

**A HYBRID RECOMMENDATION METHOD
THAT COMBINES FORGOTTEN ITEMS AND
NON-CONTENT ATTRIBUTES**

FERNANDO HENRIQUE DE JESUS MOURÃO

**A HYBRID RECOMMENDATION METHOD
THAT COMBINES FORGOTTEN ITEMS AND
NON-CONTENT ATTRIBUTES**

Thesis presented to the Graduate Program
in Computer Science of the Universidade
Federal de Minas Gerais - Departamento de
Ciência da Computação. in partial fulfill-
ment of the requirements for the degree of
Doctor in Computer Science.

ADVISOR: WAGNER MEIRA JÚNIOR

Belo Horizonte, MG

December 2014

© 2014, Fernando Henrique de Jesus Mourão.
Todos os direitos reservados.

M929h Mourão, Fernando Henrique de Jesus
A hybrid recommendation method that combines
forgotten items and non-content attributes / Fernando
Henrique de Jesus Mourão. — Belo Horizonte, MG,
2014
xix, 102 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas
Gerais - Departamento de Ciência da Computação.
Orientador: Wagner Meira Júnior

1. Computação - Teses. . 2. Sistemas de
recomendação – Teses. I. Título.

CDU 519.6*73(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO


FOLHA DE APROVAÇÃO

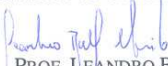
A hybrid recommendation method that combines forgotten items and non-content attributes

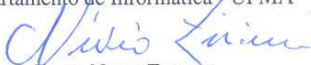
FERNANDO HENRIQUE DE JESUS MOURÃO


Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. WAGNER MEIRA JÚNIOR - Orientador
Departamento de Ciência da Computação - UFMG


PROF. LEONARDO CHAVES DUTRA DA ROCHA - Coorientador
Departamento de Ciência da Computação - UFSJ


PROF. LEANDRO BALBY MARINHO
Departamento de Informática - UFMA


PROF. NIVIO ZIVIANI
Departamento de Ciência da Computação - UFMG


PROF. RENATO MARTINS ASSUNÇÃO
Departamento de Ciência da Computação - UFMG


PROFA. SANDRA APARECIDA DE AMO
Faculdade de Computação - UFU

Belo Horizonte, 09 de dezembro de 2014.

“Veni, Vidi, Vici”
(Julius Caesar)

Abstract

Recommender Systems (RSs) play important role in many Web applications nowadays, helping users to find their favorite items amid a huge number of options. Among numerous open challenges inherent to RSs, this dissertation addressed the challenge of enhancing the discovery of potentially relevant items for each user. In this sense, we exploited two algorithmic limitations unaddressed in the literature. First, RSs fail to bring back items consumed long ago that are potentially relevant for users nowadays. Second, RSs fail to capture the whole extent on which implicit signals of preferences observed on past consumption relate to preferences observed on current consumption. We addressed the first limitation by reviewing the user’s long-term history and identifying a subset of consumed items forgotten but still re-consumable (i.e., **forgotten re-consumable items**). We mitigated the second limitation by explicitly modeling the subset of attributes derived from metadata or consumption data (i.e., **non-content attributes**). Finally, we proposed ForNonContent, a hybrid method that addresses both limitations simultaneously. Besides validating these algorithmic limitations, offline analysis on four real datasets demonstrated that recommending forgotten re-consumable items may bring diversified and novel recommendations. Also, we found that non-content attributes may enhance recommendations of six major RSs. Furthermore, we identified a complementary nature of the enhancements associated to each limitation. Finally, a user study with MovieLens members demonstrated that users appreciated the recommendations issued by ForNonContent. Thus, this work pointed out a new and promising direction to enhance the user experience with RSs.

List of Figures

1.1	ForNonContent: the hybrid method that exploits simultaneously forgotten re-consumable items and non-content attributes.	5
3.1	Representation of the item space along two dimensions: re-consumption and forgetfulness.	20
3.2	Probability distribution of normalized rank difference values between training items ranked by utility and training items ranked by <i>mem-ret</i>	34
3.3	Analysis of mean recall for re-consumed items per time interval between the test moment and their last consumption in the training data.	37
3.4	Analysis of mean rerate for recommended items per time interval between the test moment and their last consumption in the training data.	38
4.1	Visual Representation of Preference Mismatching along three non-content attributes	43
4.2	Analysis of user preferences.	50
4.3	Distributions of relative standard deviations of user consumption.	51
4.4	Analysis of preference mismatching.	52
4.5	Analysis of <i>Precision@50</i> gains loses by exploiting, individually, popularity, similarity and recency.	54
4.6	Analysis of quality gains loses by exploiting, simultaneously, popularity, similarity and recency.	55
4.7	Percentage of users for whom the non-content preference information produced any improvement.	56
4.8	Analysis of mean preference mismatching per rank in CF recommendation lists.	56
6.1	Snapshots of the Evaluation System.	71
6.2	Results of ‘Battle of Recommendation Lists’.	73
6.3	Results of ‘Rating recommended items’.	74

6.4 Results of ‘Questionnaire about the recommendations’. 75
6.5 Results of ‘Ranking recommendation lists’. 76
6.6 Results of the questionnaire about personal information of the users. 77

List of Tables

2.1	Collaborative Filtering Methods Classification.	11
3.1	Dataset information.	30
3.2	Analysis of relevance for the individual perspective of the Oblivion Problem.	31
3.3	Analysis of relevance for the collective perspective of the Oblivion Problem.	31
3.4	Pearson Correlation coefficient between rank position of re-consumed items in the utility-based rank and the <i>mem-ret</i> -based one.	35
3.5	Analysis of mean recall of re-consumed items and re-consumable rate (rerate) of recommended items.	36
3.6	Results on novelty and diversity metrics for different recommendation strategies.	39
4.1	Dataset information.	49
5.1	Probability of a RSs issue non-forgotten items to the users	60
5.2	Analysis of complementarity between Remembrall and NonContent.	67
A.1	Domains where the Oblivion Problem presents strong relevance.	87
A.2	Domains where the Oblivion Problem is not relevant.	88

List of Abbreviations & Nomenclature

Abbreviations

<i>ACT-R</i>	Adaptive Control of Thought-Rational
<i>BC</i>	Bhattacharyya Coefficient
<i>BMF</i>	Biased Matrix Factorization
<i>CF</i>	Collaborative Filtering.
<i>CB</i>	Content Based.
<i>ILS</i>	Inter-list Similarity
<i>LCCC</i>	Less Correlated to Current Context
<i>LF</i>	Latent Feature Log Linear Model
<i>LFA</i>	Less Frequently Accessed
<i>LRA</i>	Less Recently Accessed
<i>mem-ret</i>	Memory Retrievability.
<i>MF</i>	Matrix Factorization
<i>MILS</i>	Median Inter-list Similarity
<i>MPoI</i>	Median Percentage of Intersection
<i>MPoRI</i>	Median Percentage of Re-consumed Items
<i>NSD</i>	Normalized Standard Deviation
<i>PoI</i>	Percentage of Intersection

$rerate$	Re-consumption Rate.
RSD	Relative Standard Deviation

Nomenclature

u, v	A single user.
i, j, m, s, r	A single item.
a	A given attribute that describes items.
c, f, l, t	A single moment in time.
K, M, N	The size of a given list of set.
U	The set of all users of a domain.
S	A subset of users of a domain.
I	The set of all items of a domain.
C_u	The subset of all items consumed by the user u in a domain.
R_u	The subset of items recommended by a given algorithm to user u .
U	The set of all users of a domain.
$T_{u,i}$	The set of all distinct moments in which the user u consumed the item i .
V	An Euclidean vector space.
E_i	The set of all relevance values, assigned by the users, related to each distinct consumption of i in a training set.

Contents

Abstract	ix
List of Figures	xi
List of Tables	xiii
List of Abbreviations & Nomenclature	xv
1 Introduction	1
1.1 Background & Motivation	1
1.2 Thesis Statement	2
1.3 Main Contributions	5
1.4 Outline	6
2 Background Concepts & Related Work	7
2.1 Recommendation Problem	7
2.2 Quality Requirements	8
2.3 Recommendation Methods	10
2.3.1 Content-Based Methods	10
2.3.2 Collaborative Filtering	11
2.3.3 Hybrid Methods	12
2.4 Temporal Evolution	13
2.5 User Behavior Modeling	15
2.6 Summary	16
3 Forgotten Re-consumable Items	19
3.1 Problem Definition	19
3.1.1 The Oblivion Problem	19
3.1.2 Motivation	21

3.1.3	Perspectives of Oblivion	21
3.2	Scope of Relevance	22
3.3	Addressing the Oblivion Problem	25
3.3.1	Identifying Forgotten Items	25
3.3.2	Recommending Forgotten Re-consumable Items	27
3.4	Case Studies	29
3.4.1	Datasets	29
3.4.2	Analysis of Scope	30
3.4.3	Experimental Design	32
3.4.4	Exploiting Forgotten Re-consumable Items	33
3.5	Summary	39
4	Non-Content Preference Mismatching	41
4.1	Problem Definition	41
4.1.1	Non-Content Preference Attributes	41
4.1.2	Preference Mismatching	42
4.1.3	Motivation	44
4.2	Assessing Preference Mismatching	44
4.3	Exploiting Preference Mismatching	45
4.3.1	The proposed Hybrid Method	46
4.3.2	Rationale for the proposed method	48
4.4	Case Studies	48
4.4.1	Datasets	48
4.4.2	Evaluated Recommenders	49
4.4.3	Existence of Preference Mismatching	49
4.4.4	Experimental Design	52
4.4.5	Exploiting Preference Mismatching	53
4.5	Summary	57
5	Combining Forgotten items and Non-content preference	59
5.1	Motivation	59
5.2	Remembrall: A recommender of forgotten re-consumable items	61
5.3	NonContent: Mitigating non-content preference mismatching	62
5.4	ForNonContent: Combining Remembrall and NonContent	64
5.5	Case Studies	65
5.5.1	Datasets	65
5.5.2	Experimental Design	66

5.5.3	Analysis of Complementarity	67
5.6	Summary	67
6	End-user Study	69
6.1	Evaluation Goals	69
6.2	Evaluated Methods	70
6.3	Methodology of Evaluation	70
6.4	The Web-based Evaluation System	71
6.5	User-Centered Results	72
6.6	Summary	78
7	Conclusions & Future Work	79
7.1	Restatement of Thesis	79
7.2	Empirical Findings	80
7.3	Summary of Contributions	81
7.4	Limitations of the Work	82
7.5	Recommendation for Future Research	83
7.6	Final Remarks	84
Appendix A Qualitative Analysis of the Oblivion Problem		85
Appendix B Questionnaires Used in the User Studies		89
Bibliography		91

Chapter 1

Introduction

1.1 Background & Motivation

The huge amount of information available on a range of WEB applications generates a challenging scenario: users face more options than they can effectively handle [Schwartz, 2005; Adomavicius and Tuzhilin, 2005]. Commercial applications, such as Amazon, Netflix or Last.fm, among others, unwittingly hinder users to find products of their interest by providing a data collection with thousands or even millions of distinct products. Hence, tools that filter the available information, showing only what is more likely to be of user interest, are becoming increasingly important. Several studies propose strategies to recommend products, information and services to customers nowadays [Abbase and Mirrokni, 2007]. These Recommender Systems (RSs) aim to estimate potentially interesting items to users, based on a prior knowledge about user behaviors and/or about relevant characteristics of the items [Adomavicius and Tuzhilin, 2005].

A permanent challenge for RSs is how to enhance the discovery of items that users would want to consume while avoid recommending undesired ones [Anderson, 2006a]. The prospect of discovery determines the practical value of RSs in many scenarios, since RSs are useful to users when presenting potentially relevant items not easily reachable otherwise. [Vargas and Castells, 2011]. For instance, recommending new holiday destinations would be more useful than recommending favorite destinations. However, the former case is risky because there are several destinations the target user is not willing to visit at a given moment. Since users are usually interested and able to consume only a tiny portion of the available items in a domain, the challenge grows as the number of available items increases. The main goal of this dissertation is to propose new and effective RSs to enhance the discovery of potentially relevant items for each user in various domains.

Current efforts to address this challenge focus on proposing hybrid methods, which combine the strengths of distinct RSs while mitigate their weakness [Ricci et al., 2011]. As each existing RS has different strengths and weaknesses, researchers proposed a large number of combination strategies [Burke, 2002]. In general, Collaborative Filtering (CF) methods, which correlate user ratings with items, are combined to Content-based (CB) methods, which correlate user ratings with item attributes [Adomavicius and Tuzhilin, 2005; Ekstrand et al., 2011]. This particular combination has provided significant improvements due to exploiting simultaneously different perspectives of the user behavior. While CF methods assume that users with similar consumption history would share common interests, CB methods conjecture that each user exhibits a systematic preference correlated with some item attributes [Ricci et al., 2011]. Despite all advances on hybrid methods, there is still room to make them more effective or applicable to a wider range of scenarios, such as trip advice, financial services, among others [Konstan and Riedl, 2012]. This work assumes that further enhancements on hybrid methods require, at first, identifying and understanding limitations that impact the quality of the underlying RSs used to build new hybrid strategies.

Indeed, there are several limitations related to recommender systems [Ricci et al., 2011]. RSs are fundamentally based on detecting recurrent behaviors, existing on previously acquired data, and replicating these behaviors in order to predict future ones. As the human behavior and taste do not follow strict and easily predictable patterns over time, there are bounds of prediction inherent to RSs. Further, RSs face data constraints in several recommendation domains, lacking samples of data representative enough to derive adequate models [Schein et al., 2002]. Finally, we point out algorithmic limitations related to state-of-the-art RSs. As argued by Burke [2002], all of the known recommendation techniques have strengths and weaknesses, and there is no single model to handle the whole complexity of user behavior. While hybrid methods may handle the complexity of modeling distinct pieces of the user behavior due to combining different perspectives of analysis, we believe there still are pieces of such behavior uncovered by current RSs. Instead of refining the way existing hybrid methods address and combine pieces already covered by RSs, we identify and address novel pieces of user behavior unaddressed in the literature.

1.2 Thesis Statement

In the light of this context, we investigated two distinct algorithmic limitations, not perceived by the literature, through the following thesis statement:

State-of-the-art recommenders underexploit two types of information useful to enhance the discovery of relevant items: the long-term history and implicit signals of preference observed on past consumption of each user.

A closer look at the foregoing statement allows us to evince the main assumptions hereby adopted. First, although current RSs focus on discovering unknown items to users [Ricci et al., 2011], in several domains, users may be particularly interested in consuming items they have already consumed in the past, but not recently (e.g., music). Thus, the first limitation we address is to recommend known items, since they would match a piece of the current user’s taste neglected by several RSs. Second, user preferences may stem from usual content-related data, such as genres or movie actors [Jannach et al., 2010], but they may also come implicitly from metadata or consumption attributes that are not handled as “content”, such as popularity or recency of consumption. Hence, as second limitation, we investigate whether these implicit signals of preference are properly captured by RSs, once these methods do not model explicitly such signals. This work validates our statement in real scenarios; quantifies its impact on recommendations; and proposes a new hybrid method to address both limitations simultaneously, enhancing the discovery of potentially relevant items.

Aiming to handle the complexity inherent to this dissertation, we split our thesis statement into three main underlying hypotheses, which are being investigated in order, as follows:

- **Hypothesis 1:** *State-of-the-art RSs fail to bring back items consumed long ago that are potentially relevant for users nowadays.*
- **Hypothesis 2:** *State-of-the-art RSs fail to capture the whole extent on which implicit signals of preferences observed on past consumption relate to preferences observed on current consumption.*
- **Hypothesis 3:** *The two aforementioned algorithmic limitations, when addressed simultaneously, provide complementary enhancements to RSs.*

Again, the motivation to raise and address hypothesis 1 comes from the observation that users may be particularly interested in re-consuming items they already know. The challenge in this case is how to recommend properly known items when there are reasons to believe such items are no longer relevant. Instead of identifying the whole set of re-consumable items, which requires a complex modeling of user behavior and domain evolution over time, we focus on a subset easier to identify. We consolidate a new source of re-consumable items by reviewing the user’s long-term

history and identifying the subset of consumed items forgotten but still re-consumable (i.e., **forgotten re-consumable items**). Forgotten items comprise a promising source of re-consumable ones, since some of the former may still be related to the current taste of each user. While it is desirable to rescue forgotten and re-consumable items, we cannot recommend forgotten but not re-consumable ones. Formally, we define this compromise as **the Oblivion Problem**. This problem is particularly relevant for our goal of enhancing the discovery of items because RSs frequently neglect forgotten items over time [Mourão et al., 2011b]. For instance, in many scenarios RSs assume that recent data are more relevant as a consequence of a user’s taste drifting and the emergence of new items over time.

In turn, hypothesis 2 derives from four observations. First, each user exhibits a systematic preference correlated with some item’s attributes [Jannach et al., 2010]. For instance, users in a movie domain may be interested in watching movies from a specific genre (e.g., comedy). Second, a subset of these attributes may come from metadata or consumption attributes that are not usually handled as “content” (i.e., **non-content attributes**) [Mourão et al., 2013]. Considering again movie domains, a user might watch only blockbusters, which means he/she watches only popular movies. Third, non-content attributes are not explicitly modeled by state-of-the-art RSs. Fourth, non-content attributes may not be properly captured and exploited by state-of-the-art RSs. Such as expected for users, we assume that RSs prioritize a specific range of values for each non-content attribute, since they are based on inductive premises that make some assertions about items or users. Hence, a relevant question concerns the match between user non-content preference and his/her recommendation’s non-content attributes. Aiming to evaluate such match, we define **Preference Mismatching**, a metric that quantifies the difference between the actual user’s non-content preference and the recommendation’s non-content attributes. Enhancing this match would mean more accurate recommendations in practice, as well as the discovery of new items with attribute values close to the user’s preferences.

The third hypothesis raises another question in this work: could we combine the potential enhancements inherent to each limitation, providing even better recommendations? In order to answer this question, we propose ForNonContent, a new hybrid method that combines methods proposed to address each limitation (Figure 1.1). First, ForNonContent issues a list L_{For} of Top-N forgotten re-consumable items for each target user u . Also, it executes a given CF method, providing another ordered recommendation list L_{CF} of size $M \gg N$ for u . It derives item attributes based on three non-content attributes: popularity, recency, and similarity. These derived attributes conform a vector space V where each item from L_{CF} is represented.

Later, ForNonContent builds a CB model for u considering attribute space V and determines a new score for each item from L_{CF} , based on the preference mismatching metric. Then, it combines the CF score with the CB score and issues another list L_{NonC} of Top- N items that mitigate u 's non-content preference mismatching. Finally, ForNonContent combines L_{For} with L_{NonC} , intercalating the distinct items of each list and issuing a final Top- N recommendation list.

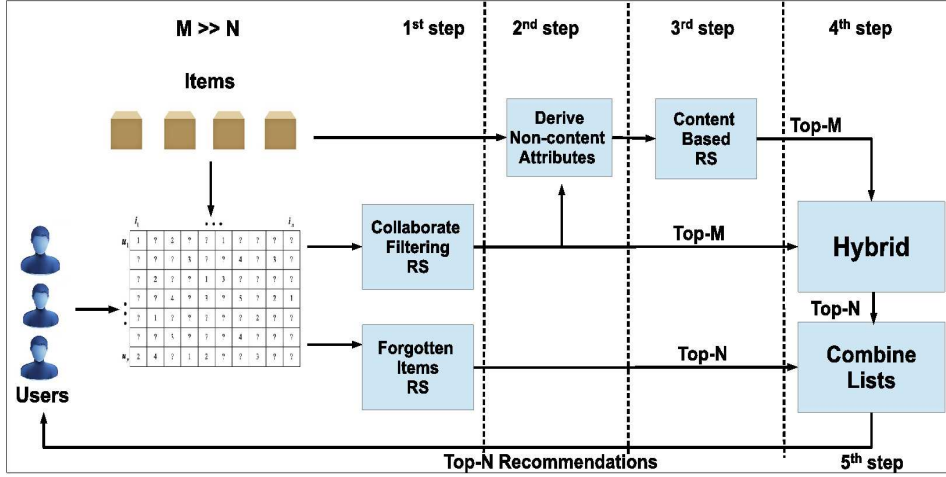


Figure 1.1: ForNonContent: the hybrid method that exploits simultaneously forgotten re-consumable items and non-content attributes.

1.3 Main Contributions

The investigation of our thesis statement represents concrete contributions for the area, pointing out a new and relevant research direction for RSs by which significant enhancements may be achieved. Further, the formalization of some concepts and problems, as well as the consolidation of new methods for addressing the raised problems and challenges are of great relevance. Hence, as main contributions of this dissertation, we highlight:

1. the formalization of a new problem in recommendation domains, namely the **Oblivion Problem**;
2. the proposal of distinct strategies to identify and recommend forgotten re-consumable items in real domains;
3. the modeling and use of information that, so far, has not been completely exploited by RSs, namely the **non-content attributes**;

4. the proposal of a hybrid method that explicitly incorporates non-content preferences into CB models;
5. the proposal of a hybrid method that exploits simultaneously non-content preferences and forgotten re-consumable items;
6. in-depth offline analyses on real domains to show the complementary benefits of addressing each limitation with regard to the discovery of items; and
7. a live study with real users that confirms the usefulness of addressing these algorithmic limitations for sake of user satisfaction.

Finally, to the best of our knowledge, this work is the first effort that effectively exploits each of the aforementioned algorithmic limitations in recommendation domains.

1.4 Outline

The remainder of this dissertation is organized as follows. Chapter 2 presents basic concepts and summarizes the main related work. Chapter 3 discusses the main issues of the Oblivion Problem, strategies to identify and recommend forgotten re-consumable items, as well as offline evaluations of the proposed strategies. Chapter 4 describes the non-content attributes and the systematic mismatching between current recommendations and the actual user taste with respect to these attributes. This chapter also introduces a hybrid method to mitigate this mismatching and discusses offline evaluations of this method. Chapter 5 presents and evaluates ForNonContent, our new hybrid method that combines solutions for the Oblivion Problem and non-content Preference Mismatching. Chapter 6 describes a live study conducted with real users to contrast the proposed methods against some state-of-the-art RSs. Finally, Chapter 7 points out the main conclusions and identifies relevant future directions for this study.

Chapter 2

Background Concepts & Related Work

In this chapter we present the general recommendation problem, as well as a summary of its main studies in the literature. Aiming at a clearer placement of this dissertation against previous works, we divide such studies according to key perspectives of analysis strongly related to it. First, we formalize the *recommendation problem* and discuss some tasks inherent to it. Then, we take into account some *quality requirements* related to our goal of enhancing discovery of relevant items in RSs. Later, we review the *recommendation methods* close to the proposed ones. Next, we summarize the set of studies most related to the Oblivion Problem, which comprise works on *temporal evolution* in recommendation domains. Thereafter, we present the main works about *user behavior modeling*, which are related to modeling user's non-content preference. Finally, we summarize the main differences between this work and previous ones, regarding all these perspectives.

2.1 Recommendation Problem

Recommendation plays important roles in e-commerce nowadays, helping users to find their favorite items and services [Abbasse and Mirrokni, 2007; Resnick and Varian, 1997; Burke, 2002]. The recommendation goal is usually defined as finding, among a potentially large number of items, those items that better suit individual interests of each user. More formally, let U be the set of all users and let I be the set of all existing items of a domain (e.g., books, movies, or songs). Let $f(u, i)$ be a utility function that measures the usefulness of each item $i \in I$ to user $u \in U$. Let also consider that each user u would demand to consume at most N distinct items, since it is not feasible to assume that users would consume an infinite number of items. Thus, we could state the recommendation problem as identifying a subset of items $R_u \subset I$ of size N (i.e.,

$|R_u| = N$) that maximizes the utility $f(u, i)$ for each user $u \in U$ and item $i \in R_u$ [Adomavicius and Tuzhilin, 2005], as shown by Equation 2.1.

$$R_u = \arg \max_{i \in I} f(u, i) \quad (2.1)$$

Recommender Systems (RSs) could be defined as any system designed to deal with the recommendation problem, producing individualized recommendations as output, or guiding users through a huge variety of options [Burke, 2002; Hoashi et al., 2003]. In RSs, the utility of an item is usually represented by a numerical rating, which indicates to what extent a given user likes a particular item. The main problem of RSs is that the utility $f(u, i)$ is usually defined only for a restricted subset of the whole space $U \times I$. It means that $f(u, i)$ needs to be extrapolated to the whole space $U \times I$, as argued in [Herlocker et al., 2004].

When taking into account the end-users goals, the recommendation problem could derive distinct tasks [Herlocker et al., 2004]. The most common tasks are *Annotation in Context* and *Find Good Items*. *Annotation in Context*, or alternatively the Rating Prediction task, aims to predict the proper rating value $f(u, i)$ that a user u would give to each unseen item i [Resnick et al., 1994]. On the other hand, *Find Good Items*, or *Top-N Recommendation*, aims to provide a ranked list of items that each user u would mostly like, not concerning to predict the actual rating values $f(u, i)$. Herlocker et al. [2004] argues that *Find Good Items* is the core recommendation task and it recurs in a range of research and commercial domains. Given such relevance, this dissertation focuses on the Top-N Recommendation task, concerning with ranking accuracy only. Thus, for sake of notation simplicity, in the following sections and chapters all mentions to recommendation refers specifically to the Top-N Recommendation task.

2.2 Quality Requirements

As discussed in Chapter 1, the main goal of this dissertation is to enhance the discovery of items potentially relevant for each user. Major developments of discovery in RSs describe this goal through three main quality requirements: usefulness, diversity and novelty. Usefulness is the primary goal of RSs and refers to the capability of RSs to identify and present to users items that match their interest [Herlocker et al., 2004]. Thus, adequate recommendation lists should contain items familiar to the user consumption habits. Diversity is related to how distinct each item in a recommendation list is with respect to the others [Vargas and Castells, 2011]. Although the domains where RSs operate have a wide variety of items, recommendations are, in general,

poorly diversified [Zhou et al., 2010; McSherry, 2002]. In turn, novelty refers to how different a piece of information is from “what has been previously seen” by a specific user [McGinty and Smyth, 2003]. Some works assume that an RS is valuable if it is able to provide new items or information to users [Sarwar et al., 2001; Kawamae, 2010].

Fulfilling these three requirements simultaneously has an appealing relevance, since it allows RSs to match properly demands of consumption with existing offers, increasing both the sales ratio and the satisfaction of users with RSs. However, previous works state diversity and novelty as requirements diametrically opposite to the notion of “familiarity” represented by usefulness [Zhang et al., 2012; Herlocker et al., 2004]. Improving usefulness, novelty and diversity simultaneously is a constant challenge for a wide range of applications [Lathia et al., 2010; Zhou et al., 2010]. A main contribution of this work relies exactly on pointing out a new direction for such achievement.

