Escuela Politécnica Superior

Departamento de Ingeniería Informática

Recommender Systems and Time Context: Characterization of a Robust Evaluation Protocol to Increase Reliability of Measured Improvements

Dissertation written by

Pedro G. Campos Soto

under the supervision of

Fernando Díez Rubio

and

Iván Cantador Gutiérrez

Madrid, 14th October 2013

PhD thesis title: Recommender Systems and Time Context: Characterization of a Robust Evaluation Protocol to Increase Reliability of Measured Improvements

Author: **Pedro G. Campos Soto**

Affiliation: Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid, Spain

Supervisors: **Fernando Díez Rubio**
Universidad Autónoma de Madrid, Spain

Iván Cantador Gutiérrez
Universidad Autónoma de Madrid, Spain

Date: 14th October 2013

Committee: **Roberto Moriyón Salomon**
Universidad Autónoma de Madrid, Spain

Pablo Castells Azpilicueta
Universidad Autónoma de Madrid, Spain

Juan Huete Guadix
Universidad de Granada, Spain

Julio Gonzalo Arroyo
Unversidad Nacional de Educación a Distancia, Spain

Roi Blanco González
Yahoo! Research Barcelona, Spain

Contents

List of figures

List of tables

Abstract

Recommender Systems (RS) aim to help users with information access and retrieval tasks, suggesting items –products or services– according to past preferences –interests, tastes– in certain contexts. For such purpose, one of the most studied contexts is the so-called temporal context, which has originated an already extensive research area, known as Time-Aware Recommender Systems (TARS).

Despite the large number of approaches and advances on TARS, in the literature, reported results and conclusions about how to exploit time information seem to be contradictory. Although several reasons could explain such contradictory findings, in this thesis we hypothesize that TARS evaluation plays a fundamental role. The existence of multiple evaluation methodologies and metrics makes it possible to find some evaluation protocol suitable for a particular recommendation approach, but ineligible or non-retributive for others. Problems that arise from this situation represent an impediment to fairly compare results and conclusions reported in different studies, making complex the identification of the best recommendation approach for a given task. Moreover, the review of published work shows that most of the existing TARS have been developed for diminishing the error in the prediction of user preferences (ratings) for items. However, nowadays the RS focus is shifting towards finding (lists of) items relevant for the target user. Also, the use of RS in diverse tasks lets develop new applications where time context information can serve as a distinctive input.

In this thesis we analyze how time context information has been exploited in the RS literature, in order to a) characterize a robust protocol that lets conduct fair evaluations of new TARS, and facilitate comparisons between published performance results; and b) better exploit time context information in different recommendation tasks. Aiming to accomplish such goals, we have identified key methodological issues regarding offline evaluation of TARS, and propose a methodological framework that lets precisely describe conditions used in the evaluation of TARS. From the analysis of these conditions, we provide a number of guidelines for a robust evaluation of RS in general, and TARS in particular. Moreover, we propose adaptations and new methods for different recommendation tasks, based on the proper exploitation of available time context information. By using fair evaluation settings, we are able to reliably assess the performance of different methods, identifying the circumstances under which some of them outperform the others.

In summary, by means of the proposed methodological characterization and the conducted experiments, we show the importance of using a robust evaluation method to measure the performance of TARS, issue which had not been addressed in depth so far.

Resumen

Los Sistemas de Recomendación (SR) tienen como objetivo ayudar a los usuarios en tareas de acceso y recuperación de información, sugiriendo ítems –productos o servicios– de acuerdo a preferencias –intereses, gustos– pasadas en contextos concretos. En los últimos años, uno de los contextos que se ha estudiado en más detalle ha sido el llamado contexto temporal, que ha dado lugar a una ya amplia área de investigación conocida como Sistemas de Recomendación Conscientes del Tiempo (SRCT).

A pesar del gran número de propuestas y avances realizados sobre SRCT, en la literatura, resultados y conclusiones sobre cómo explotar la información temporal parecen contradictorios. Aunque diversos motivos podrían explicar contradicciones existentes, en esta tesis se plantea que la evaluación de los SRCT juega un rol fundamental. La existencia de múltiples metodologías y métricas de evaluación posibilita encontrar algún protocolo de evaluación a la medida de un enfoque de recomendación particular, no necesariamente generalizable. Los problemas originados de esta situación son un impedimento para comparar imparcialmente resultados y conclusiones de diferentes estudios, dificultando la identificación de la mejor aproximación de recomendación para una tarea dada. Además de lo anterior, la revisión de los trabajos publicados muestra que la mayoría de los SRCT existentes se han desarrollado para disminuir el error en la predicción de las preferencias (*ratings*) de usuarios por ítems. Sin embargo, actualmente el foco de los SR está cambiando hacia la sugerencia de (listas de) ítems relevantes para el usuario. Por otra parte, el uso de los SR en tareas diversas posibilita nuevas aplicaciones donde la información de contexto temporal pueda ser un valor diferenciador.

Esta tesis sintetiza y analiza la forma en que la información de contexto temporal ha sido explotada en la literatura de SR, con el objetivo de a) caracterizar un protocolo de evaluación robusto que permita realizar evaluaciones imparciales de nuevos SRCT y facilitar las comparaciones entre resultados publicados; y b) explotar más adecuadamente la información de contexto temporal en diferentes tareas de recomendación. Para cumplir tales objetivos se han identificado cuestiones metodológicas clave con respecto a la evaluación *offline* de SRCT, y se propone un marco metodológico que permite describir de manera precisa las condiciones usadas en la evaluación de SRCT. Del análisis de estas condiciones, se concluye un conjunto de guías metodológicas para la evaluación robusta de SR en general y SRCT en particular. Por otro lado, se proponen adaptaciones y nuevos métodos para distintas tareas de recomendación, basadas en la adecuada explotación de la información de contexto temporal disponible. Usando escenarios de evaluación imparciales, se ha medido ecuánimemente el rendimiento de diferentes métodos, identificando las circunstancias bajo las cuales unos mejoran a otros.

En definitiva, mediante la caracterización metodológica propuesta y los experimentos realizados, se pone de manifiesto la importancia de utilizar un método de evaluación robusto para SRCT, aspecto que no había sido abordado en profundidad hasta la fecha.

Acknowledgements

I would like to thank all the people that have helped me make this thesis a reality. First, I would especially like to express my gratitude to my supervisors Fernando Díez and Iván Cantador for their guidance, patience and support through all these years. Thank you for your invaluable help in polishing my research skills, and for your care in every single aspect of my work, and moreover of my life. I feel lucky to have had the chance to collaborate with you.

I am also thankful to all the members of the Information Retrieval group at Universidad Autónoma de Madrid, IRG@UAM, especially to Pablo Castells, David Vallet, Saúl Vargas, Ignacio Fernández, and Víctor Villasante, for letting me become a part of such amazing group of people. I would like to make a special mention to Alejandro Bellogín, with whom I had the pleasure of collaborating and discussing not only about research issues, but also about life in general.

Alongside the thesis, at UAM I had the opportunity to collaborate with people from the Departamento de Ingeniería Informática. My especial thanks to Manuel Sánchez-Montañés, Silvia Teresita Acuña, José Antonio Macías, and Ruth Cobos. I also want to extend my gratitude to all the academic and administrative staff of the Escuela Politécnica Superior.

I must thank the financial support provided by the Universidad del Bío-Bío –particularly by the Departamento de Sistemas de Información, and the Facultad de Ciencias Empresariales–, and by the Government of Chile via the Becas Chile scholarship, for pursuing my PhD. I also thank the financial support provided by IRG@UAM.

Despite the distance, these years would have been far more difficult without the support from my friends, particularly from Claudio and Tatiana. I also thank Hector Valdés-González, with whom I started my research training in Chile. Additionally, I would like to thank all the people who I met during these years, and who made this long journey more enjoyable. Thanks to Luis Rojas, Alexandra Dumitrescu, Linda Barros, Ania Benítez, Raúl Cajías, Rafael Martín, and Enrique Chavarriaga.

I thank my parents, Winnie and Pedro, and my brother, Eduardo, for their love and care. I also thank my fathers-in-law, Isabel and Guillermo, for treating me as a son. I would also like to mention my brother Alex, and my grandmothers Lucía and Delia. They will always be in my mind.

Last, but not least, I endlessly thank my wife Maritza for her infinite love and patience, for her constant encouragement to accomplish my goals, and for giving me the best of all gifts: our beloved daughter Constanza. Without them all this effort would be worth nothing.

To Maritza and Constanza

Chapter 1

Introduction

In this chapter we present a general overview of the thesis, describing its main research topics, and the limitations in the subject that motivated the work, giving an outline of the conducted analysis, and reporting and discussing achieved results.

In Section 1.1 we outline the research topics that motivated this thesis. In Section 1.2 we define the scope of this work by stating the general addressed problem and research goals. Next, in Sections 1.3 and 1.4 we detail the main contributions, and list the publications originated from the conducted research. Finally, in Section 1.5 we describe the structure of this document.

1.1 Motivation: Recommendation, context, and time

Recommender Systems (RS) are software applications that aim to help users with information access and retrieval tasks on large collections of items –products or services–, by in general suggesting items according to past personal preferences.

The last decade has been fertile ground for research in the RS field and, among other issues, different recommendation problems and tasks (Adomavicius and Tuzhilin, 2005), algorithmic approaches (Herlocker et al., 1999), and evaluation metrics and methodologies (Shani and Gunawardana, 2011) have been investigated.

This amount of research has led to important advances on deployed RS, and has increased the interest in building more and better RS. On the one hand, users of RS obtain personalized suggestions about items they might be interested in, and which may be difficult for them to find. On the other hand, businesses exploiting RS obtain higher profits due to an increased consumption of suggested items. These facts have led to the creation and expansion of important personalized services supported by RS technologies in the internet, such as Amazon[1], Netflix[2], and Last.FM[3], to name a few.

During the exploitation of a RS through time, large records of user preferences – ratings and consumption logs– are collected, and these records may include information about the **context** in which the user preferences were expressed (Adomavicius and Tuzhilin, 2011). For instance, along with a particular user's preferences, a RS can record the type of device used by the user (e.g. a computer or a mobile phone), the user's location (e.g. at home or at the office), the user's mood (e.g. happy or sad), the user's social companion (e.g. alone, with relatives, or with friends), and the time at which the user expressed her preference (e.g. in the morning, or in the evening). Exploiting this information, context-aware RS (CARS) can suggest items that may fit better the user's interests under certain circumstances or situations (contexts), being thus very valuable for increasing the performance of the provided recommendations (Koren, 2009a; Adomavicius and Tuzhilin, 2011; Panniello et al., 2013).

Among the existing contextual dimensions, **time** can be considered as one of the most useful. It facilitates tracking the evolution of user preferences (Xiang et al., 2010), enabling e.g. to identify periodicity in user preferences (Baltrunas and Amatriain, 2009), and may lead to significant improvements on recommendation accuracy, as found by the winning team of the Netflix Prize competition (Koren, 2009a). Moreover, time context information is in general easy to collect without additional user efforts and strict device requirements.

[1] Amazon.com online shopping, http://www.amazon.com
[2] Netflix on-demand video streaming, http://www.netflix.com
[3] Last.FM internet radio, http://www.last.fm

Due to these benefits, recent years have been prolific in the research and development of **time-aware RS** (TARS), that is, CARS that exploit the time dimension for both user modeling and recommendation strategies. Different TARS proposals can be found in the literature, showing improvements over traditional RS on recommendation performance. However, we note that some studies have shown **divergences on assumptions in which TARS models are built**, casting doubt on the generalization of time-aware recommendation capabilities. As a matter of fact, for instance, some TARS approaches penalize old preferences data, assuming that recent data better reflect the users' current tastes, compared to older ones (Ding and Li, 2005; Ma et al., 2007; Lee et al., 2008). However, some authors, e.g. Koren (2009a) have found a decrease in recommendation performance from this type of penalization.

Although such inconsistencies could be explained by several reasons, e.g. differences in user and item characteristics, and peculiarities of the application domains, we believe that evaluation plays a prominent role. The existence of **diverse evaluation methodologies** makes it easy to find an evaluation protocol suitable for a particular algorithmic approach, but ineligible or non-retributive for others. Indeed, some authors, such as Lathia et al. (2009a, 2009b), have shown important discrepancies on recommendation performance depending on how training and test data for recommendation evaluation is chosen. Problems that arise from this situation represent an increasing impediment to fairly compare results and conclusions reported in different studies (Bellogín et al., 2011), and make the selection of the best recommendation solution for a given task more difficult (Gunawardana and Shani, 2009). The study of methodological issues that a robust evaluation of TARS should take into account in order to increase the reliability of measured improvements attributed to TARS, and facilitate the comparison of approaches, is thus a main concern in our research.

The **discovery of unexpected results in TARS studies** also shows that more research is still required to fully understand the relation between time context information and recommendation results. Baltrunas and Amatriain (2009) provide an illustrative example of this in experiments testing several time-dependent partitions of user preference data for increasing recommendation performance of a CARS. They found that the scarce {*even hours*, *odd hours*} partitioning provides higher recommendation improvements than other partitions such as {*morning*, *evening*} and {*workday*, *weekend*}. In words of Baltrunas and Amatriain, the *hours* partition corresponds to a "meaningless" partition, and thus calls for further research. What is more, the lack of studies comparing TARS performances keeps the knowledge of the circumstances under which some TARS approaches –and the time context signals exploited by them– are able to outperform the others fairly unexplored. This also prevents to adjust TARS for better exploiting the available time information in particular situations.

In addition to the above issues, a review of published work in the subject exposes that most TARS have been developed for the **rating prediction task**. Nonetheless, nowadays recommendation focus is shifting from diminishing error in rating prediction towards finding (lists of) relevant/appealing items for the target user, i.e., **the top-N recommendations task**. Moreover, the widespread use of recommender systems on diverse user tasks makes it possible to find new applications where time context information can serve as a distinctive input. All in all, understanding how time information can be exploited for improving recommendation tasks, including and beyond rating prediction, is another main goal of our research.

In summary, drawing from the state of the art on TARS approaches for generation and evaluation of contextualized recommendations as a starting point, this thesis studies, synthesizes and analyzes how time context information has been exploited in the recommender systems literature, in order to a) characterize a robust evaluation methodology that lets conduct fair evaluation of new TARS, and facilitate comparisons between TARS performance results; and b) improve the exploitation of time context information in different recommendation tasks, leading to new and better applications of time-aware recommendation technologies.

1.2 Problem statement, research goals and hypotheses

From a general point of view, the recommendation problem consists of suggesting items that should be the most appealing ones to a user according to her preferences. Traditionally, most approaches to recommender systems do not take any contextual information into account, that is, they only consider two types of entities for generating recommendations: users and items (Adomavicius and Tuzhilin, 2011). In many applications, however, contextual information may provide valuable input for improving recommendations, under the assumption that similar circumstances (i.e., contexts) are related with similar user preferences.

In this thesis, we focus on problems that incorporate **time** as a source of contextual information for both user modeling and recommendation strategies. Our final goal is to address the recommendation problem from a time-aware perspective based on two main lines of action. On the one hand, establishing a robust evaluation protocol that takes time dependencies of data into account, in order to enable an objective and rigorous evaluation of recommendation results from TARS; and on the other hand, approaching different recommendation tasks from a time-aware perspective, in order to take advantage of time context information for improving current methods' performance on such tasks. By using a robust evaluation protocol, we seek to count with a reliable assessment of the improvements obtained. For tackling these problems we aim to address the following research goals:

RG1: Characterization of the conditions involved in the evaluation of TARS. We shall develop an in-depth review and analysis of the protocols utilized for the evaluation of the current generation of TARS, aiming to distinguish and formalize the key conditions that the performed evaluations are driven by. We address this research goal in Chapter 4.

We note that in any evaluation protocol there are two fundamental components that define the setting in which a system's performance is assessed: the *evaluation metrics*, which define what to assess, and the *evaluation methodologies*, which define how to assess. In the recommender systems field, certain metrics have been accepted and are commonly used (Herlocker et al., 2004; Gunawardana and Shani, 2009). However, there is no consensus on the methodologies used (Bellogín et al., 2011). Moreover, it is a general practice to report the metrics applied to assess the performance of developed recommender systems, but it is less common to find clear descriptions of the followed evaluation methodologies. Due to this, we shall emphasize our study on methodological divergences in TARS evaluation.

RG2: Analysis of the effect of different evaluation conditions on the assessment of TARS performance. We shall study and determine whether the application of distinct evaluation conditions leads to differences in the assessment of recommendation results from TARS. From this, we shall establish a set of conditions that let conduct fair and reproducible evaluations of TARS in order to perform rigorous measurements of TARS performance. We address this research goal in Chapter 5.

As already mentioned in Section 1.1, we hypothesize that evaluation plays an important role in explaining discrepancies found in the TARS literature. However, to the best of our knowledge, the impact of different evaluation settings on assessed results has not been studied. From the analysis of such effect and the characteristics of the conditions, we shall aim to establish a set of conditions that provide reliable settings for TARS evaluation. This set of conditions shall be used throughout the experimental work in this thesis, for properly measuring the improvements achieved from the use of time context information associated to user preference data.

RG3: Adaptation of existing recommendation approaches to make better use of available time context information. We shall investigate the relation between time context information and user preferences, aiming to improve recommendation results of one or more recommendation approaches based on knowledge about time context. This knowledge will let adjust or adapt existing recommendation approaches to improve the manner in which time context knowledge is exploited. The obtained improvements will be assessed with a set of conditions that ensure a fair evaluation and comparison with other approaches. We address this research goal in Chapter 6.

Exploiting time context information has been proved to be an effective approach to improve recommendation performance, as shown e.g. by the winning team of the well-known Netflix Prize competition (Koren, 2009b). It is possible to find several approaches in the literature able to exploit time context information. Nonetheless, the shift from diminishing error prediction towards finding relevant items, and the lack of a standardized evaluation protocol, makes it difficult to establish which approaches make better use of available time context information. By counting with a fair and common evaluation setting, it would be possible to determine the circumstances in which some algorithms outperform the others. From these, we would be able to adjust or adapt the operation of some recommendation approaches in order to improve their performance.

RG4: Exploitation of time context information on a non well-established recommendation task. We aim to take advantage of the experience and insights regarding the utilization and evaluation of time-aware recommendation models, by means of developing novel applications for these techniques. With this goal in mind, we shall consider recommendation-related tasks –beyond rating prediction and top-N recommendations– where available time context information can be an important input for improvements. We shall develop new approaches based on the exploitation of time context to address a selected task, and shall use an evaluation setting that ensures a fair and robust evaluation. We address this research goal in Chapter 7.

Addressing the above research goals is based on the following hypotheses:

Hypothesis 1: Variations in the evaluation protocol lead to differences on recommendation results assessment. This hypothesis is associated with RG1 and RG2.

Hypothesis 2: The appropriate exploitation of time context information leads to improvements on assessed recommendation results. This hypothesis is associated with RG3 and RG4.

Hypothesis 3: From a temporal viewpoint, using a robust evaluation protocol of recommendation models and techniques exploiting time context information leads to a decrease on assessed performance, with respect to a less robust evaluation protocol. This hypothesis is associated with RG2, RG3 and RG4.

1.3 Contributions

The research conducted in this thesis contributes to improve the reliability of the assessment of results from time-aware recommender systems, letting a better exploitation of time context information in recommender systems. Hence, the main contributions of our research are:

- **The characterization of conditions which drive the evaluation process of TARS**. We perform a comprehensive review of TARS-related literature, identifying key methodological issues to be faced in the experimental design of an offline evaluation of TARS. From this, we formalize a number of conditions used in evaluation of TARS that address the methodological issues analyzed. The defined conditions are mostly related to the training-test data splitting process, which can be differently performed due to the existence of time context information associated to data. We also cover conditions required for evaluating specific recommendation tasks, as is detailed in Chapter 4.

- **The development of a methodological framework for describing conditions used in the evaluation of TARS**. We propose a methodological description framework that incorporates the evaluation conditions characterized in the thesis, aimed to facilitate the description and adoption of evaluation protocols, and make the evaluation process fair and reproducible. This framework, introduced in Chapter 4, includes the definition of a splitting procedure algorithm for building training-test splits of data using the formalized evaluation conditions. The usage of this framework may facilitate the comparison of results from different TARS proposals, as it lets communicate easily and formally the different evaluation conditions used to assess TARS performance.

- **The analysis of methodological issues that a robust offline evaluation of TARS in particular, and RS in general, should take into account**. We synthesize and discuss the effect of alternative conditions addressing key methodological issues involved in the evaluation of TARS throughout Chapter 4. Additionally, in Chapter 5 we classify the surveyed TARS literature in terms of the defined evaluation conditions, thus relating and analyzing the use of such conditions in a large body of research on context- and time-aware recommender systems. Furthermore, we conduct a rigorous experimental comparison of results obtained from different TARS evaluation protocols, which is also reported in Chapter 5. We evaluate a set of well-known TARS in the movie and music recommendation domains, using different types of user preference data, namely explicit and implicit ratings. This comparison is aimed to assess the influence of evaluation conditions on measured performance results, by means of accuracy and ranking metrics.

- **The proposal of a set of methodological guidelines aimed to facilitate the proper selection of conditions for offline TARS evaluation**. From the results obtained in our experiments, and the analysis of the evaluation protocols used in the TARS literature, in Chapter 5 we conclude a set of general guidelines aimed to facilitate the selection of conditions for a proper TARS evaluation. These guidelines comprise the choice of conditions for performing the training-test splitting of data required for computing evaluation metrics, and for the application of an adequate cross-validation method. We also include guidelines for selecting specific conditions required for evaluating top-N recommendations.

- **The proposal of new heuristics and adaptations for some general context-aware recommender systems to make better use of time context information.** We implement state-of-the-art CARS, and propose novel heuristics in order to improve their performance when exploiting time context information. Specifically, in Chapter 6 we propose a new impurity criterion to be used in Item Splitting (Baltrunas and Ricci, 2009a, 2009b), and develop a post-filtering strategy that let contextualize recommendations generated by the high-performing Matrix Factorization recommendation algorithm (Takács et al., 2008; Koren et al., 2009). Additionally, we adjust other impurity criteria used by Item Splitting, and adapt a contextual modeling approach by Panniello and Gorgoglione (2012). The proposed heuristics and adaptations are based on the assessment of results obtained on contextualized data from real users utilizing a common and precisely defined evaluation protocol.

- **The proposal of a novel methodology for evaluating top-N recommendations results**. We propose and use a new methodology for evaluating the top-N recommendations task in the study presented in Chapter 6, which lets build ranked list of items targeted for the same time context, including unrated items in the list, and lets provide a more realistic evaluation setting than those from other methodologies in the literature.

- **The development of novel time-aware approaches to address the identification of active users in shared user accounts task**. In Chapter 7 we propose and develop novel methods that exploit time context information to address this recently defined recommendation task, consisting of automatically identifying the active user (in a particular moment) of a shared (household) user account. We formulate this task as a classification problem, and test classifiers that exploit time features from past item consumption records of users in households. The analysis of the time features extracted show the existence of dissimilar temporal rating habits of users of household accounts, which let differentiate which user is active in a given moment.

- **The adaptation of TARS evaluation methodologies for assessing performance of methods in the identification of active users in shared user accounts task**. In Chapter 7 we describe an extension to the proposed methodological framework for TARS evaluation by defining an additional condition specific for this non well-defined task. We show that the organization of the framework lets an easy incorporation of the new condition. Based on the above, we use the conceptual structure of the framework for adapting the methodologies recommended by our methodological guidelines, in order to assess the proposed approaches for the task.

1.4 Publications

The contributions of this thesis have originated a number of publications, which are listed in the following. We group them according to the chapter and research topic they are related to.

Chapter 4

Evaluation methodologies and TARS

An initial proposal towards establishing a framework for the evaluation of time-aware recommender systems was presented in:

- Campos, P. G., Díez, F. (2010). **La Temporalidad en los Sistemas de Recomendación: Una Revisión Actualizada de Propuestas Teóricas**. *I Congreso Español de Recuperación de Información* (CERI 2010), pp. 65-76. Madrid, España.

In that work we described a review of the state of the art on TARS, from which we observed the need of improving the evaluation protocols used for TARS performance assessment. This observation motivated the main purpose of this thesis –the need to provide a more reliable evaluation of TARS performance. Aiming to accomplish that purpose, we developed a methodological framework for selecting and describing the conditions used to evaluate and compare TARS. The evaluation conditions that comprise the methodological framework introduced in the chapter are studied in:

- Campos, P.G., Díez, F., Cantador, I. (2013). **Time-Aware Recommender Systems: A Comprehensive Survey and Analysis of Existing Evaluation Protocols**. *User Modeling and User-Adapted Interaction*, Special Issue on Context-Aware Recommender Systems. In press (Online publication: 2013).

In that work we formalized a number of conditions used for the evaluation of TARS, from the analysis of evaluation protocols found in a comprehensive review of the TARS literature. These conditions let precisely describe evaluation methodologies employed in the assessment of TARS performance, facilitating the reproducibility of evaluation settings and the comparison of diverse TARS proposals.

Chapter 5

Evaluation settings and recommendation performance

Identifying the importance of the setting used for the evaluation of TARS, we studied the performance of a well-known TARS approach under different evaluation protocols. This study was presented in:

- Campos, P.G., Díez, F., Sánchez-Montañés, M. (2011). **Towards a More Realistic Evaluation: Testing the Ability to Predict Future Tastes of Matrix Factorization-based Recommenders**. *5th ACM Conference on Recommender Systems* (RecSys 2011), pp. 309-312, Chicago, IL, USA.

In that work we compared the performance of the matrix factorization (MF) algorithm –which is not time aware– against the MF with temporal dynamics approach (Koren, 2009a), under two evaluation protocols: the one used in the Netflix Prize competition, and a setting that uses a strict temporal separation of training and test data. In this study we found important differences in the relative ranking of the evaluated approaches when changing the evaluation setting, clearly showing the need for a more robust evaluation of TARS approaches. The evaluation protocols tested in this work served for defining the evaluation conditions used in the empirical comparison of TARS presented in the chapter.

Chapter 6

Evaluation of time-aware recommendation performance

Once we observed that the variability on assessed performance of distinct TARS in the literature was mainly due to the usage of different evaluation settings, we decided to implement and compare TARS proposals under a common and clear evaluation protocol. In this way, we could identify which approaches ones outperform the others, and under which circumstances. A first comparative study was presented in:

- Campos, P.G., Díez, F., Cantador, I. (2012). **A Performance Comparison of Time-Aware Recommendation Models**. *Proceedings of the 2nd Spanish Conference in Information Retrieval* (CERI 2012), Valencia, Spain.

In that work we compared TARS exploiting continuous time context information, using an evaluation methodology that takes the time order of ratings into account. However, we were limited to the use of a dataset with rating timestamps, not counting with information about the time context in which items were effectively used/consumed. In a subsequent work, we performed a user study in order to obtain reliable time context information, as well as other contextual signals, for comparing different recommendation approaches exploiting context information. This latter study is described in:

- Campos, P.G., Fernández-Tobías, I., Cantador, I., Díez, F. (2013). **Context-Aware Movie Recommendations: An Empirical Comparison of Pre-Filtering, Post-Filtering and Contextual Modeling Approaches**, *Proceedings of the 14th International Conference on Electronic Commerce and Web Technologies* (EC-Web 2013), pp. 137-149, Prague, Czech Republic.

In that work we focused on comparing general CARS approaches able to exploit time context information in the form of categorical variables. Moreover, we compared time context information against social context information, in order to study which one is more informative for the evaluated approaches, in terms of improvements on rating prediction task. The proposed methodological framework served as basis for defining the evaluation setting in that study.

Context-aware recommender systems and time context information

We studied the ability of context-aware RS for improving recommendation performance from the exploitation of time context signals modeled as categorical variables, derived from continuous time context information (in the form of timestamps) associated to ratings. We evaluated a state-of-the-art pre-filtering approach in:

- Campos, P.G., Cantador, I., Díez, F. (2013). **Exploiting Time Contexts in Collaborative Filtering: An Item Splitting Approach**, *3rd workshop on Context-Awareness in Retrieval and Recommendation* (CaRR 2013) held in conjunction with the 6th ACM International Conference on Web Search and Data Mining (WSDM 2013), pp. 3-6, Rome, Italy.

That work is focused on the analysis of the Item Splitting pre-filtering approach, looking for the best combinations of time context signals such as *period of the day* and *period of the week*, and different parameters utilized by the approach, in order to obtain improvements in rating prediction as well as in the top-N recommendations task.

