Web*, 2014.

[136] Simon Attfield, Gabriella Kazai, Mounia Lalmas, and Benjamin Piwowarski. Towards a science of user engagement (position paper). In *WSDM Workshop on User Modelling for Web Applications*, 2011.

[137] Janette Lehmann, Mounia Lalmas, Elad Yom-Tov, and Georges Dupret. Models of user engagement. In *User Modeling, Adaptation, and Personalization*, pages 164–175. Springer, 2012.

[138] Georges Dupret and Mounia Lalmas. Absence time and user engagement: evaluating ranking functions. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 173–182. ACM, 2013.

[139] Elad Yom-Tov, Mounia Lalmas, Ricardo Baeza-Yates, Georges Dupret, Janette Lehmann, and Pinar Donmez. Measuring inter-site engagement. In *Big Data, 2013 IEEE International Conference on*, pages 228–236. IEEE, 2013.

[140] Peter S Fader, Bruce GS Hardie, and Ka Lok Lee. Rfm and clv: Using iso-value curves for customer base analysis. *Journal of Marketing Research*, pages 415–430, 2005.

[141] Alex Berson, Stephen Smith, and Kurt Thearling. *Building data mining applications for CRM*. McGraw-Hill New York, 2000.

[142] Gideon Dror, Dan Pelleg, Oleg Rokhlenko, and Idan Szpektor. Churn prediction in new users of yahoo! answers. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 829–834. ACM, 2012.

[143] David C Schmittlein and Vijay Mahajan. Maximum likelihood estimation for an innovation diffusion model of new product acceptance. *Marketing science*, 1(1):57–78, 1982.

[144] Michael D Kaplowitz, Timothy D Hadlock, and Ralph Levine. A comparison of web and mail survey response rates. *Public opinion quarterly*, 68(1):94–101, 2004.

[145] O. Celma. *Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space.* Springer, 2010.

[146] Terry Therneau and Cindy Crowson. Using time dependent covariates and time dependent coefficients in the cox model. *The Survival Package (R help guide)*, 2013.

[147] CA McGilchrist and CW Aisbett. Regression with frailty in survival analysis. *Biometrics*, pages 461–466, 1991.

[148] Patrick Mair and Marcus Hudec. Analysis of dwell times in web usage mining. In *Data Analysis, Machine Learning and Applications*, pages 593–600. Springer, 2008.