Several works in the literature pursue simultaneous gains with respect to these three requirements [Smyth and McClave, 2001; McGinty and Smyth, 2003; McSherry, 2002]. For instance, Zhang et al. [2012] combined distinct RSs, balancing usefulness, diversity, novelty and serendipity. Zhou et al. [2010] also presented a combination strategy for RSs based on a bipartite user-object graph and heat spreading diffusion. Ribeiro et al. [2012] modeled this challenge as a multi-objective optimization problem and applied the Strength Pareto approach to solve it. Zhang and Hurley [2008] defined the diversification goal as a binary optimization problem and relaxed it to a trust-region problem to determine a solution. Further, McGinty and Smyth [2003] clarified the role of diversity in traditional RSs, highlighting the pitfalls of naively incorporating current diversity enhancing techniques into existing RSs. We also found studies that discuss the relationship between diversity and novelty. Vargas and Castells [2011] argued that diversity is closely related to novelty in the sense that when a set is diverse, each item is “novel” with respect to the others. On the other hand, RSs that recommend novel items also tend to promote a global diversity over time.

Our proposal differs from previous ones by exploiting two novel types of information that may affect usefulness, diversity and novelty. Matching implicit signals of user preferences observed on past consumption affects directly the usefulness of recommendations, since we filter out items that do not suit such preferences. On the other hand, recommending potentially relevant items consumed long ago affects diversity and usefulness. Some new user wishes might be met by old relevant items forgotten through time, improving diversity while keeping usefulness stable. Regarding novelty, we argue that relevant items consumed long ago may represent, in some sense, a degree of novelty, since users might not remember by themselves most of these “lost” items.

Finally, we should mention that evaluating user experience on RSs through qual-

ity requirements, such as in the foregoing discussions, represents a pure algorithmic evaluation strategy. Most studies in RSs use a metric to quantify the aforementioned requirements and perform empirical assessments on existing data [Konstan and Riedl, 2012]. For instance, distinct works quantify usefulness through the *accuracy* metric [Herlocker et al., 2004; Meyer et al., 2012], which measure how close the ranking of items issued by an RS is from the user’s true ranking in the Top-N recommendation task. However, this sort of measure does not quantify whether RSs can recommend truly *valuable items previously unknown* to the users [Ge et al., 2010; McNee et al., 2006; Adomavicius and Zhang, 2012]. Further, user experience includes, besides the delivery of personalized recommendations to users, the interaction of each user with those recommendations [Konstan and Riedl, 2012].

Despite not evaluating properly user experience, pure algorithmic evaluation strategies became a common choice since they allow us to generate and replicate results easily. Measuring user experience requires developing a complete system, which includes algorithms and user interface, and carrying out field studies with real users [Konstan and Riedl, 2012]. Recent efforts have demonstrated the value of these field studies in order to clarify the actual value of RSs for users [Cosley et al., 2003; Knijnenburg et al., 2012]. In this sense, Pu et al. [2011] synthesized and organized the accumulation of existing questionnaires and developed a well-balanced framework for live experiments in recommendation domains. Knijnenburg et al. [2012] extended and tested a user-centric evaluation framework for recommender systems proposed in [Knijnenburg et al., 2010]. Also, Konstan and Riedl [2012] presented a survey of the most important developments related to the user experience in RSs. In this work, besides pure algorithmic assessments, we contrast the proposed algorithms against some state-of-the-art methods in live experiments with real users.

2.3 Recommendation Methods

In this section, we briefly discuss the main methods belonging to the three classes of RSs most related to this work, namely Content-based (CB), Collaborative Filtering (CF) and Hybrid Methods. Broader surveys of methods related to distinct classes of RSs are presented in [Jannach et al., 2010; Candillier et al., 2009; Ricci et al., 2011].

2.3.1 Content-Based Methods

CB methods estimate the previously described utility function $f(u, i)$ of item i to user u using the known utilities $f(u, j)$ assigned by user u to each item j “similar” to i . This

similarity measure is usually defined by comparing distinct N-dimensional vectors of attributes that describe each item [Adomavicius and Tuzhilin, 2005; Mooney and Roy, 2000]. Thus, CB methods conjecture that each user exhibits a systematic preference correlated with some item attributes [Ricci et al., 2011]. Lops et al. [2011] presented detailed explanations of state-of-the-art content-based methods.

Content-based methods are strongly rooted by the Information Retrieval area, in the sense that both are based on the availability of item descriptions and a profile that assigns importance to each description [Jannach et al., 2010]. The common assumption is that users are able to formulate queries that express their interests or information needs in terms of intrinsic attributes of items [Hofmann, 2004]. However, it may be difficult, in some contexts, to identify suitable descriptors such as keywords, topics and genres, among others, that may be used to accurately describe interests.

As main advantages of this class of methods we highlight: (1) the user independence, since only ratings from the target user of recommendation are exploited; (2) transparency, since it is clear how recommendations are provided; and (3) ability to cope with the so-called Cold Start problem¹, once items are represented through describing attributes. On the other hand, the main disadvantages are related to over-specialized recommendations, since very similar items are always recommended to the same users, damaging novelty and diversity.

2.3.2 Collaborative Filtering

Collaborative Filtering methods assume that users with similar consumption history would share common interests. These methods may be classified into four distinct classes, according to the strategy and data source used, as shown by Table 2.1.

Table 2.1: Collaborative Filtering Methods Classification.

		Methodology Strategy	
		Memory-based	Model-based
Data source	User-oriented	Combines the preferences of the K most like-minded users, with similar or correlated behavior.	Exploits the user preference history to train models that estimate unknown user preferences.
	Item-oriented	Combines the ratings of the K most similar items, considering all users.	Uses past item ratings to train models that estimate unknown user ratings.

¹*Cold Start* refers to the difficulty in making recommendations on new items or for new users, since there is little information in the system about such items and users [Schein et al., 2002; Lam et al., 2008].

User-oriented Memory-based CF methods assume that people trust the recommendations from like-minded people [Yu et al., 2004]. Typically, these methods determine for each user a group of “*nearest neighbor*” users whose past ratings are similar, or highly correlated, with the user ratings. Scores for unseen items are predicted based on a combination of the scores known from the nearest neighbors. Recently, Said et al. [2013] proposed K-furthest neighbors (KFN), a KNN-based method that exploits the most dissimilar neighbors. As these Memory-based CF methods are based primarily on clusters of users, their effectiveness depends on the generated clusters express high correlations between users. *Fuzzy methods* [Wu and Li, 2008] were recently used to allow users belong to distinct clusters, reflecting the fact that users exhibit a mixture of tastes. Also, some studies extended these methods for items, since the number of distinct items is smaller than the number of distinct users in several domains [Sarwar et al., 2001; Deshpande and Karypis, 2004]. Thus, Item-oriented Memory-based CF defines, for each item, a group of the most similarly evaluated items, considering all users in the domain. Later, scores for unseen items are derived from scores given for similar items by the target user of the recommendations. A drawback is that memory-based methods do not scale well in terms of memory and computer time [Hofmann, 2004].

Model-based CF methods learn a descriptive model of user preferences and then use it to generate ratings [Yu et al., 2004]. Many of these methods are inspired on machine learning algorithms, such as neural network classifiers [Billsus and Pazzani, 1998], induction rule learning [Basu et al., 1998], Bayesian networks [Breese et al., 1998] and latent factor models [Koren, 2008]. Latent factor methods represent one of the most efficient and popular approaches in Model-based CF, since they are generally effective at estimating overall structure that relates, simultaneously, most or all items [Koren et al., 2009; Sarwar et al., 2000]. For instance, Hofmann [2004] adapted a probabilistic latent semantic analysis to the recommendation task. These techniques have proven to be efficient in recommender systems when predicting user preferences from known user-item ratings [Takács et al., 2008]. However, computational costs involved in training the models tends to be high.

A survey for CF methods was presented in [Su and Khoshgoftaar, 2009; Ekstrand et al., 2011]. Also, Cacheda et al. [2011] presented a comparative study among distinct CF techniques. Rafter et al. [2009] evaluated the limitations of neighborhood-based estimates to predict the actual taste of the users. Pennock et al. [2000] discussed axiomatic foundations of collaborative filtering based on social choice theory. In [Yu et al., 2004; Koren, 2008], the authors showed that CF methods present many advantages over CB methods such as simplicity and generality. Further, CF methods do not require us to tune many parameters neither perform extensive training

stages. In addition, these methods offer the potential to uncover implicit patterns that would be difficult or impossible to profile using CB techniques [Bell and Koren, 2007]. We also highlight their ability to achieve novel and unexpected items through neighbor users, improving novelty and diversity in RSs. As the main disadvantages of CF methods, we point out: (1) inability to cope with the Cold Start Problem; and (2) biased recommendations to popular items. Since users tend to consume mostly popular items, user similarities are heavily defined by these items.

2.3.3 Hybrid Methods

Aiming to overcome individual limitations of RSs, an increasing number of researches combine existing RSs, defining the so-called hybrid methods. In most of the cases, CF methods are combined with CB ones [Adomavicius and Tuzhilin, 2005; Balabanović and Shoham, 1997]. Thus, hybrid methods are able to attenuate the limitations of both while exploiting simultaneously the strengths for recommendation [Schein et al., 2002; Ricci et al., 2011].

Distinct strategies of hybridization could be found in the literature. Li and Murata [2012] presented a hybrid approach that incorporates multidimensional clustering into CF recommendation models. Ribeiro et al. [2012] presented an evolutionary search for hybrid models following the Strength Pareto approach, which identifies hybrid models that are in the Pareto frontier. In [Zhang et al., 2012], the authors proposed a hybrid framework that attempts to balance distinct quality requirements by combining the rank outputs of three algorithms: Artist-based LDA, Listener Diversity and Declustering. Another hybrid model, based on a bipartite user-object graph and heat spreading diffusion, is proposed in [Zhou et al., 2010]. [McAuley and Leskovec, 2013] proposed a new hybrid statistical model that combines latent dimensions in rating data, which is a CF model, with hidden topics in review texts, a CB model. Also, [Khrouf and Troncy, 2013] presented a hybrid method for event recommendations. Besides a CB system that overcomes the data sparsity, this method includes a CF model to model social aspects. Burke [2007] discussed and contrasted distinct hybridization techniques. In [Jannach et al., 2010], the authors stated that although many recommender applications are actually hybrids, little theoretical work has focused on how to hybridize algorithms. Burke [2002] presents a broad survey on hybrid methods.

Despite all advances in RSs, mainly on hybrid methods, we believe that opportunities for improvements exist, since even state-of-the-art RSs are unable to provide adequate recommendations in different real scenarios [Herlocker et al., 2004; Rafter et al., 2009]. This dissertation proposes a new hybrid method that exploits two

of these opportunities simultaneously. First, through the Oblivion Problem, we exploit the long-term history of each user. To the best of our knowledge, this is the first effort on this direction. The set of studies most related to the Oblivion Problem refers to temporal dynamics in RSs, which we discuss in the next section. Second, by taking into account the non-content attributes, we model implicit signals of preferences. Again, we did not identify any work in literature that has identified and modeled explicitly these attributes for sake of user experience in RSs. Section 2.5 presents the set of studies about user behavior modeling most related to the non-content preferences.

2.4 Temporal Evolution

RSs are based on the premise that past user behavior repeats in the future, which is not always true. The consolidation of user preference models need to find a balance between penalizing time effects that have low impact on future behavior, while capturing trends that reflect inherent recurrent patterns in the data. Thus, numerous studies started to consider a temporal constraint t on the aforementioned utility $f(u, i)$, realizing that users would present distinct demands at different moments [Xiang et al., 2010; Lathia et al., 2010; Rana and Jain, 2012]. Distinct works on temporal evolution in recommendation domains assessed the quality of RSs over time [Campos et al., 2011]. For instance, Lathia et al. [2009] evaluated the impact of temporal dynamics on recommendations. In [Zhang and Hurley, 2008], the authors measured how the diversity of recommendations is affected over time.

Recently, most works propose new strategies to deal with this problem for sake of accuracy [Lathia et al., 2010; Campos et al., 2013; Cremonesi and Turrin, 2010]. Such strategies are recognized in the literature by distinct names, such as Adaptive Information Server (AIS) [Billsus and Pazzani, 2000], Dynamic Recommender Systems (DRS) [Rana and Jain, 2012] or Time-Aware Recommender Systems (TARS) [Anand and Mobasher, 2007]. All of these works agree that static user profiles, which is the prevalent methodology in traditional RSs, cannot assess properly the preference of users over a period of time [Gauch et al., 2007]. Thus, temporal dynamics is modeled in terms of user preferences that evolve or item contents that change due to the addition of new items or deletion of older ones. Koren [2009] argued that proposing recommendation models that take time into consideration tend to be more effective than proposing complex models. Therefore, variations on the user profiles over time have been incorporated to RSs [Stern et al., 2009]. Tang and Zhou [2013] proposed a set of dynamic features for describing the evolving behavior based on time series

analysis. Ding and Li [2005] presented a novel algorithm for computing the temporal weights of items so that older items get smaller values. Rana and Jain [2012] argued that dynamics is much more complex and should be addressed in a multidimensional factor analysis model. While many time-evolving models introduced time as a universal dimension shared by all users [Sun et al., 2007], Xiang et al. [2010] argued that time is a local effect and should not be used for comparison among users. However, most of these studies constantly update the user profiles by looking at recently consumed items, adapting the input information of RSs to the most recent data [Cebrián et al., 2010].

In this work, we exploit temporal dynamics through a new strategy that enhances diversity in RSs by using the subset of items consumed by users in the remote past (i.e., forgotten items). We argue that forgotten items are promising since some user needs might be met by old items in some domains. In this sense, we evaluate distinct strategies to identify and recommend properly forgotten items. These strategies are based on ACT-R (Adaptive Control of Thought-Rational), a well-known cognitive architecture that models the human memory [Mellon et al., 2007]. We should mention that the use of psychology studies in recommendation is not novel. Indeed, ACT-R have been evaluated previously in the context of paper recommendations [Van Maanen et al., 2010; van Maanen and Marewski, 2009]. Also, Anand and Mobasher [2007] presented a novel approach to incorporate user temporal context within the recommendation process based on human memory models proposed in psychology. However, this is the first work where such models are used for evaluating forgetfulness in the context of recommendation.

2.5 User Behavior Modeling

A proper modeling of the user behavior is crucial for the success of RSs [Jannach et al., 2010]. This is a complex task given the combinatorial nature of representing each user through a subset of information available in a domain. As we are dealing with an extrapolation problem, ranking items with unknown utility, a bigger and more diversified amount of information tends to help us in this process [Adomavicius and Tuzhilin, 2001; Adomavicius et al., 2005]. Indeed, some recent studies realized the need of defining broader and more informative profiles [Bellogín et al., 2014]. However, many efforts on user modeling are still simplistic and do not take into consideration some relevant characteristics of the user behavior [Li and Kim, 2004].

Several works in RSs restrict themselves to propose increasingly complex user behavior models, such as LDA or tensors, to represent all signals of past user prefer-

ences [Xu et al., 2008; Ricci et al., 2011]. For instance, Jung et al. [2005] presented a preference model using mutual information in a statistical framework aligned with a method that combines joint features to alleviate data sparsity. Also, Liu and Jiang [2011] proposed a probabilistic matrix factorization (IPMF) algorithm that explicitly models user and item rating bias via Gaussian distributions. Such methods are usually based on content attributes from the items, such as price, or on rating information. Besides these content attributes, we assume that systematic preferences of each user may be correlated to non-content attributes derived from metadata or consumption, such as popularity. As such correlation is not modeled explicitly by RSs, we hypothesize that they are under-captured in practice. Thus, we model explicitly these correlations through CB models that are combined with traditional CF methods, in order to exploit complementary information available in the consumption data.

Specifically, we take into account three non-content attributes. The first one is the consumption popularity of each item. Popularity is a relevant type of information, since recommendation domains define scenarios of skewed consumption, where few items become popular whereas most of the remaining ones are never consumed [Anderson, 2006b; Fleder and Hosanagar, 2007]. Levy and Bosteels [2010], for instance, observed that RSs tend to reinforce popular artists, at the expense of discarding less played songs, in the Last.fm system. This conclusion was even reinforced by Yin et al. [2012], where the authors found that RSs are more prone to recommend popular items. In turn, Jambor and Wang [2010] argued that when recommending long tail items², we must take into account personalized demands of users. In this sense, the authors proposed a method that issues popular items to users interested in these items, while it provides alternative choices for users who are more prone to consume unpopular ones. Analogously to this last work, we take popularity as an item feature that should match individual preferences of each user.

The second non-content attribute is the mean pairwise similarity of all items consumed by each user. Almost all recommender systems exploit somehow the concept of similarity among users or among items [Pennock et al., 2000]. The recommendation idea is based on the premise that users trust like-minded user recommendations and interests or that future user behaviors and interests would be similar to past ones [Yu et al., 2004]. Otherwise, it would be hard to perform predictions. Hence, the search for information is biased to similar behaviors. Adomavicius and Tuzhilin [2005] found that most state-of-the-art CF methods consider solely information provided by the most similar individuals to each person in order to issue recommendations.

²Long tail items are a large number of distinct items, each one with a relatively small consumption demand [Anderson, 2006a; Park and Tuzhilin, 2008].

Also, [Herlocker et al., 2002] conducted extensive analyses about the impact of similar neighbors on RSs. None of these works, however, measure to what extent the consumption behavior of each user is affected by similar items. By taking similarity as a non-content preference, we assume that each user has individualized demands on consuming a less or more diversified set of items.

Finally, the third non-content attribute refers to the mean recency of the items consumed by each user. By recency we mean how long an item is available in a domain. Recent studies try to characterize items and users along distinct moments, defining temporal contexts that are able to determine local behaviors or needs. Xiang et al. [2010] and Yang et al. [2012], for example, distinguished short and long-term recommendations, arguing that the global needs of users would differ from some instantaneous needs. Analogously, Cebrián et al. [2010] defined ‘microprofiles’ that capture the concrete time situation of each user request. Again, we assume that such ‘local needs’ represent a feature inherent to individual preferences. Whereas some users are more interested in consuming recent items, other users would like to consume old ones or items belonging to a specific period of time (e.g., songs from ’80s). Through non-content attributes, we explicitly quantify this information and incorporate it into user behavior models.

2.6 Summary

We started this chapter by formally presenting the general recommendation problem, as well as its inherent tasks. Besides proposing recommendation as the problem of finding a set of items that maximize an individual utility function, we restrict this dissertation scope to the Top-N recommendation task, being concerned with ranking accuracy.

Later, we discussed the main quality requirements related to our main goal of enhancing the discovery of items potentially relevant for each user in RSs. This goal is intrinsically related to the challenge of fulfilling simultaneously **usefulness**, **diversity** and **novelty**. A main contribution of this dissertation relies exactly on pointing out a new direction for such achievement. Our proposal differs from previous ones by exploiting two types of information (i.e., forgotten re-consumable items and non-content attributes) that may affect these three requirements. Also, aiming at broader assessments of user experience, besides pure algorithm evaluations through these requirements, we adopted live experiments with real users to evaluate the proposed methods.

Next, we reviewed the main methods belonging to the three classes of RSs most related to this work, namely Content-based (CB), Collaborative Filtering (CF) and Hybrid Methods. This dissertation proposes a new hybrid method that exploits

two opportunities for enhancements unaddressed in the literature. This new hybrid method combines CB scores with CF scores to take into account simultaneously the long-term history (i.e., the Oblivion Problem) and implicit signals of preference (i.e., non-content attribute preferences) of each user. To the best of our knowledge, this work is the first effort that effectively exploits these two issues.

Thereafter, we summarized the main works related to temporal dynamics in RSs, which are the set of studies most related to the Oblivion Problem. We exploit temporal dynamics through a new strategy that enhances diversity in RSs by using the subset of items consumed by users in the remote past. In this sense, we evaluate distinct strategies to identify and recommend properly forgotten items. These strategies are based on ACT-R (Adaptive Control of Thought-Rational), a cognitive architecture that models the human memory. Despite psychology studies already being used in the recommendation context, this is the first work where such models are used for evaluating forgetfulness in this context.

Finally, we presented the set of studies about user behavior modeling most related to the non-content preferences. The main hypothesis with respect to non-content attributes is that RSs may under-capture them, since RSs do not model explicitly such information. Thus, we model explicitly these correlations through CB models that are combined with traditional CF methods, in order to exploit complementary information available in the consumption data. Specifically, this work takes into account three non-content attributes: consumption popularity, similarity among items and recency of each item in the system. By taking these attributes as a non-content preference, we assume that each user has individualized demands related to each of them.

Chapter 3

Forgotten Re-consumable Items

This chapter confirms our first working hypothesis: *State-of-the-art RSs fail to bring back items consumed long ago that are potentially relevant for users nowadays*. In this sense, first, we formally define the Oblivion Problem. Next, we discuss the motivations to raise and address this new problem. Later, we present two perspectives for the Oblivion Problem based on a straightforward distinction between the target user for recommendation and the user, or set of users, who have forgotten the items. Then, we discuss the scope of relevance for this problem and present a characterization methodology to assess its relevance on real domains. Next, we apply the proposed methodology to real domains and conduct traditional offline evaluations of RSs to acquire further understanding about the information we used to model forgetfulness. Finally, we summarize the main concepts, methods and conclusions discussed in this chapter.

3.1 Problem Definition

3.1.1 The Oblivion Problem

We start by defining **forgotten items** as any item that a particular user u , or subset of users, have consumed long ago and it is unlikely to be remembered or recommended to u at a recent moment t . Note that ‘be remembered’ and ‘be recommended’ are distinct concepts. While the former refers to cases where u by himself/herself, or affected by any factor external to the system, remembers an item, the latter comprises cases where the system, based on the RS or interface design, explicitly displays an item to u . These concepts raise two main issues: **(1) how to measure the probability of an item be remembered by u at t ?**; **(2) how to measure the probability of an item be recommended to u at t ?**

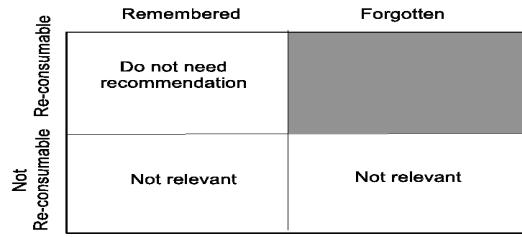


Figure 3.1: Representation of the item space along two dimensions: re-consumption and forgetfulness. We are interested in recommending the items that fit in the shaded area.

We also define **re-consumable items** as the subset of items that u have consumed and would like to re-consume at a moment t . Thus, an item might be forgotten but not necessarily re-consumable at a given moment. Furthermore, the subset of re-consumable items may include items that were not necessarily forgotten. The main issue in this case is *how to measure the probability of a user re-consuming an item at t ?* Figure 3.1 depicts the relation between forgotten and re-consumable items. Note that we do not need to concern with remembered and not re-consumable items, since users are not interested in them anymore. Further, it is not necessary to propose RSs for remembered and re-consumable items because users can reach these items by themselves. We are really concerned with items that are forgotten at a given moment, which is the main difference between this work and typical music streaming RSs [Song et al., 2012]. Observe that while it is desirable to rescue forgotten and re-consumable items, we cannot recommend forgotten but not re-consumable ones. We state this compromise as the Oblivion Problem:

Definition 1. *The **Oblivion Problem** is the problem of recommending the subset of forgotten items that are also re-consumable for a target user u at a given moment t .*

Addressing the Oblivion Problem involves two main steps:

1. Set up a model to predict whether an item is likely to have been forgotten at a given moment.
2. Set up a model to predict which forgotten items should be recommended because they are most likely to be re-consumable at a given time.

Although we may use the same information to set up both models, these steps cannot be merged into a single one since they represent non-aligned goals. For instance, whereas forgotten items tend to be those less correlated to items currently consumed by each user, re-consumable ones need to be more correlated to these items. This work proposes and evaluates distinct modeling strategies.

3.1.2 Motivation

The motivation for addressing the Oblivion Problem stems from the observation that, in many domains, users are particularly interested in re-consuming items. Our main hypothesis, however, is that there are scenarios where current RSs fail to bring back enjoyed items consumed long ago.

We point out at least three distinct scenarios that support this assumption. Aiming to depict the first one, suppose a user u has watched and enjoyed a movie m 10 years ago and eventually forgot it. Also, since re-consumption in this domain happens occasionally, new items emerge and user taste evolves, m will not be showed again to u , disregarding whether u would like to re-consume it or not. In this case, m remains forgotten because RSs assume that recent data is more relevant and will not display m anymore to u . The second scenario happens when users explicitly say to RSs that they do not want to consume an item. For instance, in music domains, popular and enjoyed songs are played many times in short periods of time. A user u may become ‘saturated’ soon and, by removing an item s from his/her playlist, says explicitly to RSs that s is no longer interesting. However, one year later, u would like to re-consume s , but most of the RSs believe that u still does not like s . The problem in this case is related to rating expiration. The final scenario arises when RSs automatically learn that users do not like an item. Suppose that u has been to Vegas two years ago and, since that time, RSs have been issuing to u deals of Vegas but he/she has never answered. RSs figure out that u is not interested in coming back to Vegas and stop recommending it. However, one year later, u is again willing to visit Vegas, but RSs still believe that Vegas is not a good recommendation. We say it is a scenario of wrong model consolidation.

Finally, we highlight that forgotten re-consumable items are promising for our goal of enhancing discovery in RSs due to three main reasons. First, the past relevance of forgotten items comprises useful evidence about the user taste. Second, some forgotten items may still be related to the current taste of each user. Third, the probability of a user to reach forgotten items within the system is small.

3.1.3 Perspectives of Oblivion

Based on the definition of forgotten items, presented in Section 3.1.1, we raise an important question: Who forgot the items? We have at least two different answers for this question, defining two distinct perspectives for the Oblivion Problem.

From an individual perspective, we provide to each target user u the items forgotten by himself/herself. Thus, we should evaluate the relevance of the Oblivion Problem individually, taking into account only the consumption history of each user

u . Hence, an item i might become relevant at different moments for distinct users, or even not be relevant for others. Rescuing forgotten items, in this case, improves the ability of personalizing recommendation services.

A second perspective is the collective one, in which items become forgotten by a subset of users and we aim to recommend these items to a target user u who have never consumed them. In this case, we define the relevance of the Oblivion Problem taking into account the aggregated consumption history of subsets of users. For instance, by considering the aggregated consumption of the X -nearest neighbors of a target user u , we can recommend novel items to u that were forgotten by his/her neighbors.

Furthermore, the collective perspective allows defining items globally forgotten by a domain, taking into account all users. Globally forgotten items may be useful to rescue “classic” items, previously desirable, but no longer consumed. Such items might interest new users in the domain.

3.2 Scope of Relevance

A fundamental concern with the Oblivion Problem refers to its applicability. We are not assuming that this problem is relevant for all domains. For instance, in book recommendations, hardly a user is interested in a book he/she has read and enjoyed in the past. Given a subset S of users from a domain ($S \subseteq U$), we measure the relevance of the Oblivion Problem for S at a given moment t through three properties:

- **Property 1:** The average probability that S at t forgets items consumed long ago;
- **Property 2:** The average probability of items issued by an RS to S at t comprise only items non-forgotten by S ;
- **Property 3:** The average probability of items consumed long ago to be re-consumable to S at t .

We state that the Oblivion Problem is as relevant as the smallest property value. For example, the Oblivion Problem has low relevance for systems that exhibit all consumed items to each user, helping them to not forget items (i.e., present low value for **Property 1**).