Chapter 7

Study of user temporal rating habits

The analysis of time context information associated to user ratings let us to address a less studied task related to recommender systems: The identification of users in shared user accounts. This task was proposed as a challenge within the 2nd Workshop on Context-aware Movie Recommendation (CAMRa 2011). The initial analysis of such data and our first approaches to the task are presented in:

- Campos, P.G., Díez, F., Bellogín, A. (2011). **Temporal Rating Habits: A Valuable Tool for Rating Discrimination**. *Proceedings of the 2nd Workshop on Context-aware Movie Recommendation* (CAMRa 2011), held in conjunction with the 5th ACM Conference on Recommender Systems (RecSys 2011), pp. 29-35, Chicago, IL, USA.

In that work we analyzed different time context variables derived from timestamps, as well as other information associated to user ratings, finding important differences in the rating behavior of different users utilizing the same shared (household) account. Moreover, we proposed a probabilistic modeling approach to the identification of the active user at a given time.

Identification of active users in shared accounts based on time context information

Motivated by the good performance of the proposed approach, we implemented and evaluated diverse methods for the above task, based exclusively on the exploitation of time context information. These methods and their performance on the task are described in:

- Campos, P.G., Bellogín, A., Díez, F., Cantador, I. (2012). **Time feature selection for identifying active household members**. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management* (CIKM'12), pp. 2311-2314 Maui, HI, USA.

The methods presented in that work were able to provide a high accuracy on the task (over 95%) using the evaluation protocol established by the CAMRa 2011 challenge organizers, which is based in the random selection of test data.

Robust evaluation of methods for the identification of active users in shared accounts

In order to test the reliability of the proposed methods, we decided to adapt and use the methodological framework proposed in this thesis to assess the methods' performance on different evaluation protocols. This evaluation is reported in:

- Campos, P.G., Bellogín, A., Cantador, I., Díez, F. (2013). **Time-Aware Evaluation of Methods for Identifying Active Household Members in Recommender Systems**, *Proceedings of the 15th Spanish Conference on Artificial Intelligence* (CAEPIA 2013), Madrid, Spain. To appear.

The study's contributions were two-fold. On the one hand, we showed that the discrimination power of the proposed methods varies considerably when assessed by different methodologies. On the other hand, we showed the flexibility and extensibility of the methodological framework proposed in this thesis, employing it for the evaluation of time-aware predictive models targeted to a different task than the ones the framework was originally designed for.

Related contributions

The observation of the difficulty in comparing distinct TARS' performance arises from a comparative study of TARS performance on diverse evaluation dimensions, conducted in the author's Master Thesis entitled "Temporal Models in Recommender Systems: An Exploratory Study on Different Evaluation Dimensions" (Campos, 2011). The review and comparison of published results made in that work showed us the need of a more reliable evaluation protocol for time-aware recommender systems. That work, thus, served as a basis for the contributions of this thesis.

Alongside the thesis additional contributions on related issues regarding recommender systems were published. Specifically, we investigated 1) heuristics for time-aware recommendation, 2) recommendation approaches able to exploit other types of context information, and 3) alternative approaches for identifying active users in shared accounts. The first proposal served as basis for exploring new TARS approaches described in Section 6.2. The second corresponds to extensions of approaches presented in Chapter 6, able to exploit all type of context information. The third corresponds to a novel approach for addressing the task described in Chapter 7.

Heuristics for time-aware recommendation

We evaluated simple heuristics to exploit time context information in:

- Campos, P.G., Bellogín, A, Díez, F., Chavarriaga, J.E. **Simple Time-Biased KNN-based recommendations**. *Workshop Challenge on Context-aware Movie Recommendation* (CAMRa 2010), held in conjunction with the 4th ACM Conference on Recommender Systems, pp. 20-23, Barcelona, Spain.

The heuristics studied in that work let adapt kNN-based recommendations by means of the exclusive exploitation of ratings in the near time of the target recommendation time. These heuristics thus help improve recommendation results provided by kNN algorithm while reduce the amount of information required to provide recommendations.

Model-based context-aware recommendation

We also investigated different model-based context-aware recommendation approaches able to exploit different types of context information. A proposal exploiting social context was presented in:

- Díez, F., Chavarriaga, J.E., Campos, P.G., Bellogín, A. (2010) **Movie Recommendations based in explicit and implicit features extracted from the Filmtipset dataset**. *Proceedings of the Workshop Challenge on Context-aware Movie Recommendation* (CAMRa 2010), held in conjunction with the 4th ACM

Conference on Recommender Systems 2010 (RecSys 2010), pp. 45-52, Barcelona, Spain.

In that work we used different collaborative filtering algorithms based on Random Walks to exploit social context information in the form of friend relationships on a movie ratings dataset. Using a different approach, we tested content-based CARS in:

- Fernández-Tobías, I., Campos, P.G., Cantador, I., Díez, F. (2013). **A Contextual Modeling Approach for Model-based Recommender Systems**, *Proceedings of the 15th Spanish Conference on Artificial Intelligence* (CAEPIA 2013), Madrid, Spain. To appear.

In that work we evaluated different machine learning algorithms exploiting user patterns including genres preferences and social context information in the form of social companion, additionally to location and time contexts, in which users prefer to watch movies and listen to music. The previous works showed the ability of the proposed approaches to improve recommendation performance from the exploitation of contextual information.

Game theoretic modeling for identifying active users in shared accounts

We tested diverse modeling approaches in order to address the novel task of identifying active users in shared accounts. One of such approaches is described in:

- Díez, F., Campos, P.G. (2012). **Identificación de usuarios en Sistemas de Recomendación mediante un modelo basado en Teoría de Juegos**. *II Congreso Español de Recuperación de Información* (CERI 2012), Valencia, España.

One of the interesting contributions of that work, besides the novelty of employing a game theoretic modeling scheme, is a proposed approach to dynamically select the best information sources independently for each shared user account.

1.5 Thesis structure

The thesis has been divided into three parts. The first part gives a literature survey in recommender systems in general, and time-aware recommender systems in particular. The second part characterizes a robust evaluation protocol for time-aware recommender systems, based on the identification and analysis of conditions that drive evaluation methodologies; and evaluates the effect of using different conditions on assessed recommendation results. The identified conditions give form to a methodological framework for the evaluation of TARS. The third part and last part presents different applications that exploit time context information, and takes advantage of the proposed framework for providing a more reliable measurement of the improvements due to the use of time-aware models. In more detail, the contents of this thesis are distributed as follows:

Part I. State-of-the-art: Recommender systems and time context

- **Chapter 2** provides an overview of the state of the art in recommender systems, considering recommendation tasks, types of user feedback, and techniques and evaluation of these systems.

- **Chapter 3** presents a comprehensive review of the state of the art in time-aware recommender systems, considering a classification of the main approaches in the literature regarding the modeling and exploitation of time context information. Additionally, the methodologies and metrics used in the evaluation of these systems are discussed.

Part II. Characterizing a robust time-aware recommendation evaluation protocol

- **Chapter 4** analyzes key methodological issues involved in the design of protocols for evaluating time-aware recommender systems, and formalizes a number of conditions addressing these issues. From the stated conditions, a methodological framework aimed to characterize the TARS evaluation process is defined.

- **Chapter 5** presents a classification of state-of-the-art TARS literature based on the key conditions used in their evaluation, and reports an empirical analysis on such conditions. From the analysis of obtained results, a number of general guidelines to select proper conditions for evaluating particular TARS are provided.

Part III. Exploiting time context information in recommendation tasks

- **Chapter 6** exposes a comparison of different TARS approaches on two important recommendation tasks, namely rating prediction and top-N recommendations. New heuristics, as well as adaptations and adjustments to some approaches that improve the exploitation of time context signals are proposed. Taking advantage of the proposed methodological framework, a fair and common evaluation setting is provided in order to obtain a reliable assessment of performance improvements. A user study performed for collecting explicit time context information from users is also detailed, which serves as input for the evaluated TARS.

- **Chapter 7** describes novel time-aware methods developed for addressing a recommender systems-related task: the identification of active users in shared (household) accounts. The proposed methods, based on the exploitation of time context information associated to rating events, are assessed under different evaluation settings provided by the adaptation of the proposed methodological framework for the evaluation of this task.

- **Chapter 8** concludes the thesis with a summary of the main contributions, and a discussion about future research lines.

Additionally to these chapters, the thesis includes the following appendixes:

- **Appendix A** contains the translation into Spanish of the *Introduction* chapter.
- **Appendix B** contains the translation into Spanish of the *Conclusions* chapter.

Part I

State-of-the-art: Recommender systems and time context

Chapter 2

Recommender systems

Nowadays, Internet and particularly Web-based services and applications bring access to almost non limited resources of information. For example, an online store may offer customers with access to millions of products. In this context, recommender Systems (RS) aim to help users with information access and retrieval tasks when large collections of items are involved. In general, these systems work by means of suggesting those items that should be the most appealing ones to the users based on their personal preferences and needs.

Different recommendation tasks defining the outcomes of a RS can be distinguished, including *rating prediction*, in which a numerical value estimating user preference for a given item is computed, and *top-N recommendations*, in which a list of the best (top-N) items is delivered. For performing these tasks, RS exploit knowledge about user preferences extracted from feedback of different forms, which are commonly classified as either *explicit feedback* or *implicit feedback*. Moreover, several techniques for computing recommendations have been proposed in the literature, being *content-based* and *collaborative filtering* techniques the most commonly recognized, and *hybrid* techniques those that combine different techniques to overcome individual limitations of each technique. Finally, in order to assess RS performance, distinct *evaluation methodologies* and *metrics* –focusing on different recommendation properties– have been proposed.

In this chapter we provide an overview of terminology, models and methods related to the building and evaluation of recommender systems. In Section 2.1 we formalize the problem of recommendation, and describe the main tasks addressed by RS. In Section 2.2 we detail the types of user feedback in RS, and in Section 2.3 we introduce main recommendation techniques. Next, in Section 2.4 we explain the methodologies and metrics used for RS evaluation. Finally, in Section 2.5 we conclude with a summary of the chapter.

2.1 Recommendation problem and related tasks

Current online service providers utilize several types of software tools to provide users with suggestions of appealing items. These tools are commonly called recommender systems. In general, the goal of these systems is to help individuals who lack of sufficient personal experience or competence to explore and evaluate a potentially overwhelming number of items, for example, those available in Web-based applications (Ricci et al., 2011). Collaborative filtering is usually considered as the first approach of recommender systems. The term was coined in the mid 90's for an email filtering application based on using different users' opinions collaboratively (Goldberg et al., 1992), following the idea of "word-of-mouth" phenomenon. Since then, diverse forms of recommendations and techniques to compute such recommendations have been proposed in the literature, and have been used in commercial and leisure applications. Moreover, several events and media have shown the growth and complexity of the field. We can mention, among others, survey papers (e.g. Adomavicius and Tuzhilin, 2005; Burke, 2007; Gunawardana and Shani, 2009; Su and Khoshgoftaar, 2009; Ekstrand et al., 2011; Pu et al., 2012), books (e.g. Jannach et al., 2010; Ricci et al., 2011), an annual conferences (Cunningham et al., 2012), workshops (e.g. Cantador et al., 2011; Castells et al., 2011; Adomavicius et al., 2012; Mobasher et al., 2012; Böhmer et al., 2013), and journal special issues (e.g. Ricci and Werthner, 2006; Jannach et al., 2008; Riedl and Smyth, 2011; Felfernig et al., 2012).

Due to the diversity of approaches for generating recommendations, it is difficult to find a general definition that holds the complexity of all existing recommender systems. Conversely, here we present simple and widely used formulations that represent the core concepts involved in common recommendation tasks.

According to Adomavicius and Tuzhilin (2005), the recommendation problem relies on the notion of *ratings* as a mechanism to capture user preferences for different items. Let U denote the set of users (known by the system), let I denote the set of items (that form the system's catalog), and let R denote a totally ordered set (e.g. non-negative integer or real numbers in a particular range) of allowed rating values. A recommender system models a function $F: U \times I \rightarrow R$ that computes a predicted rating $\hat{r}_{u,i}$ for an unknown rating $r_{u,i}$ that user $u \in U$ would assign to item $i \in I$:

$$\forall u \in U, i \in I, \hat{r}_{u,i} = F(u,i) \qquad (2.1)$$

where the rating value is interpreted as a measure of the usefulness of item i for user u.

Alternatively, the recommendation problem can be stated as the task of finding relevant items for the target user –the user for whom recommendations must be provided– (Sarwar et al., 2001). This task is consistent with the use of RS in many applications where a system does not predict ratings, but delivers lists of items that may be relevant for the user

(Shani and Gunawardana, 2011). In this case, ratings can be interpreted as a measure of relevance (score), and thus, those items scored over a certain threshold value can be considered as relevant.

For either of the above two formulations, the recommendation problem can be reduced to solve a rating prediction problem, which consists of predicting unknown ratings for pairs (u, i) by providing an estimation of the function F (Adomavicius and Tuzhilin, 2005). In this context, when a RS is required to provide an item recommendation, it could return rating predictions for a particular set of items unknown to the target user –the *rating prediction* task– or a list of top ranked items the user may prefer –the *top-N recommendations* task, also known as *recommendation ranking* task (Shani and Gunawardana, 2011). In the latter case, rating predictions[4] are generally used to rank the items, and select (for recommendation) the top ranked ones. If the order of presentation of the top-*N* items is not important, then this task is also referred to as *recommending some good items* (Herlocker et al., 2004; Gunawardana and Shani, 2009).

Figure 2.1 shows a schematic view of rating prediction (upper side) and top-N recommendations (lower side) tasks, in the context of an example movie recommender system. In the former case, the target user asks the recommender system for a prediction of her preference for a target movie in the system's catalog. The system performs an algorithm to compute the value of $F(\text{target user}, \text{target item})$, which is informed to the user. In the case of top-N recommendations, the user simply asks the system for a recommendation of movies (i.e., no target item is required), and the system computes the value of F for the target user and items in the system's catalog. The N movies with highest values of F are then delivered as recommendations for the target user.

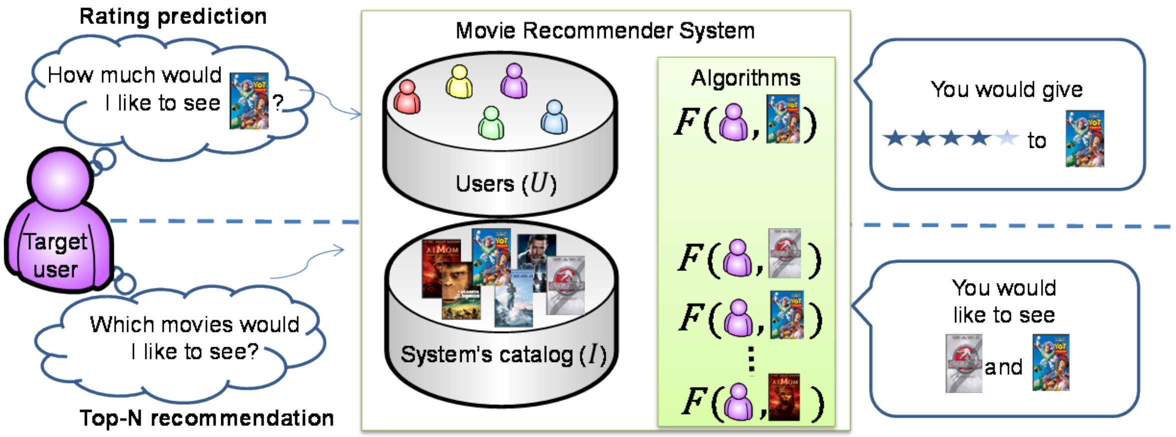


Figure 2.1. Schematic view of rating prediction and top-N recommendations tasks in a movie recommender system.

[4] In this case it may be more precise to talk about score prediction, but for the sake of simplicity we will refer to this as rating prediction.

Although top-N recommendations is perhaps the most common task of commercial RS, most RS research has been focused on the rating prediction task (Gunawardana and Shani, 2009; Cremonesi et al., 2010). Accordingly, in this thesis we mainly focus on these tasks. Nonetheless, we note that there are other recommendation tasks identified in the literature of RS. For instance, in the context of e-commerce RS, Gunawardana and Shani (2009) describe the *optimizing utility* task, in which the RS must generate recommendations that maximize the profits of a Web portal. Furthermore, Herlocker et al. (2004) provide a detailed taxonomy of user tasks for recommendation systems, which includes, among others, the *recommend sequence* task (i.e., finding a sequence of pleasant songs), the *find credible recommender* task (i.e., looking for non-serendipitous items, but items that match user tastes), and the *influence others* task (i.e., assigning high ratings to items in a given category in order to influence others to purchase items in that category).

More recently, other tasks have been explored in the literature that are not part of core functionalities of RS, but help to perform the recommendation tasks. For instance, in the TV show recommendation domain, it is common that several users in a household use only one user account for accessing a TV show RS. In this case, correctly identifying which users are requesting recommendations in a given moment –the *household member identification* task– can be important for providing personalized recommendations (Kabutoya et al., 2010; Campos et al., 2012).

2.2 Types of user feedback in recommender systems

Most recommender systems require some knowledge about user preferences and behavior. These data, however, are usually stored in transactional databases, which may not be suited for efficient recommendations. In order to properly model and exploit user knowledge, *user profiles* are usually built. A user profile stores the information that characterizes the user in a format that lets its efficient usage by a RS.

User profiles used by RS are typically built from *user feedback*, commonly classified according to how it is gathered, as *explicit feedback* and *implicit feedback* (Kelly and Teevan, 2003; Herlocker et al., 2004). Explicit feedback corresponds to information stated by the user about her preferences on items, e.g. star ratings and up/down thumbs. On the contrary, implicit feedback is automatically collected when the user interacts with the system. Examples of implicit feedback are product browsing and purchasing history in e-commerce sites, and time spent reading articles in online news sites. Figure 2.2 shows some examples of user feedback used in popular online services that use RS.

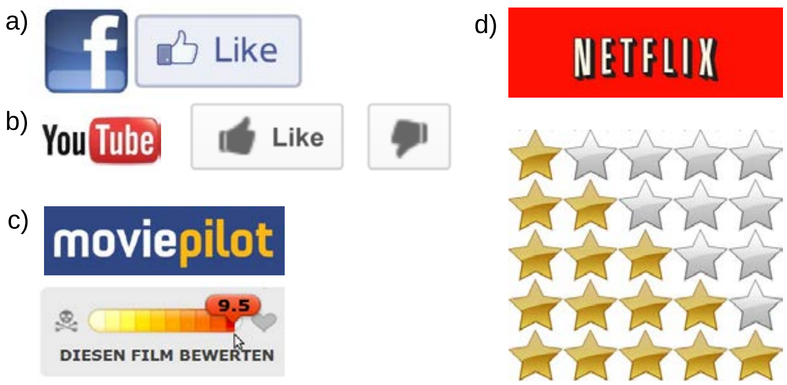


Figure 2.2. Examples of user explicit feedback in popular online services. a) Facebook's like button; b) YouTube's thumbs up and down; c) MoviePilot's rating scale; d) Netflix's stars ratings.

Traditionally, explicit feedback has been considered of higher quality than implicit feedback due to its "explicit" nature, and thus, most work on recommender systems has focused on processing such type of user feedback (Hu et al., 2008; Jawaheer et al., 2010). Explicit feedback, commonly referred to as *ratings*, is associated to a scale of values indicating the users' preferences for items. The simplest case corresponds to unary ratings, indicating a user likes an item (e.g. Facebook's[5] *like* button; see Figure 2.2a). In the binary case, there is also an indication for dislike of items (e.g. YouTube's[6] up/down thumbs; see Figure 2.2b) –note that in the unary case, an absence of a like indication is not equivalent to a dislike indication. More common are the 5-points scales (e.g. Netflix's[7] 1-5 star ratings; see Figure 2.2d) and above (e.g. MoviePilot's[8] 0 to 10 scale with step size 0.5; see Figure 2.2c), which let the user express fine-grained levels of preference. Note that whatever the scale used, a user can assign only one value to each given item. That is, an overall rating that resumes all aspects of interests. In order to enable users evaluate different dimensions of items, multi-criteria rating systems are being explored (Adomavicius and Kwon, 2007; Manouselis and Costopoulou, 2007; Adomavicius et al., 2011). These systems consider different attributes of items, and let users rate each of them independently. For instance, a multi-criteria movie RS could contemplate three criteria about movies, e.g. story, direction and acting. Although multi-criteria RS offer more flexibility, they require the users to provide more information (several ratings per item), increasing the users' effort.

Implicit feedback, on the contrary, lets RS infer user preferences from user behavior information gathered by the system (Hu et al., 2008; Knijnenburg et al., 2012). For instance, the time spent viewing a TV show or the play count of a music track, can be used as an approach of user preferences for an item. Collecting implicit feedback only requires

[5] Facebook online social networking, http://www.facebook.com
[6] YouTube video sharing, http://youtube.com
[7] Netflix on-demand video streaming, http://www.netflix.com
[8] MoviePilot movie recommendations, http://www.moviepilot.de

an initial approval to gather usage data from the user, providing a less intrusive user experience (Knijnenburg et al., 2012). This type of feedback not always reflects actual user preferences, since usage or consumption of an item does not necessarily indicates the user's preference for that item. For example, a user's TV view history may be assumed to reflect a long period of time spent by a user watching a TV show, but such assumption may be not true if the user left the TV on to do a different activity. Moreover, in implicit feedback approaches it may be hard to determine which items are disliked by the users. Not consuming an item (e.g. not seeing a TV show) cannot be inferred as negative feedback.

In general, it is more difficult to obtain explicit than implicit feedback. Some users are reluctant to provide ratings due to e.g. privacy concerns, required cognitive effort, etc. (Jawaheer et al., 2010). Moreover, some researchers have questioned the reliability of this type of feedback. For instance, Amatriain et al. (2009a, 2009b) have shown that users are inconsistent in rating the same movies through time. In fact, the concept of "magic barrier" coined by Herlocker et al. (2004), and used by other researchers (Said et al., 2012a, 2012b), refers to the limit on improvements achievable by RS due to these inconsistencies or *noise* in user ratings.

On the other hand, implicit feedback does not directly represent user preferences, and thus is considered generally as less reliable than explicit feedback. One possible way to address this problem is to derive paired magnitudes from this type of feedback for each user-item pair: an estimation of preference, together with its confidence level (Hu et al., 2008). Additionally, the lack of negative feedback leads to a bias towards positive preferences that may hamper a proper user model (Hu et al., 2008). In order to avoid such bias, some researchers have proposed methods to transform implicit feedback to explicit feedback by a proper binning of frequency values into ratings (Celma, 2008; Parra and Amatriain, 2011). These transformations relate less consumed items to negative explicit ratings, with which afterwards use recommendation techniques that require explicit feedback input.

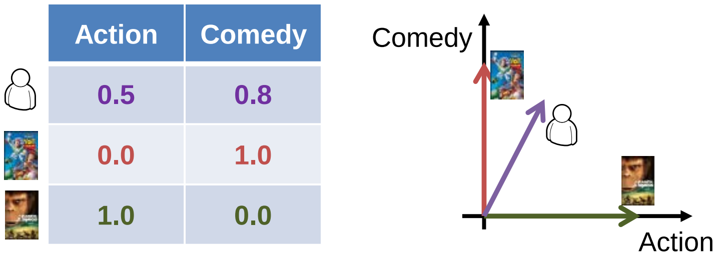
2.3 Recommendation techniques

In this section we briefly describe the main techniques used by recommender systems. Among the most general and widely used, we can distinguish between *content-based* techniques (CB), which suggest similar items to those preferred by the target user in the past, and *collaborative filtering* techniques (CF), which suggest items preferred by users with similar tastes to the target user (Adomavicius and Tuzhilin, 2005). Both CB and CF techniques exploit the target user's feedback to identify her preferences. Burke (2007) additionally identifies *demographic* techniques, which exploit the user's demographics for generating item recommendations, and *knowledge-based* techniques, which exploit specific domain knowledge about the items to recommend. Furthermore, it is possible to distinguish *hybrid* recommenders, which combine two or more of the above techniques in order to overcome some of their limitations.

Another common classification of RS considers *heuristic-based* (or *memory-based*) and *model-based* (Breese et al., 1998; Adomavicius and Tuzhilin, 2005) recommenders. Heuristic-based approaches essentially compute the rating prediction function F from the entire collection of user profiles by means of certain heuristics. That is, they compute rating predictions directly from all known ratings using a particular mathematical expression. Model-based approaches, on the other hand, learn a predictive model from the collection of known ratings, which represents an approximation of F. This requires a prior learning process to build the model, but thereafter the built model directly generates rating predictions, leading to fast response at recommendation time.

2.3.1 Content-based recommendations

Content-based (CB) recommender systems analyze and exploit the *contents* of items in order to find "similar" items to those known and preferred by the target user, assuming that such similar items are also interesting for the user (Adomavicius and Tuzhilin, 2005; Pazzani and Billsus, 2007). The contents of an item can be, for instance, the item itself in the case of text-based items (e.g. news articles, Web pages, books), the features or attributes of the item (e.g. "cuisine" and "service" features of a restaurant, and "genre" and "actors" attributes of a movie), and user generated descriptions assigned to the item (e.g. reviews and tags of an item commented and annotated in a social media). In content-based recommender systems, it is also common to represent a user profile as a weighted vector of content features, giving more weight to those features present in preferred items. Figure 2.3 shows an example of user and item content-based profiles for a movie RS (left side), and their vector representation (right side). In the figure, movie genres are used as content descriptions –only two features are included for the sake of simplicity. As shown in the right side of the figure, such profiles can be viewed as vectors in the space of content features.



	Action	Comedy
	0.5	0.8
	0.0	1.0
	1.0	0.0

Figure 2.3. Example of user and item content-based profiles in a movie recommender system.

Using user and item content-based profiles, CB recommender systems compute $F(u, i)$ as a score that represent how well the features of item i fit the preferences of user u

(Balabanović and Shoham, 1997; Adomavicius and Tuzhilin, 2005). Many methods from Information Retrieval (IR) and Machine Learning (ML) fields have been utilized to compute the above score (Adomavicius and Tuzhilin, 2005; Pazzani and Billsus, 2007; Lops et al., 2011).

Heuristic-based CB RS compute the score F using heuristic formulas that directly measure the similarity between contents of items in the system's catalog and the preferred items in the target user's profile. For instance, Lang (1995) uses the well-known *term frequency/inverse document frequency* (*tf-idf*) metric (Salton, 1989; Baeza-Yates and Ribeiro-Neto, 1999) of IR to compute weights of features (words in that case) in news articles to be recommended. More formally, let $|I|$ be the total number of items in the catalog, f_i be a feature that appears in n_i item's content descriptions, and $freq_{i,j}$ be the number of times that f_i appears in item *j*'s content description, d_j. Then, the *term frequency* $tf_{i,j}$ of f_i in d_j is:

$$tf_{i,j} = \frac{freq_{i,j}}{\max_l freq_{l,j}}$$

where the maximum is computed from the frequencies $freq_{l,j}$ of all features f_l that appear in d_j. This metric represents a normalized frequency of feature f_i in d_j. Nonetheless, if f_i appears in the descriptions of many items, it is not useful for distinguishing such items. To deal with this issue, the *inverse document frequency* idf_i of f_i is utilized; it is computed as follows:

$$idf_i = log\frac{|I|}{n_i}$$

Then, the *tf-idf* weight of f_i in d_i is defined as:

$$w_{i,j} = tf_{i,j} \times idf_i$$

Using vectors of feature weights, it is possible to find the items more similar to those preferred by the user by means e.g. of the cosine similarity:

$$cos(\vec{w}_{d_a}, \vec{w}_{d_b}) = \frac{\vec{w}_{d_a} \cdot \vec{w}_{d_b}}{\|\vec{w}_{d_a}\|_2 \times \|\vec{w}_{d_b}\|_2} = \frac{\sum_{i=1}^{K} \vec{w}_{i,d_a}\vec{w}_{i,d_b}}{\sqrt{\sum_{i=1}^{K} w_{i,d_a}^2}\sqrt{\sum_{i=1}^{K} w_{i,d_b}^2}}$$

where K is the total number of features. By representing the user profile as a vector of content features of the user's preferred items $\overrightarrow{CBProfile}(u)$ (see Figure 2.3), $F(u, i)$ can be computed as:

$$F(u,i) = cos\left(\overrightarrow{CBProfile}(u), \vec{w}_{d_i}\right)$$

Alternatively to heuristic-based approaches, model-based CB RS build a model of user preferences based on contents of items, and use the built model to compute F. For instance, Pazzani and Billsus (1997) use a Bayesian model (Duda et al., 2001) to classify Web pages as interesting or non-interesting for a user, given a set of pages previously rated by the user. This Bayesian classifier is used to compute the probability that a Web page described by d_j belongs to a class C_i (e.g. interesting or non-interesting) given the feature values of d_j:

$$P\left(C_i | f_1 = v_{1,j}, f_2 = v_{2,j}, \cdots, f_n = v_{n,j}\right)$$

Assuming that the feature values are independent, Pazzani and Billsus (1997) show that this probability is proportional to:

$$P(C_i) \prod_k P\left(f_k = v_{k,j} | C_i\right)$$

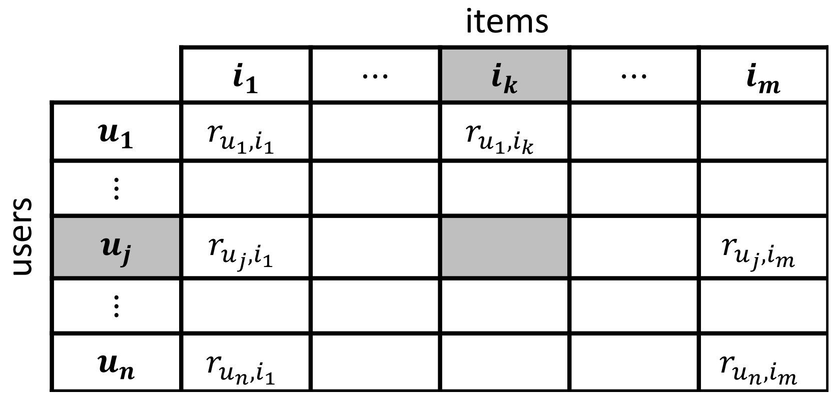
where both $P\left(f_k = v_{k,j} | C_i\right)$ and $P(C_i)$ can be estimated from training data. In this way, to classify an unrated page, the probability of each class is computed, and the page is assigned to the class with the highest probability.