Based on these properties, we propose a characterization methodology that quantifies the relevance of the Oblivion Problem in real domains. Our methodology is based on straightforward answers to five main issues:

1. *What kind of average?*

We select the median since it is more robust to outliers, besides being more informative in the absence of further knowledge about a distribution.

2. *What do we mean by ‘consumed long ago’?*

For simplicity, we say that an item ‘consumed last time long ago’ is a bygone one. We assume that an item becomes bygone when a population is not willing to consume it anymore. In this work, we approximate the minimum time interval T_{bygone} necessary to any item become bygone as the median time to popular items become non-popular in a domain. The premises are that (1) popular items require a time interval longer than non-popular ones to become bygone; (2) popular items become bygone when they are no longer popular. Formally, T_{bygone} is given as follows:

$$T_{bygone} = \overline{T_{pop}} + 1 + \overline{T_{npop}}$$

where, $\overline{T_{pop}}$ refers to the median time interval between the first occurrence of popular items in a domain and the moment they become popular; $\overline{T_{npop}}$ is the median time interval to popular items become back non-popular. An item i is popular at a given moment l if i is for the first time in the head of a distribution of items ordered decreasingly by the number of distinct users who have consumed them at l (i.e., a consumption distribution). In turn, a popular item i becomes non-popular at the first moment $c > l$ in which i is in the tail of the consumption distribution defined at c .

3. *How to measure the probability that S forgets bygone items at t ?*

We approximate this probability using a well-known psychological model of the human memory, named ACT-R (Adaptive Control of Thought-Rational) [Anderson and Schooler, 1991; Pavlik and Anderson, 2005]. ACT-R models memory as a network in which specific nodes are activated (i.e., are remembered) at a given moment and the activation spreads to linked nodes. Further details about this model are presented in Section 3.3.1. Thus, for each item deemed as bygone at t , we define its probability of being remembered by S as the *Probability Retrieval equation*¹ proposed by ACT-R. Then, we sorted all probabilities and retrieve the complement of the median value.

¹The Probability Retrieval Equation (*Prob*) is defined as $Prob = 1/(1 + e^{(-A_i - \tau)/s})$ [Mellon et al., 2007], where A_i is called Activation Score of information i ; τ is a threshold that distinguishes activated information from the rest; s is the noise level of the model and embeds uncertainty in it.

4. *How to measure the probability of an RS issue to S at t only items non-forgotten by S?*

Distinct methods recommend different subset of items, which may or may not include bygone items. Thus, measurements on this issue require us to provide a recommendation list for S at t , based on a specific method, and evaluate the probability of S remember of each recommended item. Again, we use the *Probability Retrieval equation* of each item. Then, we consider each item with retrieval probability higher than a threshold τ as non-forgotten by U . Items not consumed by S are also deemed as non-forgotten ones. We approximate the desired probability by the percentage of recommended items deemed as non-forgotten.

5. *How to measure the probability of bygone items to be re-consumable to S at t?*

We approximate this probability by the observed percentage of re-consumption over time. For each item i deemed as bygone at each time unit $l \leq t$, first, we determine the temporal distance Δ between the last moment i was consumed by S before l . Then, we calculate the percentage $RecRate_{i,\Delta}$ of users in the domain who have re-consumed i within a period of time with size at least equal to Δ . Next, we determine the mean $MeanRecRate_{\Delta}$ of the $RecRate_{i,\Delta}$ values found for all bygone items i with temporal distance Δ . We also calculate the percentage $PercentageRecItems_{\Delta}$ of distinct items that have been re-consumed within a temporal distance Δ by at least one user. Thus, the probability of re-consumption associated to each item i at each moment l is defined as the product $PercentageRecItems_{\Delta} \times MeanRecRate_{\Delta}$. Finally, we sort these probabilities and retrieve the median value.

This methodology does not take into account how the system display the items to each user, since it is difficult to quantify automatically this kind of information. Thus, a more complete analysis of relevance would, in some cases, require an inspection of the system's interfaces and whether or not the whole history of each user is exhibited to himself/herself. We evaluate the usefulness of this methodology by contrasting its results on real domains wherein the Oblivion Problem could be promptly recognized as relevant or not by common sense.

Besides a quantitative analysis on the scope of relevance of the Oblivion Problem, a qualitative analysis would reveal important aspects that describe this problem, helping to point out how it emerges over time. Appendix A presents a preliminary study wherein we discuss some existential conditions related to the Oblivion Problem.

3.3 Addressing the Oblivion Problem

As discussed in Section 3.1.1, addressing the Oblivion Problem involves two distinct steps. In the following subsections, we discuss the main issues related to each step and present intuitive strategies to perform them.

3.3.1 Identifying Forgotten Items

The key concept for identifying forgotten items is the **memory retrievability** (*mem-ret*) of items at each moment. In recommendation domains, we define $mem-ret(u, i, t)$ as the chances² of a user u to retrieve (i.e., remember by himself/herself) at a given moment t a specific item i from the set of all items consumed by him/her. The smaller the *mem-ret*, the higher the chances of i be forgotten by u at t . We propose four distinct strategies to quantify *mem-ret* based on intuitive information used by a well-know cognitive architecture for memory modeling.

3.3.1.1 Less Recently Accessed (LRA)

A simple assumption is to consider that items consumed long ago have smaller *mem-ret* than recently consumed ones. Although it does not hold for all items, we expect that most of the recently consumed items may be remembered by a user. Thus, we define $mem-ret(u, i, t)$ as shown by Equation 3.1, where $T_{u,i}$ denotes the set of all distinct moments $l < t$ in which u consumed i . In this equation, a small difference between the test moment t and each moment l defines a high *mem-ret* score, due to the negative exponential. We sum the logarithms of the differences instead of the difference values themselves in order to smooth large values.

$$mem-ret(u, i, t) = \sum_{l \in T_{u,i}} [\log(t - l)]^{-1} \quad (3.1)$$

3.3.1.2 Less Frequently Accessed (LFA)

Another intuitive assumption is that less frequently consumed items or with small utility $f(u, i)$ are less relevant for a user and consequently, over time, exhibit small *mem-ret*. In this case, we define $mem-ret(u, i, t)$ as the sum of the utility $f(u, i, l)$ assigned to i at each time $l < t$ it was consumed by u , as shown by Equation 3.2. Such as discussed in Chapter 2, $f(u, i, l)$ indicates to what extent u liked i at l . In practice,

²We use the general term ‘chances’ instead of ‘probability’ because our strategies to quantify *mem-ret* do not define true probabilities (i.e., do not sum up to 1).

this utility refers to the frequency u consumed i or the rating explicitly assigned by u to i in scenarios where there is no re-consumption. Again, $T_{u,i}$ is the set of all distinct moments $l < t$ in which u consumed i . Since we assume that $f(u, i, l)$ is a positive value higher than or equal to 1, in order to avoid zeros in the $\log(f(u, i, l))$, we add one to the utility value before applying the logarithm.

$$mem-ret(u, i, t) = \sum_{l \in T_{u,i}} \log(f(u, i, l) + 1) \quad (3.2)$$

3.3.1.3 Less Correlated to Current Context (LCCC)

Considering the set of items consumed by u during his/her c most recent training moments as his/her current context $C_{u,c}$, we expect that items less similar to this context present small $mem-ret$. This assumption stems from the observation that the human memory is associative by nature, which means we find items mostly based on associations with items currently available [Mellon et al., 2007]. In this sense, Equation 3.3 defines $mem-ret(u, i, t)$ as a weighted sum of the associations between i and each distinct item j belonging to the current context $C_{u,c}$. We measure each association through the conditional probability of i given j in the whole training set, divided by the occurrence probability of j . This is the classical Data Mining definition of *confidence* that measures the co-occurrence probability of two items [Hipp et al., 2000]. In Equation 3.3, we define the utility of each item $j \in C_{u,c}$ as the sum of all its utility values $f(u, j, l)$ assigned by u at each moment $c \leq l < t$. Also, we normalize the utility of j by the maximum utility value $max_u(C_{u,c})$ found among items of C_u .

$$mem-ret(u, i, t) = \sum_{j \in C_{u,c}} \frac{\sum_{l=c}^t f(u, j, l)}{max_u(C_{u,c})} \times \log\left(\frac{prob(i | j)}{prob(j)}\right) \quad (3.3)$$

3.3.1.4 A Cognitive Architecture for Memory Modeling (ACT-R)

The three foregoing strategies are simultaneously considered in a well-known psychological model of the human memory, named ACT-R (Adaptive Control of Thought-Rational) [Anderson and Schooler, 1991; Pavlik and Anderson, 2005]. ACT-R considers that information is stored in our long-term memory in a web-type pattern with concepts linked to each other by association. For instance, while watching the movie ‘Inception’, users would remember of ‘Titanic’ since the protagonist in both is Leonardo DiCaprio. Further, ACT-R assumes that the retrieval of a specific information involves an ‘activation process’. Specific nodes of the network are activated (i.e.,

are remembered) and the activation spreads to the linked nodes. ACT-R states that three main factors affect this activation process.

The first factor is the temporal interval between the current moment and the last time a piece of information was accessed, defined as *Retention Function*. The larger this temporal interval, the harder it is to activate a piece of information again. The second factor is the frequency of consumption of a piece of information in the past. The higher this frequency, the higher the chances of a piece of information be activated. In psychological studies, such factor is commonly referenced as *Practice Function* [Anderson et al., 1999]. Note that the Retention and Practice Functions are related to the LRA and LFA strategies, respectively. The third factor refers to the current context, that is, the set of information currently ‘consumed’ by a person. Given the associative nature of human memory, context is sometimes more important than the past history [Mellon et al., 2007]. Thus, the higher the association between a piece of information and the current context, the higher its chances of being activated.

Based on these assumptions, distinct activation process models were proposed, refined and validated for the human learning process [Mellon et al., 2007]. Through these models, it is possible to define the chances of each known information be activated or not. Our fourth strategy uses one of these models to define $mem-ret(u, i, t)$, such as given by Equation 3.4.

$$mem-ret(u, i, t) = \log \left(\sum_{l \in T_{u,i}} [t - l]^{-d} \right) + \sum_{j \in C_{u,c}} \frac{\sum_{l=c}^t f(u, j, l)}{max_u(C_{u,c})} \times \log \left(\frac{prob(i | j)}{prob(j)} \right) \quad (3.4)$$

Equation 3.4 modifies the original formula defined in the human learning context [Anderson et al., 1999] by normalizing the utility of each item belonging to the context $C_{u,c}$. The Retention and Practice functions are simultaneously addressed in the first part of this equation. The context information is assessed in the second part. Again, $T_{u,i}$ refers to all distinct moments that u consumed i ; $C_{u,c}$ denotes the set of distinct items consumed by u during his/her c most recent training moments; $f(u, j, l)$ is the utility value assigned by u to j at the moment l ; $max_u(C_{u,c})$ is the maximum utility value found among items of $C_{u,c}$; d is the parameter for memory decay over time. Through empirical evaluations, the value 0.5 has emerged as d ’s default value in several applications [Mellon et al., 2007].

3.3.2 Recommending Forgotten Re-consumable Items

Once the forgotten items have been identified, we need to distinguish re-consumable ones from the rest. Given user taste shifts, as well as the changes in the system as a whole, not all forgotten items remain re-consumable, and consequently useful for recommendation. The key concept to identify re-consumable items is the **relevance score** of the items at each moment. The higher the score related to an item, the higher its probability of being a re-consumable item. We propose four strategies to define the relevance score of each forgotten item. The goal is to recommend for each user only the Top N items, deemed as the re-consumable ones.

3.3.2.1 Context Aware Recommendation

A simple strategy is to consider that the higher the association of a forgotten item i with the current context $C_{u,c}$ of a user u , the more relevant i is for u . As done by LCCC in Section 3.3.1, the final relevance score assigned to i for u at the moment t is given as the weighted sum of the association of i with each distinct item j belonging to $C_{u,c}$, as shown by Equation 3.5. Note that we define this score exactly like the *mem-ret* in Equation 3.3. However, while we are interested in the highest score values, we aim at the lowest *mem-ret* values. Thus, we sort the items in descending order by this score and recommend the Top N items.

$$score(u, i, t) = \sum_{j \in C_{u,c}} \frac{\sum_{l=c}^t f(u, j, l)}{max_u(C_{u,c})} \times \log \left(\frac{prob(i | j)}{prob(j)} \right) \quad (3.5)$$

3.3.2.2 Temporal Distance Recommendation

Our second strategy is based on recommending items that have been forgotten for long intervals. We assume that users are more willing to re-consume items that they have consumed long ago. The longer the period an item has been forgotten, the higher its final relevance score. We define the score of each item i as the interval between the test moment t and the most recent moment that u consumed i in the training set. Then, we sort the items in descending order by this score and recommend the Top N items.

3.3.2.3 Traditional RS

Another straightforward strategy is to determine the score of each forgotten item using traditional RSs. In this sense, we use the set of identified forgotten items as input for the *UserKNN* method. *UserKNN* will define the K-nearest neighbors of the target user u and derive for each forgotten item i a score based on the mean score assigned to

it by the neighbors of u in the training set. Then, the items are sorted in descending order by such score and the Top N items are issued. We implemented our version of *UserKNN* using the Cosine measure as similarity function, such as presented in [Adomavicius and Tuzhilin, 2005]. This version also incorporates the sample bias regularization approach, with the original parameters, proposed in MyMediaLite [Gantner et al., 2011].

3.3.2.4 ACT-R Based Recommendation

Finally, we define the relevance score taking into account simultaneously the three main factors modeled by ACT-R, such as shown by Equation 3.6. As originally stated, we consider that the higher the practice in the past (second part of the equation) and the higher the association with the current context (third part of the equation), the more relevant an item is for u . However, we assume that users are more willing to re-consume items that they have consumed long ago. Thus, we take into account the retention function in a different way (first part of the equation). Instead of defining the retention function using the test moment t as basis, we consider the first moment f that u consumed any item in the training set. Thus, items consumed in the long-term history have a higher relevance than items consumed recently. Again, we sort the items in descending order by this score and recommend the Top N items.

$$\begin{aligned} score(u, i, t) = & \log \left(\sum_{l \in T_{u,i}} [l - f]^{-d} \right) + \log \left(\sum_{l \in T_{u,i}} \frac{f(u, i, l)}{t - l} \right) + \\ & \sum_{j \in C_{u,c}} \frac{\sum_{l=c}^i f(u, j, l)}{\max_{u(C_{u,c})}} \times \log \left(\frac{prob(i|j)}{prob(j)} \right) \end{aligned} \quad (3.6)$$

3.4 Case Studies

Through offline analysis, we cannot verify the effectiveness of our strategies in identifying forgotten re-consumable items. A proper evaluation would require asking explicitly to users whether or not they remember and would like to consume each recommended item. Nevertheless, offline analyses are still relevant to acquire further understanding about the information we used to model forgetfulness (step 1) and re-consumption (step 2). In this section, we discuss the necessary changes on traditional experimental design and metrics adopted in the literature to achieve such understanding, as well as our main findings.

3.4.1 Datasets

Aiming to conduct assessments on the scope of relevance and offline analysis on Top- N recommendations of forgotten re-consumable items, we used four distinct real data collections. As the first collection, we employed data collected from the *LastFm* system³, which is a UK-based Internet radio and music community website. This sample was collected through an API provided by *Last.fm*⁴. Our second dataset, ML-10M, comprises rating data samples from MovieLens⁵, gathered and made available for research purposes by GroupLens Research. As the third collection, ML-Tags, we chose a sample of tag assignment data from the MovieLens dataset. In this dataset, each user assigned a set of tags to each watched movie. Finally, we used the well-known Netflix dataset from movie domain as our fourth dataset [Bennett and Lanning, 2007]. *Netflix*⁶ is an online rental movie service that made available, for research purposes on recommendation, a dataset with information about its movies and users. Table 3.1 summarizes the main features of the evaluated datasets.

Table 3.1: Dataset information.

	LastFm	ML-10M	ML-Tags	Netflix
# Users	35,000	72,000	4,000	480,189
# Items	4 million	10,000	15,260	17,770
# Actions	85 million	10 million	95,580	100 million
# Time	281 weeks	671 weeks	157 weeks	310 weeks
Type	play count	rating	binary	rating
Domain	songs	movies	tags	movies

3.4.2 Analysis of Scope

We quantified the relevance of the Oblivion Problem in our datasets by applying the characterization methodology presented in Section 3.2. The idea is to contrast the results found by this methodology against the common sense and further knowledge about each domain, acquired by previous works [Song et al., 2012; Resnick and Varian, 1997]. We expect the Oblivion Problem to be relevant for music domains, since users tend to listen to their favorite songs many times. Also, given the huge number of available songs, we do not expect users would remember songs they used to enjoy long

³<http://www.last.fm/>

⁴<http://www.last.fm/api>

⁵<http://movielens.umn.edu>

⁶<http://www.netflix.com>

ago. The Oblivion Problem might be relevant for tag domains as well, since this kind of domain exhibits strong skewness towards recency and popularity, prioritizing to display in the system popular and recently used tags [Golder and Huberman, 2006]. Further, as tag re-consumption is the primary goal of this domain, some forgotten relevant tags might become relevant again over time. Finally, we expect movie domains to be less adherent to the Oblivion Problem. Users are more willing to watch new movies than rewatch known ones.

Tables 3.2 and 3.3 present our methodology’s results for each dataset, considering individual and collective perspectives discussed in Section 3.1.3, respectively. We derived the values related to ‘Property 2’ in these tables using the *UserKNN* method, as described in Section 3.3.2.3. We set $K = 80$ for this algorithm in our experiments, which is the default value used by MyMediaLite. Analyses on the individual perspective require defining only the parameters related to ACT-R. For all datasets, we used the default parameter values of ACT-R established in the literature [Lebiere, 1999]. That is, we set the decay factor $d = 0.5$ and the noise level $s = 0.25$. [Lebiere, 1999] argued that, while varying parameter values within a reasonable range will result in different quantitative predictions, ACT-R’s qualitative predictions are left unaltered. Thus, in order to set the parameter τ , we varied by 0.5 its value from -5.0 to -2.0 (the default search range). We set τ as the value that activates⁷ the highest number of items consumed by each user in his/her current context, which we defined as the last week of each user history. Additionally, analyses on the collection perspective require consolidating an aggregated consumption history for each target user u by merging the history of his/her X -nearest neighbors as a multiset. Aiming to avoid huge aggregated histories, we set X equals to 10 and selected the 10-nearest neighbors using again the *UserKNN* method.

Table 3.2: Analysis of relevance for the individual perspective of the Oblivion Problem.

	LastFm	ML-10M	ML-Tags	Netflix
Property 1	0.1856	0.8181	0.0329	0.6236
Property 2	1	1	1	1
Property 3	0.1186	0	0.0220	0

Results related to the individual perspective show that the Oblivion Problem is more relevant for LastFm, followed by ML-Tags and by the collections related to movies. The probability of re-consumption is zero for movies in our datasets. Also, we observed low probability of forgetfulness and re-consumption in the ML-Tags dataset.

⁷According to ACT-R, an item is activated whether its activation score is higher than τ [Mellon et al., 2007].

Table 3.3: Analysis of relevance for the collective perspective of the Oblivion Problem.

	LastFm	ML-10M	ML-Tags	Netflix
Property 1	0.0817	0.1544	0.5252	0.1198
Property 2	1	1	1	1
Property 3	0.4932	0.2921	0.0225	0.3569

We explain these values by the particular characteristics of the evaluated tagging sample. Table 3.1 points out this sample as the collection with the least amount of actions, provided by few users about many distinct tags. Hence, each user has a short profile with no reuse of tags, which results in low re-consumption probability. Since it is easy to remember each individual tag of a small set, we also found low probabilities of forgetfulness. A large sample of tag usage would reveal that users actually employ a larger number of tags, improving the probability of forgetfulness, and re-use more often a subset of tags that better represent their point of view, improving the re-consumption probability [Lipczak, 2008].

The collection perspective, on the other hand, shows that the Oblivion Problem becomes relevant even for movie domains. We observed an increment on the probability of re-consumption in all collections. Again, small increments observed in ML-Tags are related to the short and diversified profile of each user. Since MovieLens is characterized by short profiles and tags assigned by users are mostly user-specific (i.e., tags represent user points of view), profiles rarely overlap. Finally, we observed that in both perspectives a traditional *UserKNN* recommendation method does not rescue forgotten re-consumable items. It mostly issues to users unknown items. This result demonstrates our first working hypothesis for *UserKNN* methods. The proposed methodology allows us to extend such analysis to any traditional RS by simply adopting it to generate values for **Property 2**.

3.4.3 Experimental Design

We evaluated the recommendation of forgotten re-consumable items on the task of Top-N recommendations, considering the individual perspective of the Oblivion Problem. Thus, we intended to recommend for each user items that himself/herself has forgotten. As the individual perspective of the Oblivion Problem is relevant only for LastFm and ML-Tags, we restricted our analysis to these datasets. We analyzed how well the used information models forgetfulness through error measures that assess the capability of each proposed method to distinguish potentially forgotten items

from items deemed as non-forgotten. Next, we derived preliminary evidences about how ‘re-consumable’ are the recommended items by using the subset of items actually re-consumed in our datasets. In both cases, we proposed specific measurements more appropriate for evaluating forgotten re-consumable items, as described in the following sections. Finally, we investigated the diversity and novelty of recommended items. In this case, it is possible to properly evaluate diversity and novelty by offline analysis. We measured these two quality dimensions through a formal framework of analysis presented in [Vargas and Castells, 2011].

Our analyses employed the traditional training/test partition. We used 30% of the most recent weeks of each user’s history as test set and the remaining weeks as training set. We adopted a training/test partition instead of an *n-fold cross validation* since the latter would require a complex and careful design in temporally ordered data, in order to maintain the temporal properties of each dataset. Regarding parameter settings, most of the evaluated strategies do not require any parameter. Only strategies that exploit the information of context require us to define context as a parameter. For simplicity, we defined the context of each user as one single time unit, specifically, the most recent time unit of his/her training set (e.g., day, week, month – defined accordingly to each domain). Also, we set 80 as the maximum number of neighbors in the *UserKNN* method used to identify forgotten items that are re-consumable, which is presented in Section 3.3.2.3. This is the default value adopted by MyMediaLite to KNN-based RSs, since this value enabled proper results in many scenarios [Gantner et al., 2011].

3.4.4 Exploiting Forgotten Re-consumable Items

3.4.4.1 Identifying Forgotten Items

Although we cannot measure the accuracy of our strategies, it is possible to evaluate how often our strategies misidentify items as forgotten. We can define error measures to assess how well the used information distinguishes potentially forgotten items from a subset of non-forgotten ones. If an item i , consumed by a user u in the training set, was re-consumed in the test set, i was not forgotten by u during the time period spanning the test set. Hence, by assigning a *mem-ret* value for each item consumed by a specific user in the training set, and then ranking increasingly such items, according to *mem-ret*, the re-consumed items should appear at the end of this ordered list.

Based on this observation, we designed the following experiment. First, for each user u , we ranked his/her training items arranging all items not re-consumed in the test set ahead of the re-consumed ones. Then, these re-consumed items were sorted in ascending order by the utility value assigned to them by u . Moreover, we assigned to

each training item a *mem-ret* value using each strategy presented in Section 3.3.1 and generated another list sorted in ascending order according to *mem-ret*. Later, for each item re-consumed by u in the test set, we calculated the difference between its rank position in both lists. Aiming to make these differences comparable for distinct users, who have lists of distinct sizes, we normalized each value of rank difference by the list size. We repeated this process for all users and counted the frequency of occurrence of each rank difference. Finally, we derived the probability of occurrence of each difference and plot a probability distribution, such as shown by Figure 3.2. In these plots, a negative difference means that our strategies are assigning a *mem-ret* value smaller than the expected, arranging re-consumed items ahead of not re-consumed ones in the rank. We observed that LFA presented the best performance, exhibiting higher probabilities of rank differences close to zero in both datasets. All other strategies underestimated the *mem-ret* of re-consumed items. Thus, frequency or rating information, exploited by LFA, seems to be more effective to recognize re-consumed items as non-forgotten.

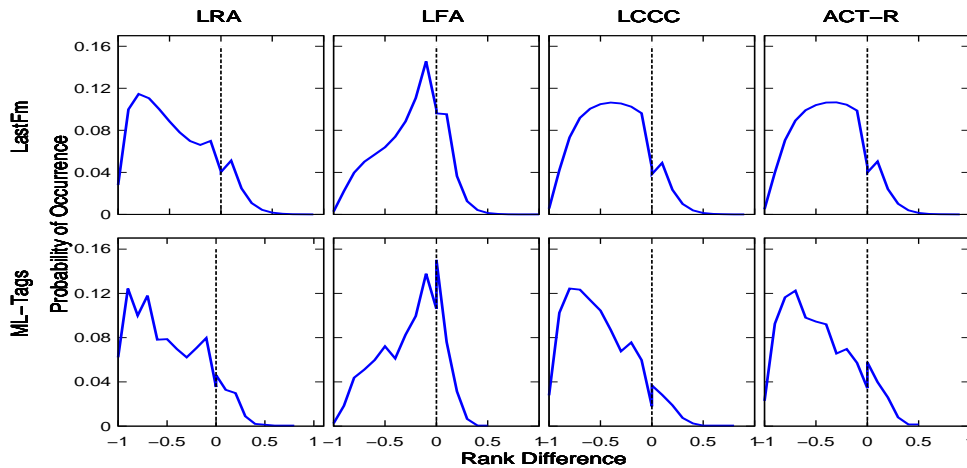


Figure 3.2: Probability distribution of normalized rank difference values between training items ranked by utility and training items ranked by *mem-ret*. A negative difference means that our strategies are assigning a *mem-ret* value smaller than expected for re-consumed items. Among all strategies, LFA presented higher probabilities of rank differences close to zero, which means that frequency or rating information is more effective to recognize re-consumed items as non-forgotten.

Further, we investigated whether there are strong correlations between the utility values assigned by users to each item re-consumed in the test set and the *mem-ret* value found for these items. We calculated the Pearson Correlation coefficient between the rank positions of each item in both lists generated for all users, shown by Table 3.4. First, almost all strategies presented significant correlation values (i.e., higher than 0.5) in both datasets. This fact reveals that the three types of information (recency, utility and association with a context) are useful to distinguish forgotten

items from re-consumed ones. Second, the information of past utility, used by LFA, presented the highest correlations, confirming that such information better predicts the current *mem-ret* of items. Since ACT-R did not exhibit gains against each of these types of information individually, other strategies for combining recency, utility and context association should be investigated. ACT-R is effective for describing learning processes, however, it must be adapted to model ‘consumption processes’ existing in recommendation domains.

In summary, we found that re-consumption is significantly correlated to three factors: information of past utility, recency of consumption, and association with current consumption. Further, past utility is more promising to differentiate re-consumed items from potentially forgotten ones. This result suggests that the most relevant items are less likely to become forgotten over time. Finally, other strategies to combine these three types of information should be investigated, since, individually, each type is correlated to the current utility of re-consumed items.

Table 3.4: Pearson Correlation coefficient between rank position of re-consumed items in the utility-based rank and the *mem-ret*-based one. The best values for each dataset are shaded.

	LastFm	ML-Tags
LRA	0.4891	0.4815
LFA	0.6596	0.7510
LCCC	0.5234	0.5452
ACT-R	0.5387	0.5587

3.4.4.2 Recommending Forgotten Re-consumable Items

This section has two distinct goals. First, we aim to raise preliminary evidences about how ‘re-consumable’ are the recommended items. Again, using the subset of items actually re-consumed in our datasets, we can derive meaningful analysis on the data used to model re-consumption. Second, we aim to evaluate the diversity and novelty of recommended items. Differently from re-consumption, we can evaluate diversity and novelty through offline analysis.

Analysis of re-consumption - We propose two distinct analyses based on items re-consumed in the test set. The first one refers to the *recall* of these items. Note that among the training items of each user, there are (1) forgotten items; (2) non-forgotten ones that users re-consumed in the test set; and (3) non-forgotten items that users did not re-consume, due to any reason. Given the whole training set as input, a recommender should assign high scores to the re-consumed items. This analysis shows

the capability of our proposals to identify items effectively re-consumed, but not to identify forgotten items potentially re-consumable. Hence, as the second analysis, we measure how ‘re-consumable’ the recommended items are, considering all users. We examine re-consumption by tallying the percentage of users who consumed an item once and then re-consumed it after x time units. We assumed that the higher this percentage, which we name re-consumption rate (*rerate*), the higher the probability of this item to be a forgotten re-consumable item after x time units. Thus, a recommender of forgotten items should present simultaneously high *recall* and *rerate* levels.

We evaluated the *recall* of re-consumed items through the following experiment. First, we used the whole training set of each user as input for each proposed recommendation strategy. Then, we retrieved as output recommendations of each strategy for each user u the Top- N items with the highest final scores. We set N as a variable value equals to 10% of the training set of u . We adopted a variable list size rather than a fixed one for all users (e.g., Top-100) since users present training and test sets with distinct sizes. Finally, for each user, we calculated the percentage of items re-consumed in the test set rescued by each strategy. In turn, we defined the *rerate* values as follows. After issuing recommendations using only the training set as input for each strategy, we merged the test and training data composing a single dataset D used only for this measurement. Then, we defined for each item $i \in D$ the percentage of users who have consumed and re-consumed it within x time units. Next, for each item r recommended to each target user u , we calculated the time interval t between the test moment of u and the last time unit in his/her training set that u consumed r . We assigned to r the *rerate* value previously calculated for the time interval t .

Table 3.5: Analysis of mean recall of re-consumed items and re-consumable rate (*rerate*) of recommended items. The best values for each metric are shaded. These results point out TempDistance as the most promising strategy for recommending forgotten items, since it presented the highest recall levels aligned with high *rerate* levels.

	LastFm		ML-Tags	
	<i>recall</i>	<i>rerate</i>	<i>recall</i>	<i>rerate</i>
CxtAware	0.1180	0.0827	0.0517	0.0164
TempDistance	0.1967	0.1015	0.1791	0.0499
<i>UserKNN</i>	0.0575	0.1171	0.0836	0.0473
ACT-R	0.1197	0.0826	0.0703	0.0273

Table 3.5 presents the mean *recall* and *rerate* values found for all users. The Temporal Distance strategy presented the best results with respect to both *recall* and *rerate* in both datasets. Besides recovering significant percentages of re-consumed items, this strategy identified items ‘more re-consumable’ by the whole set of users

in each collection. Further, ACT-R could not improve the individual information of recency, pointing out the approach used to weight and combine recency, utility and context diverges between learning and consumption scenarios. We also evaluated how *recall* and *rerate* vary according to the time interval since the last consumption of each item in the training set of each user, such as shown by Figures 3.3 and 3.4. As each dataset presents distinct number of time units, we normalized time intervals by the total number of time units of each collection, making the plots comparable. Taking into account *recall*, all recommendation strategies presented high values for items consumed long ago. This represents an important finding, since we focus on recommending relevant items that users have not consumed for a long time. On the other hand, *rerate* results showed that recommended items consumed recently are ‘more re-consumable’ by a whole set of users. Peaks on these plots, such as those observed on LastFm, are related to the underlying distribution of actions per time unit of each dataset.

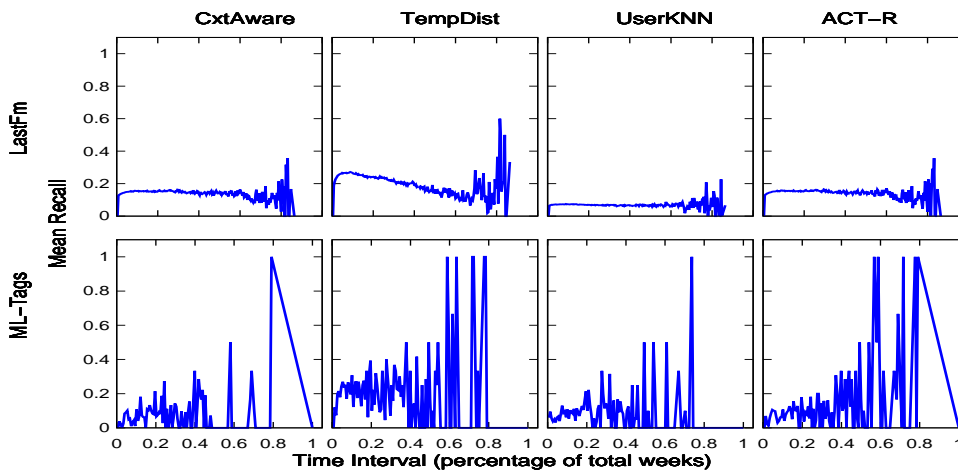


Figure 3.3: Analysis of mean recall for re-consumed items per time interval between the test moment and their last consumption in the training data. All recommendation strategies present significantly higher recall values for items consumed long ago (i.e., with larger time intervals).

This analysis demonstrates that recency of consumption allows us to recover a significant percentage of re-consumed items, while it provides forgotten items potentially ‘re-consumable’ by the whole user set of each domain. Further, all evaluated recommendation strategies are more effective in recommending items consumed long ago.

Diversity and Novelty - Although offline analysis does not assess the effectiveness of recommending forgotten re-consumable items, we still can evaluate other issues such as the diversity and novelty achieved by the proposed strategies. In this sense, we employed a formal framework of analysis presented in [Vargas and Castells, 2011].

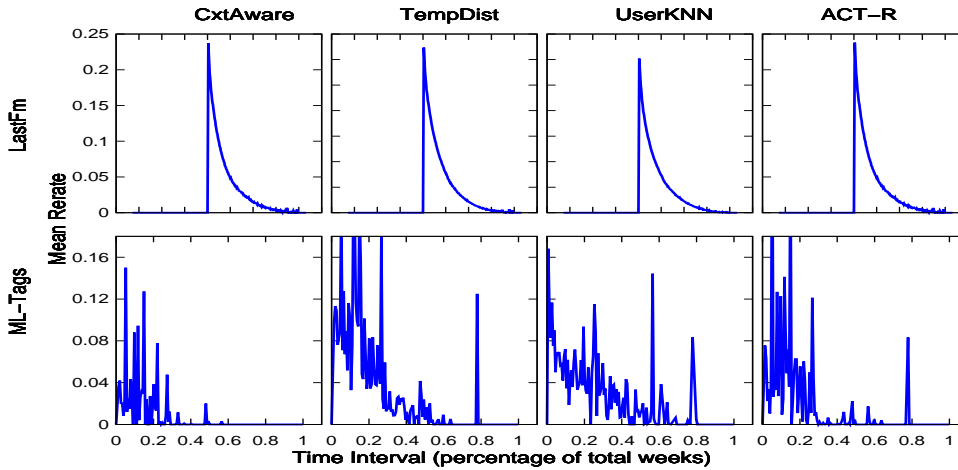


Figure 3.4: Analysis of mean rerate for recommended items per time interval between the test moment and their last consumption in the training data. Items consumed more recently (i.e., with smaller time intervals) tend to be more ‘re-consumable’ in both datasets.

Specifically, we used EPC and EILD to measure novelty and diversity, respectively, considering in both cases the discount function ($disc(K)$) equals to 0.85^{k-1} for relevance aware analysis and Pearson correlation as similarity distance measure [Vargas and Castells, 2011]. While EPC measures the complement of the expected popularity for the relevant recommended items, EILD generalizes the average intra-list distance with the introduction of rank-sensitivity and relevance. We applied these metrics to the output recommendations of each proposed strategy, whose size corresponds to 10% of the training set of each target user. Further, we proposed as a gold standard the diversity and novelty levels found in the test set of each user, since these levels were defined by the actual user behavior. Aiming to ensure that the gold standard has the same number of items that was recommended to each user, we randomly selected a subset of each user’s test set of size equals to 10% of his/her training set.

Table 3.6 presents the results for each metric. Considering diversity, strategies that use information about context association (i.e., CxtAware and ACT-R) presented higher values of diversity than the others. This result shows that, although both strategies focus on recovering items more similar to the current context of each user, the rescued items are more distinct from each other. Regarding novelty, TempDistance and *UserKNN* exhibited the best results in both collections, pointing out that these techniques rescue some unpopular items. As expected, novelty values related to the recommendations were smaller than those related to the gold standard, since we are recommending items already known by each user. However, we found meaningful values showing that forgotten items may be considered somehow novel in the present.

Table 3.6: Results on novelty and diversity metrics for different recommendation strategies. We take relevance and rank discount considering the discount with base 0.85, as done in [Vargas and Castells, 2011]. The best values for each metric are shaded.

	No relevance				Relevance			
<i>Metric</i>	LastFm		ML-Tags		LastFm		ML-Tags	
	EPC	EILD	EPC	EILD	EPC	EILD	EPC	EILD
CxtAware	0.9997	0.4174	0.9980	0.1969	0.0005	0.0071	0.0023	0.0002
TempDistance	0.9997	0.3842	0.9980	0.1743	0.0010	0.0063	0.0024	0.0002
<i>UserKNN</i>	0.9997	0.3971	0.9971	0.2000	0.0009	0.0060	0.0024	0.0002
ACT-R	0.9997	0.4175	0.9980	0.1940	0.0005	0.0071	0.0023	0.0002
Test set	0.9999	0.4202	0.9986	0.2640	0.0025	0.0151	0.0036	0.0005

3.5 Summary

We started this chapter by formalizing the main concepts related to the Oblivion Problem. Besides delimiting the differences between forgotten and re-consumable items, we formalized the **Oblivion Problem** as the problem of recommending the subset of forgotten items that are also re-consumable for a target user u at a given moment t .

Next, we depicted some scenarios where users are particularly interested in re-consuming items but current RSs may fail to bring back enjoyed items consumed long ago. These scenarios represent the motivation to raise and address the Oblivion Problem.

Later, based on the distinction between the target user and the set of users who forgot the items, we discussed the individual and collective perspectives of the Oblivion Problem. While addressing the individual perspective improves the ability of personalizing recommendation services, the collective one allows presenting to users items enjoyed in a remote past as novel recommendations.

Concerning with the applicability of the Oblivion Problem, we also discussed its scope of relevance on real domains. Based on properties inherent to the problem definition, we proposed a characterization methodology that quantifies the relevance of the Oblivion Problem in distinct domains.

Thereafter, we described intuitive strategies to address the two main steps related to the Oblivion Problem. The first step refers to predict whether an item is likely to have been forgotten at a given moment. In the second step, we need to predict which forgotten items should be recommended because they are most likely to be re-consumable at a given time. The proposed models for both steps are based on intuitive information used by a well-known cognitive architecture for memory modeling.

Finally, we evaluated the proposed methodology and models through offline

analysis considering four datasets. First, the methodology demonstrated our first working hypothesis. Also, we acquired further understanding about the information we used to model forgetfulness (step 1) and re-consumption (step 2). While past relevance is a more promising information for identifying forgotten items, recency of consumption allows us to recover high percentages of re-consumed and re-consumable items. Additionally, we found that besides enhancing diversity, the recommendation of forgotten re-consumable items may bring items deemed as novel in the present, since they are not consumed for a long time.

Chapter 4

Non-Content Preference Mismatching

This chapter evaluates our second working hypothesis: *State-of-the-art RSs fail to capture the whole extent on which implicit signals of preferences observed on past consumption relate to preferences observed on current consumption*. We start by defining formally non-content preference attributes and the *Preference Mismatching* metric. This metric quantifies how the recommendations provided by a given CF match the user previous consumption with respect to the values along non-content attributes. Also, we discuss the motivation to raise and address this working hypothesis. Next, we present a characterization methodology to evaluate the preference mismatching metric for any CF method in real domains. Later, we propose a method to build CB models by using non-content attribute information and combine these new models with existing CF methods to produce better recommendations. Then, we validate our working hypothesis and evaluate the proposed method in real domains. The chapter ends with a summary of the main conclusions and contributions related to non-content attributes.

4.1 Problem Definition

4.1.1 Non-Content Preference Attributes

A formal description of the Preference Mismatching requires, first, to define non-content preference attributes. A non-content preference attribute is derived from previous consumption data and quantifies a criterion by which an item might have been chosen previously. For instance, how long an item has been consumed in a domain or the item's popularity are non-content preference attributes available in almost all domains. Specifically, such attributes fulfill three requirements:

1. The consumption data are enough for allowing the computation of the attribute values;
2. There is a function that maps the represented criterion to a numeric spectrum of values, so that it is possible to assign a value to each item in the population;
3. The per user consumption must be related to a short range of values along the spectrum. In practice, we focus on attributes that may be exploited for sake of prediction. For instance, the last moment at which an item was consumed in a domain would not be relevant whether the user consumption disregards this information.

4.1.2 Preference Mismatching

Aiming to measure how well RSs capture a specific non-content preference attribute, we describe each item by $|D|$ non-content attributes and define the statistical measure *Expected Value* ($E(D_a)$) for any subset of items along each attribute D_a . In practice, we can approximate this expected value by the mean value observed in training data. Thus, we mathematically define user's preference in recommendation domains through Definition 2.

Definition 2. *The **user's preference** is a $|D|$ -dimensional vector that quantifies, for each attribute D_a , the relative difference between the expected value $E(D_a|C_u)$, given the subset C_u of items consumed by a specific user u , and the expected value $E(D_a|I)$, given the entire item set I .*

More formally, let $E(D_a|C_u)$ be the expected value of the set C_u for each $D_a \in D$. The preference $Preference[u, D_a]$ is given by Equation 4.1.

$$Preference[u, D_a] = \begin{cases} \frac{E(D_a|C_u) - E(D_a|I)}{E(D_a|I)}, & \text{if } E(D_a|I) \neq 0 \\ \lim_{E(D_a|I) \rightarrow 0} \frac{E(D_a|C_u)}{E(D_a|I)}, & \text{otherwise} \end{cases} \quad (4.1)$$

We assume that consumption is not random and users present a systematic preference on items from a specific range of values for distinct non-content attributes. Therefore, explicitly considering such non-content preference allows us to refine the identification of user interests. For instance, the information that a specific user's preference is towards old and unpopular items allows selecting a subset of items that better suit his/her interests.

Analogously, we define the per user recommendation's non-content description. Let $E(D_i|R_{\mathcal{A}_n, u})$ be the expected value of the subset $R_{\mathcal{A}_n, u}$ of items recommended by a

given method \mathcal{A}_n to the user u , along each $D_a \in D$. The description $Desc[\mathcal{A}_n(u), D_a]$ is the relative difference between the expected value $E(D_a | R_{\mathcal{A}_n, u})$ of the subset $R_{\mathcal{A}_n, u}$ and the expected value $E(D_a | I)$ of the entire item set I , such as given by Equation 4.2.

$$Desc[\mathcal{A}_n(u), D_a] = \begin{cases} \frac{E(D_a | R_{\mathcal{A}_n, u}) - E(D_a | I)}{E(D_a | I)}, & \text{if } E(D_a | I) \neq 0 \\ \lim_{E(D_a | I) \rightarrow 0} \frac{E(D_a | R_{\mathcal{A}_n, u})}{E(D_a | I)}, & \text{otherwise} \end{cases} \quad (4.2)$$

Such as expected for users, we assume that recommenders prioritize a specific range of values for each non-content attribute, since they are based on inductive premises that make some assertions about items or users. Based on this perspective, a relevant question concerns the match between user non-content preference and recommendation non-content attributes. Aiming to evaluate such match, we define a metric named **Preference Mismatching** that measures the difference between our mathematical definition of user's preference and the recommendation's non-content description. Figure 4.1 depicts the concepts of user's preference, recommendation's description and Preference Mismatching hereby introduced.

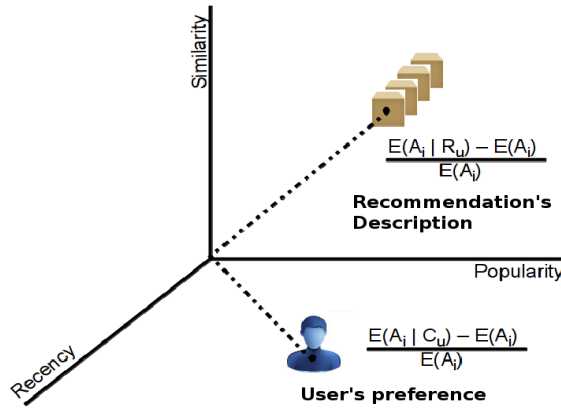


Figure 4.1: Visual Representation of Preference Mismatching along three non-content attributes (i.e., popularity, similarity and recency). Preference Mismatching quantifies the distance between the vector that represents user's preference and the vector that represents recommendation's non-content description.

Therefore, we assume that this difference can be calculated using different approaches and it is significant when its absolute value is higher than a minimum positive value ϵ that determines whether a recommendation description is similar to a user preference in a given domain. Therefore, a relevant hypothesis to be assessed is whether Preference Mismatching is usually significant in real domains.

4.1.3 Motivation

The motivation to evaluate and address Preference Mismatching stems from two observations: (1) non-content preferences might influence the behavior of some users; (2) state-of-the-art RSs do not guarantee the issued recommendations are aligned with such preferences. Current RSs do not exploit explicitly non-content attributes. CB methods consider only attributes related to the items, disregarding further information about consumption. Correlations modeled by these methods are restricted to the set of attributes explicitly used to describe each item. Hence, correlations involving non-content attributes remain hidden in the data. CF methods, on the other hand, use consumption information to model user behaviors. However, there is no evidence in the literature whether non-content attributes are somehow captured by CF models. Since consumption information is modeled by CF methods, we restricted our analyses about non-content attributes derived from consumption and metadata to this type of RSs.

4.2 Assessing Preference Mismatching

Aiming to verify the existence of non-content preference mismatching in real domains, we present a characterization methodology that answers some crucial questions:

1. Is the user consumption associated with a short range of values for each attribute D_a ?
2. What is the user's preference with respect to each attribute D_a ?
3. Does the user consumption present high variability on his/her individual preference for each attribute D_a ?
4. What is the recommendation's non-content description with regard to each attribute D_a ?
5. What is the preference mismatching with respect to each attribute D_a ?

We conduct analyses related to each of these questions on a set of transactions T , which comprises the transactional history of users in a domain. Further, we divide T into two disjoint sets, a training set T_a and a test set T_e , such that $T = T_a \cup T_e$. In all steps, we use only the test set T_e to calculate the expected values and all other measures related to each non-content attribute. In turn, we use the training set T_a to assign to each test item the values of each attribute D_a . For instance, the popularity inherent to each test item is defined as its prior popularity on T_a .

We evaluate the correlation between the user consumption and a given non-content attribute D_a through the *Normalized Standard Deviation (NSD)* defined for

each user u , as described in Equation 4.3, where $\sigma(D_a|C_u)$ denotes the standard deviation of D_a values in the set of consumed items C_u . This metric assumes that a non-content attribute provides predictive relationship with respect to user consumption whether the dispersion of values observed for the consumed items is significantly smaller than the dispersion observed in the whole spectrum of values. Thus, the smaller the NSD , the more the user consumption is correlated to a subset of values.

$$NSD(u) = \frac{\sigma(D_a|C_u)}{|\max(D_a) - \min(D_a)|} \quad (4.3)$$

Considering assessments of non-content preference in real domains, we measure, for each user u , his/her $Preference[u, D_a]$ for each attribute D_a , as described by Equation 4.1. Besides preference, the consumption variability exhibited by each user is also relevant. Predicting consumption of attributes that present small variability with regard to its values tends to be easier, since the consumption becomes similar to the user preference. Conversely, when this variability is more pronounced, information about expected values becomes less useful and predicting future consumption becomes more challenging. We measure the contribution of each attribute D_a to the user consumption variability through the *Relative Standard Deviation* (RSD) defined for each user u , as presented in Equation 4.4. The higher the RSD , the higher the variability that D_a brings to u consumption.

$$RSD(u) = \frac{\sigma(D_a|C_u)}{E(D_a|C_u)} \quad (4.4)$$

Similarly to measurements of user preferences, we measure the per user recommendation's non-content description through $Desc[a_n(u), D_a]$, defined by Equation 4.2. Finally, we evaluate the Preference Mismatching for each user u as the difference between $Desc[a_n(u), D_a]$ and $Preference[u, D_a]$. Thus, we can verify whether CF models fail to incorporate accurate information about non-content preferences. The relevance of such analysis is that whenever the per user recommendation's non-content description differs significantly from the user consumption, the user interests are not satisfied, affecting the quality of the recommendations.

4.3 Exploiting Preference Mismatching

In this section, we present our hybrid recommendation method for the Top- N recommendation task that combines rating information, assigned by traditional CF models to items, with the score defined by CB models that represent how well an item

matches the user non-content preference. In this sense, we assume that the smaller the preference mismatching value, the better an item matches the user non-content preference. Therefore, explicitly approximating the recommendation non-content description to the user preference would mean a significant recommendation improvement. From this perspective, an item should be recommended to a specific user whenever, besides exhibiting a high rating, it has a high probability of matching this user non-content preference for a set of selected attributes.

4.3.1 The proposed Hybrid Method

Our method consists of four main steps. First, we execute a given CF method \mathcal{A}_n , such as *Matrix Factorization*, in order to obtain an initial list of M items deemed as relevant by the CF model, such that $M \gg N$. In the second step, we derive non-content attribute values and define a vector attribute space composed of these derived attributes. Then, we represent each item present in the \mathcal{A}_n 's list within this space by computing the item value along each attribute. In this dissertation, we derived three attributes: **popularity**, **similarity** and **recency**. We selected these attributes based on some economic and social theories currently employed in RSs [Anderson, 2006a], which suggest that similarity, recency and popularity may be related to the user's taste. Further, previous studies have pointed out evidences of systematic trends along these attributes, reinforcing their relevance for this study. For instance, Yin et al. [2012] argued that RSs are more apt to recommend popular items, while recommending unpopular ones remains a challenge.

Formally, popularity refers to the receptivity of items in a domain, with respect to the desire of consumption. We measure its values as the percentage of distinct users who have consumed each item, regardless when, in a data sample. Similarity measures to what extent the items consumed by each user u are similar to each other, using the pairwise cosine distance of the item consumption vectors. The u 's preference for similarity is then computed as the mean of the similarity scores of all pairwise combinations of items consumed by u . When assessing an item i that is candidate for recommendation, its similarity score is the mean of the similarity scores of i with all items already consumed by u . Finally, recency refers to how long an item is available in a domain. We measure its values as the difference between a reference timestamp and the timestamp when the item was first consumed in a domain. By these definitions, it is straightforward that the selected non-content attributes fulfill the two first requirements discussed in Section 4.1.1. Since the third requirement is domain dependent, we evaluate it in the case study section.

In the third step, we define a CB model for preference on the vector space using a multivariate Gaussian because of its computational simplicity and the lack of evidences to adopt a more specific model. Thus, the non-content preference of each user u is a function $\mathcal{N}(\mu_u, \sigma_u^2)$ derived from the user non-content preference information. Along each attribute, we define the mean value of all items already consumed by u as the mean μ_u . We also derive the covariance matrix σ_u^2 from u 's consumption history. Then, for each item i issued by \mathcal{A}_n in the first step, we define a new score that quantifies the preference mismatching between the item representation and the user preference model. In this case, such score is simply the probability defined by the function $\mathcal{N}(\mu_u, \sigma_u^2)$ at the point defined by the vector that represents i . The adoption of a probabilistic perspective for measuring preference mismatching in this case stems from the need of models to take into account distinct attributes simultaneously. Also, we need to capture both the user non-content preference and the variability around this preference. Differently from the characterization methodology, where we calculate preference mismatching through an Euclidean perspective with the single goal of measuring the mismatching along each attribute, individually, we use a more robust perspective of analysis.

The last step combines the rating information provided by \mathcal{A}_n with this probabilistic score, generating a final score used to re-rank the recommendations. Among the possible combination strategies, we choose a simple linear combination between ratings and probabilities to define the final score of each item i , such as presented in Equation 4.5.

$$Score(u, i, t) = \alpha \times \frac{f_{\mathcal{A}_n}(u, i, t)}{\max_{-u}(R_{\mathcal{A}_n, u})} + (1 - \alpha) \times \frac{\mathcal{N}(\mu_u, \sigma_u^2)_i}{\max[\mathcal{N}(\mu_u, \sigma_u^2)]} \quad (4.5)$$

where $f_{\mathcal{A}_n}(u, i, t)$ denotes the utility (e.g., the rating) predicted by \mathcal{A}_n to item i at the test moment t , considering the target user u ; $R_{\mathcal{A}_n, u}$ is the set of items recommended by \mathcal{A}_n to u ; $\max_{-u}(R_{\mathcal{A}_n, u})$ represents the maximum utility assigned by \mathcal{A}_n to any item belonging to $R_{\mathcal{A}_n, u}$; $\mathcal{N}(\mu_u, \sigma_u^2)_i$ refers to the score assigned by u 's preference model to item i ; and α represents a weighting factor in the linear combination. Aiming to evaluate the relevance of the non-content preference information on this combination, we perform an exhaustive evaluation of several α values between 0 and 1. Also, given the complexity of evaluating individual α values for each user, we adopted a single global α for all users, although it is expected that distinct users require different combination weights. Furthermore, we normalize each rating $f_{\mathcal{A}_n}(u, i, t)$ and probability $\mathcal{N}(\mu_u, \sigma_u^2)_i$, since they vary on distinct scale of values.

4.3.2 Rationale for the proposed method

We employ a somewhat unusual approach to compute our CB scores, in the third step, to reflect the fact that CF recommendations may or may not adequately represent user’s preferences in the dimensions we model. Instead of directly computing a CB score, and then a hybrid recommender mix the CF and CB scores, we compute a CB delta score that already has built into it a reflection of the degree to which the CF recommendation reflects the modeled user preference. Hence, if a highly-ranked item already reflects the user’s preferences in non-content attributes, the CB delta will leave this item where it is in the recommendation list. But if an item is over or under recommended relative to the preference dimensions, the CB recommendation may move it down or up as appropriate.

Additionally, we highlight that such hybrid method can be easily incorporated to the traditional recommendation process, regardless the domain or adopted CF method. Whenever it is possible to identify any significant non-content attribute that users follow, our approach is able to incorporate it explicitly into the recommendations. Also, we can apply distinct strategies for building preference models, calculating preference mismatching and combining CF and CB scores.