The main advantage of CB RS is their ability to recommend items that have no rating assigned –avoiding the *new item* problem of collaborative filtering RS –because they only require knowing the contents of new items. CB recommendations are thus useful when data sparsity is very high, or the item catalog rapidly changes, such as in the news recommendation domain (Balabanović and Shoham, 1997). On the contrary, CB RS require some form of item content descriptions to generate recommendations. CB techniques are also limited by the number and type of features associated with the items, i.e., the *limited content analysis* problem; no CB RS can provide good recommendations if analyzed content does not contain enough and useful information. Moreover, CB RS suffer from *over-specialization*, as they suggest items similar to those items already known by the user (which are in her profile), and thus cannot provide *novel* nor *serendipitous* recommendations, which is also referred to as the *portfolio effect* (Burke, 2002). Finally, CB RS require an enough number of items preferred by the target user –the *new user* problem–, in order to have a proper knowledge about the user's preferences (Adomavicius and Tuzhilin, 2005; Pazzani and Billsus, 2007; Lops et al., 2011).

2.3.2 Collaborative filtering

Collaborative filtering (CF) recommender systems aim to find items that are liked by users with similar preferences to the target user (Adomavicius et al., 2005; Su and Khoshgoftaar, 2009; Ekstrand et al., 2011). These RS extend the social process of "word-of-mouth" phenomenon –in which people ask their peers (or look for experts' advice) about e.g. books to read or where to go on vacation– thus generating *collaborative* suggestions. Hence, CF RS do not require descriptions of item contents, but some quantitative measure of preferences from users for different items.

In CF RS, a user profile is usually represented as a vector of numeric ratings, and the set of vector profiles from all users gives form to the so-called *rating matrix* M. Figure 2.4 shows a simple example of a rating matrix. A rating matrix is usually very sparse because a typical user rates a small portion of the available items. A blank empty cell corresponds to an item that has not been rated by a particular user, and whose rating is estimated by computing the value of function F for such user and item. In this case, $F(u_j, i_k)$ represents the preference user u_j might have for item i_k based on the preferences expressed by similar-minded users (represented as the gray cell in the middle).


items
users

	$\boldsymbol{i_1}$	$\cdots$	$\boldsymbol{i_k}$	$\cdots$	$\boldsymbol{i_m}$
$\boldsymbol{u_1}$	r_{u_1,i_1}		r_{u_1,i_k}		
$\vdots$					
$\boldsymbol{u_j}$	r_{u_j,i_1}				r_{u_j,i_m}
$\vdots$					
$\boldsymbol{u_n}$	r_{u_n,i_1}				r_{u_n,i_m}

Figure 2.4. Example of rating matrix $\boldsymbol{M}$.

Heuristic-based CF RS are based on heuristic formulas that compute F directly from the ratings in matrix M. One of the most used heuristics is the neighbor-based or *k-Nearest Neighbors* (kNN) heuristic, which computes the preference of u for i as an aggregation of the ratings given to i by the u's most similar users (the nearest neighbors) (Adomavicius and Tuzhilin, 2005):

$$F(u,i) = \underset{v \in N(u)}{\text{aggr}}\ r_{v,i} \tag{2.2}$$

where $N(u)$ is the set of nearest neighbors of u, and $r_{v,i}$ is the rating given by neighbor v to item i. In this context, to find the nearest neighbors of u, a similarity or distance metric is needed (Amatriain et al., 2011; Desrosiers and Karypis, 2011). A common choice is to use a correlation metric (Adomavicius and Tuzhilin, 2005; Su and Khoshgoftaar, 2009), such as the Pearson's correlation coefficient ρ (Desrosiers and Karypis, 2011):

$$\rho(u,v)=\frac{\sum_{i\in I_u\cap I_v}\left(r_{u,i}-\bar{r}_u\right)\left(r_{v,i}-\bar{r}_v\right)}{\sqrt{\sum_{i\in I_u\cap I_v}\left(r_{u,i}-\bar{r}_u\right)^2\sum_{i\in I_u\cap I_v}\left(r_{v,i}-\bar{r}_v\right)^2}} \tag{2.3}$$

where $I_u$is the 2.3 set of items rated by u (items in u’s profile) and $\bar{r}_u$ is the average rating of u. For instance, Resnick et al. (1994) use the correlation coefficient (2.3) and the following instantiation of the aggregation function (2.2) for computing rating predictions in the GroupLens RS:

$$F(u,i)=\bar{r}_u+\frac{\sum_{v\in N(u)}\left(r_{v,i}-\bar{r}_v\right)\rho(u,v)}{\rho(u,v)}$$

These formulas can be improved to obtain more precise rating predictions. Herlocker et al. (1999) discuss several variations of the aggregation function (2.2) and the correlation coefficient (2.3), as well as other similarity metrics such as the Spearman’s correlation and the cosine similarity.

Furthermore, the above heuristics are commonly called *user-based* CF because their computations are based on sets of user neighbors. Alternatively, *item-based* CF (Sarwar et al., 2001; Linden et al., 2003) explore relationships between items. In this case, heuristics like (2.3) are modified to find items similar to the target item –the item for which a rating prediction is required–, and the rating prediction is computed as an aggregation of the ratings given to the items in the neighborhood of the target item. Figure 2.5 shows a schematic view of both approaches. The left side of the figure shows the user-based approach in which user rating vectors are compared, in order to find users similar to the target user (u_j). The right side of the figure shows the item-based approach in which item rating vectors are compared, in order to find items similar to the target item (i_k).

	$\boldsymbol{i_1}$	⋯	$\boldsymbol{i_k}$	⋯	$\boldsymbol{i_q}$	⋯	$\boldsymbol{i_m}$
$\boldsymbol{u_1}$	r_{u_1,i_1}		r_{u_1,i_k}		r_{u_1,i_q}		
⋮							
$\boldsymbol{u_j}$	r_{u_j,i_1}		?		r_{u_j,i_q}		r_{u_j,i_m}
⋮							
$\boldsymbol{u_p}$	r_{u_p,i_1}		r_{u_p,i_k}		r_{u_p,i_q}		r_{u_p,i_m}
⋮							
$\boldsymbol{u_n}$			r_{u_n,i_k}		r_{u_n,i_q}		r_{u_n,i_m}

	$\boldsymbol{i_1}$	⋯	$\boldsymbol{i_k}$	⋯	$\boldsymbol{i_q}$	⋯	$\boldsymbol{i_m}$
$\boldsymbol{u_1}$	r_{u_1,i_1}		r_{u_1,i_k}		r_{u_1,i_q}		
⋮							
$\boldsymbol{u_j}$	r_{u_j,i_1}		?		r_{u_j,i_q}		r_{u_j,i_m}
⋮							
$\boldsymbol{u_p}$	r_{u_p,i_1}		r_{u_p,i_k}		r_{u_p,i_q}		r_{u_p,i_m}
⋮							
$\boldsymbol{u_n}$			r_{u_n,i_k}		r_{u_n,i_q}		r_{u_n,i_m}

Figure 2.5. Schematic view of user-based (left side) and item-based (right side) kNN.

Model-based CF RS, in contrast, learn a predictive model from the collection of known ratings, and afterwards use this model to compute F. A successful example of model-based CF is the matrix factorization (MF) technique (Takács et al., 2008; Koren et al., 2009), an extension of the Singular Value Decomposition (SVD) technique that models user-item interactions in a latent factor space, where latent factors are used to efficiently predict unknown ratings. In general, MF techniques iteratively approximate the rating matrix M by user and item latent factor matrices P and Q of lower dimension (d in our notation). Using such latent factors, the function F can be computed as:

$$F(u,i) = \sum_{j=0}^{d} P_{j,u} Q_{j,i} = p_u^T q_i \tag{2.4}$$

where p_u and q_i are the u-th column of P and the i-th column of Q, and represent the latent factor vectors of user u and item i, respectively. Values of P and Q are computed by minimizing an estimation of a rating prediction error, such as the Frobenius norm, $\min\|M - PQ\|_F^2$.

Several other techniques from the ML field have been used for building CF rating models, including clustering (Breese et al., 1998; Rashid et al., 2007), Bayesian classifiers (Chien and George, 1999), neural networks (Salakhutdinov et al., 2007), and Latent Semantic Analysis (Hofmann, 2003, 2004), to name a few.

The main advantage of CF RS is the ability to deal with any type of item, since CF does not require item contents descriptions. Additionally, they have better chances to provide novel and serendipitous item suggestions, since they generate recommendations based on preferences of multiple users, and thus include items dissimilar to those used by the target user in the past. Nonetheless, they suffer from the new user/item problem (i.e., the *cold-start* problem), and may find difficult to generate good recommendations in case of rating *sparsity*, i.e., when only a small fraction of ratings is available. Finally, since CF relies on finding similarities between users (or items), when the target user has unusual preferences, CF may find difficult to find (an enough number of) other similar-minded users –the *gray sheep* problem– (Burke, 2002; Adomavicius and Tuzhilin, 2005; Su and Khoshgoftaar, 2009).

2.3.3 Hybrid recommender systems

Due to the different characteristics, together with the advantages and drawbacks, of existing recommendation techniques, a common practice in RS is to combine two or more of such techniques in hybrid approaches, aiming to overcome individual limitations. An alternative hybrid approach consist of combining different implementations of a particular technique (Burke, 2002, 2007; Adomavicius and Tuzhilin, 2005). Research in ML has shown that combining multiple predictive models –such as the ones used by RS– often yields better results than using an isolated model (Bishop, 2006). A well-known example of this is the Netflix Prize competition (Bennett and Lanning, 2007), where the best performing recommendation approaches corresponded to large *ensembles* of recommendation algorithms (Koren, 2009b; Piotte and Chabbert, 2009; Töscher et al., 2009).

Burke (2002, 2007) presents a detailed taxonomy of hybrid RS, identifying seven different types:

- **Weighted**: The system numerically combines the scores provided by different recommendation algorithms, by means e.g. of a voting scheme or a linear combination, to produce a single recommendation.

- **Switching**: The system switches (i.e., selects) among available recommendation algorithms depending on the current "recommendation situation." This type of hybrid approach requires some reliable criterion with which base the switching decision.

- **Mixed**: The system presents together the results from different recommendation algorithms. In this case, an appropriate combination method is required.

- **Feature Combination**: The system performs a single recommendation algorithm, which is fed with combined features derived from different knowledge sources.

- **Feature Augmentation**: The system performs several recommendation algorithms, using the output of one of them as additional input for other algorithm in turn.

- **Cascade**: The system utilizes some recommendation algorithms to refine recommendations given by others (i.e., break ties), using a pre-defined priority of algorithms.

- **Meta-level**: The system uses the model learned by one recommendation algorithm as input for another algorithm in turn. The original knowledge source is completely replaced by the model built by the contributing recommender.

Adomavicius and Tuzhilin (2005) further identify the following hybrid recommendation approaches that combine CB and CF techniques as follows:

- Implementing content-based and collaborative filtering algorithms separately and combining their predictions. This corresponds to a weighted or a switching hybrid according to the taxonomy of Burke (2002, 2007).

- Incorporating some content-based characteristics into a collaborative filtering approach. One example of this hybrid technique consist of applying a CF algorithm with user profiles that include content-based information, as done by Balabanović and Shoham (1997). This corresponds to a specific case of the feature combination or a feature augmentation hybrid approach, according to Burke's taxonomy (Burke, 2002, 2007).

- Incorporating some collaborative characteristics into a content-based approach. One example of this hybrid technique is to use a dimensionality reduction technique (Bishop, 2006), e.g. applying Latent Semantic Indexing (Deerwester et al., 1990) on content-based profiles in order to exploit their commonalities, creating a collaborative view of a collection of content-based user profiles, as done by Soboroff and Nicholas (1999). This corresponds to a specific case of the feature combination or a feature augmentation hybrid approach, according to Burke’s taxonomy (Burke, 2002, 2007).

- Constructing a general unifying model that incorporates both content-based and collaborative characteristics. The aim of this type of hybrid technique is to exploit user and item information in a single model, such as the model described by Ansari et al. (2000), which integrates user preferences and item characteristics into a Bayesian model. This corresponds to a specific case of the feature combination hybrid approach, according to Burke’s taxonomy (Burke, 2002, 2007).

The selection of the best hybrid approach for a particular situation depends on the characteristics of the recommenders to combine, the data available, and the run-time efficiency requirements (Burke, 2002, 2007). For instance, some hybrid approaches, such as the weighted approach, assume that the individual recommenders have uniform performance, which is not always the case.

2.3.4 Other techniques

Content-based and collaborative filtering recommendations are the most used techniques in RS (Adomavicius and Tuzhilin, 2005; Su and Khoshgoftaar, 2009; Ekstrand et al., 2011; Lops et al., 2011; Ricci et al., 2011), but other techniques have been proposed in the literature. Burke (2002, 2007) distinguishes three types of alternative RS: *demographic-based*, *knowledge-based*, and *community-based* (recently called as *social-based*). Ricci et al. (2011) also emphasize *context-aware* techniques as an additional approach to RS.

Demographic-based RS utilize demographic information available in the user profile, e.g. age, gender, educational level, and country of residence, to produce recommendations targeted to specific demographic niches. For instance, Pazzani (1999) describes how to exploit the users’ gender, age, and area code (location) to identify types of users that prefer certain restaurants. As noted by Ricci et al. (2011), this technique is popular in the marketing literature, but has attracted relatively little attention in RS community.

Knowledge-based RS exploit specific domain knowledge about how certain item features meet the users’ needs and preferences. Examples of knowledge-based RS are *case-based reasoning systems* that use examples of user goals and related items as a source to find items with similar features (Burke, 2000), and *constraint-based systems* that apply explicit rules about how to relate user goals with item features (Felfernig et al., 2011). Knowledge-based RS do not suffer from cold-start, but require experts' domain knowledge encoded in the system, a problem known as *knowledge acquisition bottleneck* (Felfernig et al., 2011).

Social-based RS focus on exploiting preferences from the user's friends, as opposed to exploiting preferences from all the community of users. This technique is based on the idea that people rely more on recommendations from their friends than from other unknown users (Sinha and Swearing, 2001). The growing popularity of social networks such as Facebook and Twitter[9] has been attracting interest in this approach within the RS field, being an open research topic (Ricci et al., 2011).

Context-aware RS (CARS) exploit the *context* (e.g. location, time, weather, device, and mood) in which users use or consume items (Adomavicius et al., 2005; Adomavicius and Tuzhilin, 2011). In this way, CARS can discriminate the interest a user may have in a particular item within different contexts and situations. Several approaches have been proposed to deal with contextual information (Adomavicius et al., 2005; Adomavicius and Tuzhilin, 2011; Baltrunas, 2011; Panniello and Gorgoglione, 2012). In general, CARS require one or more of the explained basic CB and CF techniques as underlying methods for computing recommendations, and somehow take into account contextual information in the process of generating recommendations. Information exploited by other techniques can also be used as a proxy of context, e.g. demographic data such as age and gender (Baltrunas and Ricci, 2009a). Hence, we consider CARS techniques as enhancements for improving other RS techniques, rather than a pure technique for computing recommendations. We deepen into CARS techniques in Chapter 3.

2.4 Evaluation of recommender systems

In the past most research on RS has focused on designing and improving the performance of proposed recommendation algorithms (Herlocker et al., 2004; Shani and Gunawardana, 2011). In order to compare and select the best performing among several alternative algorithms, it is necessary to measure and compare their performance. This comparison is usually made empirically based on experiments that test the algorithms performance either in an *online* or an *offline* setting, by applying a particular *evaluation protocol*, that is, by using certain *evaluation metrics* –which define what to assess– and a given *evaluation methodology* –which defines how to assess. In this section we briefly describe these concepts.

[9] http://www.twitter.com

2.4.1 Online and offline evaluation

Broadly speaking, two types of evaluations can be performed to assess the performance of recommender systems, namely online evaluation (also called user-based evaluation), and offline evaluation (also called system evaluation) (Herlocker et al., 2004; Gunawardana and Shani, 2009; Shani and Gunawardana, 2011).

Whatever the evaluation case, a RS is built with information about the users –such as preferences and demographics– and the items –such as content descriptions and attributes. Then, user responses to received recommendations are tracked, and are used to compute certain metrics related to one or more desired properties of the recommendations, e.g. accuracy, diversity, and novelty of rating predictions, and user satisfaction.

In online evaluation, users interact with several settings of the system under evaluation, and may fill questionnaires regarding their experience with the system and the received recommendations. Evaluation results are then obtained by recording and comparing the users' behavior (ratings, activity logs, etc.) over different system settings (Kohavi et al., 2009), by means of subjective user perceptions gathered in the questionnaires (Knijnenburg et al., 2012), or by combinations of both. In offline evaluation, on the other hand, datasets containing past user behavior are used to simulate how users would have behaved if they had used the evaluated system. In this case, evaluation results are obtained by comparing predicted and actual ratings for users and items of the dataset (Herlocker et al., 2004; Shani and Gunawardana, 2011).

Online evaluation may be considered preferable to offline evaluation, mainly due to its ability to take into account the user's experience (Knijnenburg et al., 2012; Konstan and Riedl, 2012). That is, the user's perceptions about the interaction with the system. Moreover, some studies have shown differences between offline metric results and user perceived quality (Cremonesi et al., 2011). Although there is no a clear explanation, variations in user interfaces (Cosley et al., 2003), data selection (Cremonesi et al., 2011), and situational and personal characteristics of users (Knijnenburg et al., 2012) may be related with such differences.

Despite its advantages, online evaluation is more difficult and expensive to perform, as it requires counting with (fully) functional implementations of the system's settings to evaluate. Moreover, users have to be recruited and probably be paid for testing the system. Offline evaluation, on the other hand, only requires implementing the system's algorithmic settings to be tested. If a dataset is already available, no user recruiting is needed. Thanks to historical data availability, offline evaluation brings a low cost, and easy to reproduce experimental environment for testing new algorithms, and distinct settings of a particular algorithm. Thus, a common practice is to test new recommendation algorithms by offline evaluations, especially as a preceding step to online evaluation, in which only the best (offline) performing algorithms would be tested. In this way, overall experimentation costs are reduced (Kohavi et al., 2009; Shani and Gunawardana, 2011). Because of these issues, in this thesis we focus on offline evaluation.

2.4.2 Offline evaluation methodologies

In the literature a large variety of evaluation protocols –metrics and methodologies– has been proposed for offline evaluation of RS. In order to facilitate the analysis of existing RS offline evaluation approaches, in this section we describe the main steps that should be followed to conduct an offline evaluation of a RS. Particular implementations of these steps give form to distinct evaluation methodologies.

In general, a recommendation model is built (or equivalently a recommendation heuristic is computed) with available user data. Afterwards, its ability to deliver good[10] recommendations is assessed somehow with additional user data. In an offline evaluation scenario, we have to simulate the users' actions after receiving recommendations. This behavior is reproduced by splitting the set of available ratings into a *training set* (Tr) –which serves as historical data to learn the users' preferences– and a *test set* (Te) –which is considered as knowledge about the users' decisions when faced with recommendations, and is commonly referred to as *ground truth* data. Since test data should not be accessible during the model/heuristic building process, in general, the only restriction that must be hold is to avoid pairwise user-item rating overlaps between training and test sets, i.e., $Tr \cap Te = \emptyset$.

Figure 2.6 shows a schematic view of the generic stages of an offline evaluation protocol for RS. In the figure the ratings matrix M is partitioned into a training set Tr and a test set Te, using a training-test splitting process. Once a model (or heuristic) is built with Tr, the recommendation process is performed to generate a set of item suggestions for each user. These item suggestions are then compared against the ground truth Te using a number of metrics. In this context, additional processing of data may be required depending on the recommendation task at hand, among rating prediction and top-N recommendations tasks (Herlocker et al., 2004; Gunawardana and Shani, 2009). In the former task, recommendations correspond to rating predictions, and the metrics are computed by comparing predicted and actual values of test ratings. In the latter task, recommendations consist of a ranked list of items predicted as the most appealing for the user. Here, metrics take into account the ranking positions of relevant and non-relevant test items in the generated list (Cremonesi et al., 2010; Bellogín et al., 2011). In this case the notion of item relevance can take multiple definitions, e.g. by considering relevant items those whose actual rating values are over certain threshold.

[10] There is no general definition of what *good* recommendations are. Nonetheless, a commonly used approach is to establish the quality (goodness) of recommendations by computing different metrics that assess various desired characteristics of a RS output.

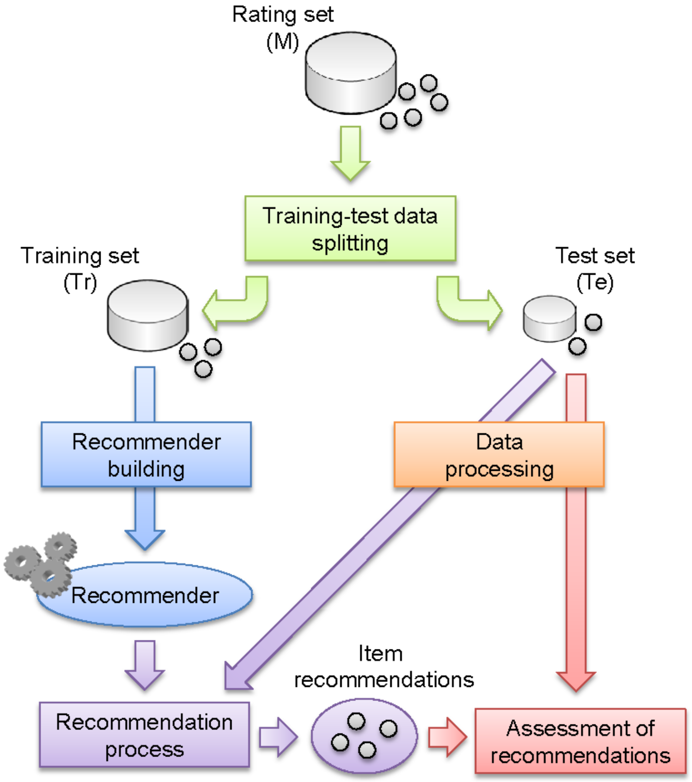


Figure 2.6. Schematic view of the generic stages followed in an offline evaluation protocol for recommender systems.

2.4.3 Evaluation metrics

A wide array of metrics has been proposed and used to evaluate and compare recommendation algorithms, attempting to assess different properties of generated recommendations (Herlocker et al., 2004; Gunawardana and Shani, 2009). In the literature most of the published evaluations of RS have focused on rating prediction accuracy metrics, such as the Mean Absolute Error (MAE) and the Root Mean Squared Error ($RMSE$), which measure how well a RS can predict the ratings of particular items (Gunawardana and Shani, 2009):

$$MAE = \frac{\sum_{r_{u,i} \in Te} \left| \hat{r}_{u,i} - r_{u,i} \right|}{|Te|} \tag{2.5}$$

$$RMSE = \sqrt{\frac{\sum_{r_{u,i} \in Te} \left(\bar{r}_{u,i} - r_{u,i} \right)^2}{|Te|}} \tag{2.6}$$

These metrics provide an estimation of the deviation of prediction values from true ones, being RMSE more sensitive to large errors (Herlocker et al., 2004), and being the lower values those that indicate better accuracy. Recently, it has been argued, however, that ranking precision metrics are better suited for recommendation purposes, as RS are typically required to present a limited number of the most appealing items for a user – instead of rating predictions for individual items (Konstan and Riedl, 2012). For such purpose, in general, an item ranking for user u –denoted by I_{rank_u}– is generated by comparing (and sorting) rating predictions, and the top-N items in the ranking I_{topN_u} are delivered as recommendations. Then, ranking precision metrics measure to what degree the list of recommendations contains relevant items for the users –we denote by I_{rel_u} the set of relevant items for u– (Herlocker et al., 2004). These metrics usually correspond to adaptations of metrics used in the IR field, such as Precision (P), Recall (R), F-measure (F), and normalized Discounted Cumulative Gain ($nDCG$) (Baeza-Yates and Ribeiro-Neto, 1999), and metrics used in the ML field, such as the Receiver Operating Characteristic curve, and the Area Under the Roc curve (AUC) (Ling et al., 2003). P, R, F and $nDCG$ can be computed by:

$$P = \frac{1}{|U_{Te}|} \cdot \sum\nolimits_{u \in U_{Te}} \frac{\left|I_{rel_u} \cap I_{topN_u}\right|}{\left|I_{topN_u}\right|} \tag{2.7}$$

$$R = \frac{1}{|U_{Te}|} \cdot \sum\nolimits_{u \in U_{Te}} \frac{\left|I_{rel_u} \cap I_{topN_u}\right|}{\left|I_{rel_u}\right|} \tag{2.8}$$

$$F = \frac{2PR}{(P + R)} \tag{2.9}$$

$$nDCG = \frac{1}{|U_{Te}|} \cdot \sum\nolimits_{u \in U_{Te}} \frac{DCG_u}{IDCG_u} \tag{2.10}$$

being U_{Te} the set of users with ratings in Te, $DCG_u = rel_{u,1} + \sum_{pos=2}^{N} \frac{rel_{u,pos}}{\log_2 pos}$, $rel_{u,pos}$ the relevance value for user u of the item at position pos in I_{rank_u}, and $IDCG_u$ the *ideal* DCG_u, that is, DCG_u computed over a full known ranked relevance items list. It is also common to use the notation $P@N$, $R@N$, $F@N$, and nDCG@N respectively, to indicate the ranking position or cutoff N up to which items are considered recommended in the computation of these metrics. In ML literature the items $I_{rel_u} \cap I_{topN_u}$ are usually referred to as *true positives* because they represent the set of recommended (positive) items that are truly relevant. The AUC (Ling et al., 2003) can be computed as:

$$AUC = \frac{1}{|U_{Te}|} \cdot \sum\nolimits_{u \in U_{Te}} \frac{S_0 - \left|I_{rel_u}\right|\left(\left|I_{rel_u}\right| + 1\right)/2}{\left|I_{rel_u}\right|\left|I \backslash I_{rel_u}\right|} \tag{2.11}$$

where $S_0 = \sum_{i \in I_{rel_u}} rank(i)$, and $rank(i)$ is the rank position of item i. Note that, in general, rating prediction accuracy metrics are used to assess a rating prediction task, while ranking precision metrics are used to assess a top-N recommendations task.

Apart from prediction accuracy and ranking precision, other recommendation properties are recently under research. This is the case of novelty and diversity (Vargas and Castells, 2011), by means of metrics like Self Information (*SI*) (Zhou et al., 2010), and Intra List Similarity (ILS) (Ziegler et al., 2005) metrics,

$$SI = \frac{1}{|U_{Te}|} \cdot \sum\nolimits_{u \in U_{Te}} \frac{\sum_{i \in I_{topN_u}} \log_2 \left(\frac{|U|}{U_{rel_i}} \right)}{\left| I_{topN_u} \right|} \tag{2.12}$$

$$ILS = \frac{1}{|U_{Te}|} \cdot \sum\nolimits_{u \in U_{Te}} \frac{\sum_{i \in I_{topN_u}, j \in I_{topN_u}, i \neq j} sim(i, j)}{2} \tag{2.13}$$

where U_{rel_i} denotes the set of users to whom item i is relevant, and $sim(i, j)$ denotes a similarity metric between i and j, e.g. ρ.

Novelty metrics aim to capture the degree in which unknown items (for a user in particular or for the overall community) are recommended, whereas diversity metrics assess how similar the items in a recommendation list are.

2.5 Summary

Recommender systems are successful tools that help users to find items suited to their preferences and needs on overwhelming collections, available through e.g., Web-based applications. In this chapter we have revised the most common concepts of the recommendation problem and tasks, as well as their sources of knowledge, the main techniques developed in their implementation, and the evaluation methodologies and metrics used in their evaluation.

Despite the advances in the field, there are several open problems that require attention from the research community. For instance, as noted by Adomavicius and Tuzhilin (2005), not all transactional information available in databases is exploited by most RS. In fact, most techniques reviewed in this chapter can be modeled through a rating estimation function $F(u, i)$ that depends only on the user and the item, leaving out any other contextual information. An example of contextual information is the time information associated to preferences. In Chapter 3 we shall review RS techniques particularly suited for exploiting time information. The evaluation of RS also represents an open area of research. The existence of multiple ways to implement evaluation protocols shows the need of working towards a standardization of evaluation methodologies, in order to facilitate reproducibility and comparability of RS.

Chapter 3

Time-aware recommender systems

Time-aware recommender systems are a type of context-aware recommender systems that take advantage of contextual information in the form of time. A wide range of approaches on modeling and exploiting such information for recommendation purposes have been proposed in the literature.

In this chapter we present a comprehensive review of the literature on time-aware recommender systems, starting with a description of the relation between them and the more general context-aware recommendation approaches. In Section 3.1 we describe a generic approach to incorporate contextual information in the recommendation process, and in Section 3.2 we discuss particularities of time as context information for recommendation. Next, in Section 3.3 we detail the different time-aware recommendation approaches revised, classifying them according to how time is treated, and in Section 3.4 we describe the methodologies followed to evaluate such approaches. Finally, in Section 3.5 we conclude with a summary of the chapter.

3.1 Incorporating contextual information into recommender systems

Context is a multifaceted concept that has been studied across different research disciplines, and has been defined in multiple ways (Adomavicius and Tuzhilin, 2011). Hence, Bazire and Brézillon (2005) compile 150 definitions of context from various disciplines such as computer science, economy, and philosophy. Dey (2001) states that "context is any information that can be used to characterize the situation of an entity", where in the case of a recommender system an entity can be a user, an item, or an experience the user is evaluating (Baltrunas and Ricci, 2013). Bazire and Brézillon (2005) observe that, in psychology, it is common to analyze a person performing a task in a given situation, aiming to state which context is relevant from the context of the person, the context of the task, and the context of the interaction. In the case of recommendation, the interaction between users and items is the key piece of information. Thus, in most of context-related recommender systems (RS) research, any information regarding the situation in which a user experiences (interacts with) an item – e.g. location, time, weather, device, and mood – is considered as context.

The importance of including context in the recommendation process can be observed from a practical viewpoint through a classic, simple example in the tourism domain: Although a user may love skiing, recommending her a ski resort during summer is questionable. Such recommendation could be detrimental for the user's trust in the system if interpreted as 'out of context.' In fact, in a user study comparing several RS that use and do not use contextual information, Gorgoglione et al. (2011) found that the former provide more user trust in the delivered recommendations. Moreover, they detected that trust affect purchasing behavior, and observed that RS exploiting context information increase the user's trust and levels of sales.

Recommender systems that exploit any of the above types of information are known as context-aware RS (CARS) (Adomavicius and Tuzhilin, 2011). Different approaches and techniques have been proposed for developing CARS. Most of them follow a *representational* view of context (Dourish, 2004), which assumes that context can be described by means of attributes that can be observed. In this view, context can be represented as a set of variables or contextual attributes known *a priori*. For instance, Panniello et al. (2009a) present a hierarchical structure of contextual information for the context *period of the year* of purchases on an e-commerce portal. Figure 3.1 shows this hierarchy, which considers two broad contexts: winter and summer. In a second level, each of these contexts is further split into more specific periods.

In contrast, the *interactional* view (Dourish, 2004) assumes that context is not a property that can be defined in advance, but "it arises from the activity," i.e., it assumes that observable user behavior is influenced by an underlying context (that is not necessarily observable). In this view, context is a property discovered from the user's behavior, whose scope is defined dynamically, and may or may not be relevant to some particular activity. Hence, most work in RS using this view does not define a set of fixed context variables, but infers them from the user's actions with the system. Herlocker and Konstan (2001) provide an example of this, aiming to performing task-focused recommendations. In that work, the context of interest is the user task, which is inferred from a number of example items related to the task. The example item set, called the *task profile*, is then used to find task-associated items –we note that no explicit description of the task is generated. For instance, if the example items are a hammer and a screwdriver, the system may infer that woodwork is the current task. Then, the system may recommend nails and screws to the user.

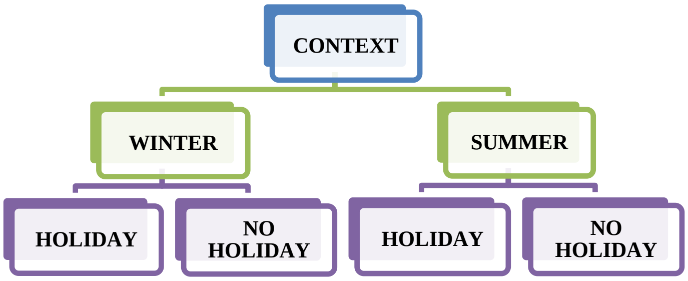


Figure 3.1. Hierarchical structure of contextual information used in (Panniello et al., 2009a)**.**

In this thesis we follow the representational view because it provides a computationally feasible approach (Baltrunas, 2011), and is the approach mostly used in time-aware RS research.

Based on the representational view of context, and extending the definition of recommendation problem given in Chapter 2 (Eq. 2.1), Adomavicius et al. (2005) present a generic model for CARS that incorporate additional dimensions of contextual information C into the rating computation formula F:

$$F: U \times I \times C \rightarrow R \tag{3.1}$$

This model assumes that the context can be *known* and represented as a set of contextual dimensions $C_1, C_2, \dots, C_n \in C$, where each dimension C_i has its own type and domain. Moreover, a dimension C_i may have different representations reflecting the complex nature of its contextual information (Adomavicius and Tuzhilin, 2011). For instance, the contextual dimension *location* can be defined as a plain list of values, e.g. {home, abroad}, or can be defined by means of a hierarchical structure such as room → building → neighborhood → city → country. In general, according to Palmisano et al. (2008) and Adomavicius et al. (2005), each contextual dimension $C_i \in C$ can be defined as a set of attributes $C_i = \{C_i^1, C_i^2, \dots, C_i^q\}$ that may be independent or related through some kind of structure.

Adomavicius and Tuzhilin (2011) established a classification of CARS based on the algorithmic approach for contextual information treatment, considering *contextual pre-filtering*, *contextual post-filtering* and *contextual modeling* systems. In contextual pre-filtering, the target recommendation context –i.e., the context in which the target user expects to consume the recommended items– is used to filter user profile data relevant to such context before the rating prediction computation. In contextual post-filtering, rating predictions are adjusted according to the target context after being computed (on entire user profiles). In both cases, traditional non-contextualized recommendation algorithms can be performed, as the contextualization involves an independent pre- or post- processing computation. On the other hand, contextual modeling incorporates context information directly into the model used to compute rating predictions. In this way contextual modeling lets effectively extend and exploit user-item relations with context information without the need of discarding (valuable) data or adapting generated recommendations for providing contextualized suggestions.

3.2 Specifying time context in recommender systems

Among existing contextual dimensions, the *time dimension* –i.e., the contextual signals related to time, such as *period of the day*, *day of the week*, and *season of the year*– has the advantage of being easy to collect, since almost any system can record item usage/consumption/rating timestamps. Moreover, as noted in the ski resort recommendation example given in the previous section, the time dimension can serve as a valuable input for improving recommendation quality (Baltrunas and Amatriain, 2009; Koren, 2009a; Panniello et al., 2009b).

Time-aware recommender systems (TARS) can be considered as a specialized type of CARS. Their main characteristic is the usage of time context information at some stage of the rating prediction process, being able to provide differentiated recommendations depending on the target recommendation time –i.e., the desired time for item usage or consumption, which may be different from the recommendation delivery time– according e.g. to the preferences expressed by the users at similar time contexts in the past. Thus, the general formulation of context-dependent rating prediction can be particularized for the time dimension of context, T, as follows:

$$F\colon U \times I \times T \to R \tag{3.2}$$

where T can be represented in several ways. According to Merriam-Webster[11], time is defined as “a non-spatial continuum that is measured in terms of events that succeed one another from past through present to future.” From this definition, it follows that it is possible to establish an order between time events (or time values), e.g. night is after

[11] http://www.merriam-webster.com/dictionary/time

evening, and Monday is before Tuesday. A second sense of time is “the measured or measurable period during which an action, process, or condition exists or continues.” Given the huge differences in duration of various processes (consider e.g. the duration of a movie, and the human lifetime), several time units have been used, e.g. hours, days, months and years (Whitrow, 1988), together with hierarchies of time units (e.g. a day “is formed” by 24 hours, and a week “is formed” by 7 days). This hierarchical structure and the fact that time is a continuum, lead to a cyclic conception of time where its values repeat periodically.

Due to this flexibility in the time conception and measurement, different representations of time context information can be used. For example, time may be modeled as a continuous variable whose values are the specific times at which items are rated/consumed (e.g. a *timestamp* like “*January 1st, 2000 at 00:00:00*”). Another option is to specify categorical values, regarding time periods of interest in the recommendation domain at hand. For instance, in the tourism domain, a seasonal variable like *season of the year* = {*hot_season*, *cold_season*} may be convenient, whereas in the music and movie domains, the variable *period of the week* = {*workday*, *weekend*} may have more sense. A hierarchical modeling could also be used, enabling to control the degree of granularity of the time context information (e.g. *period of the week* can be disaggregated into *day of the week* = {*Monday, Tuesday, … , Sunday*}). In this sense, it has to be noted that storing the *timestamp* of user actions is the most flexible option, since it lets exploit diverse representations of time context, including both continuous and (maybe overlapping) categorical values.

In general, collecting time information of user interactions does not require additional user effort nor impose strict system/device requirements. Many time-aware recommendation models exploit collected time information related to past user preferences, e.g. the timestamps associated to ratings and item consumptions by users. Moreover, it has been used as a key input to achieve significant improvements on recommendation accuracy (Koren, 2009a). Hence, the timestamps of collected user preferences are valuable, easy-to-collect data for improving recommendations.

At this point, we note that when a RS exploits ratings instead of usage/consumption data, the collected rating timestamps do not necessarily correspond to real usage/consumption time, and thus may not be considered as the context in which users prefer using/consuming items. Nonetheless, Said et al. (2011) found that users tend to rate items shortly after consuming them –a fact that lets relate rating preferences with some time context signals.

Other sources of time information can also be collected and exploited. Examples of interesting events are the time of the items’ incorporation into the system’s catalog, and the time of the users’ registration into the system’s community. We denote by $\mathcal{T}(e)$ the function returning the time associated to an event e.

Due to the benefits and flexibility of time context, recent years have been prolific in the research and development of TARS exploiting explicit and implicit user feedback. In the next section we detail on the approaches to time-aware recommendation.

3.3 Time-aware approaches in recommender systems

A wide array of approaches on modeling and exploiting time context information has been proposed in the RS literature. In order to get a comprehensive landscape on existing approaches, in this section we review and classify a large number of papers about TARS.

First approaches considering time information in RS date back to 2001. Zimdars et al. (2001) treated the CF recommendation problem as a time series prediction problem, encoding the time-dependent order of the data. In a more generic perspective, Adomavicius and Tuzhilin (2001) proposed the use of a multi-dimensional representation of RS in order to deal with contextual information, including the time dimension among others.

Despite this early work, the topic recalled the attention of researchers more recently; proposals by Adomavicius et al. (2005) and Koren (2009a) have had a strong influence in the field, which is producing an increasing number of RS approaches exploiting some form of time information.

For the sake of simplicity, we roughly group related work by the type of treatment given to time information. In this sense, we identify approaches that adapt rating predictions depending on the target recommendation time, and represent time as a continuous contextual variable *–continuous time-aware RS–* or as categorical context variables *–categorical time-aware RS*. Additionally, there are approaches that exploit time context information in a more subtle way, without differentiating rating prediction according to the target recommendation time, but rather adjusting some parameters or data dynamically *–time adaptive RS*.

Additionally, for each TARS approach, we distinguish between *heuristic-based* (or memory-based) and *model-based* approaches, following a common classification of recommender systems (Breese et al., 1998; Adomavicius and Tuzhilin, 2005). As described in Chapter 2, heuristic-based approaches use the collection of ratings for computing predictions, whereas model-based approaches learn a model from rating data, which is afterwards used to compute predictions.

3.3.1 Continuous time-aware recommender systems

In this type of TARS, time information T is represented as a continuous variable. The rating prediction becomes an explicit function of the target recommendation time t, $\hat{r} = F(u, i, t)$, where $t \in T$ is measured in time units such as seconds, days, and years. Note that this formulation lets define a target recommendation time different from the current time; the recommendation may be required for a time different from the requesting time, e.g. the user can ask “what movie could I see tomorrow?”

In the case of heuristic-based continuous TARS, heuristics for computing rating predictions incorporate continuous time information into their analytic formulas. A common approach of this type of TARS is to differently weight ratings according to their “age” (distance, in terms of time) with respect to the target time, generally in the form of an increasing penalization on older data, under the assumption that more recent ratings better reflect current user tastes and interests. In user-based kNN (see Eq. 2.2) this leads to:

$$F(u,i,t) = \underset{v \in N(u)}{\mathrm{aggr}}\, r_{v,i} \cdot w_t\left(t, \mathcal{T}\left(r_{v,i}\right)\right) \tag{3.3}$$

where $t \in T$ denotes the target (recommendation) time, and $w_t(\cdot,\cdot)$ returns a time-dependent weight. For instance, Ding and Li (2005) proposed an exponential time decay weight $w_t\left(t, \mathcal{T}\left(r_{v,i}\right)\right) = e^{-\lambda \cdot \left(t - \mathcal{T}\left(r_{v,i}\right)\right)}$, being λ a constant value representing the decay rate. In this formulation, the value of λ is computed as $\lambda = 1/T_0$, being T_0 the *half life*, that is, the weight of a rating reduces approximately by $1/2$ after T_0 days. Figure 3.2 shows some typical examples of weight curves generated with Ding and Li’s exponential time decay model, using different values of T_0.

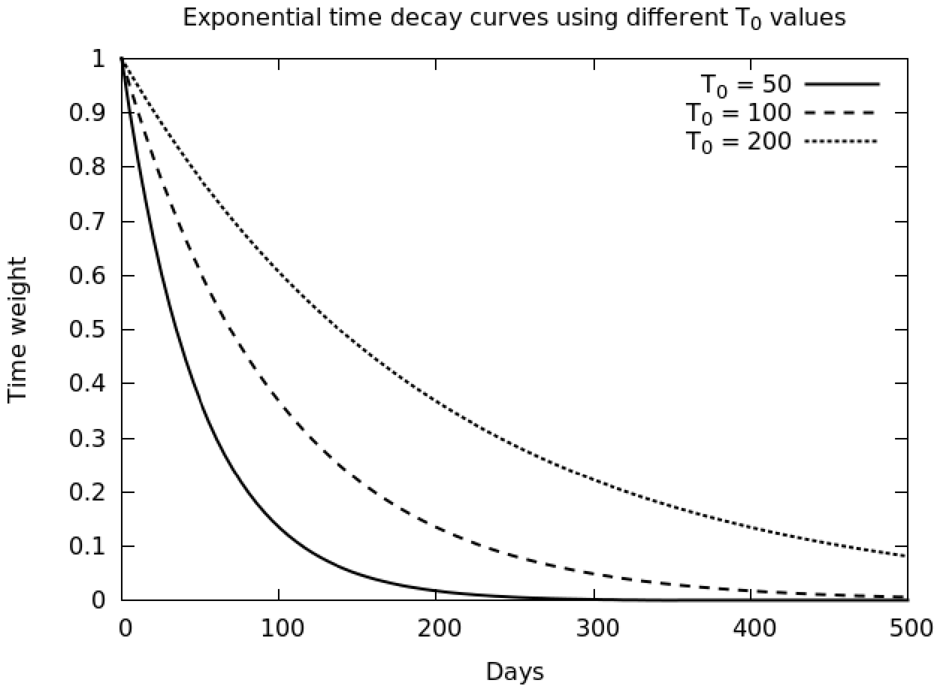


Figure 3.2. Example of exponential time decay weights using different T_0 values.

The notion of time weight can also be used to estimate the similarity between users or items. For example, Hermann (2010) used the heuristic $s_t\left(t, \mathcal{T}(r_{u,i}), \mathcal{T}(r_{u,j})\right) = 1/\left(\left|\mathcal{T}(r_{u,i}) - \mathcal{T}(r_{u,j})\right| + \left|min\left(\mathcal{T}(r_{u,ji}), \mathcal{T}(r_{u,j})\right) - t\right|\right)$ as a measure of time similarity between items consumed by user u. The most extreme case of this approach is that in which $w_t(\cdot,\cdot)$ is 0 (i.e., the data is discarded) if the time distance between $\mathcal{T}(r)$ and t is over some specified threshold (Gordea and Zanker, 2007; Campos et al., 2010). This is sometimes referred to as instance selection, time window, or time truncation.

Model-based continuous TARS build models from users rating data, taking the dynamics of such data into account. As in heuristic-based continuous TARS, time is represented as a continuous variable, and the target time is explicitly considered for rating prediction. There is not a general formulation for these approaches, whose formulations strongly depend on the proposed models. One of the best known examples is the temporal dynamics model proposed by Koren (2009), which corresponds to a MF model (see Eq. 2.4). In order to take time effects in the MF model into consideration, Koren incorporates into 2.4 static and dynamic bias terms as follows:

$$F(u, i, t) = \mu + b_u(t) + b_i(t) + p_u^T(t)q_i$$

where μ denotes the overall mean rating, $b_u(t)$ and $b_i(t)$ are user- and item-specific time-dependent biases. User factors represented as $p_u(t)$ are assumed to change through time, becoming time-aware. Note that this model assigns latent factor vector(s) to a user at each time t. Figure 3.3 shows a schematic view of a static user factor vector, and the corresponding bi-dimensional time-aware factor vector. For each user and factor, there are several values, one per time unit. In the formulation of (Koren, 2009a), t is measured in days, being able to detect changes in the users' preferences with a daily granularity, and in (Rendle et al., 2011), an additional factorization-based model including time-variant factors is described.

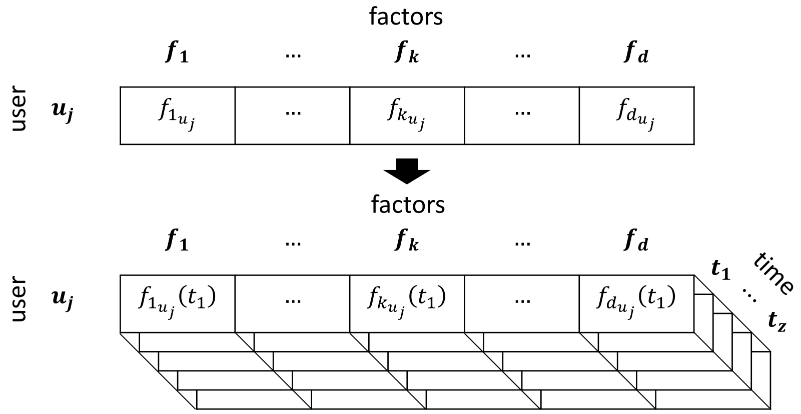


Figure 3.3. Schematic view of static user factors (upper side) and time-aware user factors (lower side) vectors.

Another example of a continuous time-based model is given in (Xiong et al., 2010), where a Bayesian probabilistic Tensor Factorization (TF) model is proposed. In that work Xiong and colleagues incorporate time as an additional feature vector associated to each user-item pair, instead of to each user and factor as in (Koren, 2009a). In this way, the $|U| \times |I|$ rating matrix M is extended into a three dimensional *tensor* $\mathfrak{M} \in \mathbb{R}^{|U|\times|I|\times|T|}$. We note that a tensor is a generalization of the matrix concept to three or more dimensions. Figure 3.4 shows a schematic view of a ratings tensor. In the figure the tensor is composed of user, item, and time dimensions.

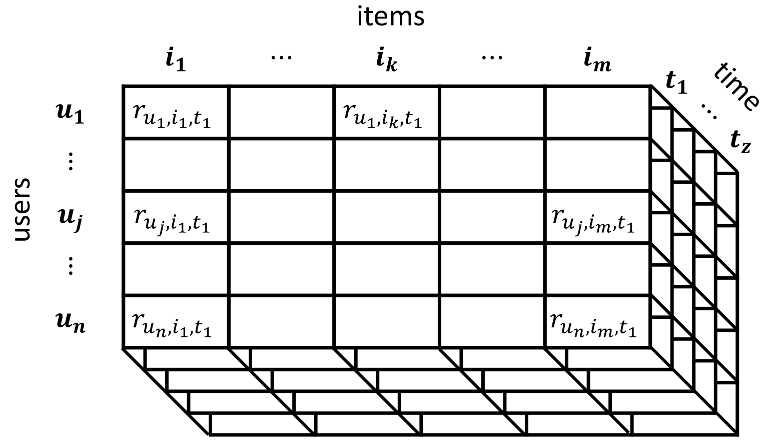


Figure 3.4. Schematic view of a three dimensional rating tensor.

Under this scheme, users, items and time are modeled via probabilistic latent factor vectors that are computed by means of the TF approach. Hence, rating prediction is computed by the scalar product:

$$F(u,i,t) = \sum_{j=0}^{d} P_{j,u} Q_{j,i} W_{j,t}$$

where $P_{\cdot,u}$, $Q_{\cdot,i}$ and $W_{\cdot,t}$ denote the feature vectors of user u, item i, and time t, respectively. This formulation avoids an expensive increase of time-related factors associated with entities, as in the case of MF models.

A different modeling scheme is presented by Koenigstein et al. (2011), where they use *session* factors to model specific user behavior in music listening sessions. Such sessions are inferred from time information associated to ratings, in such a way that a session is defined as a set of consecutive ratings with no more than 5 hours of difference. The authors found that users tend to rate songs in a session very similarly.

We note that, from the temporal perspective, the main disadvantage of this type of models is the inability to extrapolate future temporal dynamics. Authors, however, argue that these models isolate persistent signals from transient noise, thus helping to predict future user behavior.

3.3.2 Categorical time-aware recommender systems

In this type of TARS, the time dimension T is modeled as one or more discrete variables $T^1, T^2, \cdots T^n \in T$ that let treat ratings differently depending on their contextual values. Under this formulation, the possible target time is one of the values of the contextual variables, and no references to time ordering can be made (e.g. a user can ask "what movies may I see in the weekends?", but not "what movies may I see the *next* weekend?"). The main difference between continuous TARS and categorical TARS is given by the domain of the time information they use; in categorical TARS, time information is represented as discrete contextual values.

Heuristic-based categorical TARS include discrete time context information in their heuristics. Generic CARS algorithms exploiting time by *contextual pre-filtering* and *contextual post-filtering* strategies belong to this category. A particular time context is represented as $t = \bigcup_j t^j \mid t^j \in T^j$. For instance, given $T^1 = \{morning, evening\}$ and $T^2 = \{workday, weekend\}$, two allowed values of t are $t_1 = \{morning, weekend\}$, and $t_2 = \{evening, workday\}$. Thus, in this case, there is no possibility to order the data by means of their timestamps, that is, there is no "older" data, but rather data relevant (or not) for a particular context t. This change in modeling time information leads to a different type of heuristics for computing $F(u, i, t)$ than those used in continuous TARS. In this case, a time-dependent filter z_t is used, which can be viewed as a penalty applied to data non-relevant for the target context t. Depending on the contextualization strategy, this filters a different role in the computation of $F(u, i, t)$.

In contextual pre-filtering, z_t is used as a filter to select relevant ratings for prediction computations, being computed in general as $z_t\big(t, \mathcal{T}(r)\big) = 1$ when $t = \mathcal{T}(r)$, and $z_t\big(t, \mathcal{T}(r)\big) = 0$ otherwise. In this way, a set M^t of ratings relevant to context t, $M^t = \{\, r_{u,i} | r_{u,i} \in M, z_t\left(t, \mathcal{T}\left(r_{u,i}\right)\right) = 1\}$, is selected, and prediction computations are performed using M^t and a model like 2.1. Figure 3.5 shows an example of ratings selection in contextual pre-filtering. In the figure, two contextual values are considered. The ratings in M^t corresponding to the target context (those shadowed) are the only ones used in prediction computations.

For instance, Baltrunas and Amatriain (2009) created contextual micro-profiles, each of them containing ratings of a user in a particular context, as a pre-filtering strategy aimed to better detect the user's preferences for specific time contexts. Only those micro-profiles that correspond to the target context are used for computing recommendations. The authors tested several contextual schemes, such as *timeOfTheDay* = {*morning*, *evening*}, *timeOfTheWeek* = {*workday*, *weekend*} and *timeOfTheYear* = {*hot_season*, *cold_season*}, obtaining improvements on accuracy metrics. Figure 3.6 shows a schematic view of contextual micro-profiles created from a user profile.

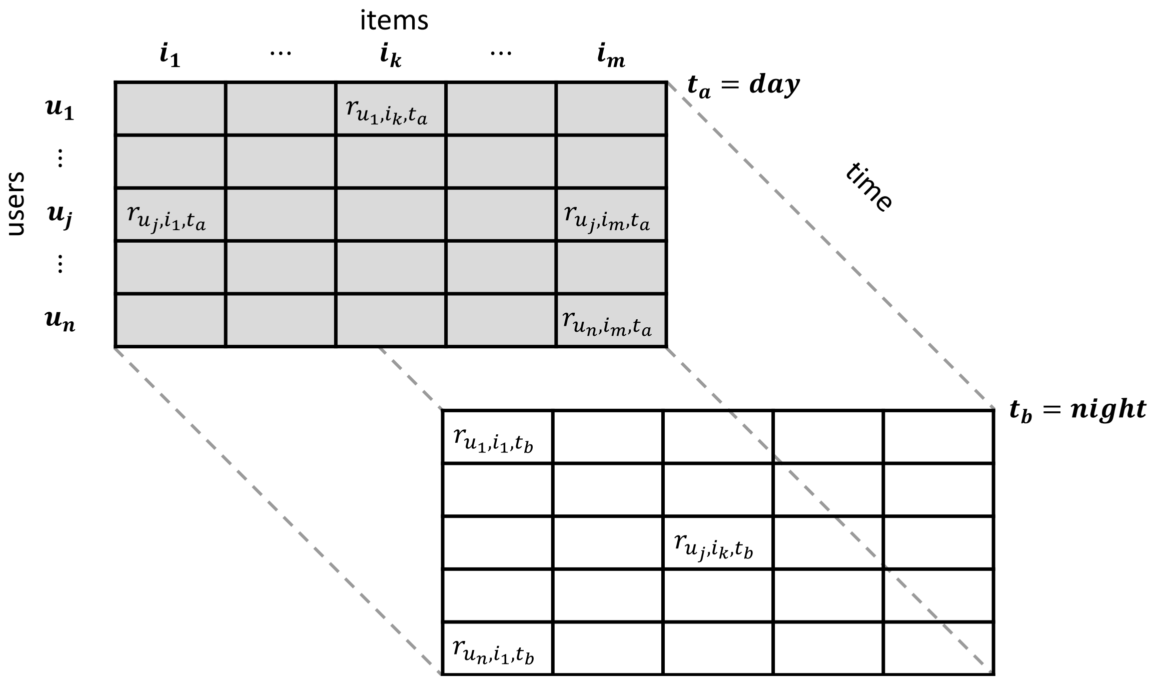


Figure 3.5. Selection of ratings in contextual pre-filtering.