4.4 Case Studies

4.4.1 Datasets

We performed empirical evaluations considering five real data collections. Besides the three dataset used to evaluate the first hypothesis in Chapter 3, namely Netflix, LastFm and ML-10M, we adopted two other datasets, ML-1M and Million. ML-1M is another rating data sample from MovieLens (<http://movielens.umn.edu>), gathered and made available for research purposes by GroupLens Research. Million is a random sample from the Million Song Dataset (<http://labrosa.ee.columbia.edu/millionsong/tasteprofile>) [Bertin-Mahieux et al., 2011], made available recently for research purposes on recommendation. We did not use the dataset ML-Tags in the following experiments on account of its restricted amount of binary data, which hinders consolidating non-content attributes from consumption. Table 4.1 summarizes the main features of each evaluated dataset. As Million does not provide temporal information about user actions, we cannot evaluate the non-content attribute of recency on it. Further, for the calculation of each non-content attribute, rating based datasets were transformed into consumption data simply by considering all ratings as consumption, disregarding the rating.

Table 4.1: Dataset information.

	Netflix	LastFm	ML-1M	ML-10M	Million
# Users	480,189	35,000	6,000	72,000	200,000
# Items	17,770	4 million	4,000	10,000	348,360
# Actions	100 million	85 million	1 million	10 million	19 million
# Time	310 weeks	281 weeks	149 weeks	671 weeks	-
Type	rating	play count	rating	rating	play count
Domain	movies	songs	movies	movies	songs

4.4.2 Evaluated Recommenders

Our analyses took into account six representative CF techniques, both memory-based and model-based, for the Top- N recommending task. Specifically, the set A of evaluated methods comprises the algorithms *Matrix Factorization* (MF), *Latent Feature Log Linear Model* (LF), *Biased Matrix Factorization* (BMF), *SVDPlusPlus* (SVD) implemented and distributed by the MyMediaLite project [Gantner et al., 2011]. For simplicity of analysis, we used the default parameters of each algorithm in the library on all evaluations. Furthermore, since the memory-based implementations of MyMediaLite were not able to handle the analyzed datasets, we implemented our versions of the traditional algorithms *UserKNN* and *ItemKNN* using the Cosine measure as similarity function, such as presented in [Adomavicius and Tuzhilin, 2005]. Also, for both algorithms, we incorporated the sample bias regularization with the original parameters used in MyMediaLite and 80 as the maximum number of neighbors. Our experiments were performed in octa-core machines with 96 GB of RAM. However, these machines were not able to run LF, SVD and ItemKNN methods on LastFm neither LF on our MillionSongs data sample, on account of the inability of these methods to scale to huge volumes of data.

4.4.3 Existence of Preference Mismatching

Starting our analyses by the measurements of consumption correlation, we plotted a *Complementary Cumulative Distribution Function* (CCDF) of the NSD values found for each user. This distribution shows that, in general, users consume items belonging to a restricted range of values along each attribute. In all datasets, we observed that more than 80% of the users exhibit a normalized standard deviation smaller than 25%, 30% and 15% for popularity, recency and similarity, respectively. Thus, the variability of consumption exhibited by each user usually relies on a range of values smaller

than one quarter of the whole spectrum. Despite not being sufficient for determining accurate preferences for each user, values in each of these non-content attributes may help to filter out irrelevant items.

By taking into account the user preference analysis, we plotted a *CCDF* of the $Preference[u, D_a]$ values found for each user in our data collections, such as presented by Figure 4.2. This distribution shows that users exhibit distinct preferences for each attribute. Further, the absence of gaps in these plots evinces that there is no predominant preference in the evaluated attributes. We observed low probabilities related even to zero, marked as dashed lines in the plots, although there is a concentration of preferences in a range near to zero (-0.5 and 0.5) in almost all cases. This behavior points out that user non-content preferences mostly deviate from a single and global expected value $E(D_a)$ half of the $E(D_a)$ value, for each attribute. Therefore, a single expected value is not enough to describe accurately all users. Finally, we highlight that users exhibited a slightly higher interest towards more popular, similar and recent items than the expected in almost all datasets, since the probability of positive preference values along popularity and similarity is 60% and negative preference values along recency is 65%.

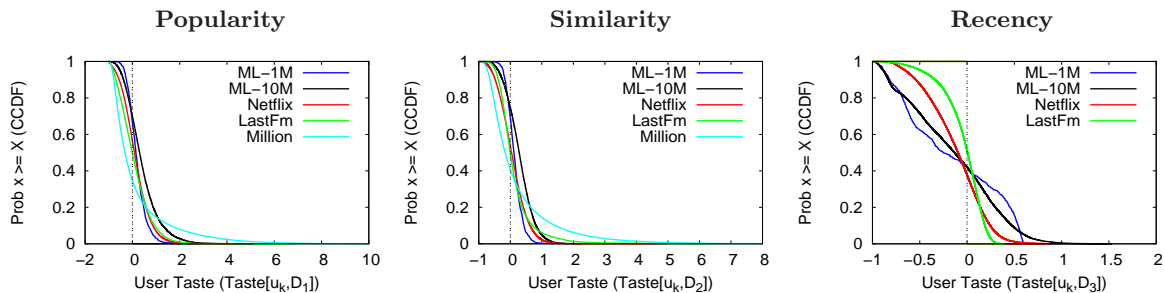


Figure 4.2: Analysis of user preferences. Users exhibited a slightly higher interest towards more popular, similar and recent items than the expected in almost all datasets.

By plotting a *CCDF* of the *RSD* values found for each user, we evinced large variabilities in all evaluated datasets, for the three selected attributes, as shown by Figure 4.3. Variabilities larger than 50% had probabilities of occurrence higher than 70% for almost all datasets and attributes. For some cases, we observed variabilities even larger than the expected value estimated from user consumption histories (i.e., larger than 100% in the plots). Thus, besides presenting distinct preference values, users also consume a range of items with different characteristics regarding each attribute. Based on these results, we conclude that users are not strongly tied to their individual preferences, presenting high variability of consumption in all evaluated collections.

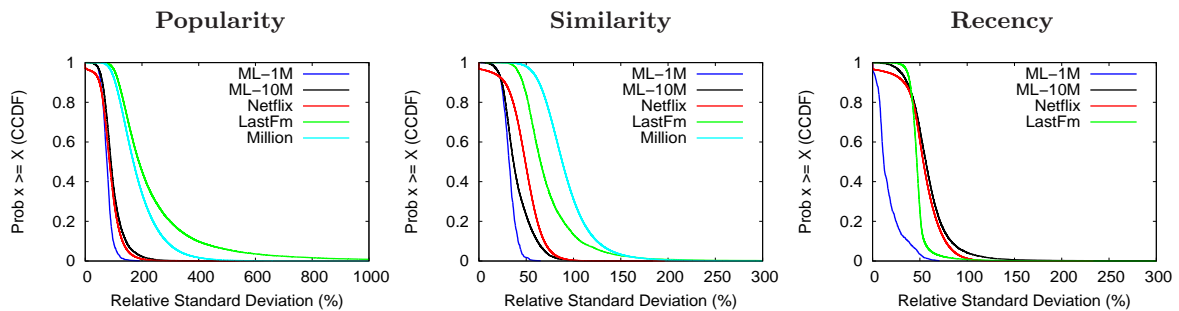


Figure 4.3: Distributions of relative standard deviations of user consumption. These plots show that users consumed a range of items with different characteristics with regard to each non-content attribute in the evaluated datasets.

In order to assess the recommendation non-content description values in the evaluated datasets, we plotted a *CCDF* of $Desc[\mathcal{A}_n(u), D_a]$ for each user u . Similarly to users, RSs provided distinct non-content descriptions, which also vary according to each dataset. Starting by recency, despite presenting distinct descriptions, we observed almost consensual behaviors among the six evaluated RSs. Most of them prioritized recent items, presenting over than 60% of probability for negative description values. For popularity and similarity, we observed a more diversified scenario. The same methods exhibited distinct behaviors on different datasets. For instance, whereas LF and SVD presented positive popularity and similarity description values in ML-1M and ML-10, they exhibited negative ones in Netflix. Further, most of these non-content description values lay between -0.5 and 1.0 for both attributes, demonstrating a high diversity of descriptions. Also, the results showed an unexpected behavior for UserKNN and ItemKNN with respect to popularity and similarity. Differently from previously stated [Rafter et al., 2009], KNN-based methods prioritized items less popular and similar than those usually consumed by each user, exhibiting negative popularity and similarity description values. Such divergence stems from the fact that the consolidation of neighborhoods is heavily based on more popular and similar items, since similarity is usually defined over a consumption intersection between user transactions. However, the items actually recommended, which are outside of this intersection, tend to be less popular and similar.

Finally, analyses on the preference mismatching demonstrated that all evaluated methods provided recommendations that systematically deviate from the user preferences. Figure 4.4 shows a *CCDF* of the difference $Desc[\mathcal{A}_n(u), D_a] - Preference[u, D_a]$ defined for each user u . We observed a high concentration of difference values near to zero within ranges of ± 1 , ± 0.5 and ± 0.05 for popularity, similarity and recency, respectively. Thus, an ϵ value of 0.33, for instance, would be enough to define a sig-

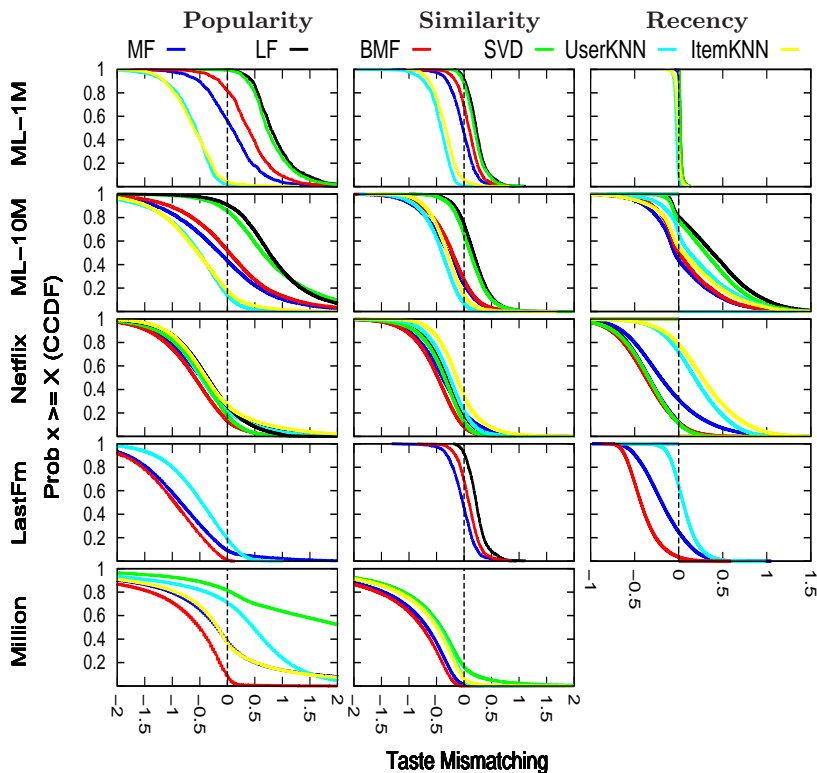


Figure 4.4: Analysis of preference mismatching. All evaluated methods provided recommendations that systematically deviate from the actual user preferences.

nificant preference mismatching along similarity for more than half of users in almost all datasets, methods and attributes. Indeed, these are expressive differences when we take into account the expected values. For example, for a user who exhibits a similarity preference of 0.30, such a difference means that RSs usually recommend items with non-content description from 0.20 to 0.40. Considering each attribute individually, through recency we observed that, besides presenting description values towards recent items (i.e., negative values), most RSs presented recommendation descriptions stronger than the user preferences, recommending to them items more recent than they usually consume. For popularity and similarity, RSs presented diversified behaviors. Whereas we observed positive description values in some datasets, for the same RSs, we observed negative ones in other datasets. We believe these behaviors result from inherent characteristics of each dataset and a deeper analysis in this direction would be required.

4.4.4 Experimental Design

Besides demonstrating the existence of significant non-content preference mismatching in real domains, we need to verify the utility of reducing this mismatching towards better recommendations. In this sense, we evaluated the proposed hybrid method. Since

a proper *n-fold cross validation* design would require a careful design in temporally ordered data and demand huge execution time for the evaluated datasets and algorithms, the following analyses employed a traditional training (70%) / test (30%) partition.

Aligned with the main goal of this dissertation of enhancing the discovery of relevant items in the recommendations, our analyses took into account three distinct quality dimensions for the Top-50 recommendation task: accuracy, novelty and diversity. Assessments on accuracy were based on the classical *Precision@50* and precision was measured by counting the number of distinct items of the Top-50 recommendation that appears in the per user test set. Such as done in Chapter 3, we measured novelty and diversity through a formal framework of analysis presented in [Vargas and Castells, 2011]. Specifically, we used the EPC_rank, and the EILD for measuring novelty and diversity, respectively, considering in both cases the discount function ($disc(K)$) equals to 0.85^{k-1} , Pearson correlation as similarity distance measure and relevance aware recommendations [Vargas and Castells, 2011]. Also, we set the parameter M given as input for our hybrid method to 500, aiming to exploit significantly larger lists of items than the final recommendation list (ten times larger in this case) while keeping computationally feasible the experimentation. Finally, we point out that our strategy of analysis was based on contrasting the results of each original CF method \mathcal{A}_n against the results of our hybrid model when performed with \mathcal{A}_n . Our primary goal is to identify the relevance of non-content preference attributes for improving traditional CF methods, rather than contrasting it against other hybrid methods.

4.4.5 Exploiting Preference Mismatching

We started our analyses by investigating the individual usefulness of each selected non-content attribute, for providing better recommendations. Figure 4.5 shows the gains and loses of *Precision@50* when building a probabilistic model for each attribute individually and using distinct combination weights. We observed expressive gains when exploiting popularity in MF, BMF, UserKNN and ItemKNN for all datasets. However, the methods LF and SVD, which exhibited the highest popularity preference mismatching values in Figure 4.4, could not be improved through this attribute. As they exhibited such a strong deviation towards popular items, reducing preference mismatching among the 500 recommended items was not enough to improve the results. Taking into account similarity and recency, we could not improve the CF results in most cases. In summary, our hybrid method was not able to effectively exploit alone each of these attributes in order to improve recommendations.

An immediate question is what happens when we take into account the three se-

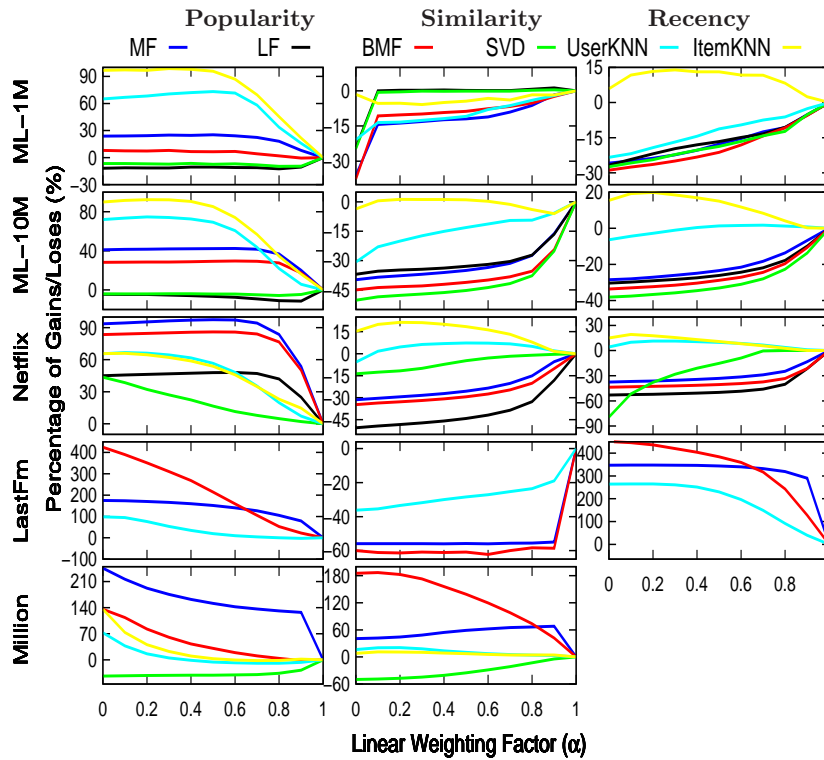


Figure 4.5: Analysis of $Precision@50$ gains|loses by exploiting, individually, popularity, similarity and recency. While we achieved expressive gains when exploiting popularity in MF, BMF, UserKNN and ItemKNN for all datasets, we could not improve the CF results when exploiting similarity and recency, individually.

lected attributes simultaneously. Figure 4.6 answers this question regarding accuracy, novelty and diversity. Besides even higher gains in terms of accuracy in almost all CF algorithms and datasets, we also achieved simultaneous gains regarding novelty and diversity. In general, the most expressive gains were observed in CF methods with the worse performance in each dataset. For instance, we observed gains over than 200% on LastFm and Million datasets. In these cases, the original CF results were actually not significant. However, among the Top-500, the CF methods rescued several relevant items for each user and the non-content preference information was enough to identify these items. On the other hand, gains around 10% were consistently related to CF methods focused on suggesting items more popular, similar and older than the user expected interest (i.e., LF and SVD). Although they could achieve high accuracy rates, several items recommended by these methods not necessarily suit the user non-content preference. This fact explains why the gains in these cases were not as expressive as for the other methods. As the lists provided by LF and SVD exhibited non-content descriptions far from the non-content preferences of each user u , items closer to the user preference in the Top-500 were still far from the actual preference of u .

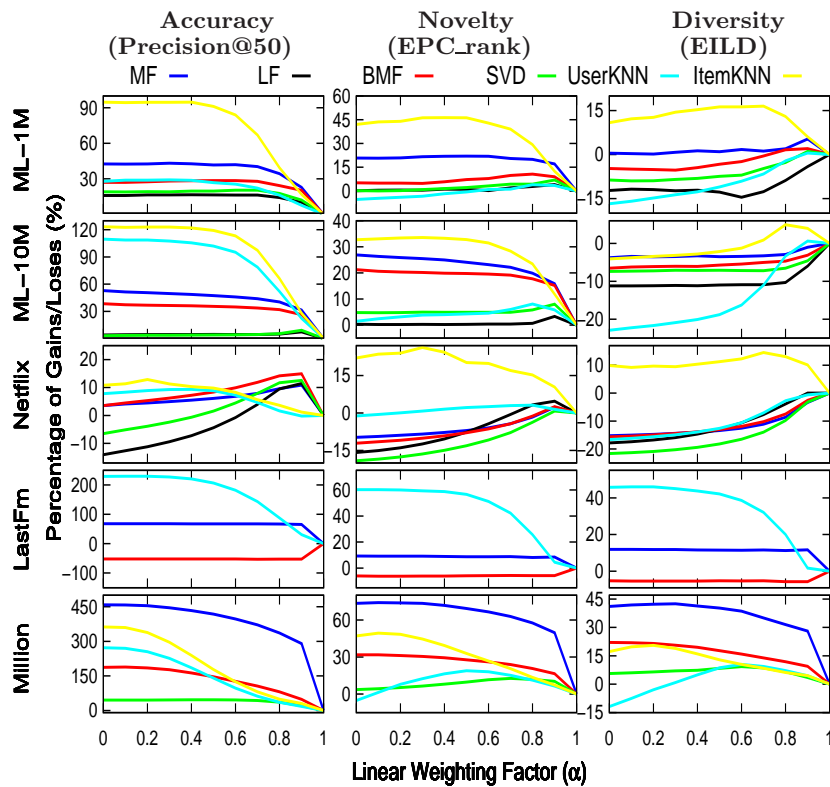


Figure 4.6: Analysis of quality gains|loses by exploiting, simultaneously, popularity, similarity and recency. In this case, we achieved simultaneous gains with regard to accuracy, novelty and diversity in almost all methods and datasets.

Besides verifying the strength of the improvements provided by exploiting non-content preference in RSs, it is also important to investigate how often these gains happen. In this sense, we evaluated the percentage of users in each database for whom our hybrid method was able to produce any improvement in a Top- N recommendation, in terms of accuracy. As our method processes a prior recommendation list of size $M \gg N$ provided by a CF method for each user, such percentage is limited by the percentage of these prior lists that contain more relevant items than those present among its N first items. Figure 4.7 presents the percentage of possible improvements as the number of distinct users for whom our hybrid method made enhancements, divided by the number of users for whom the recommendation list provided by each CF could be improved. Our method, even adopting a global linear combination weight, was able to improve recommendation for more than 40% of these users in most cases. Further, the percentage of users for whom our method produced losses was at most 10% in all datasets.

In summary, our hybrid method allowed us to verify the relevance of user non-content preferences in practical scenarios. By exploiting this type of information, we provided expressive gains in terms of accuracy, novelty and diversity for six major

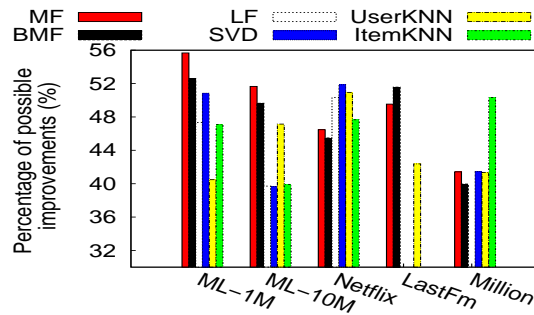


Figure 4.7: Percentage of users for whom the non-content preference information produced any improvement. Besides improving recommendation for more than 40% of the users in all datasets, the proposed method produced losses for at most 10% of the users.

CF methods, considering the Top-50 recommendation task. We explain these gains by the fact that non-content preference information is able to filter out items that seem to suit user preferences, but mismatch non-content characteristics from the items usually consumed by each user. Aiming to evince the existence of such mismatch items among the sorted Top- N list originally recommended by each CF method, we calculated the non-content preference mismatching, such as performed in our methodology, but considering now each rank in this list. For sake of brevity, we show the mean of this deviation per rank among all users for one dataset (ML-10M), such as presented in Figure 4.8, although the same behavior was observed in all other collections. The preference mismatching varied significantly along the ranks, not presenting any monotonic behavior. It reinforces that CF recommendations do not capture the systematic preference existing along each non-content attribute.

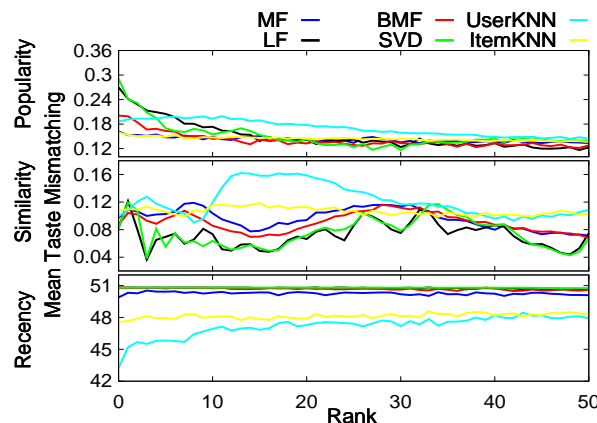


Figure 4.8: Analysis of mean preference mismatching per rank in CF recommendation lists. The preference mismatching varied significantly along the ranks, not presenting any monotonic behavior.

Additionally, we evaluated to what extent Gaussian functions are appropriate

for modeling user non-content preferences for the selected attributes. By plotting the probability distribution of the mean expected value of all users, along each attribute, and its standard deviation, we found that, indeed, the global behavior in our domains presents a Gaussian shape. However, these distributions point out two main issues. First, users exhibited behavior significantly distinct from each other, demonstrated by high standard deviations in all points. Hence, distinct users would require, besides different parameters, different model functions. For instance, while a set of users exhibit a Gaussian-like behavior with respect to the preference, others would present a power law like. In this case, non-parametric solutions may be applied. Second, the evaluated attributes do not vary in the same way, requiring distinct models. While recency seems much more an exponential function, popularity and similarity present a Gaussian shape. Therefore, besides all gains achieved by our simple method, these issues point out a room for even more improvements.

4.5 Summary

Aiming to evaluate our second working hypothesis, in this chapter, we formally defined the non-content preference attributes, as well as the *Preference Mismatching* metric. We introduced this metric to measure the difference between mathematical definitions of user's preference and the recommendation description along non-content attributes.

We also discussed the motivation to evaluate and address Preference Mismatching, which stems, first, from the fact that non-content preferences might influence the behavior of some users. Second, we observed that state-of-the-art RSs do not guarantee the issued recommendations are aligned with such preferences derived from consumption or metadata. Since consumption information is modeled by CF methods, we restricted our analyses about non-content attributes and Preference Mismatching to this type of RSs.

Thereafter, the chapter described a characterization methodology to verify the existence of non-content preference mismatching in real domains. This methodology answers crucial questions related to our definition of non-content preference, using traditional statistics metrics of data variability and dispersion along each non-content attributed selected for analysis.

Later, we proposed a hybrid method to build CB models by using non-content attribute information and combine these new models with existing CF methods to produce better recommendations. This methods assumes that an item should be recommended to a specific user whenever, besides exhibiting a high rating, it has a high probability of matching this user non-content preference for a set of selected attributes.

Thus, whenever it is possible to identify any significant non-content attribute that users follow, our approach is able to incorporate it explicitly into the recommendations.

Next, using five real collections, we validated two main issues related to Preference Mismatching: its existence and usefulness for recommendation. By exploiting non-content preferences explicitly, we provided expressive gains regarding accuracy, novelty and diversity in six major CF methods, considering the Top-50 recommendation task. We explain these gains by the fact that non-content preference information filters out items that seem to suit user preferences but mismatch non-content characteristics from items consumed by each user.

Chapter 5

Combining Forgotten items and Non-content preference

This chapter discusses our third working hypothesis: *The limitations related to forgotten re-consumable items and non-content preference, when addressed simultaneously, provide complementary enhancements to RSs.* We start by presenting the motivation to combine solutions proposed for each limitation individually. Thereafter, we introduce Remembrall, a method that consolidates the further knowledge we acquired about exploiting forgotten re-consumable items. Next, we present NonContent, a novel hybrid method that refines the method proposed in Section 4.3 to mitigate Preference Mismatching. Later, we describe ForNonContent, a hybrid method that combines Remembrall and NonContent. Then, we evaluate the complementarity between recommendation lists provided by Remembrall and NonContent. Finally, we summarize the main discussions and conclusions raised in this chapter.

5.1 Motivation

Chapters 3 and 4 demonstrated our hypotheses about two unaddressed algorithmic limitations in RSs: the Oblivion Problem and non-content preference mismatching. While we verified, in Chapter 3, that UserKNN fails to bring back forgotten re-consumable items, Chapter 4 demonstrated the existence of non-content Preference Mismatching on the recommendations of UserKNN and other five major RSs. Actually, we can show these five RSs also fail to bring back forgotten re-consumable items. Table 5.1 presents the probability of RSs issue non-forgotten items. This probability is the **Property 2** of the methodology proposed in Section 3.2. In this experiment, we used: (1) the six RSs evaluated in Chapter 4 (Section 4.4.2); (2) the datasets used to study

the Oblivion Problem (Section 3.4.1); (3) and the methodology’s parameters adopted in Section 3.4.2. All evaluated methods presented high probability of recommending only items non-forgotten by the target user in the present.