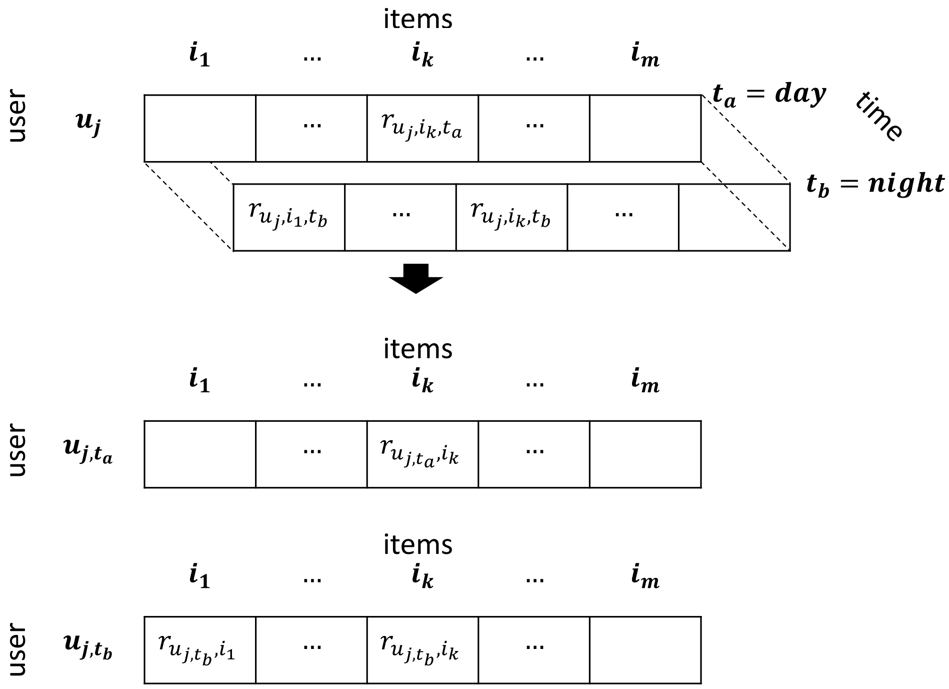


Figure 3.6. Schematic view of contextual micro-profiles.

In contextual post-filtering, z_t is used to adapt rating prediction values previously computed with the original rating matrix M and a model like (2.1):

$$F(u,i,t) = F(u,i) \cdot z_t(t,u,i)$$

An example of post-filtering using categorical time context variables is given in (Panniello et al., 2009a). The time context information used is presented in Figure 3.1. In that work, rating predictions $\hat{r}_{u,i}$ are computed by means of a heuristic as 2.2. After that, the rating predictions are contextualized based on the contextual probability $P_t(u,i,t)$ that user u chooses a certain type of item i in context t as follows:

$$\hat{r}_{u,i,t} = \begin{cases} \hat{r}_{u,i} & \text{if} \quad P_t(u,i,t) \geq P^* \\ 0 & \text{if} \quad P_t(u,i,t) < P^* \end{cases}$$

where $P_t(u,i,t)$ is computed as the number of u's neighbors who purchased i in context t divided by the total number of neighbors, and P^* is a threshold value. In this way, $\hat{r}_{u,i}$ is interpreted as relevant for the target time t when $P_t(u,i,t) \geq P^*$. This is equivalent to set $z_t(t,u,i) = 1$ when $P_t(u,i,t) \geq P^*$, and $z_t(t,u,i,) = 0$ otherwise.

It is important to note that if rating timestamps are available, multiple time context attributes can be exploited. For instance, Lee et al. (2010) derived the time variables *season* = {*fall*, *winter*, *spring*, *summer*}, *dayOfWeek* = {*sun*, *mon*, *tue*, *wed*, *thu*, *fri*, *sat*}, and *timeOfDay* = {*midnight*, *dawn*, *morning*, *AM*, *noon*, *PM*, *evening*, *night*}, and used all these attributes together for recommendation computation.

Given the flexibility of the categorical context representation, it is easy to incorporate other contextual dimensions beyond time, and use more complex representations of time context. For instance, Palmisano et al. (2008) used a hierarchical structure for the contextual variable *intent of purchase* at a food distributor. The hierarchy presents three levels: a first, more general level that considers the intents *personal* and *gift*, a second, specific level for the intent gift that considers the values {*event*, *no_event*}, and a third level for the intent (gift, event) that takes the values {*christmas*, *easter*}. More examples combining different discrete contextual dimensions (including time) can be found in (Adomavicius et al., 2005; Gorgoglione and Panniello, 2009; Panniello et al., 2013).

Model-based categorical TARS learn models from user preference data that include discrete time context attributes. One of the first approaches on model-based categorical TARS is presented in (Oku et al., 2006), where several contextual dimensions including time, social companion, and weather are incorporated into a Support Vector Machine model (Vapnik, 1995) for restaurant recommendation.

Karatzoglou et al. (2010) used TF to model n-dimensional contextual information. They called this approach *multiverse* recommendation because of its ability to bridge data pertaining to different contexts (universes of information) into a unified model. In this approach, the rating information is represented as an n-dimensional tensor $\mathfrak{M} \in \mathbb{R}^{|U|\times|I|\times|C_1|\times|C_2|\times\cdots\times|C_{n_c}|}$, where $C_1, C_2, \dots, C_{n_c}$ represent contextual dimensions of information. By applying the High Order SVD decomposition approach (Lathauwer et al., 2000), $\mathfrak{M}$ is factorized into factor matrices $P \in \mathbb{R}^{d_P \times |U|}$, $Q \in \mathbb{R}^{d_Q \times |I|}$, $A_k \in \mathbb{R}^{d_{C_k} \times |C_k|}$, and a central tensor $\mathfrak{S} \in \mathbb{R}^{d_P \times d_Q \times d_{C_1} \times \cdots \times d_{C_{n_C}}}$, in which d_k denotes the number of latent factors describing each dimension k. Figure 3.7 shows a schematic view of a 3-dimensional High Order SVD TF. In the figure, $\mathfrak{M}$ is composed of users, items, and one contextual dimension (C_1). The result of the factorization are a matrix of user factors (P), a matrix of item factors (Q), a matrix of context C_1 factors (A_1) and the central tensor $\mathfrak{S}$.

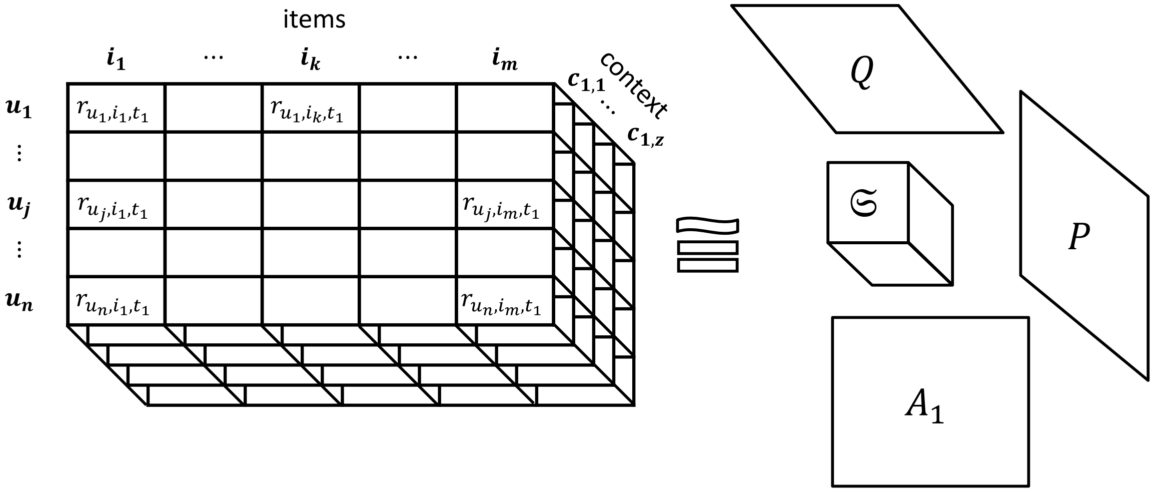


Figure 3.7. Schematic view of (3-dimensional) High Order SVD tensor factorization.

Once obtained the factor matrices, the rating prediction formula becomes a function of the target user, item, and context:

$$F(u, i, c_1, c_2, \cdots, c_{n_c}) = \mathfrak{S} \times_P P_{\cdot,u} \times_Q Q_{\cdot,i} \times_{A_1} A_{1\cdot,c_1} \times_{A_2} A_{2\cdot,c_2} \times \cdots \times_{A_{n_c}} A_{n_C\cdot,c_{n_C}}$$

where $\times_D$ denotes a tensor-matrix multiplication operator, and the subscript shows the direction on the tensor on which to multiply. Another example of categorical time-aware model is given in (Rendle et al., 2011), where Factorization Machines (FMs) were used to combine continuous and categorical time information.

3.3.3 Time-adaptive recommender systems

In this type of TARS, the rating prediction does not depend on the target recommendation time. In general, time-adaptive RS exploit time information from past user preferences in order to adjust parameters or data according to changes of some data characteristics through time. This is an important difference with respect to continuous and categorical TARS approaches, as the rating prediction is not targeted for a particular time context.

Heuristic-based time-adaptive RS generally penalize older preferences that are presumed to be not/less valid at recommendation time, and usually utilize a continuous time representation. Thus, they could be considered as a particular case of time decay heuristics, but, as noted before, they do not target a specific recommendation time. An example of time adaptive heuristics can be found in (Lee et al., 2008, 2009), where implicit purchase information is transformed into explicit ratings by assigning increasing weights to more recent ratings. This is modeled as a special time-dependent weight function $w'_t\big(\mathcal{T}(r)\big)$ that assigns a *weight* to each rating r according to its timestamp:

$$F(u,i) = \underset{v \in N(u)}{\text{aggr}}\, r_{v,i} \cdot w'_t\left(\mathcal{T}\left(r_{v,i}\right)\right)$$

Note that, differently to the time-dependent weight $w_t(\cdot)$ discussed in heuristic-based continuous TARS, the function $w'_t\big(\mathcal{T}(r)\big)$ only depends on the rating time, but not on the target recommendation time. These heuristics can also use, for instance, the time an item is incorporated into the system's catalog for the weight computation. We remark again that, under this formulation, the rating prediction is a function of the users and items, but not of the target (recommendation) time. A particular formulation of $w'_t\big(\mathcal{T}(r)\big)$ is given in (Ding et al., 2006), where item ratings are weighted according to their deviation from the target user's latest ratings on similar items. The underlying assumption is that the user's latest ratings on a neighborhood of similar items show her current trend on such items.

Following the idea of detecting user interest drift, Min and Han (2005) and Cao et al. (2009) developed approaches that derive time series of user ratings, aiming to establish current user interests. In order to build the time series ratings, items are grouped according to a certain heuristic, e.g. by category, as done in (Min and Han, 2005) –leading to several time series for each user, one per category– or grouping all ratings using an interest measure that takes into account item similarity, as done in (Cao et al., 2009) –leading to a unique time series for each user. Figure 3.8 shows examples of time series generated with those methods for the same user. The left side of the figure shows two time series of the user, each of them corresponding to items in two different categories (Min and Han, 2005). In this case, an interest drift is observed when any of the curves shows a trend shift. The right side of the figure shows the time series corresponding to the user's all rated items (Cao et al., 2009). In this case, an interest drift is observed when the curve shows a peak – we note that this latter method includes the items' similarity into the interest computation, and thus, a peak shows an increase in interest on certain group of similar items, followed by a decrease in interest on those items, or in the similarity of items.

An additional form of time adaptive heuristics is described in (Lathia et al., 2009a), where the number k of neighbors to be used in a kNN approach is dynamically adjusted, looking for values of k that diminish the error on previous predictions. Other approaches performing time adaptive heuristics are (Zimdars et al., 2001), where Web logs are coded as time series, and (Tang et al., 2003), where the production year of movies is used to reduce dimensionality in a CF system by means of an "old" movie pruning strategy.

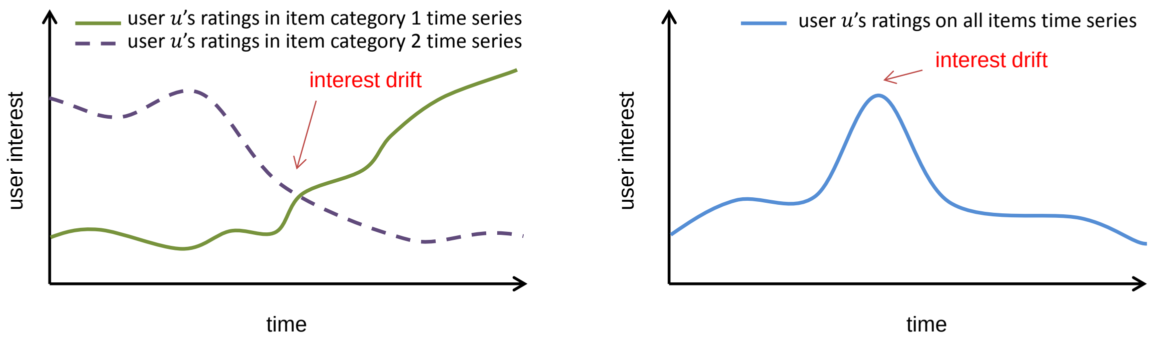


Figure 3.8. Example of rating time series for the same users, from two alternative methods. In the left part of the figure, Min and Han's method (Min and Han, 2005) generates several time series, one per rated items' category . In the right part of the figure, Cao's method (Cao et al., 2009) generates a unique time series from all the user's rated items.

In model-based time-adaptive RS, rating estimations are improved by means of exploiting temporal ordering of ratings rather than temporal closeness and relevance with respect to the target recommendation time. An example of time adaptive model is described in (Karatzoglou, 2011), where a temporal order of ratings is incorporated into a MF model, by means of learning differentiated item factors according to the rating timestamps, thus extending Eq. (2.4) to:

$$F(u,i) = \sum_{j=1}^{d} P_{j,u} Q_{j,i}^{s}\, Q_{j,a}^{s-1} Q_{j,b}^{s-2} \cdots Q_{j,N}^{s-N}$$

where $Q_{\cdot,i}^{s}$ is the item factors vector learned for item i with user preference information recorded until time s, and $a, b, \cdots, N$ denote the items consumed at times $s-1$, $s-2$ and so on. This model is referred to as a multiplicative model. The authors also proposed a summative model in which factor products $\langle p_u, q_i^s \rangle, \langle p_u, q_a^{s-1} \rangle, \langle p_u, q_b^{s-2} \rangle, \cdots, \langle p_u, q_N^{s-N} \rangle$ are summed up to compute the rating prediction $F(u,i)$.

One more example of time adaptive model can be found in (Jahrer et al., 2010), where several time-unaware models are learned. In order to blend such models, training data are split into several bins according to rating times (among other variables), and different weights are assigned to each time bin. In this way the blending process becomes time-dependent.

3.3.4 Overview of time-aware recommendation approaches

There has been a considerable amount of research on TARS, as described in Sections 3.3.1, 3.3.2 and 3.3.3. As discussed before, most TARS can be categorized as 1) *continuous*, *categorical*, or *time-adaptive*, according to the treatment given to time information, and as 2) *heuristic-based* or *model-based*, according to the type of recommendation techniques used for rating estimation. Table 3.1 groups representative TARS approaches using these two orthogonal dimensions, including example references.

Table 3.1. Overview of time-aware recommender systems in terms of algorithmic approaches and time treatment.

		Algorithmic approach	
		Heuristic-based	**Model-based**
Time treatment	**Continuous TARS**	• Time decay (Ding and Li, 2005) • Time window (Gordea and Zanker, 2007) • Temporal similarity (Hermann, 2010)	• Matrix Factorization with temporal dynamics (Koren, 2009a) • MF with temporal dynamics and session factors (Koenigstein et al., 2011) • Tensor Factorization (Xiong et al., 2010)
	Categorical TARS	• kNN-based Pre-Filtering (Adomavicius et al., 2005) • kNN-based Post-Filtering (Panniello et al., 2009b) • Micro-profiles (Baltrunas and Amatriain, 2009)	• Support Vector Machines (Oku et al., 2006) • Tensor Factorization (Karatzoglou et al., 2010)
	Time-Adaptive RS	• Time-based CF with implicit feedback (Lee et al., 2008, 2009) • Recency-based CF (Ding et al., 2006) • Time series of ratings (Zimdars et al., 2001; Min and Han, 2005; Cao et al., 2009) • Time-based pruning (Tang et al., 2003) • Adaptive Neighbors (Lathia et al., 2009a)	• Temporal Order Modeling (Karatzoglou, 2011) • Time-dependent blending (Jahrer et al., 2010)

We note that the most flexible TARS category corresponds to categorical TARS, since their modeling scheme lets include other type of categorical context dimensions, such as location and social companion; in fact, most TARS in this category are actually CARS that incorporate some time context information. However, its main drawback is the difficulty to model changes in user preferences through time; in general, it only models periodicity of preferences. This disadvantage is addressed by some model-based TARS that combine continuous and categorical time context variables. In particular, the matrix factorization with temporal dynamics model by Koren (2009) can handle factors associated with categorical context data, thus enabling exploit both time representations (categorical time variables can be derived from continuous time information). Other factorization-based methods such as Tensor Factorization can also handle both representations. However, more research is required in order to find the best modeling approach that properly integrates continuous and categorical time context information.

The methods presented in the table have shown a superior performance when compared against time-unaware baselines. However, little work has been done in comparing different TARS proposals, to determine which ones outperform the others, and under which circumstances. Moreover, there is a great diversity in the evaluation protocols used for TARS, which difficult their comparison. In the next section, we review the most common of such evaluation protocols.

3.4 Time-aware recommendation evaluation

As discussed in the previous section, there are multiple approaches to time-aware recommendation. Moreover, there are several methodologies and metrics that have been used to evaluate such approaches. In this section, we present some representatives examples of those methodologies, and discuss evaluation issues arising from them.

3.4.1 Time-aware evaluation methodologies

Diverse methodologies for the evaluation of RS have been developed, and TARS evaluation has not been the exception. Based on the generic stages of offline evaluation introduced in Chapter 2, we may observe that the training-test splitting process is the most influential step because it defines the (training) data that will be used for building a recommender system, and the main (test) data that will be used for performing recommendation evaluations. Due to these facts, in the following we focus our discussion on the training-test splitting process.

One of the most widely used methodologies for TARS evaluation is the one utilized in the Netflix Prize competition (Bennett and Lanning, 2007). It has been used in publications related with the competition, as well as other publications using the Netflix Prize dataset. In this methodology, ratings from each user are sorted according to their timestamps. Then, a fixed number n_f of ratings from each user are assigned to the test set, and the remaining ratings are assigned to the training set (Bennett and Lanning, 2007). Figure 3.9 shows a schematic view of this training-test splitting process. The figure shows the items rated by each user (represented as triangles) sorted by rating time, indicating which ratings are assigned to the test set. In the figure, $n_f = 3$, and the shadowed ratings represent the ones assigned to the test set.

We observe that this methodology ensures that all users have an equal number of test ratings[12]. However, we also observe that the timestamps of test ratings from some users may be lower (i.e., earlier) than the timestamps of training ratings from other users. In case

[12] In strict sense, the described methodology ensures that *most* users will have an equal number of test ratings. In those cases where a user has few ratings, assigning n_f ratings to the test set may leave the user with no or few training ratings. In such a case, an alternative condition may be defined, e.g. assigning only the half of the user's ratings to the test set.

of CF-based TARS, this may imply that some ratings whose values would not be known in a "real-world" setting (one that respects time order of the whole set of ratings) can be used to predict some other ratings.

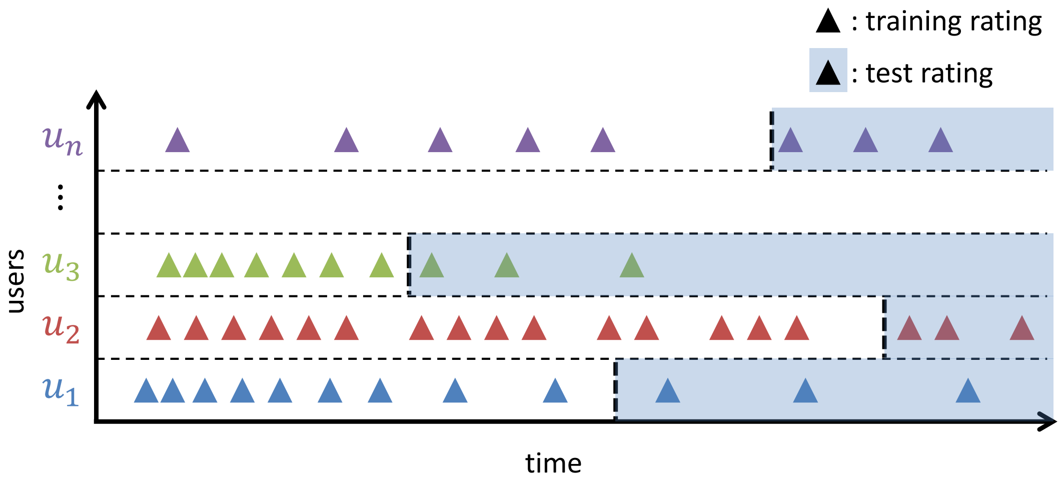


Figure 3.9. Schematic view of training-test splitting of ratings performed in the evaluation methodology used in the Netflix Prize competition, among many others.

An alternative methodology used in (Lathia et al., 2009a) and other works attempts to mimic how the training-test splitting would be in real-world operation of RS, that is, using a strict time-based splitting. In this case, a particular date/time is selected as splitting point, and all ratings prior to that time are used as training data, while ratings after that time are used as test data. This is equivalent to start the evaluation of a deployed RS in the defined splitting date/time. All data stored by the RS prior to that time can be used to compute predictions, but none after that time. Figure 3.10 shows a schematic view of this training-test splitting process. In the figure the dotted vertical line represents the splitting date/time, and the shadowed ratings represent the ones assigned to the test set.

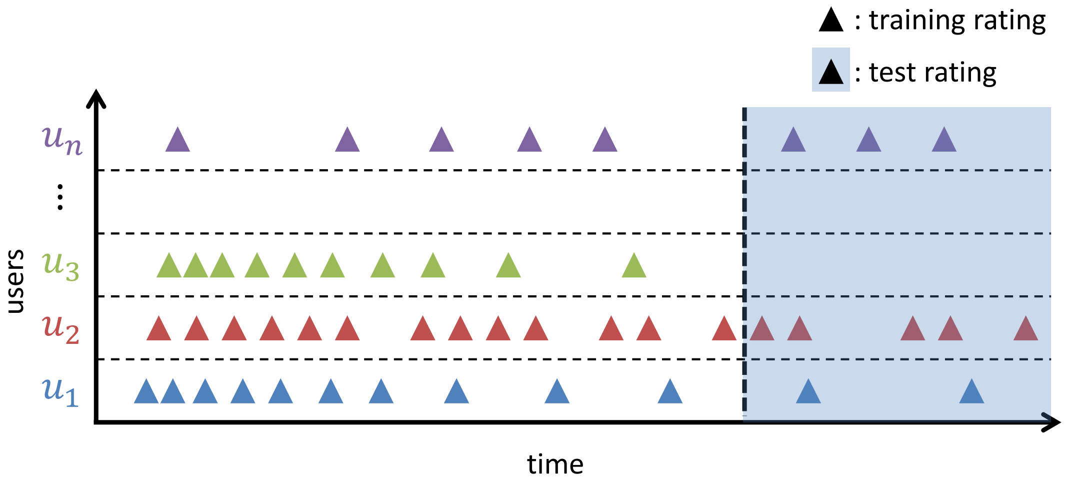


Figure 3.10 Schematic view of a training-test splitting of ratings that mimic real-world operation of RS.

We observe that in this case, there are a varying number of test ratings for the users. Moreover, some users may have all their ratings assigned to the test set, if they rated all items consumed after that time; or conversely, all their ratings may be assigned to the training set if the users provided all their ratings before that time. Despite this, we note that this scenario seems more realistic than the one used in the Netflix Prize competition, from a temporal point of view.

A third methodology that has been used in several works corresponds to a random (time-independent) training-test splitting. In this case, a random selection of ratings is assigned to the test set, and the remaining ratings are assigned to the training set. Figure 3.11 shows a schematic view of a random training-test splitting. In the figure, the shadowed ratings represent the ones assigned to the test set.

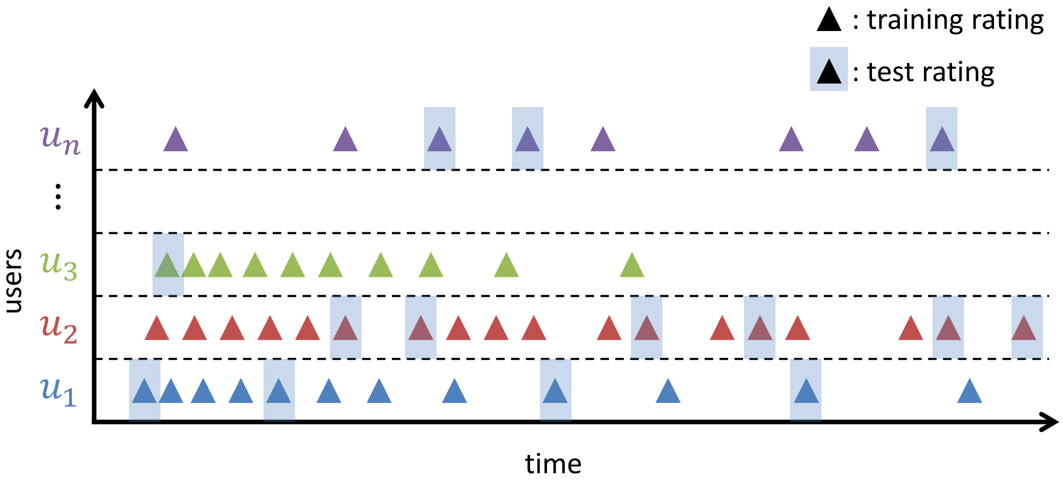


Figure 3.11. Schematic view of a random training-test splitting of ratings.

We observe that in this case there are no restrictions on which ratings can be used for training a RS. Hence, although this methodology does not take the time of ratings into consideration, it has been used for evaluating some TARS, mainly those exploiting time in a categorical representation.

3.4.2 Time-aware evaluation metrics

There are a few metrics in RS literature that explicitly consider time in their formulations. In general, recommendation results from TARS are assessed by means of traditional metrics such as RMSE and Precision, varying the evaluation methodology followed. Despite this, in the following we review some proposed performance metrics for time-aware recommendation.

Lathia et al. (2009b) propose a time-aware accuracy metric based on RMSE, which they called *time-averaged RMSE* ($RMSE_{TA}$). This metric is computed as the RMSE on ratings made until a particular time t:

$$RMSE_{TA} = \sqrt{\frac{\sum_{r_{u,i} \in Te_t} \left| \bar{r}_{u,i} - r_{u,i} \right|}{|Te_t|}}$$

where Te_t is the set of ratings in Te made until time t, i.e., $Te_t = \{r_{u,i,t'} \colon r_{u,i,t'} \in Te, t' \leq t\}$. This metric is intended to be applied iteratively during a long period of time, in order to observe the evolution of rating prediction accuracy through time.