Table 5.1: Probability of RSs issue non-forgotten items to the users. All evaluated methods present high probability of recommending non-forgotten items, confirming our first working hypothesis.

	LastFm	ML-10M	ML-Tags	Netflix
UserKNN	1	1	1	1
ItemKNN	-	1	1	1
MF	0.99	0.85	1	0.96
LF	-	0.78	0.99	0.97
BMF	0.99	0.82	1	0.97
SVD	-	0.75	0.99	0.96

Besides validating our hypotheses, we proposed and evaluated different strategies to model forgetfulness and re-consumption in recommendation domains. Also, we characterized Preference Mismatching in real domains and proposed a novel hybrid method that mitigates existing mismatching. Although each of these proposals provided significant enhancements on the Top- N recommendation task, we raise a question in this chapter: could such enhancements be combined, providing even better recommendations?

This question stems from further knowledge we acquired about both limitations. The definitions of forgotten re-consumable items and non-content attributes show no direct correlation with each other. While the former relies on individual consumption of each single item over time, by each user u , non-content attributes refer to common characteristics observed among the whole set of items consumed by u . On the other hand, both concepts are not orthogonal, since recency of consumption is somehow related to the consumption of individual items over time. Furthermore, the strategies we proposed to address each limitation exploit portions of the item space of each domain that, although overlap, are not the same. Addressing the Oblivion Problem requires exploiting the long history of u . Conversely, mitigating Preference Mismatching includes recommending items not known by u . Thus, addressing one limitation does not necessarily represent to cope with the other. As this dissertation considers that both limitations may affect the discovery of items in RSs, we intend to address them simultaneously.

Therefore, we propose in this chapter ForNonContent, a new hybrid method that exploits simultaneously the long-term history, through forgotten re-consumable

items, and implicit preference signals, using non-content preferences. We designed ForNonContent to perform the Top- N recommendation task through three main steps:

1. Issue a list of Top- N forgotten re-consumable items;
2. Issue a list of Top- N items with reduced non-content preference mismatching;
3. Combine the previous two lists providing a single Top- N recommendation list.

Based on our main findings about forgotten re-consumable items and non-content attributes, we proposed efficient methods to handle each of these three steps in practice. The following sections describe in details each method.

5.2 Remembrall: A recommender of forgotten re-consumable items

As discussed in Chapter 3, recommending forgotten re-consumable items involves two steps with non-aligned goals: (1) identify the set of forgotten items; and (2) identify the subset of re-consumable ones. Remembrall uses the concept of *memory retrievability* (*mem-ret*) to perform the first step, such as defined by Equation 5.1. Since we realized ACT-R should be adapted to better model consumption domains, we evaluated distinct strategies to combine recency of consumption, past utility and association with currently consumed items (i.e., context). For sake of simplicity, this work restricted the search for proper combinations to linear models of these three types of information, taken two by two or all together. An exhaustive analysis on all linear combinations pointed out the combination of recency of consumption and context as the best one for most of the datasets described in Section 3.4.1. Hence, Equation 5.1 assumes that the less recently consumed and the less correlated to a current context, the higher the chances of an item be forgotten. However, these two types of information might not be equally important to predict these chances.

$$\begin{aligned}
 \text{mem-ret}(u, i, t) = & \alpha \times \left[\log \left(\sum_{l \in T_{u,i}} [t - l] \right) \right] + \\
 & (1 - \alpha) \times \left[\sum_{j \in C_{u,c}} \frac{\sum_{l=c}^t f(u, i, l)}{\max_{u'}(C_{u',c})} \times \log \left(\frac{\text{prob}(i|j)}{\text{prob}(j)} \right) \right]
 \end{aligned} \tag{5.1}$$

where, $T_{u,i}$ refers to all distinct moments that u consumed i ; $C_{u,c}$ denotes the set of distinct items consumed by u during his/her c most recent training moments; $f(u, i, l)$

is the utility value assigned by u to i at the moment l ; $\max_u(C_{u,c})$ is the maximum utility value found among items of $C_{u,c}$.

In the second step, given the set of selected forgotten items, Remembrall derives a *relevance score* to distinguish the subset of re-consumable ones, such as shown by Equation 5.2. The higher this score, the higher the chances of an item be re-consumable. Again, we evaluated the best linear combination of recency of consumption, past utility and context. In this case, the combination of recency and past utility presented the best results for most of the evaluated datasets. Hence, Equation 5.2 assumes that the less recently consumed and the higher its past utility, the higher the chances of an item be re-consumable. However, both types of information have different relevance to derive each item score.

$$\text{score}(u, i, t) = \alpha \times \left[\log \left(\sum_{l \in T_{u,i}} [t - l] \right) \right] + (1 - \alpha) \times \left[\log \left(\sum_{l=c}^{t-1} \frac{f(u, i, l)}{t - l} \right) \right] \quad (5.2)$$

5.3 NonContent: Mitigating non-content preference mismatching

This section introduces NonContent, a hybrid method that refines the method proposed in Section 4.3 to mitigate Preference Mismatching. NonContent determines to what extent non-content preference mismatching affects users in a personalized manner. Such as the original method, NonContent has four main steps. The three first steps are identical in both methods. Thus, NonContent first executes a given CF method \mathcal{A}_n in order to obtain an initial list of $M \gg N$ items deemed as relevant by \mathcal{A}_n . This CF method could be seen as a parameter of NonContent and any known method could be used. In the second step, NonContent derives three non-content attribute values (**popularity**, **similarity** and **recency**). We presented a formal definition of these attributes in Section 4.3. Again, these three attributes compose a multidimensional vector space used in the next steps.

The third step defines a CB model for preference on the defined vector space using a multivariate Gaussian. The non-content preference of each user u is a function $\mathcal{N}(\mu_u, \Sigma_u)$ derived from the user non-content preference information. The mean value μ_u along each attribute and the covariance matrix Σ_u are derived from u 's consumption history. Then, for each item i issued by \mathcal{A}_n in the first step, we define a

new score that quantifies the preference mismatching between the item representation and the user preference model. This score is simply the probability defined by function $\mathcal{N}(\mu_u, \Sigma_u)$ at the point defined by the vector that represents i .

The last step combines the rating information provided by \mathcal{A}_n with this probabilistic score, generating a final score used for re-ranking the recommendations. We use a linear combination between ratings and probabilities to define the final score of each item i . The original method adopts a global linear combination weight for all users. NonContent modifies this strategy by determining an individual weight α_u for each user u . We define α_u as the *Hellinger Distance* between the multivariate Gaussian function $\mathcal{N}(\mu_u, \Sigma_u)$ that models u 's non-content preference and the Gaussian function $\mathcal{N}(\mu, \Sigma)$ that models the non-content attributes of all items, such as shown by Equations 5.3 to 5.5¹. This metric quantifies the similarity between two probability distributions [Liese and Miescke, 2008]. We assume that non-content attributes are useful for modeling u 's preference when they allow us to differentiate u 's past consumption from non-consumed items. Thus, the more distinct the two distributions are, the more useful non-content attributes are for modeling u and higher should be α_u .

$$\Sigma_M = \frac{\Sigma + \Sigma_u}{2} \quad (5.3)$$

$$BC = \exp\left(\frac{1}{8}(\mu - \mu_u)^T \Sigma_M^{-1}(\mu - \mu_u) + \frac{1}{2} \ln\left(\frac{\det \Sigma_M}{\sqrt{\det \Sigma \det \Sigma_u}}\right)\right)^{-1} \quad (5.4)$$

$$\alpha_u = \sqrt{1 - BC} \quad (5.5)$$

Equation 5.6 presents the linear combination used to derive ForNonContent's final score. We issue the Top- N items with the highest score to each target user u .

$$Score(u, i, t) = \alpha_u \times \frac{f_{\mathcal{A}_n}(u, i, t)}{\max_{u}(R_{\mathcal{A}_n, u})} + (1 - \alpha_u) \times \frac{\mathcal{N}(\mu_u, \sigma_u^2)_i}{\max[\mathcal{N}(\mu_u, \sigma_u^2)]} \quad (5.6)$$

where $f_{\mathcal{A}_n}(u, i, t)$ denotes the utility (e.g., the rating) predicted by \mathcal{A}_n to item i at the test moment t , considering the target user u ; $R_{\mathcal{A}_n, u}$ is the set of items recommended by \mathcal{A}_n to u ; $\max_{u}(R_{\mathcal{A}_n, u})$ represents the maximum utility assigned by \mathcal{A}_n to any item belonging to $R_{\mathcal{A}_n, u}$; $\mathcal{N}(\mu_u, \sigma_u^2)_i$ refers to the score assigned by u 's preference model to item i ; and α_u represents a personalized weighting factor in the linear combination. We normalize each rating $f_{\mathcal{A}_n}(u, i, t)$ and probability $\mathcal{N}(\mu_u, \sigma_u^2)_i$, since they vary on distinct scale of values.

¹ In Equation 5.4, BC stands for the *Bhattacharyya Coefficient*.

5.4 ForNonContent: Combining Remembrall and NonContent

Once we have issued a Top- N list of forgotten re-consumable items and another one of items that reduce the mismatching with regard to non-content preferences, we need to combine both lists. Although forgotten items comprise a promising source of recommendation, we are aware that users are not interested in consuming only known items. On the other hand, the refined recommendations provided by NonContent do not allow us to rediscover known items, since most of the CF methods neglect forgotten items. Our intent is to enhance simultaneously the discovery of known and unknown items potentially relevant for each user. ForNonContent adopts straightforward answers for three main issues related to this goal.

The first issue is *How do we combine known and unknown recommended items?* Basically, there are two strategies. In the first one, we could combine numerically the scores derived by Remembrall and NonContent for each item. This strategy, however, presents several drawbacks. First, the scores provided by each method are in different numeric scales. Second, the variability and distribution of values may also be different. Third, it is hard to determine a semantic correspondence between scores on these two scales. For instance, could we state that the item with the highest score for Remembrall is as important as the item with the highest score for NonContent? Thus, defining a robust numeric combination of both scores is a difficult task. The second strategy is to consider the recommendation lists as ordered lists that we want to merge somehow, like in a Merge-sort algorithm. Hence, we need to define a sort function that determines the relative order of each pair of items. ForNonContent adopts this second strategy because it avoids the aforementioned problems of comparing scores.

Our second issue is *How to determine this sort function for merging the lists of items.* Actually, the item ordering plays an important role for user experience in RSs. Knijnenburg et al. [2012] argued that the order in which the items are presented to users affect the user's perception of quality with respect to the issued recommendations. Further, Hu and Pu [2011] argued that categorizing the recommendations leads to higher satisfaction and decision confidence. However, the effects of the order in which recommendations are presented are unclear and a further understanding requires detailed evaluations about user experience when presented to distinct ordering alternatives. Since it goes beyond the scope of this work, we restricted ForNonContent to a simple merge of the two recommendation lists. First, we remove duplicate items, maintaining the best ranked occurrence only. Then, we intercalate the lists, first

presenting an item issued by NonContent, followed by a recommendation of Remembrall, another one from NonContent and so on. This strategy ensures a homogeneous balance of both kind of recommendations, not prioritizing one over the other.

Finally, we decide *how many forgotten items to recommend for each user*. Some users are more willing to consume a larger amount of forgotten items than others. For instance, while some users would like to listen again to songs from the '90s, others would prefer to listen to this year's Billboard hits. Determining the proper rate of forgotten items to recommend is a challenge. Even asking the users directly would not be enough, since users do not know all items they have forgotten. Further, the willingness to consume such items would vary over time and according to the items available at each moment. This work defines a heuristic to determine this rate, based on the percentage of known items actually re-consumed in the test set, as follows. First, we identify the most recent moment t in the training set of each user u in which u consumed any of the items deemed as forgotten by him/her (i.e., any item present in the u 's recommendation list issued by Remembrall). Then, we count the number N_t of items consumed in the test set that was consumed in the training set at a moment before or equal to t . We determine the percentage of forgotten items to be recommended as N_t divided by the total number of test items of u . This rate represents the percentage of consumed items that are as old as the recommended forgotten items. The premise is that the recommended forgotten items would work as substitute items for these 'old' consumed items with respect to age. Substitute items have similar value and fulfill the same user needs [Nicholson and Snyder, 2011].

5.5 Case Studies

5.5.1 Datasets

Since our analyses included the execution of both Remembrall and NonContent methods, we used datasets that present two properties. First, they must provide temporal information about user actions, in order to allow us exploit the user history over time. Second, they must provide enough information to consolidate non-content attributes from consumption, such as discussed in Section 4.1.1. The subset of datasets used in our previous analyses that ensure both properties are: Netflix, LastFm, ML-1M and ML-10M. Further details about these datasets were presented in Section 4.4.1.

5.5.2 Experimental Design

Our offline analysis intend to demonstrate that both algorithmic limitations bring complementary enhancements. In this sense, we designed three distinct experiments. First, we evaluated for each user the intersection between the recommendation lists issued by Remembrall and NonContent divided by the list size (i.e., the Percentage of Intersection – PoI). Then, we determined the median PoI value ($MPoI$) found for all users. Higher intersection values mean that NonContent recommendations include Remembrall ones. Second, we calculated the median percentage of items re-consumed in the test set and recovered by Remembrall that were also recommended by NonContent for each target user (i.e., Median Percentage of Re-consumed Items – $MPoRI$). This percentage reveals the amount of actually re-consumed items rescued by Remembrall also identified by NonContent. Finally, we determined an inter-list similarity (ILS) as the median similarity of all pairs of items from a Cartesian product of the recommendation lists issued by Remembrall and NonContent for each user. Similarity between two items was the pairwise Cosine distance of the corresponding consumption vectors. We summarize the ILS found for all users through the median ILS value ($MILS$).

We evaluated Remembrall and NonContent individually, in order to assess the gains each one could provide on the evaluated datasets. We implemented Remembrall as described in Section 5.2. The only parameter it requires is the linear combination weight α . In order to define α , we varied its values by 0.05 from 0.05 to 0.95 and chose the value that presented the best results in most datasets. Through this process, we set $\alpha = 0.75$. In turn, we implemented NonContent according to Section 5.3. This method does not require any parameter. Finally, we evaluated ForNonContent, which was implemented as described in Section 5.4. ForNonContent does not require any other parameter than the parameter α required by Remembrall.

Again, our analyses employed the traditional training/test partition. We used 30% of the most recent weeks of each user history as a test set and the remaining weeks as a training set. We adopted a training/test partition instead of *n-fold cross validation* since the latter would require a complex and careful design in temporally ordered data, in order to maintain the temporal properties of each dataset.

Finally, we highlight that it is not possible to assess through offline analysis the gains or losses of ForNonContent over Remembrall and NonContent. On one side, accuracy measures are not valid for forgotten items, since we expect that users do not consume them in the test set. On the other hand, recall analysis on recommendations issued by NonContent are not useful, since, through this method, we aim to present to users novel items rather than to rescue a large number of known ones. We left such

analysis for a user study in a real domain, discussed in Chapter 6.

5.5.3 Analysis of Complementarity

Table 5.2 shows the complementarity observed between Remembrall and NonContent in our data collections. The intersection of items recommended by each method (*MPoI*) is less than 10% for all datasets. Also, the intersection of the subset of items recommended by Remembrall and NonContent that were re-consumed by each target user (*MPoRI*) is even smaller (i.e., smaller than 1% for all collections). In large datasets, such as LastFm, the intersection is close to zero. Thus, each method presented to users distinct sets of items, exploiting different portions of the available items in each domain. Further, the median similarity among items belonging to distinct recommendation lists is also small in all datasets (i.e., smaller than 0.1400). Besides presenting recommendation lists with low intersection, the recommended items belonging to distinct lists are not similar. These results point out that by combining Remembrall and NonContent we could improve the diversity of the recommendations. Through this combination, we also could address distinct pieces of each user taste, enhancing his/her experience with the recommender system.

Table 5.2: Analysis of complementarity between Remembrall and NonContent. Besides presenting low intersection of recommended items, the recommendation lists present low inter-list similarity. These results evince the complementarity of Remembrall’s and NonContent’s recommendations.

	Netflix			LastFm			ML-1M			ML-10M		
	MPoI	MPoRI	MILS	MPoI	MPoRI	MILS	MPoI	MPoRI	MILS	MPoI	MPoRI	MILS
MF	0.0183	0.0046	0.0671	0	0	0.0184	0.0342	0.0093	0.0810	0.0210	0.0071	0.0719
LF	0.0089	0.0021	0.0792	-	-	-	0.0174	0.0061	0.0881	0.0106	0.0049	0.0662
BMF	0.0146	0.0035	0.0709	0	0	0.0197	0.0208	0.0087	0.0653	0.0117	0.0060	0.0574
SVD	0.0097	0.0017	0.0556	-	-	-	0.0149	0.0059	0.0590	0.0185	0.0051	0.0489
UserKNN	0	0	0.0872	0	0	0.036	0	0	0.1233	0	0	0.1098
ItemKNN	0	0	0.1041	-	-	-	0	0	0.1304	0	0	0.1175

5.6 Summary

We started this chapter by pointing out that addressing the Oblivion Problem does not necessarily represent to cope with the Preference Mismatching. Strategies proposed to address each of these algorithmic limitations exploit portions of the item space available on each domain that, although overlap, are not the same. As this dissertation considers that both limitations may affect the discovery of items in RSs, we address them simultaneously.

Next, we introduced Remembrall, a novel method that better combines information of past utility, recency of consumption and current context of consumption to model forgetfulness and re-consumption. Since we realized ACT-R should be adapted to better model consumption domains, we evaluated distinct strategies to combine these three types of information. Remembrall assumes that the less recently consumed and the less correlated to a current context, the higher the changes of an item be forgotten. Also, the more recently consumed and the higher its past utility, the higher the chances of an item be re-consumable.

Thereafter, we presented NonContent, a hybrid method that refines the method proposed in Section 4.3 to mitigate Preference Mismatching. NonContent determines to what extent non-content preference mismatching affects users in a personalized manner. This method assumes that non-content attributes are useful for modeling u 's preference whenever they allow us to differentiate u 's past consumption from items not consumed.

Finally, we evaluated the complementarity between Remembrall and NonContent. Besides presenting recommendation lists with low intersection, the recommended items belonging to distinct lists are not similar. Thus, by combining both lists we could address distinct pieces of each user's taste, enhancing his/her experience with the recommender system. We leave to online analysis the assessments on gains or losses of ForNonContent over Remembrall and NonContent, given the inability to perform such evaluation through offline analysis.

Chapter 6

End-user Study

This chapter discusses the end-user study we conducted in a well-known and relevant recommendation domain. First, we present the evaluation goals of this study. Then, we briefly describe the methods compared in our live analysis. Next, we present the adopted methodology of evaluation. Thereafter, we introduce the web-based system we implemented for this study. Later, we discuss the main findings on user feedback. The chapter ends with a summary of the main concepts and findings hereby discussed.

6.1 Evaluation Goals

We conducted a user study to estimate the perceived value of recommendations issued by the methods proposed in this dissertation. In this sense, we designed a survey for MovieLens users to evaluate our recommendations. The survey presents to each participant recommendations issued by five distinct RSs, asks him/her to rate the movies, compare distinct recommendation lists and fill small questionnaires about the recommendations and himself/herself. Participants were unaware of how we generated recommendations. Despite the analysis of scope pointed out that the individual perspective of the Oblivion Problem has low relevance for MovieLens users, Section 3.4.2, observing how such users would react when exposed to recommendations of items consumed long ago would be relevant. This is a relevant issue since movie recommenders, in general, do not issue these items as new recommendations, although MovieLens' interfaces allow users to verify their set of rated movies. A positive feedback in this case would reinforce the usefulness of recommending forgotten items in several domains where this problem becomes equally or more relevant.

6.2 Evaluated Methods

The study includes recommendations issued by five distinct RSs. The first one is a matrix factorization method (*Latent Feature Log Linear Model* - LF). We used the version of LF implemented and distributed by the MyMediaLite project with its default parameters [Gantner et al., 2011]. The second RS is STREAM, a well-known hybrid method [Bao et al., 2009]. We implemented STREAM as a combination of three distinct methods: LF (a user-based CF), ItemKNN with Pearson Correlation (an item-based CF) and ItemAttributeKNN with Cosine Distance (a content-based RS). Again, we used versions of these methods implemented and distributed by MyMediaLite with their default parameters. Such as suggested by the STREAM's authors, we used as content for each movie: *title, release year, genres, keywords, plot, actors, directors*, which were gathered from the IMDB dataset¹. Further, due to high computational costs, we adopted the following simplifications. First, we trained STREAM using a random sample of 25% of the training examples. Second, we combined the three methods through a multivariate linear regression. Third, we did not use any 'runtime metric' proposed by the authors. As the third RS we implemented Remembrall such as described in Section 5.2. Our fourth RS is the NonContent method describe in Section 5.3. Finally, the fifth RS is ForNonContent, the proposed hybrid method that combines the recommendations issued by Remembrall and NonContent, Section 5.4. We adopted the same parameter configurations of our offline analysis for Remembrall, NonContent and ForNonContent, such as discussed in Section 5.5.2.

6.3 Methodology of Evaluation

It is noteworthy that we did not optimize any evaluated RS with the training set used to issue the recommendations for the survey. Such training set comprises about 17.5 million ratings that 82,000 distinct MovieLens' users assigned to 21,600 movies. Aiming to compose the test set, we randomly picked 1,000 distinct users who have rated at least 50 movies in the system and have signed in MovieLens from 01/01/2013 to 10/01/2013. We sent an email to each selected user with the link for the evaluation system and gathered along two months all answers willingly submitted by 235 users. We did not require users to perform all survey tasks.

¹<http://www.imdb.com/interfaces>

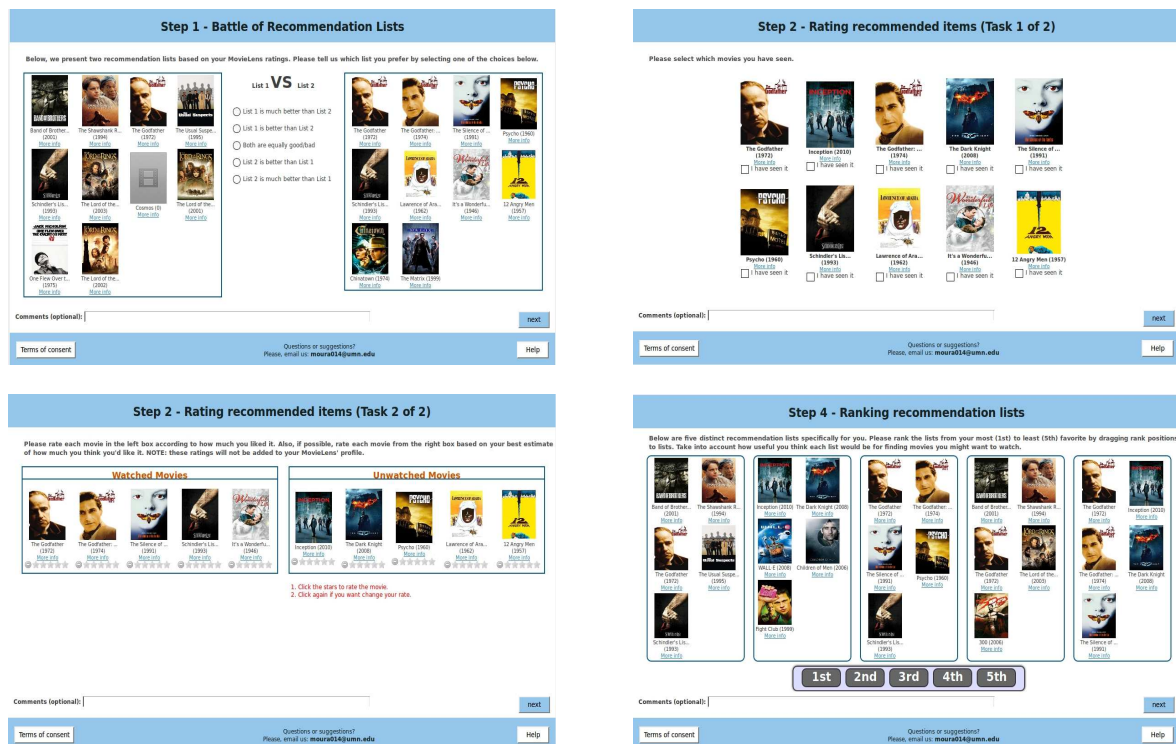


Figure 6.1: Snapshots of the Evaluation System.

6.4 The Web-based Evaluation System

We divided the survey into five distinct steps, such as described below. Figure 6.1 presents some snapshots of the evaluation system.

1. **Battle of Recommendation Lists** - We presented to each participant two distinct lists, each one containing 10 recommendations issued by different methods, and asked him/her to compare the lists. We compared each pair of methods considering all possibilities of arrangements in order to avoid unwilling effects related to the order in which the lists were presented in the interface. So, the pair $\langle LF, Remembrall \rangle$, for instance, becomes different from $\langle Remembrall, LF \rangle$. Since we were not interested in contrasting traditional RSs (i.e., LF and STREAM) against each other, we evaluated 18 distinct pairs of methods and each one was evaluated by thirteen users.
2. **Rating recommended items** - We presented to each participant a recommendation list issued by a method picked at random and ask him/her to perform two tasks. First, we asked the participants to identify recommended movies previously seen. Second, we asked them to rate each previously watched movie according to how much he/she liked it. Also, if possible, they should rate each

unwatched movie based on the best estimate of how much he/she would like it. Each method was evaluated by 46 distinct users in this step.

3. **Questionnaire about the recommendations** - We asked each participant to answer eight questions about the recommendation list presented to him/her in the previous step. We presented the answers of each question using the Likert scale [Albaum, 1997]. The whole questionnaire is presented in Appendix B.
4. **Ranking recommendation lists** - We presented to each participant five distinct lists, which contained five items recommended to him/her by each evaluated RS. Then, we asked each participant to assign to each list only one of five possible rank positions according to how useful he/she thinks each list is for finding movies he/she might want to watch. We found that 214 users completed this step.
5. **Tell us a little about yourself** - Finally, we asked each participant to fill a questionnaire about basic personal and behavioral information, regarding the activity of watching movies. We present the questionnaire used in this step and the options for each question in Appendix B. Among all participants, 211 of them filled the questionnaire.

6.5 User-Centered Results

Analyses on the ‘Battle of Recommendation Lists’, Figure 6.2 (a), show that almost 20% of the participants selected Remembrall as ‘much better’ than any other method. This value was about 10% higher than the second most voted method, ForNonContent. Further, both traditional recommenders (LF and STREAM) were pointed less frequently as a ‘much better’ option. Considering the options ‘much better’ and ‘better’ as positive feedback, we found that Remembrall had the highest percentage of positive feedback (46%), followed by ForNonContent (41%). Also Remembrall and ForNonContent had the lowest percentage of negative feedback (less than 25%), where negative feedback comprises the options ‘much worse’ and ‘worse’. On the other hand, one third of the evaluated users said that all methods, except STREAM, provided similar recommendations.