Alternatively, Lathia et al. (2010) address the problem of measuring diversity and novelty of recommendations through time. They use set theoretic differences in order to assess such metrics. In the case of time-aware diversity, they compare the differences between consecutive recommendation lists presented to users, aiming to measure (and avoid) the repetition of recommendations:

$$diversity@N\left(I_{topN_{u,t_1}}, I_{topN_{u,t_2}}\right) = \frac{\left| I_{topN_{u,t_1}} \setminus I_{topN_{u,t_2}} \right|}{N}$$

where $I_{topN_{u,t}}$ is the set op top-N items recommended at time t. As noted by the authors, one limitation of this metric is that it measures the diversity between two lists, highlighting the extent to which users are sequentially offered the same recommendations, but does not provide take into account how recommendations change in terms of new items. They also propose a novelty metric that compares a recommendation list to the set of all items that have been recommended until time t, $I^*_{recomm_{u,t}}$:

$$novelty@N\left(I_{top-N_{u,t}}\right) = \frac{\left| I_{top-N_{u,t}} \setminus I^*_{recomm_{u,t}} \right|}{N}$$

These metrics represent alternatives to traditional metrics in order to measure recommendation properties in a time-aware manner. However, they have been rarely used in recommendation evaluation, probably due to the difficulty for delivering a unique resume value –they provide different values in different points of time. Most of the revised TARS have been evaluated by means of traditional error metrics such as MAE and RMSE for measuring rating prediction, and ranking accuracy metrics such as Precision and Recall for assessing the top-N recommendations task. In this context, it is important to note that the majority of TARS research has been focused on the rating prediction task.

3.4.3 Open problems in time-aware recommendation evaluation

We observe from the literature review that TARS evaluation presents important differences in the methodologies followed for assessing recommendation quality properties. The availability of rating timestamps can be considered as the source of major differences in the evaluation of TARS compared to other types of RS. In particular, the ability to order ratings according to timestamps before training-test data splitting lets define this splitting in various ways, as discussed in Section 3.4.1. The possibilities range from maintaining a random data split–as mostly done in time-unaware RS evaluation– to a strict time-aware split. In the latter, a test rating $r_{Te} \in Te$ has a timestamp posterior to any training rating $r_{Tr} \in Tr$, i.e., a time-dependent order of rating data is used: $\forall r_{Te}, r_{Tr}, \mathcal{T}(r_{Te}) > \mathcal{T}(r_{Tr})$ (see Figure 3.10). We note that this case is the most similar to a real-world setting, where a RS may only use past recorded data in order to estimate future user preferences.

The methodologies used for TARS evaluation make use of several intermediate approaches that highly differ from one work to another, and we hypothesize that existing methodological differences may have a significant effect on the assessment of TARS performance. For instance, in many methodologies, time-dependent order of data is not used to perform the training-test splitting, which may represent an unfair setting for TARS with respect to time-unaware methods unable to exploit time information.

As a matter of fact, some studies have shown divergences on the ground assumption in which recommendation models are built, casting doubt on the generalization of time-aware recommendation capabilities. The results from (Ding and Li, 2005) show recommendation improvements when applying a time decay weight, while experiments on the Netflix Prize dataset (Koren, 2009a) indicate that better rating prediction is achieved when no time weight is applied. In experiments testing several time-dependent rating data partitioning for creating contextual micro-profiles, Baltrunas and Amatriain (2009) found that the scarce {*even hours*, *odd hours*} partitioning provides higher recommendation improvements than other partitions such as {*morning*, *evening*} and {*workday*, *weekend*}. In words of the authors, the hours partition correspond to a “meaningless” partition, and calls for further research. Additionally, Lathia et al. (2009a) found that improvements obtained by some non-contextualized algorithms on the Netflix Prize dataset do not hold when computing predictions on an iterative basis by strictly using past ratings to predict future ratings –the actual setting for a real-world recommender systems.

Despite the fact that a number of reasons could be enumerated for explaining such contradictory findings (e.g. different application domains, item characteristics and contextualization schemas for time information), we believe evaluation plays a prominent role. The existence of multiple evaluation methodologies, each of them with distinct assumptions and purposes, makes it easy to find an evaluation protocol suitable for a particular algorithmic approach, but ineligible or non-retributive for others. Problems that arise from this situation thus represent an increasing impediment to fairly compare results and conclusions reported in different studies, and make the selection of the best TARS solution for a given task more difficult.

3.5 Summary

Exploiting the context in which users express their preferences has been proven very valuable for increasing the performance of recommendations. Among existing contextual dimensions, time information can be considered as one of the most useful ones. Moreover, time context information is in general easy to collect without additional user efforts and strict device requirements. Due to these benefits, recent years have been prolific in the investigation and development of time-aware recommender systems. In this chapter, we have revised and classified state-of-the-art literature on TARS.

Despite the benefits of time-aware recommendation, some studies have shown important divergences regarding the results achieved by different approaches. We hypothesize that methodological differences plays a prominent role in explaining these divergences. In the next Part of this thesis, we present the research conducted to get a deeper understanding of the impact of existing differences in TARS evaluation, by means of the development of a methodological framework that lets define and analyze the key conditions that drive TARS evaluation methodologies.

Part II

Characterizing a robust time-aware recommendation evaluation protocol

Chapter 4

A methodological framework for time-aware recommendation evaluation

A wide range of approaches dealing with time context information in user modeling and recommendation strategies has been proposed. In the literature, however, reported results and conclusions about how to incorporate and exploit time information within the recommendation process are contradictory in some cases. The existence of multiple evaluation methodologies seems to have a key role in explaining such opposing outcomes. Moreover, the lack of standardization in evaluation of TARS represents an impediment to fairly compare results from different studies.

In this chapter we propose a descriptive methodological framework aimed to characterize the TARS evaluation process, and make it fair and reproducible under different circumstances. The framework is based on a set of key evaluation conditions defined from the analysis of the TARS literature. These conditions address a number of general methodological issues to be faced in the experimental design of an offline evaluation of TARS. Moreover, the formalism of the framework includes the definition of a splitting procedure that lets precisely build and replicate data splits for evaluation. In Section 4.1 we briefly analyze methodological differences among TARS evaluation protocols, and pose methodological questions and related evaluation conditions, namely *data splitting conditions* –which are related to the training-test data splitting process–, *cross-validation conditions*, and *top-N recommendations conditions* – which are specific for the top-N recommendations task. In Section 4.2 we introduce the main concepts of the proposed framework. In Section 4.3 we detail the data splitting conditions, which are related to the rating ordering of training and test sets, the size of these sets, and the base data user for building the splits. In Section 4.4 we present cross-validation conditions, which include time-independent and time-dependent conditions. In Section 4.5 we describe the conditions that are specific for top-N recommendations evaluation, namely the formation of the set of items to be ranked, and the identification of items relevant to the user. Finally, in Section 4.6 we end the chapter with partial conclusions regarding the evaluation conditions defined.

4.1 Evaluation conditions for time-aware recommender systems

A review of proposed TARS and protocols followed to evaluate such systems shows important methodological differences in the assessment of recommendation results across studies. Although the diversity in evaluation protocols is not a problem per se, it makes the comparison of TARS –and consequently the selection of the appropriate recommendation approach for a particular application or domain– difficult. In order to deal with this situation, a formal description of decisions on existing alternatives in the evaluation of TARS is needed. In this section we identify the main decisions to be made from the analysis of divergences in methodologies described in Chapter 3 (Section 3.4.1), and state a set of *methodological questions* to be addressed in the design of a TARS evaluation protocol.

An important source of methodological variations is the training-test rating splitting process, particularly when rating timestamps are available, as is the case in most TARS studies. As a matter of fact, in the literature one can find diverse implementations of the *hold-out* method (Duda et al., 2001), which has been widely used for rating data splitting (Gunawardana and Shani, 2009). A first decision to be made when designing an evaluation setting is whether the rating splits should be based on some *rating ordering* criterion. For instance, we may use a time-dependent ordering in which all ratings are first ordered according to their timestamps. Next, those ratings prior to a particular date are selected for training, whereas the remaining ratings are selected for test, as done in (Panniello et al., 2009a). We may, on the other hand, use a random selection of training and test ratings, without considering the ratings' timestamps or any ordering criterion, as done in (Stormer, 2007). Between these two extreme cases, there are intermediate options. For example, ordering the ratings by timestamp separately for each user, and assigning the most recent ratings of each one to the test set, as done in the Netflix Prize competition (Bennett and Lanning, 2007).

A careful analysis of differences among the reviewed evaluation protocols shows that the ordering –and the overall splitting process– can also have different *base rating sets*. That is, the splits can be created on the base of the whole rating matrix, as done in (Karatzoglou, 2011), or can be created independently over each user's ratings, as done in (Koenigstein et al., 2011).

The *number of ratings* selected for the test set is also chosen differently among TARS-related papers. An example is the typical proportion-based schema (e.g. 80% of the ratings for the training set, and the remaining 20% for the test set) used e.g. in (Lee et al., 2010). Other strategies, in contrast, select a fixed number of ratings per user, as done in (Ding et al., 2006), or set a threshold date to select ratings before (after) that date for the training (test) set, as done in (Lathia et al., 2009a), to name a few.

Additionally, there are *cross-validation* strategies (Stone, 1974) aimed to increase the generalization of the evaluation results on independent data. A popular strategy is to use some variant of the resampling method, such as *X-fold cross-validation*, but other strategies have also been used in TARS evaluation. These techniques let average results over several test sets extracted from M, by means of repeatedly splitting the data (Gunawardana and Shani, 2009).

Another important source of methodological differences is related to specific requirements for assessing particular recommendation tasks. For instance, in order to evaluate the top-N recommendations task, we have to establish a set of *target items* a recommender has to rank. As described in (Bellogín et al., 2011), several approaches have been used to generate the set of target items. For example, ranking only those items for which the user's relevance can be determined (Adomavicius et al., 2005); or mixing items considered relevant for the user (e.g. highly rated items) with other items considered as non-relevant (e.g. unrated items), as done in (Cremonesi et al., 2010).

Moreover, for the top-N recommendations task, the concept of item *relevance* –an estimation of the user's interest in an item– has been interpreted differently. One interpretation is that all items in the target user's test set are relevant. It has been argued, however, that some items in the user's test set should be treated as non-relevant; consider for example a one-time-played song, or a movie rated with 1 in a 1-5 rating scale. Furthermore, as noted by Parra and Amatriain (2011), in some scenarios, such information may be treated as evidence of the user's lack of interest in an item. For instance, the authors argue that one can assume that a user did not like a TV series she watched only once.

The above discussion expresses that there are several potential sources of divergence in protocols used for evaluating TARS. These potential sources of differences lead to a set of methodological questions regarding the design of a TARS evaluation protocol:

- MQ1: What *base rating set* is used to perform the training-test splitting?
- MQ2: What *rating ordering* is used to assign ratings to the training and test sets?
- MQ3: How many *ratings* comprise the training and test sets?
- MQ4: What *cross-validation method* is used for increasing the generalization of evaluation results?

In the case of top-N recommendations task evaluation, there are a set of additional methodological questions that must be answered:

- MQ5: Which items are considered as *target items*?
- MQ6: Which items are considered *relevant* for each user?

In the subsequent sections we describe the possible ways to address each of the above methodological questions by means of a number of *evaluation conditions*, which we have defined from the revision of evaluation protocols used in the TARS literature. These evaluation conditions are *base rating set conditions*, addressing MQ1; *rating ordering conditions*, addressing MQ2; *rating set size conditions*, addressing MQ3; *cross-validation conditions*, addressing MQ4; *target item conditions*, addressing MQ5; and *relevant item conditions*, addressing MQ6. We describe and formalize these conditions in the context of a generic descriptive methodological framework, aiming to facilitate a precise communication of the evaluation conditions that drive an evaluation process.

4.2 A methodological description framework for TARS evaluation conditions

In the following we define the methodological framework aimed to address the methodological questions stated in the previous section. This framework is constituted by a set of evaluation conditions, and a procedure to make the evaluation process fair and reproducible under different circumstances. We begin by giving some general definitions of concepts used in the framework, and next we provide detailed descriptions of each evaluation condition involved in the framework.

Definition 1 A ***split*** Σ of the rating matrix M is a partition of M into a training set Tr and a test set Te, that is, $\Sigma = \langle Tr, Te \rangle \mid Tr \cap Te = \emptyset$.

Definition 2 A ***splitting procedure*** is an algorithm that takes as input the 4-tuple $\langle M \times \mathscr{b} \times \mathscr{o} \times \mathscr{s} \rangle$, where $\mathscr{b} \in \mathfrak{B}$ is a condition in the set $\mathfrak{B}$ of conditions to define a base rating set, $\mathscr{o} \in \mathcal{O}$ is a condition in the set $\mathcal{O}$ of conditions to define an ordered set of ratings, and $\mathscr{s} \in \mathcal{S}$ is a condition in the set $\mathcal{S}$ of conditions to define the size of the training and test sets of a split. The output of a splitting procedure is a split Σ.

Definition 3 A ***base rating set condition*** $\mathscr{b} \in \mathcal{B}$ specifies the set of base datasets $M^{\mathscr{b}} = \{M_1, M_2, \cdots, M_m\}$ generated from M, being $M = \bigcup_k M_k\,, M_k \subseteq M$.

Definition 4 A ***rating ordering condition*** $\mathscr{o} \in \mathcal{O}$ establishes an ordered sequence for a set of ratings. The sequence Seq_k defined by the ordering condition $\mathscr{o}$ over the ratings in M_k is:

$$Seq_k = \left\{r_{(1)}, r_{(2)}, \cdots, r_{(|M_k|)}\right\} \mid r_{(j)} \in M_k$$

where $r_{(j)}$ denotes the j-th rating in the sequence.

Definition 5 A ***rating set size condition*** $\mathscr{s} \in \mathcal{S}$ sets a criterion for computing the number of ratings from Seq_k that will be assigned to the training and test sets Tr and Te, denoted by $\mathscr{s}^{Tr}(Seq_k)$ and $\mathscr{s}^{Te}(Seq_k)$.

Algorithm 4.1 describes the steps of a splitting procedure. First, the base rating sets are built (step 1). Then, according to the rating ordering condition $\mathscr{o}$, a sequence of ratings is generated from each base dataset (step 2). After that, according to the rating set size condition $\mathscr{s}$ (and a set of parameter values depending on the value of $\mathscr{s}$), the number of training and test ratings is established (step 3). Taking these sizes into account, the first ratings in each sequence are assigned to the training set, and the following ratings are assigned to the test set (step 4).

```
Splitting Procedure($M$, $\mathscr{b}$, $\mathscr{o}$, $\mathscr{s}$)
Input: Rating matrix $M$, base rating set condition $\mathscr{b}$, rating
    ordering condition $\mathscr{o}$, rating set size condition $\mathscr{s}$
Output: Split $\Sigma = \langle Tr, Te \rangle$
Step 1: According to $\mathscr{b}$, build the base  rating sets $M^{\mathscr{b}} = \{M_1, M_2, \cdots, M_m\}$.
Step 2: According to $\mathscr{o}$, generate the sequences $Seq_k$ from the rating sets $M_k$.
Step 3: According to $\mathscr{s}$, compute the sizes $\mathscr{s}^{Tr}(Seq_k)$ and $\mathscr{s}^{Te}(Seq_k)$.
Step 4: For each ordered sequence $Seq_k$:
    4.1: Find the rating $r_{(p_k)}$ such that the subsequence
         $\{r_{(1)}, r_{(2)}, \cdots, r_{(p_k)}\}, r_{(j)} \in Seq_k$ contains $\mathscr{s}^{Tr}(Seq_k)$ ratings.
    4.2: Assign the first $p_k$ ratings to the training set $Tr$,
         and the remaining ratings to the test set $Te$,
         forming the split $\Sigma = \langle Tr, Te \rangle$.
```

Algorithm 4.1. Splitting procedure to generate training and test rating sets.

According to the generic stages of offline evaluation protocols, described in Chapter 2 (Section 2.4.2), in order to perform the assessment of recommendations, we also need to process the test set data, build (train) the recommender, and perform the recommendation process. The training set obtained from the splitting procedure is used as input data for building the RS. In case of assessing a top-N recommendations task, the test set is further processed for obtaining a set of target items to rank, and selecting the set of relevant items for each user. Then, recommendations are computed to obtain rating predictions for items in the test set, or generate a ranking of items in case of top-N recommendations. When cross-validation (CV) methods are utilized, the above stages are repeated according to the CV condition used.

In the next sections we provide specific formulations for these evaluation conditions. We group conditions in $\mathfrak{B}$, $\mathcal{O}$ and $\mathcal{S}$ as *data splitting* conditions, and specific conditions for ranking items as *top-N recommendations* conditions.

4.3 Data splitting conditions

Data splitting conditions are involved in the training-test splitting process, namely the base rating set ($\mathfrak{B}$), the rating ordering ($\mathcal{O}$), and the rating set size ($\mathcal{S}$) conditions, which are detailed in the following.

4.3.1 Base rating set conditions

The base rating set conditions state whether the rating ordering and rating set size conditions in steps 2 and 3 of the splitting procedure (described in Algorithm 4.1) are applied on the whole set of ratings in M, or independently in different rating sets $M_k \subseteq M$. Specifically, we consider two conditions, namely the community-centered ($\mathscr{b}_{cc}$) and the user-centered ($\mathscr{b}_{uc}$) base rating set conditions.

Community-centered base rating set condition. A single base rating set with all the ratings in M is used:

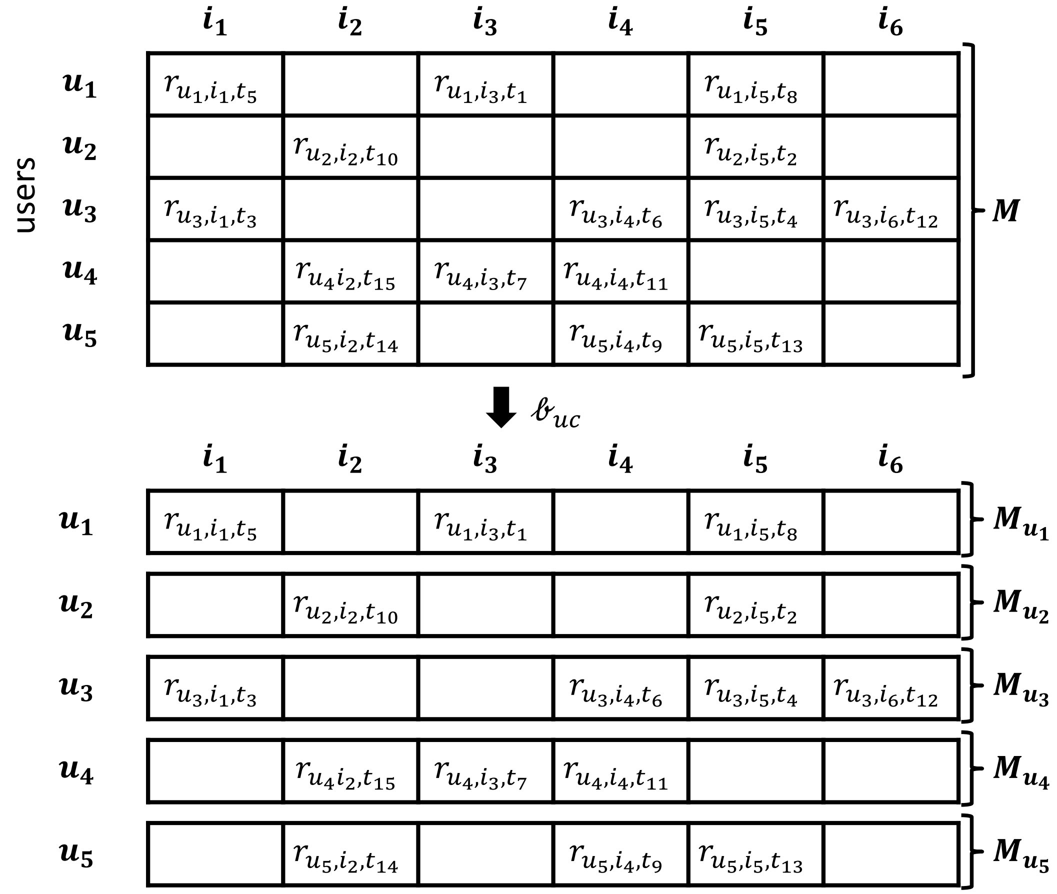
$$M^{\mathscr{b}_{cc}} = M$$

As a result of the application of rating ordering and rating set size conditions on the full set of ratings when $\mathscr{b}_{cc}$ condition is applied, some users may have all or none of their ratings in the test set. This makes no possible to assess the RS performance for such users, as in general CF strategies cannot generate recommendations for users without profiles (i.e., without training ratings), and metrics cannot be computed without ground truth data (i.e., test ratings). This problem is due to large differences on rating patterns between users, existing some users with many more ratings than others, and/or different rating distributions across time. A solution for this is to split each user's ratings separately.

User-centered base rating set condition. A base rating set M_u is built with the ratings of each user u:

$$M^{\mathscr{b}_{uc}} = \{M_u \mid u \in U\}, M_u = \left\{r_{u,\cdot} \mid r_{u,\cdot} \in M\right\}$$

Figure 4.1 shows a schematic view of base datasets generated by a user-centered base rating set conditions. The rating matrix M in the figure is a quite simple one, with only five users and six items, in order to facilitate the visualization.



	i_1	i_2	i_3	i_4	i_5	i_6
u_1	r_{u_1,i_1,t_5}		r_{u_1,i_3,t_1}		r_{u_1,i_5,t_8}	
u_2		$r_{u_2,i_2,t_{10}}$			r_{u_2,i_5,t_2}	
u_3	r_{u_3,i_1,t_3}			r_{u_3,i_4,t_6}	r_{u_3,i_5,t_4}	$r_{u_3,i_6,t_{12}}$
u_4		$r_{u_4i_2,t_{15}}$	r_{u_4,i_3,t_7}	$r_{u_4,i_4,t_{11}}$		
u_5		$r_{u_5,i_2,t_{14}}$		r_{u_5,i_4,t_9}	$r_{u_5,i_5,t_{13}}$	

	i_1	i_2	i_3	i_4	i_5	i_6	
u_1	r_{u_1,i_1,t_5}		r_{u_1,i_3,t_1}		r_{u_1,i_5,t_8}		M_{u_1}
u_2		$r_{u_2,i_2,t_{10}}$			r_{u_2,i_5,t_2}		M_{u_2}
u_3	r_{u_3,i_1,t_3}			r_{u_3,i_4,t_6}	r_{u_3,i_5,t_4}	$r_{u_3,i_6,t_{12}}$	M_{u_3}
u_4		$r_{u_4i_2,t_{15}}$	r_{u_4,i_3,t_7}	$r_{u_4,i_4,t_{11}}$			M_{u_4}
u_5		$r_{u_5,i_2,t_{14}}$		r_{u_5,i_4,t_9}	$r_{u_5,i_5,t_{13}}$		M_{u_5}

Figure 4.1. Schematic view of the application of a user-centered base condition.

In the figure the ratings in M from each user u_j form an independent rating set, M_{u_j}, that will serve for the application of other data splitting conditions. By performing the splitting independently on each user's ratings, we can ensure that all users will have ratings in both the training and test sets[13].

The above defined base rating set conditions correspond to the most common settings used in TARS evaluation. Other possible conditions, such as item-centered, are less practical since recommendation performance is usually assessed for each user. After the selection of one base rating set condition, a rating ordering condition has to be chosen, as detailed in the next subsection.

4.3.2 Rating ordering conditions

The rating ordering conditions establish the type of ordering to apply in the generation of the rating sequence(s) used to make the training-test set splitting. We define two rating

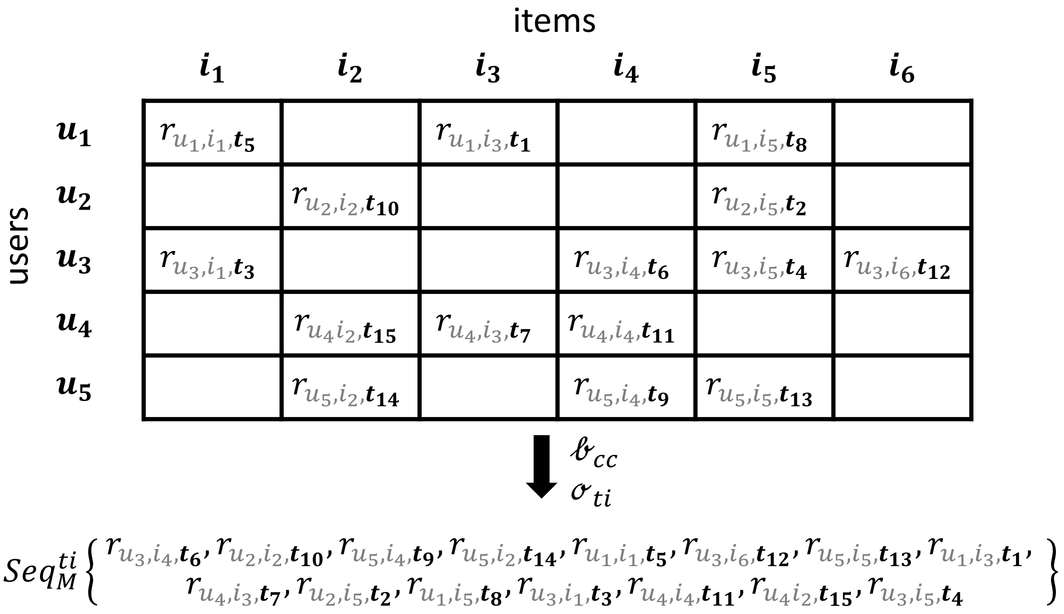
[13] In a strict sense, a user-centered split ensures that *most* users will have training and test data, but there may be some users without enough ratings for both training and test sets. This will depend not only on the number of ratings of each user, but also on the definition of other evaluation conditions like the *size* condition.

ordering conditions related with the evaluation settings found in the TARS literature review: a time-independent (i.e., random) ordering (σ_{ti}), and a time-dependent ordering (σ_{td}).

Time-independent rating ordering condition. The timestamps associated to ratings are not considered for ordering the latter in the training and test datasets. Other ordering criteria may be used, but in general, the sequence[14] they generate consists of a random selection of ratings from the base rating set M_k:

$$M_k \xrightarrow{\sigma_{ti}} Seq_k^{ti}$$

Figure 4.2 shows an example rating sequence (lower side) built from a community-centered base rating set and a time-independent rating ordering condition. In the upper side of the figure, the base rating set –which is equivalent to the rating matrix M in this case– remarks the timestamps of the ratings. Only one sequence is formed, and timestamps are not considered in the sequence's order (random sequence).



	i_1	i_2	i_3	i_4	i_5	i_6
u_1	r_{u_1,i_1,t_5}		r_{u_1,i_3,t_1}		r_{u_1,i_5,t_8}	
u_2		$r_{u_2,i_2,t_{10}}$			r_{u_2,i_5,t_2}	
u_3	r_{u_3,i_1,t_3}			r_{u_3,i_4,t_6}	r_{u_3,i_5,t_4}	$r_{u_3,i_6,t_{12}}$
u_4		$r_{u_4 i_2,t_{15}}$	r_{u_4,i_3,t_7}	$r_{u_4,i_4,t_{11}}$		
u_5		$r_{u_5,i_2,t_{14}}$		r_{u_5,i_4,t_9}	$r_{u_5,i_5,t_{13}}$	

$$Seq_M^{ti}\left\{\begin{matrix} r_{u_3,i_4,t_6}, r_{u_2,i_2,t_{10}}, r_{u_5,i_4,t_9}, r_{u_5,i_2,t_{14}}, r_{u_1,i_1,t_5}, r_{u_3,i_6,t_{12}}, r_{u_5,i_5,t_{13}}, r_{u_1,i_3,t_1}, \\ r_{u_4,i_3,t_7}, r_{u_2,i_5,t_2}, r_{u_1,i_5,t_8}, r_{u_3,i_1,t_3}, r_{u_4,i_4,t_{11}}, r_{u_4 i_2,t_{15}}, r_{u_3,i_5,t_4} \end{matrix}\right\}$$

Figure 4.2. Example of rating sequence built from a community-centered base rating set using a time-independent rating ordering condition.

The main advantage of this rating ordering condition is its applicability, since it does not require timestamp information. Its main drawback, from a contextual point of view, is that time dependencies between training and test ratings do not hold. This means that the timestamp of some training ratings could be more recent than the timestamp of test ratings, as shown in Figure 3.11. That is, some ratings included in the training set may have been

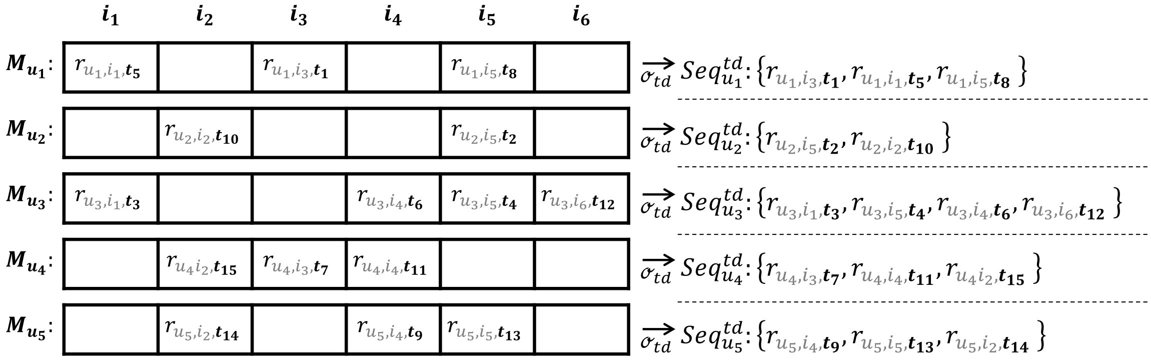
[14] Note that each user's rating sequence Seq_u^{ti} is generated independently in case of using a user-centered base rating set condition.

produced after some ratings in the test set. This situation can be interpreted as an evaluation of TARS that have knowledge about “future” preferences of the users, which is far away from a real-world setting, and may give TARS unfair advantages in an offline evaluation (Campos et al., 2011b). Using a time-dependent order can help avoid this problem.

Time-dependent rating ordering condition. The rating sequence is ordered according to the rating timestamps by means of a time-dependent rating ordering σ_{td}:

$$M_k \xrightarrow{\sigma_{td}} Seq_k^{td}$$

being Seq_k^{td} ordered by increasing rating timestamp, i.e., $\mathcal{T}\left(r_{(j)}\right) \leq \mathcal{T}\left(r_{(j+1)}\right), \forall r_{(j)} \in Seq_k^{td}$ [15]. Figure 4.3 shows different rating sequences built from user-centered base rating sets, using a time-dependent rating ordering condition. The left part of the figure shows the base rating sets with the timestamps of their ratings. Note that base rating sets in Figures 4.2 and 4.3 are built from the same rating matrix M showed in Figure 4.1, but using different base rating set conditions. In Figure 4.3, one rating sequence is built from each base rating set –that is, for each user’s ratings a sequence is built–, and each sequence is strictly ordered according to the ratings’ timestamps, as shown in the right side of the figure.



	$\boldsymbol{i_1}$	$\boldsymbol{i_2}$	$\boldsymbol{i_3}$	$\boldsymbol{i_4}$	$\boldsymbol{i_5}$	$\boldsymbol{i_6}$	
$\boldsymbol{M_{u_1}}$:	$r_{u_1,i_1,\boldsymbol{t_5}}$		$r_{u_1,i_3,\boldsymbol{t_1}}$		$r_{u_1,i_5,\boldsymbol{t_8}}$		$\overrightarrow{\sigma_{td}}\ Seq_{u_1}^{td}$: $\{r_{u_1,i_3,\boldsymbol{t_1}}, r_{u_1,i_1,\boldsymbol{t_5}}, r_{u_1,i_5,\boldsymbol{t_8}}\}$
$\boldsymbol{M_{u_2}}$:		$r_{u_2,i_2,\boldsymbol{t_{10}}}$			$r_{u_2,i_5,\boldsymbol{t_2}}$		$\overrightarrow{\sigma_{td}}\ Seq_{u_2}^{td}$: $\{r_{u_2,i_5,\boldsymbol{t_2}}, r_{u_2,i_2,\boldsymbol{t_{10}}}\}$
$\boldsymbol{M_{u_3}}$:	$r_{u_3,i_1,\boldsymbol{t_3}}$			$r_{u_3,i_4,\boldsymbol{t_6}}$	$r_{u_3,i_5,\boldsymbol{t_4}}$	$r_{u_3,i_6,\boldsymbol{t_{12}}}$	$\overrightarrow{\sigma_{td}}\ Seq_{u_3}^{td}$: $\{r_{u_3,i_1,\boldsymbol{t_3}}, r_{u_3,i_5,\boldsymbol{t_4}}, r_{u_3,i_4,\boldsymbol{t_6}}, r_{u_3,i_6,\boldsymbol{t_{12}}}\}$
$\boldsymbol{M_{u_4}}$:		$r_{u_4 i_2,\boldsymbol{t_{15}}}$	$r_{u_4,i_3,\boldsymbol{t_7}}$	$r_{u_4,i_4,\boldsymbol{t_{11}}}$			$\overrightarrow{\sigma_{td}}\ Seq_{u_4}^{td}$: $\{r_{u_4,i_3,\boldsymbol{t_7}}, r_{u_4,i_4,\boldsymbol{t_{11}}}, r_{u_4 i_2,\boldsymbol{t_{15}}}\}$
$\boldsymbol{M_{u_5}}$:		$r_{u_5,i_2,\boldsymbol{t_{14}}}$		$r_{u_5,i_4,\boldsymbol{t_9}}$	$r_{u_5,i_5,\boldsymbol{t_{13}}}$		$\overrightarrow{\sigma_{td}}\ Seq_{u_5}^{td}$: $\{r_{u_5,i_4,\boldsymbol{t_9}}, r_{u_5,i_5,\boldsymbol{t_{13}}}, r_{u_5,i_2,\boldsymbol{t_{14}}}\}$

Figure 4.3. Example of rating sequences built from user-centered base rating sets, using a time-dependent rating ordering condition.

This rating ordering condition aims to maintain time dependencies between ratings, and is only applicable when ratings have timestamp information. Thus, when the desired evaluation setting is aimed to mimic real-world conditions (as illustrated in Figure 3.10), a time-dependent ordering may be preferred.

It is important to note that a strict time-dependent ordering between training and test ratings can be generated only by using a community-centered base rating set condition. In

[15] In case of ties, they could be broken by sorting the tied ratings by user id. If still there are tied ratings, then they could be sorted by item id (note that in real datasets it is possible to find users with several ratings with the same timestamp due to e.g. inconsistencies in the log subsystem).

such a case, all the test ratings have timestamps more recent than the timestamp of any training rating, that is, $\forall r_{Tr} \in Tr, r_{Te} \in Te, \mathcal{T}(r_{Tr}) < \mathcal{T}(r_{Te})$. This combination generates an evaluation setting similar to a real-world setting, where a deployed RS can only be trained with data available up to a particular moment, and its effectiveness is usually evaluated with user feedback provided afterwards. Examples of this approach are presented in (Ardissono et al., 2004) and (Panniello et al., 2009a). A drawback of this combination – $\mathscr{b}_{cc}$ and $\mathscr{o}_{td}$– is that, due to different user rating distributions through time, there may be many users without ratings either in the training or in the test set.

When a time-dependent rating ordering condition is used with a user-centered base rating set –$\mathscr{b}_{uc}$ and $\mathscr{o}_{td}$–, each user's ratings are time-sorted *independently from other users' ratings* (see Figure 4.3). This means that time-dependent ordering of ratings is maintained for each user independently, and thus cross-user time dependencies are not maintained; some users may have training ratings subsequent to test ratings of other users. On the other hand, by using this combination TARS do not have access to future knowledge of the target user (in contrast to using $\mathscr{o}_{ti}$), and the problem of leaving many users with only training or test ratings is avoided (in contrast to using $\mathscr{b}_{cc}$).

The combination of user-centered and time-dependent ordering conditions has been one of the most used in TARS evaluation, probably because it was used for building the training and test sets of the Netflix Prize competition (Bennett and Lanning, 2007). It has been argued, however, that TARS that make recommendations for a target user by exploiting knowledge about other users' "future" preferences (with respect to the target recommendation time) may have unfair advantages in an evaluation (Campos et al., 2011b).

Once the rating ordering condition has been established, only the size of the training and test rating sets remains to be decided in order to perform the training-test splitting procedure. The conditions defining these sizes are described in the next subsection.

4.3.3 Rating set size conditions

The rating set size evaluation conditions establish how many ratings from each rating sequence Seq_k are included in the training and test sets, Tr and Te. In the literature, it is common to establish and report the size of the test set, and there are several ways to set that size; some of them can be used in combination with any other evaluation conditions, while others can only be used with a particular combination of conditions, as we shall explain below. The covered conditions include proportion-based size, fixed size, and date-based size. Note that, as described in Algorithm 4.1, the rating ordering is considered when assigning ratings to training and test sets. In general, the first $\mathscr{s}^{Tr}(Seq_k)$ ratings of Seq_k are assigned to Tr, and the remaining $\mathscr{s}^{Te}(Seq_k)$ ratings are assigned to Te.

Proportion-based size. Denoted by $\mathscr{s}_{prop}$, this condition establishes that a proportion q_{prop} of the ratings in each Seq_k is used as test data, and the remaining ratings are used as training data, that is, $\mathscr{s}_{prop}^{Te}(Seq_k) = q_{prop} \cdot |Seq_k|$ and $\mathscr{s}_{prop}^{Tr}(Seq_k) = \left(1 - q_{prop}\right) \cdot |Seq_k|$, with $q_{prop} \in [0,1]$[16].

When a user-centered base rating set condition ($\mathscr{b}_{uc}$) is used –i.e., a rating sequence is built for each user– a proportion-based rating set size ensures that different users have a similar proportion of their ratings in the training and test sets. An evaluation setting defined with this combination on conditions is used in (Zheng and Li, 2011).

Fixed size. Denoted by $\mathscr{s}_{fix}$, this condition establishes that a fixed number q_{fix} of the ratings in each Seq_k are used as test data, that is, $\mathscr{s}_{fix}^{Te}(Seq_k) = q_{fix}$ and $\mathscr{s}_{fix}^{Tr}(Seq_k) = |Seq_k| - q_{fix}$.

When using $\mathscr{b}_{uc}$ a fixed rating set size condition ensures that the same number of ratings is assigned to the users' test sets, regardless the number of training ratings of each user. Figure 4.4 shows an example of the application of $\mathscr{s}_{fix}$. In the figure, the rating sequences are built using $\mathscr{b}_{uc}$ and $\mathscr{o}_{td}$, and $q_{fix} = 1$. The red shadowed ratings are assigned to Te, and the remaining (green shadowed) ratings are assigned to Tr.

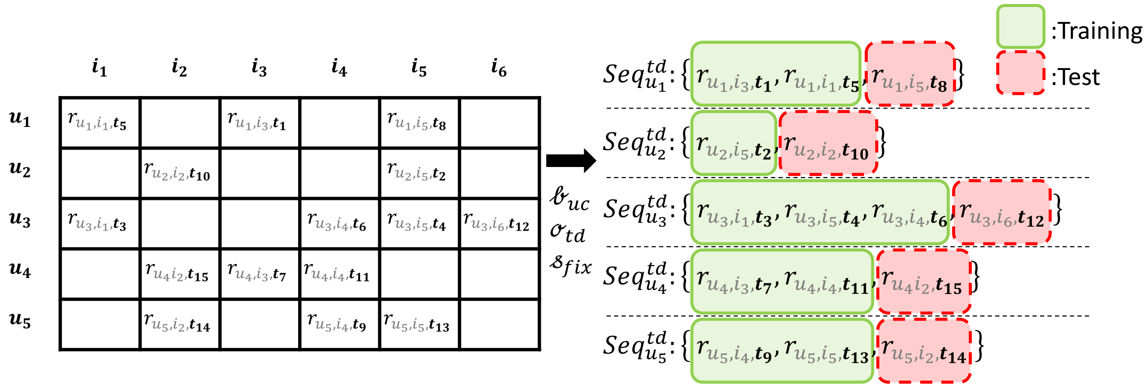

:Training
:Test
i_1 i_2 i_3 i_4 i_5 i_6
u_1: r_{u_1,i_1,t_5}, r_{u_1,i_3,t_1}, r_{u_1,i_5,t_8}
u_2: $r_{u_2,i_2,t_{10}}$, r_{u_2,i_5,t_2}
u_3: r_{u_3,i_1,t_3}, r_{u_3,i_4,t_6}, r_{u_3,i_5,t_4}, $r_{u_3,i_6,t_{12}}$
u_4: $r_{u_4,i_2,t_{15}}$, r_{u_4,i_3,t_7}, $r_{u_4,i_4,t_{11}}$
u_5: $r_{u_5,i_2,t_{14}}$, r_{u_5,i_4,t_9}, $r_{u_5,i_5,t_{13}}$
$\mathscr{b}_{uc}$
$\mathscr{o}_{td}$
$\mathscr{s}_{fix}$
$Seq_{u_1}^{td}: \{r_{u_1,i_3,t_1}, r_{u_1,i_1,t_5}, r_{u_1,i_5,t_8}\}$
$Seq_{u_2}^{td}: \{r_{u_2,i_5,t_2}, r_{u_2,i_2,t_{10}}\}$
$Seq_{u_3}^{td}: \{r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_3,i_4,t_6}, r_{u_3,i_6,t_{12}}\}$
$Seq_{u_4}^{td}: \{r_{u_4,i_3,t_7}, r_{u_4,i_4,t_{11}}, r_{u_4,i_2,t_{15}}\}$
$Seq_{u_5}^{td}: \{r_{u_5,i_4,t_9}, r_{u_5,i_5,t_{13}}, r_{u_5,i_2,t_{14}}\}$

Figure 4.4. Example of the application of a fixed size condition ($q_{fix} = 1$) on sequences of ratings built with a user-centered base rating set and a time-dependent rating ordering conditions.

Given that some users may have less than q_{fix} ratings, the rating set size condition can be changed for such users. For instance, in the Netflix Prize competition dataset, a fixed number of $q_{fix} = 9$ ratings of each user was selected for building the test sets, but in cases where $|Seq_u| < 18$ only half of a user's ratings were selected as test data (Bennett and Lanning, 2007). This can be interpreted as a mechanism switching from a fixed into a proportion-based rating set size condition with $q_{prop} = 0.5$.

[16] In case of a non-integer size value, it could be rounded to the nearest integer value.

If the fixed size condition refers to the number of training ratings, all the users have the same number of training ratings q_{fix}^{Tr}, leaving the remaining ratings for the test dataset (whose size would vary from user to user). This latter case has been referred to as *given N* (Ding and Li, 2005), where $N = q_{fix}^{Tr}$.

Time-based size. Denoted by $\mathscr{s}_{time}$, this rating set size condition can only be used with a time-dependent rating ordering condition. It establishes a threshold time q_{time} that is used to assign the ratings of each Seq_k into training and test sets. In this case the ratings with a timestamp after q_{time} are assigned to Te, and the ratings with a timestamp before q_{time} are assigned to Tr. Hence, given the last index p_k in Seq_k that satisfies $\mathcal{T}\left(r_{(p_k)}\right) \leq q_{time}$, the sizes $\mathscr{s}_{time}^{Tr}(Seq_k) = \left|\{r_{(1)}, r_{(2)}, \dots, r_{(p_k)}\}\right|$ and $\mathscr{s}_{time}^{Te}(Seq_k) = |Seq_k| - \mathscr{s}_{time}^{Tr}(Seq_k)$ are established, as done in (Lu et al., 2009).

Using the time-based size condition in combination with either a community-centered or a user-centered base rating set condition yields equivalent training and test sets if the threshold q_{time} is the same for all users. This particular case is similar to the combination of a community-centered base rating set, a time-dependent rating ordering, and a proportion-based rating set size conditions (with an appropriate q_{prop} value).

The time-based size condition can be enhanced by incorporating an ending time limit q_{end_time}. In this case, only the ratings whose timestamps are between q_{time} and q_{end_time} are assigned to the test set. Hence, given the last index l_k in Seq_k that satisfies $\mathcal{T}\left(r_{(l_k)}\right) \leq q_{end\_time}$, the sizes $\mathscr{s}_{time}^{Tr}(Seq_k) = p_k$ and $\mathscr{s}_{time}^{Te}(Seq_k) = l_k - p_k$ are assigned, and the ratings with timestamp subsequent to q_{end_time} are discarded, as done in (Liu et al., 2010b; Pradel et al., 2011). In this case the assumption is that the user's preferences (manifested as ratings) that were produced long after a target recommendation time should not be considered for assessing the quality of recommendations. Figure 4.5 shows training and test sets formed by applying a user-centered (right side of the figure) and community-centered (bottom side of the figure) base rating set conditions, time-dependent rating ordering and time-based rating set size conditions. In the figure $q_{time} = t_8$ and $q_{end\_time} = t_{13}$. The final training and test sets formed by using either $\mathscr{b}_{uc}$ or $\mathscr{b}_{cc}$ are equivalent. As shown in the figure, a disadvantage of using a $\mathscr{s}_{time}$ condition is that some users may have no ratings assigned to the training or the test set (u_1 and u_5 in the figure).

An alternative way to specify a time-based size is establishing a period of time (or window size) q_{time_window} to the timespan of the test ratings (e.g. $q_{time\_window} = 10$ days). Hence, the ratings assigned to the test set are those starting from the last known rating time in each Seq_k minus q_{time_window}. In this case, given the last index p_k in Seq_k that satisfies $t\left(r_{(p_k)}\right) \leq t\left(r_{(|Seq_k|)}\right) - q_{time\_window}$, the sizes $\mathscr{s}_{time\_window}^{Tr}(Seq_k) = p_k$ and $\mathscr{s}_{time\_window}^{Te}(Seq_k) = |Seq_k| - p_k$ are assigned, as done in (Zhan et al., 2006).

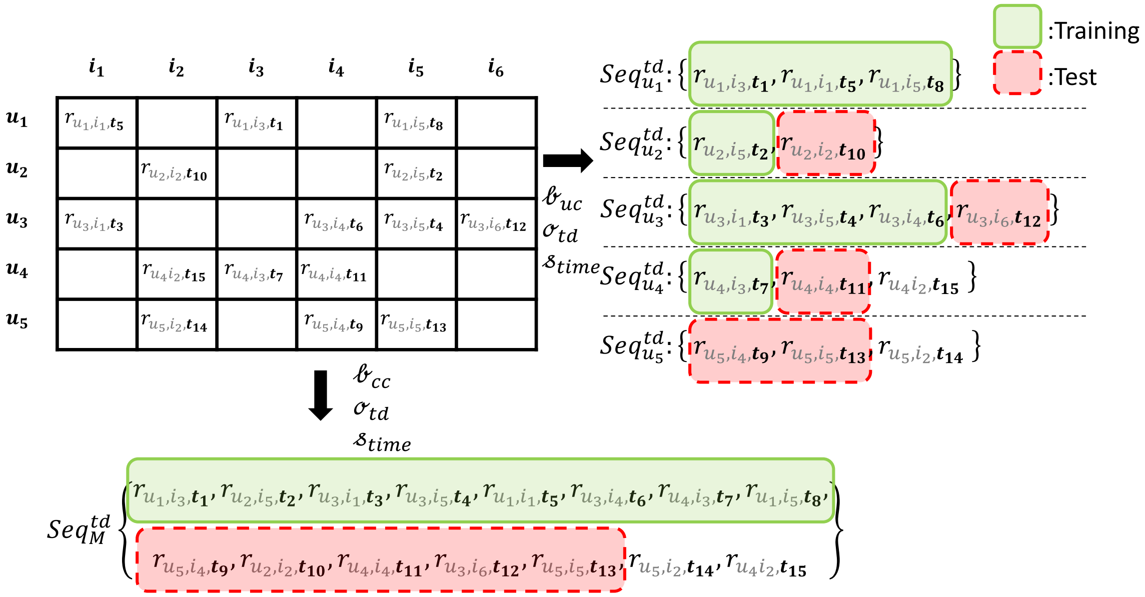

:Training
:Test
i_1 i_2 i_3 i_4 i_5 i_6
u_1 u_2 u_3 u_4 u_5
r_{u_1,i_1,t_5} r_{u_1,i_3,t_1} r_{u_1,i_5,t_8}
$r_{u_2,i_2,t_{10}}$ r_{u_2,i_5,t_2}
r_{u_3,i_1,t_3} r_{u_3,i_4,t_6} r_{u_3,i_5,t_4} $r_{u_3,i_6,t_{12}}$
$r_{u_4,i_2,t_{15}}$ r_{u_4,i_3,t_7} $r_{u_4,i_4,t_{11}}$
$r_{u_5,i_2,t_{14}}$ r_{u_5,i_4,t_9} $r_{u_5,i_5,t_{13}}$
$\mathcal{b}_{uc}$ $\mathcal{o}_{td}$ $\mathcal{s}_{time}$
$Seq^{td}_{u_1}: \{r_{u_1,i_3,t_1}, r_{u_1,i_1,t_5}, r_{u_1,i_5,t_8}\}$
$Seq^{td}_{u_2}: \{r_{u_2,i_5,t_2}, r_{u_2,i_2,t_{10}}\}$
$Seq^{td}_{u_3}: \{r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_3,i_4,t_6}, r_{u_3,i_6,t_{12}}\}$
$Seq^{td}_{u_4}: \{r_{u_4,i_3,t_7}, r_{u_4,i_4,t_{11}}, r_{u_4,i_2,t_{15}}\}$
$Seq^{td}_{u_5}: \{r_{u_5,i_4,t_9}, r_{u_5,i_5,t_{13}}, r_{u_5,i_2,t_{14}}\}$
$\mathcal{b}_{cc}$ $\mathcal{o}_{td}$ $\mathcal{s}_{time}$
Seq^{td}_M $\{r_{u_1,i_3,t_1}, r_{u_2,i_5,t_2}, r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_1,i_1,t_5}, r_{u_3,i_4,t_6}, r_{u_4,i_3,t_7}, r_{u_1,i_5,t_8}, r_{u_5,i_4,t_9}, r_{u_2,i_2,t_{10}}, r_{u_4,i_4,t_{11}}, r_{u_3,i_6,t_{12}}, r_{u_5,i_5,t_{13}}, r_{u_5,i_2,t_{14}}, r_{u_4,i_2,t_{15}}\}$

Figure 4.5. Example training and test sets formed by applying a time-based size condition ($q_{time} = t_8, q_{end\_time} = t_{13}$).

We note that when used in combination with a user-centered base rating set condition, $\mathcal{T}\left(r_{(|Seq_k|)}\right)$ is different for each user. In such a case, there would be a different starting date for the ratings in the test set of each user. Because of this, combining the $\mathcal{s}_{time\_window}$ and $\mathcal{b}_{uc}$ conditions has a disadvantage similar to that of the $\mathcal{s}_{fix}$ condition when it is used with user-centered base rating set and time-dependent rating ordering conditions, since some users may have all their ratings within the test timespan.

These conditions describe the ways in which training and test set sizes are defined in TARS evaluation. By using different combinations of the three types of conditions addressed in this section –base rating set, rating ordering, and rating set size conditions– it is possible to replicate most of the evaluation settings that have been used in TARS literature. In the following subsection we show examples of use of the conditions in order to reproduce some common data splits used in TARS evaluation.

4.3.4 Examples of use of data splitting conditions

In order to show the use of evaluation conditions under the splitting procedure (Algorithm 4.1), in the following we reproduce two commonly used data splits for TARS evaluation, namely, the one used in the Netflix Prize competition, and one avoiding temporal overlap of ratings.

The splitting procedure (Algorithm 4.1) requires as input the rating matrix M, and the base rating set, rating ordering and rating set size conditions. According to the Netflix Prize competition setting, the last n_f ratings of each user are assigned to the test set, and the remaining ratings are assigned to the training set. When n_f is larger than the half of the user profile size, the last half of user ratings are assigned to test. From this description, we determine that we must use a) a user-centered base rating set condition $\mathcal{b}_{uc}$ (because we need to independently select ratings from each user), b) a time-dependent rating ordering condition $\mathcal{o}_{td}$ (because we need to find the last ratings of each user), and c) a fixed rating set size condition $\mathcal{s}_{fix}$ with $q_{fix} = n_f$, and a proportion rating set size $\mathcal{s}_{prop}$ with $q_{prop} = 0.5$ in cases where n_f is larger than the half of the user profile size. To facilitate the visualization of the example, we use the small rating matrix showed in Figure 4.6, composed of four users and seven items, and set $n_f = 2$.

users \ items	$\boldsymbol{i_1}$	$\boldsymbol{i_2}$	$\boldsymbol{i_3}$	$\boldsymbol{i_4}$	$\boldsymbol{i_5}$	$\boldsymbol{i_6}$	$\boldsymbol{i_7}$
$\boldsymbol{u_1}$	$r_{u_1,i_1,\boldsymbol{t_5}}$		$r_{u_1,i_3,\boldsymbol{t_1}}$		$r_{u_1,i_5,\boldsymbol{t_8}}$		$r_{u_1,i_7,\boldsymbol{t_{14}}}$
$\boldsymbol{u_2}$		$r_{u_2,i_2,\boldsymbol{t_{10}}}$			$r_{u_2,i_5,\boldsymbol{t_2}}$		
$\boldsymbol{u_3}$	$r_{u_3,i_1,\boldsymbol{t_3}}$			$r_{u_3,i_4,\boldsymbol{t_6}}$	$r_{u_3,i_5,\boldsymbol{t_4}}$	$r_{u_3,i_6,\boldsymbol{t_{12}}}$	$r_{u_3,i_7,\boldsymbol{t_9}}$
$\boldsymbol{u_4}$		$r_{u_4i_2,\boldsymbol{t_{15}}}$	$r_{u_4,i_3,\boldsymbol{t_7}}$	$r_{u_4,i_4,\boldsymbol{t_{11}}}$			$r_{u_4,i_7,\boldsymbol{t_{13}}}$

Figure 4.6. Example rating matrix.