Additionally, Figure 6.2 (b) shows the percentage of predilection for another method when compared to each baseline. Almost 40% of the participants preferred LF rather than NonContent, but only 27% of the participants preferred LF rather than ForNonContent. Thus, the inclusion of forgotten re-consumable items in the recommendation list seems to bring items perceptible and appreciated for some users. Also,

almost half of the users preferred Remembrall rather than NonContent, which shows an equilibrium of preferences between these methods. On the other hand, ForNonContent was lightly preferred over both. Finally, except for STREAM, which was disliked by most of the users, there was no predominant preference for a single method.

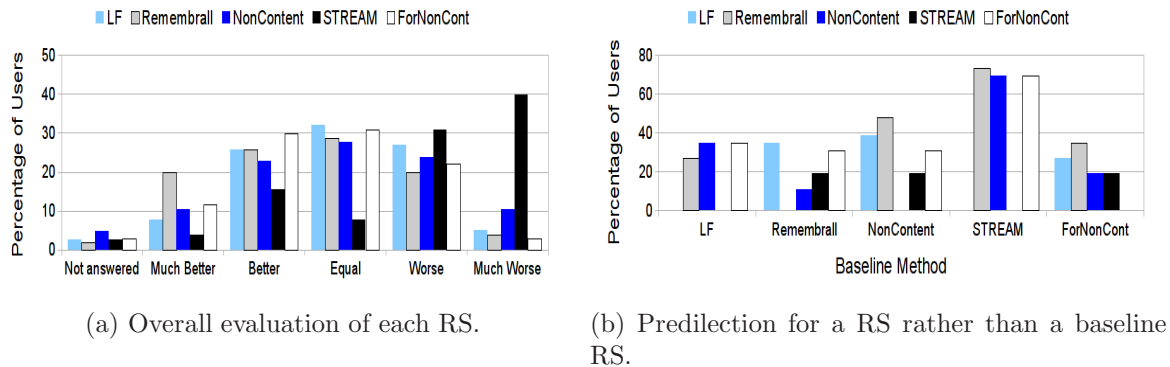
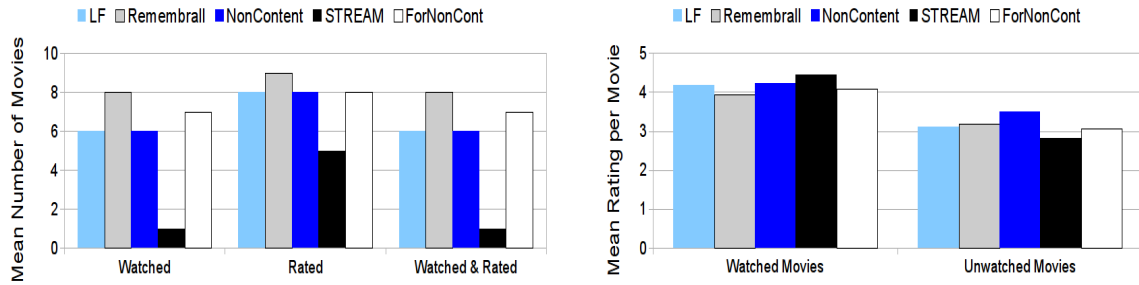


Figure 6.2: Results of ‘Battle of Recommendation Lists’.

Through Figure 6.3, we observe that STREAM presented the smallest number of watched movies per recommendation list. This behavior is sometimes related to ‘risky’ recommendations, providing many new items to users [Ricci et al., 2011]. This would be an explanation for the poor results found for STREAM in the first step. As expected, Remembrall presented the highest number of watched movies. However, we also observed that the number of watched movies is not equal to 10. It means that some users forgot they have watched a movie. We may interpret this fact as an evidence of ‘novelty’ when reintroducing to these users items that they used to like but cannot remember nowadays. Considering the mean rating of the rated movies, there was no relevant difference among the evaluated methods. These results show that rating was not adequate to verify whether users prefer one method over the others, since we have verified, for instance, that most of the users did not like the recommendations provided by STREAM.

The questionnaire about the recommendation lists evinced the perceived value of our recommendation methods, such as shown by Figure 6.4. Remembrall was the method with highest percentage of users for whom recommendations matched their interest (85%). Further, all proposed methods (i.e., Remembrall, NonContent and ForNonContent) matched the interest of at least 72% of the users. LF matched the interest of 62% of the users while STREAM matched the interest of less than 30% of the users. More than half of the users have watched most of the items recommended by our methods (Remembrall, NonContent and ForNonContent) and by LF. On the other hand, recommendations of STREAM were mostly new to most participants.



(a) Mean number of watched and/or rated movies.

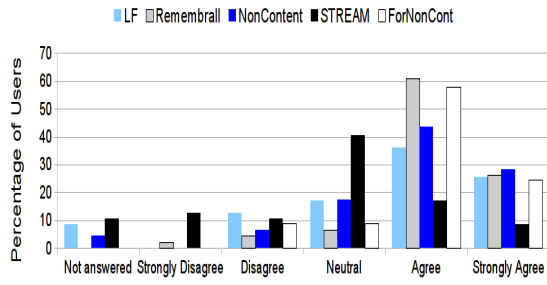
(b) Mean Rating per Movie.

Figure 6.3: Results of 'Rating recommended items'.

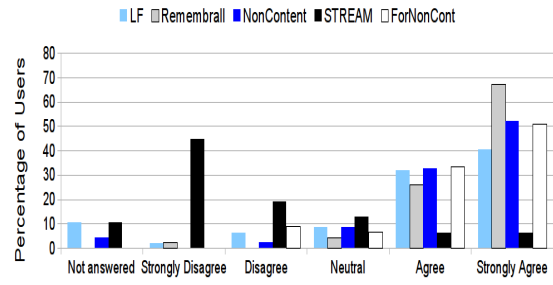
Also, all evaluated methods presented diversified recommendations. Finally, we found that over 60% of participants were satisfied and would reuse systems that issue recommendations based on one the proposed methods. While Remembrall was the method liked by most participants, ForNonContent was deemed as the most useful method for finding movies that users might want to watch.

Figure 6.5 shows results related to the step of 'Ranking recommendation lists'. Figure 6.5 (a) shows that when taking into account the top-5, Remembrall was the method that appeared most often in first rank, followed by LF. In the second rank position, LF and ForNonContent were the most frequently occurring methods. ForNonContent also was the most frequent one in the third rank position. Thus, improvements made by NonContent on LF was not perceptible in the top-5 for some users. Figure 6.5 (b) shows the percentage of time that each method appeared ahead of a baseline in the ranking defined by the participants. Except for STREAM, which was most disliked, there was no absolute predilection for a method over any other. The percentage of times that a given method was ranked ahead of another is around 40% to 50% for most pairs of methods.

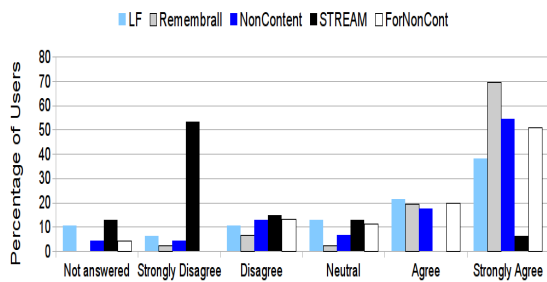
Finally, the questionnaire about personal information revealed important characteristics of the participants, such as shown by Figure 6.6. About 60% are between 25 and 44 years old, and 60% use MovieLens at least monthly. Almost half of the participants watch movies weekly. Around 70% of the participants re-watch at most 25% of all movies him/her have watched. Also, only 6% do not re-watch movies. More than half of the participants reported needing to wait at least one year before re-watching a movie. Less than 50% of the watched movies are blockbusters for 71% of the evaluated users. These statistics show that even for movie recommendations, where the 're-consumption' of items is not common, movies already seen represent recommendations of interest for more than half of the evaluated users. Moreover, as it represents a



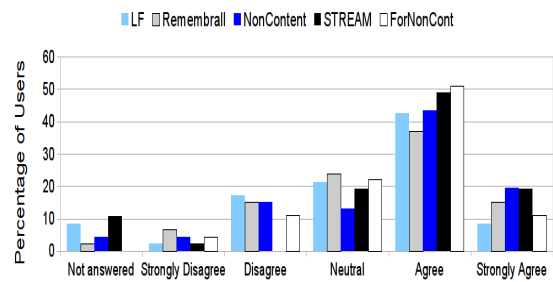
(a) The items recommended to me matched my interests.



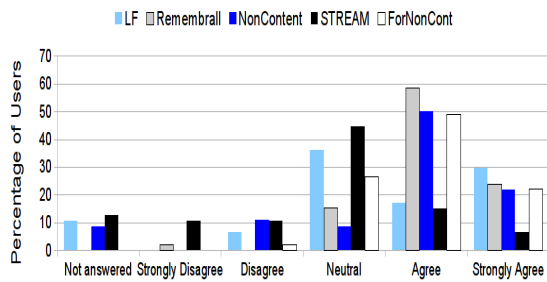
(b) I was familiar with the recommended movies.



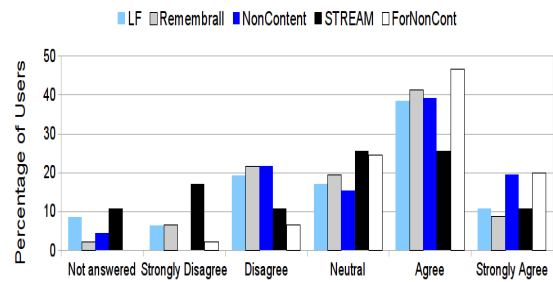
(c) I have watched most of the recommended movies.



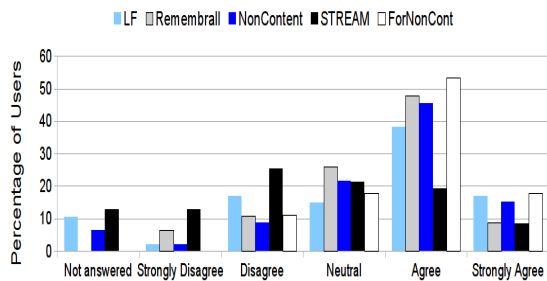
(d) It is a diverse set of movies.



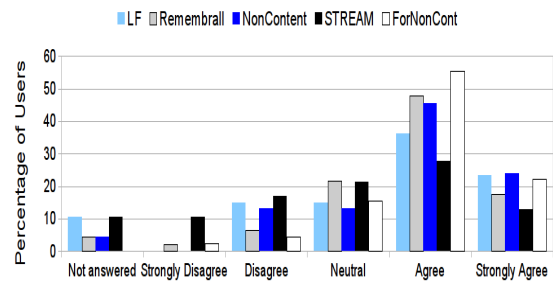
(e) I liked the items recommended to me.



(f) These recommendations are useful for finding movies I might want to watch.

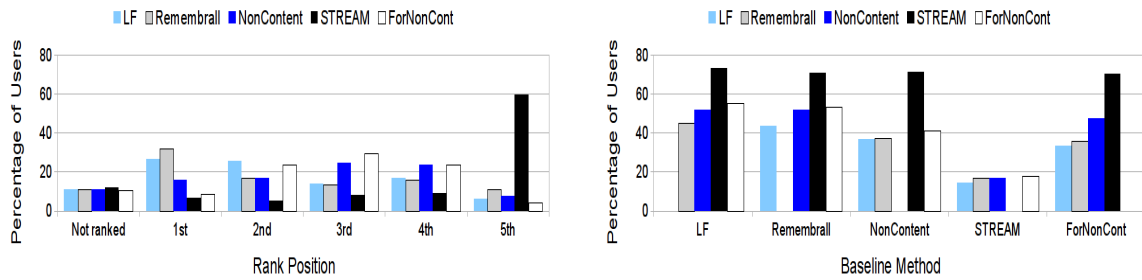


(g) Overall, I am satisfied with the recommendations.



(h) I am willing to use the system that issued these recommendations again.

Figure 6.4: Results of ‘Questionnaire about the recommendations’.

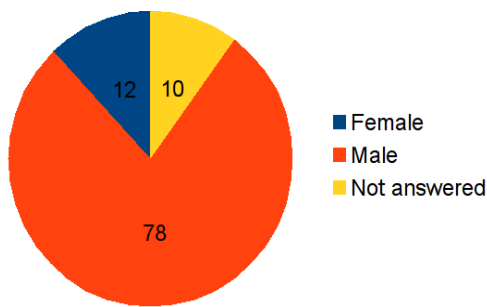


(a) Histogram of ranking assignments.

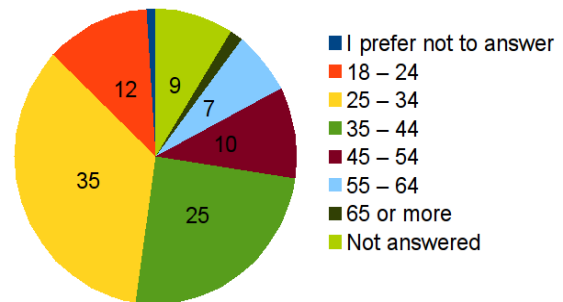
(b) Predilection for a method rather than for the baseline.

Figure 6.5: Results of 'Ranking recommendation lists'.

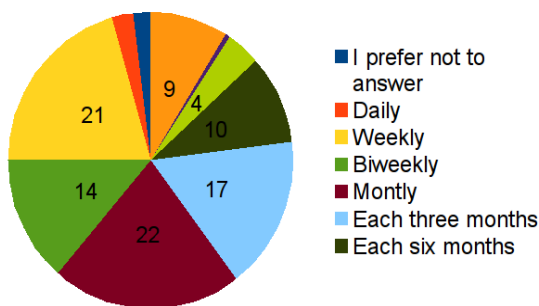
domain with high consumption rate (i.e., weekly for almost half of the users), diversity becomes an important characteristic. Also, users are open to watch movies that are not the most popular in the system, allowing RSs to rescue items close to the tail of the distribution. All of these characteristics demonstrate the usefulness of RSs that address simultaneously forgotten re-consumable items and non-content preference mismatching.



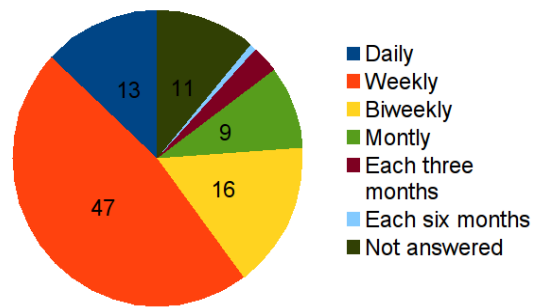
(a) Gender



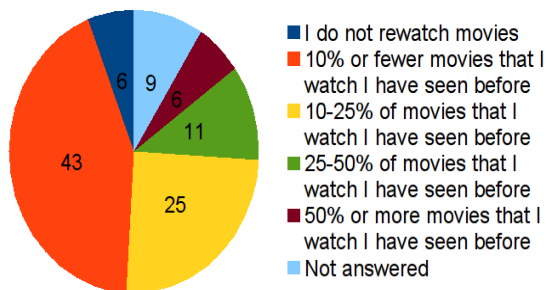
(b) Age



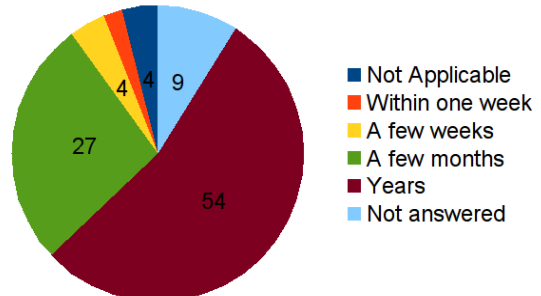
(c) How often do you use MovieLens?



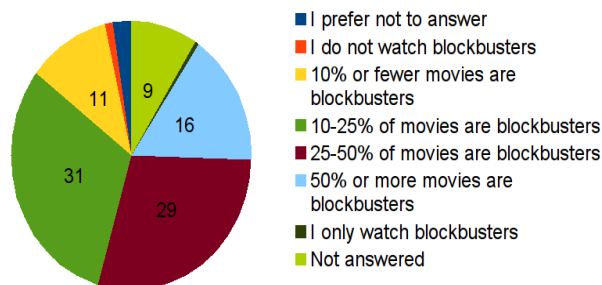
(d) How often do you watch movies?



(e) What percentage of movies that you watch are movies that you have seen before?



(f) In general, how long is it after watching a movie before you would watch it again?



(g) What percentage of movies that you have watched are blockbusters?

Figure 6.6: Results of the questionnaire about personal information of the users.

6.6 Summary

We discussed in this chapter the user study conducted to estimate the perceived value of recommendations issued by the methods proposed in this dissertation. We designed a survey for MovieLens users to contrast the recommendations issued by Remembrall, NonContent and ForNonContent against traditional RSs.

First, we discussed the motivation for this live study. Then, we briefly described the evaluated methods, as well as implementation issues and parameter configurations related to each method. Next, we discussed the methodology of evaluation adopted in the survey. We did not optimize any evaluated RS with the training set used to issue the recommendations for the survey. Also, we consolidated the test set by randomly picking 1,000 distinct users from MovieLens, active in 2013.

Thereafter, we presented the web-based system implemented for this survey. The system includes five steps by which the users should rate recommended movies, rank recommendation lists and fill questionnaires about the recommendations and himself/herself.

Finally, we discussed the main findings on user feedback. While Remembrall was the method liked by most participants, ForNonContent was deemed as the most useful method for finding movies that users might want to watch. We also found that even for movie recommendations, where the ‘re-consumption’ of items is not common, movies already seen represent recommendations of interest for more than half of the evaluated users. In summary, the survey demonstrated the usefulness of RSs that address simultaneously forgotten re-consumable items and non-content preference mismatching.

Chapter 7

Conclusions & Future Work

We start this chapter by restating our research question, reinforcing its context and relevance. Then, we provide a synthesis of empirical findings with respect to the underlying hypotheses evaluated in previous chapters. Next, we summarize the main contributions of this dissertation. Thereafter, we discuss its main limitations. Then, we present promising research directions. The chapter ends with our final remarks.

7.1 Restatement of Thesis

Recommender Systems (RSs) play an important role in many Web applications nowadays, helping users to find their favorite items among a huge number of options. Despite recent advances, there still are several open challenges inherent to RSs, such as proper user taste modeling and data sparsity, among others. This dissertation had as main goal addressing one of these challenges: how to enhance the discovery of items that users would want to consume while not recommending undesirable ones. The prospect of discovery determines the practical value of RSs in many scenarios, since RSs are useful to users when presenting potentially relevant items not easily reachable otherwise. Current efforts to address this challenge focus on proposing hybrid methods, which combine the strengths of distinct RSs and mitigate their weakness. However, even state-of-the-art RSs are still unable to provide adequate recommendations in different real scenarios. Thus, there are opportunities to improve hybrid RSs. We exploited two opportunities unaddressed in the literature, which we stated through the following thesis statement:

State-of-the-art recommenders underexploit two types of information useful to enhance the discovery of relevant items: the long-term history and implicit signals of preference observed on past consumption of each user.

7.2 Empirical Findings

Aiming to handle the complexity inherent to the above statement, we adopted a divide-and-conquer investigation strategy, splitting it into three main underlying hypotheses. We extensively investigated each of these hypotheses in one of the previous chapters. Thus, our main empirical findings are chapter specific and were summarized within the respective empirical chapters: *Forgotten Re-consumable items*; *Non-content Preference Mismatching*; *Combining Forgotten items and Non-content preference*; and *End-user Study*. This section synthesizes these findings to validate the study's three working hypothesis.

1. *State-of-the-art RSs fail to bring back items consumed long ago that are potentially relevant for users nowadays.*
 - a. The evaluated recommendation methods did not even recommend items consumed long ago in two datasets. Meanwhile, about 25% of the recommended items were previously consumed by the target users in the other datasets.
 - b. Analysis of scope confirmed common sense that recommending forgotten items is more relevant for music domains, followed by tag and movie domains.
 - c. We could model forgetfulness and re-consumption using three factors: information of past utility, recency of consumption, and association with current consumption.
 - d. Although the ACT-R framework is effective to describe learning processes, it cannot model 'consumption processes' existing in recommendation domains.
 - e. Besides enhancing diversity, recommending forgotten re-consumable items may bring items deemed as novel in the present, since they have not been consumed for a long time.
2. *State-of-the-art RSs fail to capture the whole extent on which implicit signals of preferences observed on past consumption relate to preferences observed on current consumption.*
 - a. Users exhibited distinct preferences with high variability along each evaluated non-content attribute. We also found that users demonstrated a slightly higher interest towards more popular, similar and recent items than the expected in almost all datasets.

- b. RSs provided recommendations with different biases along each non-content attribute, which also varied according to each dataset.
 - c. All evaluated methods provided recommendations that systematically deviate from the observed user preferences. Indeed, RSs presented recommendation biases stronger than the user preferences, recommending to them items more recent than they usually consume.
 - d. By explicitly modeling non-content preferences, we achieved simultaneous gains with regard to accuracy, novelty and diversity in six major CF methods.
 - e. The proposed method improved recommendations for more than 40% of the users in all datasets, whereas it produced losses for at most 10% of the users.
3. *The two aforementioned algorithmic limitations, when addressed simultaneously, provide complementary enhancements to RSs.*
- a. The methods proposed to rescue forgotten re-consumable items and mitigate non-content preference mismatching provided complementary recommendations in our experiments. Besides low intersection of recommended items, the recommendation lists presented low inter-list similarity.
 - b. The inclusion of forgotten re-consumable items in Top-10 recommendation lists brought items appreciated by users in our live study.
 - c. All proposed methods provided recommendations that matched the interest of at least 72% of the evaluated users in the live study.
 - d. Even for movie recommendations, where the ‘re-consumption’ of items is not common, movies seen long ago represented recommendations of interest for more than half of the evaluated users in the live study.

7.3 Summary of Contributions

The main contributions of this dissertation could be classified into three main groups.

- **Concepts and Problems:** We introduced new concepts and problems inherent to the recommendation task. We formalized the concepts of forgetfulness and re-consumption to model the interest of users in a subset of items consumed long ago. Further, the existence of systematic preferences hidden in metadata and consumption data inspired the definition of non-content attributes.

- **Methods:** We proposed distinct methods to address each of the problems raised by this dissertation. First, we proposed a novel method that better combines information of past utility, recency of consumption and current context of consumption to model forgetfulness and re-consumption. Then, we presented an approach to incorporate explicitly into the recommendations any significant non-content attribute that users follow. Finally, we proposed a hybrid method that combines these two methods.
- **Understanding and Knowledge:** We acquired further knowledge about how to model two unaddressed phenomena that affect the user behavior. Offline and online evaluations evinced the effectiveness of three factors, proposed by a psychological framework for human memory, to model forgetfulness and re-consumption. We also found that all proposed methods effectively may enhance the discovery of potentially relevant items, matching the user interest. Finally, these evaluations confirmed that users, from a real domain, may perceive and appreciate the value of recommendations issued by the proposed methods.

Therefore, we point out a new and relevant research direction for RSs by which significant enhancements may be achieved. To the best of our knowledge, this work is the first effort that effectively exploits forgotten re-consumable items and non-content attributes in recommendation domains. We reported all of these findings along distinct publications [Mourão et al., 2014b,a, 2013, 2011b].

7.4 Limitations of the Work

The study has offered an evaluative perspective on two algorithmic limitations of RSs unaddressed by the literature. As a direct consequence of this methodology, the study faced a number of limitations, which need to be discussed.

1. **Extent of results:** Our findings are purely based on empirical assessments. Hence, we are unable to make strong claims about the best approach to address forgotten re-consumable items and no-content preferences. Also, we cannot ensure that the achieved results are extensible to all domains. Finally, it is still unclear the necessary conditions on which the implemented methods provide gains.
2. **Implementation decisions:** For sake of efficiency, we adopted several simplistic decisions that should be refined to handle the actual conditions of recommendation domains. For instance, we adopted a global linear combination

weight in the algorithm proposed to recommend forgotten re-consumable items. Also, gains related to the algorithm that address non-content attributes are limited by the Top-500 recommendation lists provided by each CF. Whether the items recommended by a given CF do not cover those with characteristics similar to the user non-content interests, our method will not be able to bring improvements. In addition, we need to investigate more elaborate strategies to combine distinct recommendation lists.

3. **Experimental Design:** Finally, we point out the need of designing more robust experiments that balance time demanding and statistical robustness of analysis. Indeed, the experimental design of temporally ordered data in recommendation domains is not well established.

7.5 Recommendation for Future Research

Aligned with the foregoing discussion about the limitations of this dissertation, we highlight as immediate future work three main branches.

1. **Theoretical Analysis:** Deriving theoretically the extent on which each RS incorporates forgetfulness, re-consumption and non-content attributes into its recommendations has important implications. This analysis may allow us to propose even better strategies to model these three pieces of information. Theoretical analysis is a powerful tool to enlarge the scope of our results.
2. **Temporal Evolution:** Although several works agree that static user profiles cannot assess properly the preference of users over time, we did not find studies in recommendation concerned with explaining such temporal evolution [Gauch et al., 2007]. Understanding this evolution would help us to refine the proposed methods to handle forgetfulness, re-consumption and changes on non-content preferences over time. Indeed, we started preliminary studies on these direction, aiming to define temporally robust user profiles [Cardoso et al., 2011; Mourão et al., 2011a].
3. **Learning to rank:** Learning to rank (LTR) provides a framework particularly useful to combine distinct recommendation lists [Liu, 2009; Shi et al., 2010; Sun et al., 2012]. By using the user histories as training data, LTR based-methods produce a rank function to better order the subset of items selected by RSs. This strategy could also be personalized, defining a distinct rank function for each

user. Thus, we can exploit these methods to combine forgotten re-consumable items with items that suit the user non-content preferences.

7.6 Final Remarks

In summary, this dissertation showed that discovering potentially relevant items goes beyond presenting unknown items to users. Known items forgotten over time and items that better suit preference signals hidden in consumption data are promising to enhance the discovery capability of RSs. Indeed, online analyses allowed us to conclude that, when combining these items, we may address distinct pieces of the user taste, enhancing the user experience with RSs.

Appendix A

Qualitative Analysis of the Oblivion Problem

Besides quantifying the relevance of the Oblivion Problem in real domains, we are concerned with the reasons by which it would emerge. Based on the definition of the Oblivion Problem, presented in Section 3.1.1, we derive three main requirements to qualify its existence:

1. **Usefulness:** Recommendation is useful for rescuing items not reachable by users;
2. **Oblivion:** Users forget consumed items through time;
3. **Re-consumption:** Users want to re-consume some already known items.

The Usefulness requirement is the basic assertion of recommendation scenarios. Whether recommendation is not useful at all, rescuing forgotten items also would not be. The Oblivion requirement refers to the need to handle ‘forgetful’ items, otherwise there is nothing to be remembered. Finally, users must desire to re-consume what he/she enjoyed in the past. Aiming to provide a qualitative framework of existence inspection for the Oblivion Problem, we analyze the strength of each requirement along a non-exhaustive set of conditions¹:

- **Usefulness** - Recommendation is useful when:

1. The total amount of available items is huge; (e.g., song recommendation)

¹This is a preliminary set of conditions related to the characteristics of recommendation domains, types of available items and ways that users interact with the domain and consume items. We raised them by revisiting some conclusions, results and characterizations discussed in the literature [Ricci et al., 2011; Herlocker et al., 2004].

2. Decision making is difficult due to a large number of variables or to the technical nature of such variables; (e.g., car recommendation)
 3. Users exhibit interest in non-trivial items. (e.g., non-popular landmark recommendation)
- **Oblivion** - Users forget items when:
 1. Users consume individually a large amount of items over time; (e.g., movie recommendation)
 2. Users exhibit episodic interest about items; (e.g., TV buying recommendation)
 3. The domain exhibits a temporal skewness towards recently consumed or released items. (e.g., news recommendation)
 - **Re-consumption** - Users want to re-consume some items when:
 1. Users are more willing to repeat positive experiences than trying novel ones; (e.g., hotel recommendation)
 2. Items belong to ordinary habits of the users; (e.g., grocery recommendation)
 3. A known item is strongly associated to a current context; (e.g., tag recommendation)
 4. Specific contexts demand consumption of known items. (e.g., restaurant recommendation for celebration of a special date)

Although these conditions are intuitive and qualitatively easy to be identified in distinct domains, we believe that quantifying some of them is a challenging task. Tables A.1 and A.2 illustrate the evaluation of the proposed conditions on distinct domains taken as relevant and not relevant for the Oblivion Problem, respectively.

Table A.1: Domains where the Oblivion Problem presents strong relevance.

Domain	Usefulness	Oblivion	Re-consumption
Songs	<p>Very Strong</p> <ol style="list-style-type: none"> 1. Musical domains have hundreds of thousands or even millions of distinct items. 2. Users have no difficulty in deciding which tracks they would like to listen. 3. The consumption is mostly focused on popular items, although the search for few non-trivial items is common as well as the existence of small and specialized niches of non-trivial items. 	<p>Very Strong</p> <ol style="list-style-type: none"> 1. Users listen to a large set of tracks. 2. User interest is not episodic, since users listen to songs frequently. 3. The domain exhibits strong skewness towards recency and popularity, making more available in the system new and recently consumed tracks. 	<p>Very Strong</p> <ol style="list-style-type: none"> 1. Users frequently listen to songs they like repeatedly. Some users listen more often to few distinct tracks than many different ones. 2. Rather than ordinary habits, tracks represent momentary tastes. However, we may observe seasonal musical habits, such as listening to Christmas carols in December or romantic songs during the Valentine’s Day. 3. The context defined by an artist, a band or an album often affects the desire for re-listening to known tracks associated with this context.
Tags	<p>Very Strong</p> <ol style="list-style-type: none"> 1. There is a huge and dynamic set of tags since they comprise free-content generated by users. 2. The large amount of options, the need to better organize items and find tags easily recognized by other users as appropriate often make tagging a challenging process. 3. Tagging is mostly based on using most popular tags. However, non-trivial tags, related to subjective qualifiers or to non-trivial items, may be used. 	<p>Very Strong</p> <ol style="list-style-type: none"> 1. As users assign more than one tag to each consumed item, the total number of tags used by each user is even greater than the number of consumed items in several domains. 2. Tags are not used episodically, since users assign tags to a large percentage of consumed items. 3. These domains exhibit strong temporal skewness towards recency and popularity, prioritizing to display in the system popular and recently used tags. 	<p>Strong</p> <ol style="list-style-type: none"> 1. Since a main goal of tags is to connect closely related items, re-consumption becomes a primary characteristic of tag usage. 2. Whether the items being tagged represent ordinary habits or are highly correlated to frequently consumed items, tags would indirectly represent habits. 3. Tagging is mostly based on using known tags strongly associated to a current context.
Grocery	<p>Strong</p> <ol style="list-style-type: none"> 1. Markets have dozens of thousands of items usually related to distinct brands. 2. The decision on which type of item to buy is generally related to well-defined personal needs. Sometimes there is a difficulty that stems from brand choice, given differences in price, quality, popularity, among others. 3. Specialized items, such as exotic culinary or luxury products, are examples of non-trivial items usually consumed. 	<p>Strong</p> <ol style="list-style-type: none"> 1. Users buy a large number of items periodically according to personal needs. 2. Grocery shopping is not episodic, however few items could be bought episodically (e.g., light bulbs) 3. There is a strong skewness toward items consumed more often. 	<p>Strong</p> <ol style="list-style-type: none"> 1. Users tend to keep buying known and appreciated items. Sales and discounts of specific items, however, would make users try new products. 2. Most of the consumed items represent ordinary habits of consumption. 3. Several products are naturally correlated to others in the user’s purchases, such as pasta and sauce, bread and butter, meat and seasoning.

Table A.2: Domains where the Oblivion Problem is not relevant.

Domain	Usefulness	Oblivion	Re-consumption
Movies	<p>Very Strong</p> <ol style="list-style-type: none"> 1. Movie domains have dozens of thousands of distinct items. 2. Deciding which movie to watch would be time consuming, since it usually involves evaluating features such as genre, director, casting, synopsis, among others. 3. Consumption of non-trivial items is less common but possible in this domain. 	<p>Strong</p> <ol style="list-style-type: none"> 1. Users watch several distinct movies over long periods of time. 2. User interest is not episodic, since users watch movies frequently. 3. The domain exhibits strong temporal skewness towards recency and popularity, prioritizing to display in the system new and popular items. 	<p>Very Weak</p> <ol style="list-style-type: none"> 1. Users are more willing to watch new movies than rewatch known ones. 2. Rather than an ordinary habit, movies represent taste. 3. In general, there are associations between distinct items, such as sharing the same director, but these associations do not induce users to re-consume. 4. Specific and infrequent contexts, such as Valentine’s Day, may cause users to watch a movie again.
News	<p>Very Strong</p> <ol style="list-style-type: none"> 1. There is a huge and dynamic volume of news, such as observed in streaming data. 2. Users easily decide to read a news report or not based on the title or associated keywords. 3. For sake of credibility, users usually look for popular news or those published by ‘authorities’ related to each topic. Thus, non-trivial news are less frequently consumed. 	<p>Very Strong</p> <ol style="list-style-type: none"> 1. Users read a huge amount of distinct news within short periods of time. 2. User interest is not episodic, since users read news frequently. 3. Domain is completely skewed towards recency and popularity, prioritizing to display in the system new and popular items. 	<p>Almost Nonexistent</p> <ol style="list-style-type: none"> 1. News items are highly correlated with time, since most of its value lies in providing new information. For this reason, users rarely want to re-read news. 2. News items do not comprise habits. Although users may habitually read specific sections, such as economics, the news articles themselves are new. 3. While recent news are commonly similar to past ones, this phenomenon does not reflect on willingness to reread known news items. 4. Research contexts or specific interest in real world entities or concepts may be seen as contexts that require re-reading some already known news.
Courses	<p>Strong</p> <ol style="list-style-type: none"> 1. There is a large number of online courses in this kind of domain. 2. Evaluating the quality of a course is time consuming and often difficult due to the limited amount of available information. 3. In general, users look for more reputable courses which consequently become more popular, enrolling less frequently in less popular courses. 	<p>Weak</p> <ol style="list-style-type: none"> 1. Users do not enroll in a large number of courses, since a course is a time demanding activity. 2. Some users exhibit an episodic interest about courses. 3. Domain with a strong skew towards popularity and authority. 	<p>Almost Nonexistent</p> <ol style="list-style-type: none"> 1. Users are more willing to enroll in new courses than revisiting known ones. 2. Courses do not comprise habits. 3. There are strongly connected courses (e.g., calculus I and calculus II) but such association does not bring a desire for revisiting known courses. 4. In restricted scenarios, such as new professors consolidating their teaching material, users may be interested in enrolling again in a known course.

Appendix B

Questionnaires Used in the User Studies

Questionnaire for the third step of the survey:

1. The items recommended to me matched my interests.
2. I was familiar with the recommended movies.
3. I have watched most of the recommended movies.
4. It is a diverse set of movies.
5. I liked the items recommended to me.
6. These recommendations are useful for finding movies I might want to watch.
7. Overall, I am satisfied with the recommendations.
8. I am willing to use the system that issued these recommendations again.

Questionnaire for the fourth step of the survey:

1. Gender
2. Age
3. How often do you use MovieLens?

I prefer not to answer; Daily; Weekly; Biweekly; Monthly Each three months; Each six months; Never

4. How often do you watch movies?

I prefer not to answer; Daily; Weekly; Biweekly; Monthly Each three months; Each six months; Never

5. What percentage of the movies that you watch are movies that you've seen before?

I prefer not to answer; I don't rewatch movies; 10% or fewer of movies that I watch I've seen before; 10-25% of movies that I watch I've seen before; 25-50% of movies that I watch I've seen before; 50% or more movies that I watch I've seen before

6. In general, how long is it after watching a movie before you would watch it again?

N/A; I prefer not to answer; Within one week; A few weeks; A few months; Years

7. What percentage of movies that you have watched are blockbusters?

I prefer not to answer; I don't watch blockbusters; 10% or fewer of movies are blockbusters; 10-25% of movies are blockbusters; 25-50% of movies are blockbusters; 50% or more movies are blockbusters; I only watch blockbusters

Bibliography

- Abbasse, Z. and Mirrokni, V. S. (2007). A recommender system based on local random walks and spectral methods. In *Proceedings of the 9th WebKDD and 1st SNA-KDD*, pages 102--108. Springer.
- Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23(1):103--145.
- Adomavicius, G. and Tuzhilin, A. (2001). Extending recommender systems: A multi-dimensional approach. In *Proceedings of the Workshop on Intelligent Techniques for Web Personalization (ITWP2001)*, pages 4--6.
- Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 17(6):734--749.
- Adomavicius, G. and Zhang, J. (2012). Stability of recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 30(4):23:1--23:31.
- Albaum, G. (1997). The likert scale revisited. *Journal-Market research society*, 39:331-348.
- Anand, S. S. and Mobasher, B. (2007). From web to social web: Discovering and deploying user and content profiles. chapter Contextual Recommendation, pages 142--160. Springer-Verlag, Berlin, Heidelberg.
- Anderson, C. (2006a). *The long tail*. Gramedia Pustaka Utama.
- Anderson, C. (2006b). *The long tail: How endless choice is creating unlimited demand*. Random House Business Books.

- Anderson, J. R., Fincham, J. M., and Douglass, S. (1999). Practice and retention: A unifying analysis. *Journal of Experimental Psychology Learning Memory and Cognition*, 25:1120--1136.
- Anderson, J. R. and Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological science*, 2(6):396--408.
- Balabanović, M. and Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of ACM*, 40(3):66--72.
- Bao, X., Bergman, L., and Thompson, R. (2009). Stacking recommendation engines with additional meta-features. In *Proceedings of the 3th ACM conference on Recommender systems (RecSys)*, pages 109--116. ACM.
- Basu, C., Hirsh, H., and Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the National Conference on Artificial Intelligence (NCAI)*, pages 714--720. JOHN WILEY & SONS LTD.
- Bell, R. M. and Koren, Y. (2007). Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proceedings of the Seventh International Conference on Data Mining (ICDM)*, pages 43--52. IEEE.
- Bellogín, A., Said, A., and de Vries, A. P. (2014). The magic barrier of recommender systems – no magic, just ratings. In Dimitrova, V., Kuflik, T., Chin, D., Ricci, F., Dolog, P., and Houben, G.-J., editors, *User Modeling, Adaptation, and Personalization*, volume 8538 of *Lecture Notes in Computer Science*, pages 25--36. Springer International Publishing.
- Bennett, J. and Lanning, S. (2007). The netflix prize. In *Proceedings of KDD Cup and Workshop*, volume 2007. Citeseer.
- Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., and Lamere, P. (2011). The million song dataset. In *Proceedings of the 12th International Society for Music Information Retrieval (ISMIR)*.
- Billsus, D. and Pazzani, M. (1998). Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML)*, volume 46--54, page 98.
- Billsus, D. and Pazzani, M. J. (2000). User modeling for adaptive news access. *User modeling and user-adapted interaction*, 10(2-3):147--180.

- Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, UAI'98, pages 43--52, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction (UMUAI)*, 12(4):331--470.
- Burke, R. (2007). The adaptive web. chapter Hybrid Web Recommender Systems, pages 377--408. Springer-Verlag, Berlin, Heidelberg.
- Cacheda, F., Carneiro, V., Fernández, D., and Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web (TWEB)*, 5(1):2:1--2:33.
- Campos, P., Díez, F., and Sánchez-Montañés, M. (2011). Towards a more realistic evaluation: Testing the ability to predict future tastes of matrix factorization-based recommenders. In *Proceedings of the fifth ACM conference on Recommender systems (RecSys)*, pages 309--312. ACM.
- Campos, P. G., Díez, F., and Cantador, I. (2013). Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction (UMUAI)*, pages 1--53.
- Candillier, L., Jack, K., Fessant, F., and Meyer, F. (2009). State-of-the-art recommender systems. *Collaborative and Social Information Retrieval and Access Techniques for Improved User Modeling*, pages 1--22.
- Cardoso, A., Rocha, D., Mourão, F., Sachetto, R., Rocha, L., and Meira Jr., W. (2011). A characterization methodology of evolutionary behavior in recommender systems. In *Proceedings of the 7th International Conference on Web Information Systems and Technologies (WEBIST)*, Noordwijkerhout, Netherlands.
- Cebrián, T., Planagumà, M., Villegas, P., and Amatriain, X. (2010). Music recommendations with temporal context awareness. In *Proceedings of the 4th ACM conference on Recommender systems (RecSys)*, RecSys '10, pages 349--352, New York, NY, USA. ACM.
- Cosley, D., Lam, S. K., Albert, I., Konstan, J. A., and Riedl, J. (2003). Is seeing believing?: how recommender system interfaces affect users' opinions. In *Proceedings*

- of the SIGCHI Conference on Human Factors in Computing Systems*, pages 585--592. ACM.
- Cremonesi, P. and Turrin, R. (2010). Controlling consistency in top-n recommender systems. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 919--926. IEEE.
- Deshpande, M. and Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143--177.
- Ding, Y. and Li, X. (2005). Time weight collaborative filtering. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM)*, CIKM '05, pages 485--492, New York, NY, USA. ACM.
- Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011). *Collaborative filtering recommender systems*. Now Publishers Inc.
- Fleder, D. M. and Hosanagar, K. (2007). Recommender systems and their impact on sales diversity. In *Proceedings of the 8th ACM Conference on Electronic Commerce (CEC)*, pages 192--199. ACM.
- Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2011). Mymedialite: a free recommender system library. In *Proceedings of the 5th ACM conference on Recommender systems (RecSys)*, RecSys '11, pages 305--308, New York, NY, USA. ACM.
- Gauch, S., Speretta, M., Chandramouli, A., and Micarelli, A. (2007). The adaptive web. chapter User profiles for personalized information access, pages 54--89. Springer-Verlag, Berlin, Heidelberg.
- Ge, M., Delgado-Battenfeld, C., and Jannach, D. (2010). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems (RecSys)*, RecSys '10, pages 257--260, New York, NY, USA. ACM.
- Golder, S. A. and Huberman, B. A. (2006). Usage patterns of collaborative tagging systems. *Journal of information science*, 32(2):198--208.
- Herlocker, J., Konstan, J., and Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, 5(4):287--310.

- Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5--53.
- Hipp, J., Güntzer, U., and Nakhaeizadeh, G. (2000). Algorithms for association rule mining—a general survey and comparison. *ACM sigkdd explorations newsletter*, 2(1):58--64.
- Hoashi, K., Matsumoto, K., and Inoue, N. (2003). Personalization of user profiles for content-based music retrieval based on relevance feedback. In *Proceedings of the Eleventh ACM International Conference on Multimedia (MULTIMEDIA)*, MULTIMEDIA '03, pages 110--119, New York, NY, USA. ACM.
- Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):89--115.
- Hu, R. and Pu, P. (2011). Enhancing recommendation diversity with organization interfaces. In *Proceedings of the 16th international conference on Intelligent user interfaces (IUI)*, pages 347--350. ACM.
- Jambor, T. and Wang, J. (2010). Optimizing multiple objectives in collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems (RecSys)*, pages 55--62. ACM.
- Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender systems: an introduction*. Cambridge University Press.
- Jung, S. Y., Hong, J.-H., and Kim, T.-S. (2005). A statistical model for user preference. *Transactions on Knowledge and Data Engineering (TKDE)*, 17(6):834--843.
- Kawamae, N. (2010). Serendipitous recommendations via innovators. In *Proceedings of the 33rd international ACM Conference on Research and development in information retrieval (SIGIR)*, SIGIR '10, pages 218--225, New York, NY, USA. ACM.
- Khrouf, H. and Troncy, R. (2013). Hybrid event recommendation using linked data and user diversity. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys)*, RecSys '13, pages 185--192, New York, NY, USA. ACM.
- Knijnenburg, B., Meesters, L., Marrow, P., and Bouwhuis, D. (2010). User-centric evaluation framework for multimedia recommender systems. In Daras, P. and Ibarra, O., editors, *User Centric Media*, volume 40 of *Lecture Notes of the Institute for*

- Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 366--369. Springer Berlin Heidelberg.
- Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., and Newell, C. (2012). Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction (UMUAI)*, 22(4-5):441--504.
- Konstan, J. A. and Riedl, J. (2012). Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction (UMUAI)*, pages 1--23.
- Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 426--434, New York, NY, USA. ACM.
- Koren, Y. (2009). Collaborative filtering with temporal dynamics. In *Proceeding of the 15th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 447--456. ACM New York, NY, USA.
- Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30--37.
- Lam, X., Vu, T., Le, T., and Duong, A. (2008). Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd ACM International Conference on Ubiquitous Information Management and Communication (ICUIMC)*, pages 208--211. ACM.
- Lathia, N., Hailes, S., and Capra, L. (2009). Evaluating collaborative filtering over time. In *In Proceedings of the ACM SIGIR Workshop on the Future of IR Evaluation*, Boston, USA.
- Lathia, N., Hailes, S., Capra, L., and Amatriain, X. (2010). Temporal diversity in recommender systems. In *Proceeding of the 33rd international ACM Conference on Research and development in information retrieval (SIGIR)*, pages 210--217, New York, NY, USA. ACM.
- Lebiere, C. (1999). The dynamics of cognition: An act-r model of cognitive arithmetic. *Kognitionswissenschaft*, 8(1):5--19.
- Levy, M. and Bosteels, K. (2010). Music recommendation and the long tail. In *Proceedings of the 1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys*, Barcelona, Spain.

- Li, Q. and Kim, B. (2004). Constructing user profiles for collaborative recommender system. *Advanced Web Technologies and Applications*, pages 100--110.
- Li, X. and Murata, T. (2012). Multidimensional clustering based collaborative filtering approach for diversified recommendation. In *Proceedings of the 7th International Conference on Computer Science & Education (ICCSE)*, pages 905--910. IEEE.
- Liese, F. and Miescke, K.-J. (2008). *Statistical decision theory: estimation, testing, and selection*. Springer.
- Lipczak, M. (2008). Tag recommendation for folksonomies oriented towards individual users. *ECML PKDD discovery challenge*, 84.
- Liu, M. and Jiang, X. (2011). Modeling user and item biases with gaussian distribution for collaborative filtering. In *Proceedings of the Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, volume 3, pages 2070--2073. IEEE.
- Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225--331.
- Lops, P., Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 73--105. Springer US.
- McAuley, J. and Leskovec, J. (2013). Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys)*, RecSys '13, pages 165--172, New York, NY, USA. ACM.
- McGinty, L. and Smyth, B. (2003). On the role of diversity in conversational recommender systems. *Case-Based Reasoning Research and Development*, pages 1065--1065.
- McNee, S. M., Riedl, J., and Konstan, J. A. (2006). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *Proceedings of the CHI Extended Abstracts on Human Factors in Computing Systems (CHI EA)*, CHI EA '06, pages 1097--1101, New York, NY, USA. ACM.
- McSherry, D. (2002). Diversity-conscious retrieval. *Advances in Case-Based Reasoning*, pages 27--53.

- Mellon, J. R. A. R. K. et al. (2007). *How can the human mind occur in the physical universe?* Oxford University Press, USA.
- Meyer, F., Fessant, F., Clérot, F., and Gaussier, E. (2012). Toward a new protocol to evaluate recommender systems. *arXiv preprint arXiv:1209.1983*.
- Mooney, R. and Roy, L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195--204. ACM.
- Mourão, F., Araújo, C., Rocha, L., Konstan, J. A., and Meira Jr, W. (2014a). On the pursuit of forgotten re-consumable items in recommendation domains. *Manuscript submitted for publication*.
- Mourão, F., Cardoso, A., Rocha, L., and Meira Jr., W. (2011a). Extraction of evolutive features in recommendation domains. In *Proceedings of the Iadis International Conference WWW/Internet*, Rio de Janeiro, Brazil.
- Mourão, F., Fonseca, C., Araújo, C., and Meira Jr, W. (2011b). The oblivion problem: Exploiting forgotten items to improve recommendation diversity. In *Proceedings of the DiveRS: Workshop on Novelty and Diversity in Recommender Systems*, Chicago, IL, USA.
- Mourão, F., Rocha, L., Araújo, C., Konstan, J. A., and Meira Jr, W. (2014b). Combining forgotten items and non-content attributes through hybrid recommendation method. *Manuscript submitted for publication*.
- Mourão, F., Rocha, L., Konstan, J. A., and Meira Jr, W. (2013). Exploiting non-content preference attributes through hybrid recommendation method. In *Proceedings of the 7th ACM conference on Recommender systems (RecSys)*, pages 177--184. ACM.
- Nicholson, W. and Snyder, C. M. (2011). *Microeconomic Theory: Basic Principles and Extensions*. CengageBrain. com.
- Park, Y. and Tuzhilin, A. (2008). The long tail of recommender systems and how to leverage it. In *Proceedings of the ACM conference on Recommender systems (RecSys)*, pages 11--18. ACM.
- Pavlik, P. I. and Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science*, 29(4):559--586.

- Pennock, D., Horvitz, E., Giles, C., et al. (2000). Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *Proceedings of the National Conference on Artificial Intelligence (NCAI)*, pages 729--734. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
- Pu, P., Chen, L., and Hu, R. (2011). A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems (RecSys)*, pages 157--164. ACM.
- Rafter, R., O'Mahony, M. P., Hurley, N. J., and Smyth, B. (2009). What have the neighbours ever done for us? a collaborative filtering perspective. In *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization (UMAP)*, UMAP '09, pages 355--360, Berlin, Heidelberg. Springer-Verlag.
- Rana, C. and Jain, S. K. (2012). A study of the dynamic features of recommender systems. *Artificial Intelligence Review*, pages 1--13.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the ACM conference on Computer supported cooperative work (CSCW)*, pages 175--186. ACM.
- Resnick, P. and Varian, H. (1997). Recommender systems. *Communications of the ACM*, 40(3):56--58.
- Ribeiro, M., Lacerda, A., Veloso, A., and Ziviani, N. (2012). Pareto-efficient hybridization for multi-objective recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems (RecSys)*, pages 19--26. ACM.
- Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 1--35. Springer US.
- Said, A., Fields, B., Jain, B. J., and Albayrak, S. (2013). User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm. In *Proceedings of the 2013 conference on Computer supported cooperative work (CSCW)*, pages 1399-1408. ACM.
- Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). Application of dimensionality reduction in recommender system—a case study. In *Proceedings of the ACM Web Mining for E-Commerce Workshop (WebKDD)*. Citeseer.

- Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web (WWW)*, pages 285--295. ACM.
- Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th ACM Conference on Research and development in information retrieval (SIGIR)*, pages 253--260. ACM Press.
- Schwartz, B. (2005). *The paradox of choice: Why more is less*. Harper Perennial.
- Shi, Y., Larson, M., and Hanjalic, A. (2010). List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems (RecSys)*, pages 269--272. ACM.
- Smyth, B. and McClave, P. (2001). Similarity vs. diversity. *Case-Based Reasoning Research and Development*, pages 347--361.
- Song, Y., Dixon, S., and Pearce, M. (2012). A survey of music recommendation systems and future perspectives. In *Proceedings of the 9th international symposium on computer music modelling and retrieval (CMMR)*, pages 19--22.
- Stern, D. H., Herbrich, R., and Graepel, T. (2009). Matchbox: large scale online bayesian recommendations. In *Proceedings of the 18th international conference on World wide web (WWW)*, pages 111--120, New York, NY, USA. ACM.
- Su, X. and Khoshgoftaar, T. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4--23.
- Sun, J., Faloutsos, C., Papadimitriou, S., and Yu, P. (2007). Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 687--696. ACM.
- Sun, J., Wang, S., Gao, B. J., and Ma, J. (2012). Learning to rank for hybrid recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM)*, pages 2239--2242. ACM.
- Takács, G., Pilászy, I., Németh, B., and Tikk, D. (2008). Investigation of Various Matrix Factorization Methods for Large Recommender Systems. In *Proceedings of the 2nd Netflix-KDD Workshop*. ACM New York, NY, USA.

- Tang, X. and Zhou, J. (2013). Dynamic personalized recommendation on sparse data. *Transactions on Knowledge and Data Engineering (TKDE)*, 25(12):2895--2899.
- van Maanen, L. and Marewski, J. N. (2009). Recommender systems for literature selection: A competition between decision making and memory models. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*.
- Van Maanen, L., Van Rijn, H., Van Grootel, M., Kemna, S., Klomp, M., and Scholtens, E. (2010). Personal publication assistant: Abstract recommendations by a cognitive model. *Cognitive Systems Research*, 11(1):120--129.
- Vargas, S. and Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the 5th ACM conference on Recommender systems (RecSys)*, pages 109--116. ACM.
- Wu, J. and Li, T. (2008). A Modified Fuzzy C-Means Algorithm For Collaborative Filtering.
- Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., and Sun, J. (2010). Temporal recommendation on graphs via long-and short-term preference fusion. In *Proceedings of the 16th ACM ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 723--732. ACM.
- Xu, G., Zhang, Y., and Yi, X. (2008). Modelling user behaviour for web recommendation using lda model. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 3, pages 529--532. IEEE.
- Yang, D., Chen, T., Zhang, W., Lu, Q., and Yu, Y. (2012). Local implicit feedback mining for music recommendation. In *Proceedings of the sixth ACM conference on Recommender systems (RecSys)*, RecSys '12, pages 91--98, New York, NY, USA. ACM.
- Yin, H., Cui, B., Li, J., Yao, J., and Chen, C. (2012). Challenging the long tail recommendation. *Proceedings of the VLDB Endowment*, 5(9):896--907.
- Yu, K., Schwaighofer, A., Tresp, V., Xu, X., and Kriegel, H. (2004). Probabilistic memory-based collaborative filtering. *Transactions on Knowledge and Data Engineering (TKDE)*, 16(1):56--69.

- Zhang, M. and Hurley, N. (2008). Avoiding monotony: improving the diversity of recommendation lists. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 123--130, New York, NY, USA. ACM.
- Zhang, Y. C., Séaghdha, D. Ó., Quercia, D., and Jambor, T. (2012). Auralist: introducing serendipity into music recommendation. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 13--22. ACM.
- Zhou, T., Kuscsik, Z., Liu, J., Medo, M., Wakeling, J., and Zhang, Y. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511--4515.