Following the steps of the splitting procedure (Algorithm 4.1), we first apply the base rating set condition $\mathcal{b}_{uc}$ to build the base rating sets. This lead to the rating sets $M^{\mathcal{b}_{uc}} = \{M_{u_1}, M_{u_2}, M_{u_3}, M_{u_4}\}$ with $M_{u_j} = \left\{r_{u_j,\cdot} \mid r_{u_j,\cdot} \in M\right\}$. That is:

$$
\begin{aligned}
M_{u_1} &= \{r_{u_1,i_1,t_5}, r_{u_1,i_3,t_1}, r_{u_1,i_5,t_8}, r_{u_1,i_7,t_{14}}\}\\
M_{u_2} &= \{r_{u_2,i_2,t_{10}}, r_{u_2,i_5,t_2}\}\\
M_{u_3} &= \{r_{u_3,i_1,t_3}, r_{u_3,i_4,t_6}, r_{u_3,i_5,t_4}, r_{u_3,i_6,t_{12}}, r_{u_3,i_7,t_9}\}\\
M_{u_4} &= \{r_{u_4,i_2,t_{15}}, r_{u_4,i_3,t_7}, r_{u_4,i_4,t_{11}}, r_{u_4,i_7,t_{13}}\}
\end{aligned}
$$

In the second step, we apply the rating ordering condition $\mathcal{o}_{td}$ to generate the sequences of ratings from each rating set. This lead to the following time ordered sequences $Seq_{u_1}^{td}, Seq_{u_2}^{td}, Seq_{u_3}^{td}$ and $Seq_{u_4}^{td}$:

$$
\begin{aligned}
Seq_{u_1}^{td} &= \{r_{u_1,i_3,\boldsymbol{t_1}}, r_{u_1,i_1,\boldsymbol{t_5}}, r_{u_1,i_5,\boldsymbol{t_8}}, r_{u_1,i_7,\boldsymbol{t_{14}}}\}\\
Seq_{u_2}^{td} &= \{r_{u_2,i_5,\boldsymbol{t_2}}, r_{u_2,i_2,\boldsymbol{t_{10}}}\}\\
Seq_{u_3}^{td} &= \{r_{u_3,i_1,\boldsymbol{t_3}}, r_{u_3,i_5,\boldsymbol{t_4}}, r_{u_3,i_4,\boldsymbol{t_6}}, r_{u_3,i_7,\boldsymbol{t_9}}, r_{u_3,i_6,\boldsymbol{t_{12}}}\}\\
Seq_{u_4}^{td} &= \{r_{u_4,i_3,\boldsymbol{t_7}}, r_{u_4,i_4,\boldsymbol{t_{11}}}, r_{u_4,i_7,\boldsymbol{t_{13}}}, r_{u_4,i_2,\boldsymbol{t_{15}}}\}
\end{aligned}
$$

In the third step, we apply the rating set size condition $\mathscr{s}_{fix}$ (or alternatively $\mathscr{s}_{prop}$ in cases where n_f is less than the half of the user profile size) to compute the sizes of the training and test sets from each sequence, $\mathscr{s}^{Tr}(Seq_k)$ and $\mathscr{s}^{Te}(Seq_k)$. That is:

$$
\begin{aligned}
&\mathscr{s}_{fix}^{Tr}\left(Seq_{u_1}^{td}\right) = \left|Seq_{u_1}^{td}\right| - q_{fix} = 4 - 2 = \mathbf{2}, \\
&\mathscr{s}_{fix}^{Te}\left(Seq_{u_1}^{td}\right) = q_{fix} = \mathbf{2} \\
&\mathscr{s}_{prop}^{Tr}\left(Seq_{u_2}^{td}\right) = \left(1 - q_{prop}\right) \cdot |Seq_k| = (1 - 0.5)\ \cdot 2 = \mathbf{1}, \\
&\mathscr{s}_{prop}^{Te}\left(Seq_{u_2}^{td}\right) = q_{prop} \cdot |Seq_k| = 0.5 \cdot 2 = \mathbf{1} \\
&\mathscr{s}_{fix}^{Tr}\left(Seq_{u_3}^{td}\right) = \left|Seq_{u_3}^{td}\right| - q_{fix} = 5 - 2 = \mathbf{3}, \\
&\mathscr{s}_{fix}^{Te}\left(Seq_{u_3}^{td}\right) = q_{fix} = \mathbf{2} \\
&\mathscr{s}_{fix}^{Tr}\left(Seq_{u_4}^{td}\right) = \left|Seq_{u_4}^{td}\right| - q_{fix} = 4 - 2 = \mathbf{2}, \\
&\mathscr{s}_{fix}^{Te}\left(Seq_{u_4}^{td}\right) = q_{fix} = \mathbf{2}
\end{aligned}
$$

In the fourth step, we look for the rating $r_{(p_k)}$ such that the subsequence $\left\{r_{(1)}, r_{(2)}, \cdots, r_{(p_k)}\right\}, r_{(j)} \in Seq_k$ contains $\mathscr{s}^{Tr}(Seq_k)$ ratings. We show in bold the rating $r_{(p_k)}$ for each Seq_k^{td}:

$$
\begin{aligned}
p_{u_1} &= 2\colon Seq_{u_1}^{td} = \left\{r_{u_1,i_3,t_1}, \boldsymbol{r_{u_1,i_1,t_5}}, r_{u_1,i_5,t_8}, r_{u_1,i_7,t_{14}}\right\} \\
p_{u_2} &= 1\colon Seq_{u_2}^{td} = \left\{\boldsymbol{r_{u_2,i_5,t_2}}, r_{u_2,i_2,t_{10}}\right\} \\
p_{u_3} &= 3\colon Seq_{u_3}^{td} = \left\{r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, \boldsymbol{r_{u_3,i_4,t_6}}, r_{u_3,i_7,t_9}, r_{u_3,i_6,t_{12}}\right\} \\
p_{u_4} &= 2\colon Seq_{u_4}^{td} = \left\{r_{u_4,i_3,t_7}, \boldsymbol{r_{u_4,i_4,t_{11}}}, r_{u_4,i_7,t_{13}}, r_{u_4,i_2,t_{15}}\right\}
\end{aligned}
$$

Finally, we assign the first p_k ratings in each Seq_k to the training set Tr, and the remaining ratings to the test set Te, forming the split $\Sigma = \langle Tr, Te\rangle$:

$$
\begin{aligned}
Tr &= \left\{r_{u_1,i_3,t_1}, r_{u_1,i_1,t_5}, r_{u_2,i_5,t_2}, r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_3,i_4,t_6}, r_{u_4,i_3,t_7}, r_{u_4,i_4,t_{11}}\right\} \\
Te &= \left\{r_{u_1,i_5,t_8}, r_{u_1,i_7,t_{14}}, r_{u_2,i_2,t_{10}}, r_{u_3,i_7,t_9}, r_{u_3,i_6,t_{12}}, r_{u_4,i_7,t_{13}}, r_{u_4,i_2,t_{15}}\right\}
\end{aligned}
$$

Let us suppose now the case of a setting avoiding temporal overlaps. In this case, we require all ratings time-sorted, assigning the last ratings to the test set. From this description, we determine that we must use a) a community-centered base rating set condition $\mathscr{b}_{cc}$ (because we need to time-sort the full set of ratings), b) a time-dependent rating ordering condition $\mathscr{o}_{td}$ (because we need to find the last ratings), and c) a proportion rating set size condition $\mathscr{s}_{prop}$ with $q_{prop} = 0.2$. We could use another rating set size condition, but this one lets easily select a proportion of the full set of ratings. We use the example rating matrix showed in Figure 4.6.

Following the steps of the splitting procedure (Algorithm 4.1), we first apply the base rating set condition $\mathscr{b}_{cc}$ to build the base rating set. This leads to the rating set $M^{\mathscr{b}_{cc}} = \{r | r \in M\}$. That is:

$$M^{b_{cc}} = \left\{ \begin{matrix} r_{u_1,i_1,t_5}, r_{u_1,i_3,t_1}, r_{u_1,i_5,t_8}, r_{u_1,i_7,t_{14}}, r_{u_2,i_2,t_{10}}, r_{u_2,i_5,t_2}, r_{u_3,i_1,t_3}, r_{u_3,i_4,t_6}, \\ r_{u_3,i_5,t_4}, r_{u_3,i_6,t_{12}}, r_{u_3,i_7,t_9}, r_{u_4,i_2,t_{15}}, r_{u_4,i_3,t_7}, r_{u_4,i_4,t_{11}}, r_{u_4,i_7,t_{13}} \end{matrix} \right\}$$

In the second step, we apply the rating ordering condition σ_{td} to generate the ordered sequence of ratings from the rating set. This lead to the sequence:

$$Seq_M^{td} = \left\{ \begin{matrix} r_{u_1,i_3,\boldsymbol{t_1}}, r_{u_2,i_5,\boldsymbol{t_2}}, r_{u_3,i_1,\boldsymbol{t_3}}, r_{u_3,i_5,\boldsymbol{t_4}}, r_{u_1,i_1,\boldsymbol{t_5}}, r_{u_3,i_4,\boldsymbol{t_6}}, r_{u_4,i_3,\boldsymbol{t_7}}, r_{u_1,i_5,\boldsymbol{t_8}}, \\ r_{u_3,i_7,\boldsymbol{t_9}}, r_{u_2,i_2,\boldsymbol{t_{10}}}, r_{u_4,i_4,\boldsymbol{t_{11}}}, r_{u_3,i_6,\boldsymbol{t_{12}}}, r_{u_4,i_7,\boldsymbol{t_{13}}}, r_{u_1,i_7,\boldsymbol{t_{14}}}, r_{u_4,i_2,\boldsymbol{t_{15}}} \end{matrix} \right\}$$

In the third step, we apply the rating set size condition $\mathscr{s}_{prop}$ to compute the sizes of the training and test sets from the sequence, $\mathscr{s}_{prop}^{\mathrm{Tr}}(Seq_k)$ and $\mathscr{s}_{prop}^{\mathrm{Te}}(Seq_k)$. That is:

$$\begin{aligned} &\mathscr{s}_{prop}^{Tr}\left(Seq_M^{td}\right) = \left(1 - q_{prop}\right) \cdot |Seq_k| = (1 - 0.2) \cdot 15 = \mathbf{12}, \\ &\mathscr{s}_{prop}^{Te}\left(Seq_M^{td}\right) = q_{prop} \cdot |Seq_k| = 0.2 \cdot 15 = \mathbf{3} \end{aligned}$$

In the fourth step, we look for the rating $r_{(p_M)}$ such that the subsequence $\{r_{(1)}, r_{(2)}, \cdots, r_{(p_M)}\}, r_{(j)} \in Seq_M^{td}$ contains $\mathscr{s}^{Tr}\left(Seq_M^{td}\right)$ ratings. We show in bold the rating $r_{(p_M)}$:

$$p_M = 12:$$

$$Seq_M^{td} = \left\{ \begin{matrix} r_{u_1,i_3,t_1}, r_{u_2,i_5,t_2}, r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_1,i_1,t_5}, r_{u_3,i_4,t_6}, r_{u_4,i_3,t_7}, r_{u_1,i_5,t_8}, \\ r_{u_3,i_7,t_9}, r_{u_2,i_2,t_{10}}, r_{u_4,i_4,t_{11}}, \boldsymbol{r_{u_3,i_6,t_{12}}}, r_{u_4,i_7,t_{13}}, r_{u_1,i_7,t_{14}}, r_{u_4,i_2,t_{15}} \end{matrix} \right\}$$

Finally, we assign the first p_M ratings to the training set Tr, and the remaining ratings to the test set Te, forming the split $\Sigma = \langle Tr, Te \rangle$:

$$\begin{aligned} Tr &= \left\{ \begin{matrix} r_{u_1,i_3,t_1}, r_{u_2,i_5,t_2}, r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_1,i_1,t_5}, r_{u_3,i_4,t_6}, \\ r_{u_4,i_3,t_7}, r_{u_1,i_5,t_8}, r_{u_3,i_7,t_9}, r_{u_2,i_2,t_{10}}, r_{u_4,i_4,t_{11}}, r_{u_3,i_6,t_{12}} \end{matrix} \right\} \\ Te &= \left\{ r_{u_4,i_7,t_{13}}, r_{u_1,i_7,t_{14}}, r_{u_4,i_2,t_{15}} \right\} \end{aligned}$$

These examples show how different data splits for evaluation of TARS can be defined by using the evaluation conditions and splitting procedure included in the proposed framework. We note that, additionally, some works have performed cross-validation methods in the evaluation process. We review the related conditions in the following section.

4.4 Cross-validation conditions

Cross-validation conditions state whether one or more data splits (i.e., pairs of training-test sets) are built with the ratings in M. Research in Statistics (Arlot and Celisse, 2010) and Machine Learning (Dietterich, 1998) has shown that the variability of evaluation results is diminished by repeating the evaluation process several times by using a different data split each time. This procedure is commonly referred to as *cross-validation*. In this section we describe two general cross-validation conditions, namely time-independent and time-dependent cross-validation, which can be approached by diverse methods that have been used in the revised TARS literature. First of all, we introduce the hold-out procedure, as it is the basic building block for the above methods, and can deal with time dimension in different ways according to other evaluation conditions explained in previous subsections.

Hold-out splitting. One training set and one test set are built according to the evaluation conditions *base rating set*, *rating ordering*, and *rating set size*, and avoiding pairwise (user, item) rating overlap, i.e., $Tr \cap Te = \emptyset$. The performance of TARS is measured by training the recommendation algorithm with ratings in Tr and comparing generated recommendations with the ground truth Te, as done e.g. in (Panniello et al., 2009a). The rationale for having separated, non-overlapping training and test sets is that measuring performance of rating predictions on training data may produce an underestimated prediction error (Arlot and Celisse, 2010).

4.4.1 Time-independent cross-validation condition

The cross-validation methods that satisfy this condition make use of a time-independent rating ordering condition, and build X different splits $\Sigma_x = (Tr_x, Te_x), x \in \{1, \cdots, X\}$ with the ratings in M, avoiding pairwise (user, item) rating overlap on each split, i.e., $Tr_x \cap Te_x = \emptyset$. These methods are described in the following.

Repeated sampling. This method repeats the hold-out splitting procedure X times, according to some *base rating set* and *rating set size* conditions. By using a random, time-independent rating ordering condition (i.e., a different sequence is generated in each repetition, due to the use of a random ordering) it is ensured that each split will be different from the rest. This method has been applied in (Gordea and Zanker, 2007).

User resampling. This method randomly samples a subset of users $U_x \subset U$ in each repetition, and then applies the hold-out splitting procedure on each dataset $M_x = \{r_{u,\cdot} | u \in U_x\}$, according to some *base rating set* and *rating set size* condition. This method has been used in (Zheng and Li, 2011).

X-fold cross validation. This is a commonly used method that takes a time-independent ordered sequence of ratings, and splits it into X disjoint sets (called *folds*). Then, X different training and test sets are built in each repetition, by assigning the ratings in one fold to the test set, and the ratings in the remaining $X - 1$ folds to the training set, as e.g. done in (Adomavicius et al., 2005). In general, the folds are equally sized, and thus the value of X is used to determine the size of training and test sets. Note that this is similar to use a proportion-based size condition, with $q_{prop} = 1/X$ .Furthermore, this method can be applied with a community-centered or a user-centered base rating set conditions. Figure 4.7 shows example folds generated by a 3-fold cross-validation method when using a user-centered (right side of the figure) and a community-centered (bottom side of the figure) base rating set conditions. We note that using a user-centered base rating set condition ensures that most users will have ratings in each fold, while a community-centric base condition does not. Despite this difference, details about the used base rating set are rarely given in the literature; in general, only the usage of an X-fold cross validation method and the value of X are reported.

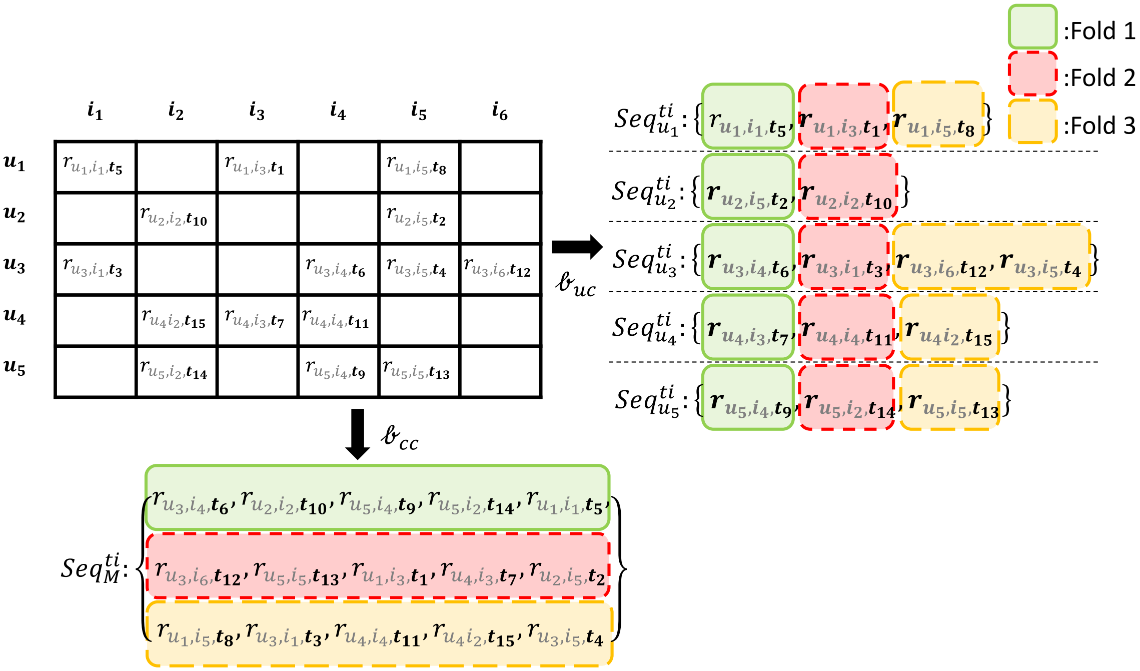
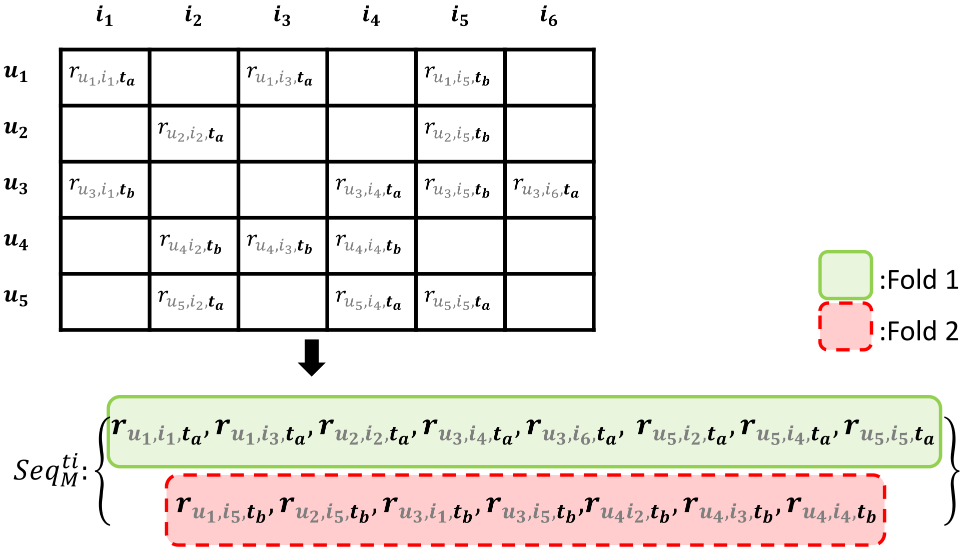

🟩 :Fold 1
🟥 :Fold 2
🟨 :Fold 3
i_1 i_2 i_3 i_4 i_5 i_6
u_1: r_{u_1,i_1,t_5}, r_{u_1,i_3,t_1}, r_{u_1,i_5,t_8}
u_2: $r_{u_2,i_2,t_{10}}$, r_{u_2,i_5,t_2}
u_3: r_{u_3,i_1,t_3}, r_{u_3,i_4,t_6}, r_{u_3,i_5,t_4}, $r_{u_3,i_6,t_{12}}$
u_4: $r_{u_4,i_2,t_{15}}$, r_{u_4,i_3,t_7}, $r_{u_4,i_4,t_{11}}$
u_5: $r_{u_5,i_2,t_{14}}$, r_{u_5,i_4,t_9}, $r_{u_5,i_5,t_{13}}$
b_{uc}
$Seq^{ti}_{u_1}: \{r_{u_1,i_1,t_5}, r_{u_1,i_3,t_1}, r_{u_1,i_5,t_8}\}$
$Seq^{ti}_{u_2}: \{r_{u_2,i_5,t_2}, r_{u_2,i_2,t_{10}}\}$
$Seq^{ti}_{u_3}: \{r_{u_3,i_4,t_6}, r_{u_3,i_1,t_3}, r_{u_3,i_6,t_{12}}, r_{u_3,i_5,t_4}\}$
$Seq^{ti}_{u_4}: \{r_{u_4,i_3,t_7}, r_{u_4,i_4,t_{11}}, r_{u_4,i_2,t_{15}}\}$
$Seq^{ti}_{u_5}: \{r_{u_5,i_4,t_9}, r_{u_5,i_2,t_{14}}, r_{u_5,i_5,t_{13}}\}$
b_{cc}
Seq^{ti}_{M}:
$r_{u_3,i_4,t_6}, r_{u_2,i_2,t_{10}}, r_{u_5,i_4,t_9}, r_{u_5,i_2,t_{14}}, r_{u_1,i_1,t_5},$
$r_{u_3,i_6,t_{12}}, r_{u_5,i_5,t_{13}}, r_{u_1,i_3,t_1}, r_{u_4,i_3,t_7}, r_{u_2,i_5,t_2}$
$r_{u_1,i_5,t_8}, r_{u_3,i_1,t_3}, r_{u_4,i_4,t_{11}}, r_{u_4,i_2,t_{15}}, r_{u_3,i_5,t_4}$

Figure 4.7. Examples of folds created by a 3-fold cross-validation method using a user-centered (upper side) and a community-centered (lower side) base rating set conditions.

Leave-one-out. This is a particular X-fold cross validation method in which $X = |M|$. Each rating in M is considered as the test set, and the remaining ratings are used for training. Although this method has showed the lowest variability in results in generic prediction problems (Arlot and Celisse, 2010), its high computational cost (the algorithms must be trained and evaluated $|M|$ times) makes it unfeasible in many situations. This method has been applied in (Cremonesi and Turrin, 2009).

Category-based cross validation. This is a X-fold cross validation method that has been used to evaluate categorical TARS. In this case, a rating set is partitioned according to the value of one or more categorical context variables. Afterwards, training-test splits are built with ratings from one partition each time. The rationale for this is twofold: on the one hand, making independent evaluations of TARS performance in different categorical contexts; and, on the other hand, facilitating the computation of a single value of performance across contexts for a given metric.

The category-based cross validation condition can be applied with a time-independent or a time-dependent[17] rating ordering condition, but requires the availability of categorical context information associated to the ratings. Note that, when the categorical variable corresponds to a time context variable, this method lets evaluate TARS performance separately on different time contexts (e.g. weekday vs. weekend). This method, in contrast, does not ensure that time dependencies between ratings in a given training-test set pair hold (unless a time-dependent rating ordering condition is used). The number of different splits that can be generated with this method is limited by the number of different categorical values of the contextual variables. Figure 4.8 shows example partitions generated when applying this condition using a community-based base rating set condition. The training-test splitting procedure is performed afterwards on each of these partitions individually. This method has been applied in (Baltrunas and Amatriain, 2009).



	i_1	i_2	i_3	i_4	i_5	i_6
u_1	r_{u_1,i_1,t_a}		r_{u_1,i_3,t_a}		r_{u_1,i_5,t_b}	
u_2		r_{u_2,i_2,t_a}			r_{u_2,i_5,t_b}	
u_3	r_{u_3,i_1,t_b}			r_{u_3,i_4,t_a}	r_{u_3,i_5,t_b}	r_{u_3,i_6,t_a}
u_4		$r_{u_4 i_2,t_b}$	r_{u_4,i_3,t_b}	r_{u_4,i_4,t_b}		
u_5		r_{u_5,i_2,t_a}		r_{u_5,i_4,t_a}	r_{u_5,i_5,t_a}	

Seq_M^{ti}: $\{ r_{u_1,i_1,t_a}, r_{u_1,i_3,t_a}, r_{u_2,i_2,t_a}, r_{u_3,i_4,t_a}, r_{u_3,i_6,t_a}, r_{u_5,i_2,t_a}, r_{u_5,i_4,t_a}, r_{u_5,i_5,t_a}$
$r_{u_1,i_5,t_b}, r_{u_2,i_5,t_b}, r_{u_3,i_1,t_b}, r_{u_3,i_5,t_b}, r_{u_4 i_2,t_b}, r_{u_4,i_3,t_b}, r_{u_4,i_4,t_b} \}$

Figure 4.8. Example of a two-fold partition generated by category-based cross-validation.

In general, time-independent cross-validation methods present two characteristics that must be handled carefully when evaluating TARS. On the one hand, they may produce overlapping training and test sets from the time ordering point of view. On the other hand, they may produce pairwise (user, item) rating overlaps between different training or test sets, which may make the application of statistical tests difficult (Dietterich, 1998). The first issue can be addressed by using one of the time-dependent cross-validation methods described in the following subsection.

4.4.2 Time-dependent cross-validation condition

The cross-validation methods that satisfy this condition aim to ensure that time dependencies between ratings in each training-test set pair hold, i.e., $\forall r_{Tr} \in Tr_x, \forall r_{Te} \in$

[17] The application of a time-dependent rating ordering condition requires the availability of rating timestamps.

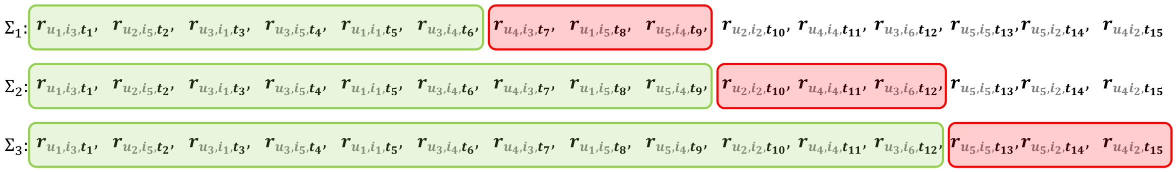
$Te_x, \mathcal{T}(r_{Tr}) < \mathcal{T}(r_{Te})$. This is accomplished by using the combination of a community-centered base rating set and a time-dependent rating ordering conditions, and some time-based rating set size condition. The time-based size condition can be iteratively updated to form time-evolving training and test sets. These methods are described in the following.

Time-dependent resampling. This is a simple method that selects X different ratings $r^x \in M \mid x \in \{1,2,\cdots,X\}$, and builds splits Σ_x by using a $\mathcal{S}_{time}$ condition with the time-based size threshold $q^x_{time} = \mathcal{T}(r^x)$. This method has been used in (Hermann, 2010).

Time-dependent users resampling. This method is similar to the time-independent user resampling, but uses a time-dependent rating ordering condition. In this case, X different splits are built because the users (and thus the ratings) vary from sample to sample. This method has been used in (Cremonesi and Turrin, 2010).

Increasing-time window. This method builds different splits by means of increasing the timespan of training sets. It requires the definition of a training window size $q_{time\_window\_Tr}$ and a test window size $q_{time\_window\_Te}$, measured in some time unit, e.g. *days* or *weeks*. For building the initial training and test sets, this method uses a $\mathcal{S}_{time}$ condition with $q_{time} = \mathcal{T}\left(r_{(1)}\right) + q_{time\_window\_Tr}$ and $q_{end\_time} = q_{time} + q_{time\_window\_Te}$. The method builds subsequent training and test sets by iteratively updating q_{time} and q_{end_time} as follows: $q'_{time} = q_{time} + q_{time\_window\_Te}$ and $q'_{end\_time} = q_{end\_time} + q_{time\_window\_Te}$. Setting $q_{time\_window\_Tr}$ and $q_{time\_window\_Te}$ such that $q_{time\_window\_Tr} + X \cdot q_{time\_window\_Te} = \mathcal{T}\left(r_{(|M|)}\right)$, the method builds X training and test sets, being the timespans in the test sets equally sized. This method has been used in (Lathia et al., 2009a).

Figure 4.9 shows example splits generated by increasing-time window. Green shadowed ratings are assigned to the training set, and the red shadowed to the test set. In the figure, $q_{time\_window\_Tr} = 6$ and $q_{time\_window\_Te} = 3$ (for the purpose of this example, we assume that time indexes used correspond to certain time measure unit, e.g. days). The upper side of the figure shows the first split, the split in the middle represents the second one, and the bottom side of the figure shows the third split. We note that ratings not shadowed are not used in the corresponding split.



Σ_1: r_{u_1,i_3,t_1}, r_{u_2,i_5,t_2}, r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_1,i_1,t_5}, r_{u_3,i_4,t_6}, r_{u_4,i_3,t_7}, r_{u_1,i_5,t_8}, r_{u_5,i_4,t_9}, $r_{u_2,i_2,t_{10}}$, $r_{u_4,i_4,t_{11}}$, $r_{u_3,i_6,t_{12}}$, $r_{u_5,i_5,t_{13}}$, $r_{u_5,i_2,t_{14}}$, $r_{u_4 i_2,t_{15}}$

Σ_2: r_{u_1,i_3,t_1}, r_{u_2,i_5,t_2}, r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_1,i_1,t_5}, r_{u_3,i_4,t_6}, r_{u_4,i_3,t_7}, r_{u_1,i_5,t_8}, r_{u_5,i_4,t_9}, $r_{u_2,i_2,t_{10}}$, $r_{u_4,i_4,t_{11}}$, $r_{u_3,i_6,t_{12}}$, $r_{u_5,i_5,t_{13}}$, $r_{u_5,i_2,t_{14}}$, $r_{u_4 i_2,t_{15}}$

Σ_3: r_{u_1,i_3,t_1}, r_{u_2,i_5,t_2}, r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_1,i_1,t_5}, r_{u_3,i_4,t_6}, r_{u_4,i_3,t_7}, r_{u_1,i_5,t_8}, r_{u_5,i_4,t_9}, $r_{u_2,i_2,t_{10}}$, $r_{u_4,i_4,t_{11}}$, $r_{u_3,i_6,t_{12}}$, $r_{u_5,i_5,t_{13}}$, $r_{u_5,i_2,t_{14}}$, $r_{u_4 i_2,t_{15}}$

Figure 4.9. Example of increasing-time window splits, using $q_{time\_window\_Tr} = 6$ and $q_{time\_window\_Te} = 3$. Green shadowed ratings correspond to training data, and red shadowed ratings correspond to test data.

Fixed-time window. This cross-validation method is a variation of the increasing-time window method. The timespan size of each training set is maintained by means of discarding "old" ratings. In this case, the first training and test sets are built as in the increasing-time window method, and the subsequent training sets are pruned by discarding those ratings out of the training time window $q_{time\_window\_Tr} : r \mid \mathcal{T}(r) < q_{time} - q_{time\_window\_Tr}$. Figure 4.10 shows example splits generated by fixed-time window. In this example we use the same setting as in Figure 4.9, only varying the cross-validation method. In this case, initial ratings are discarded in subsequent splits.

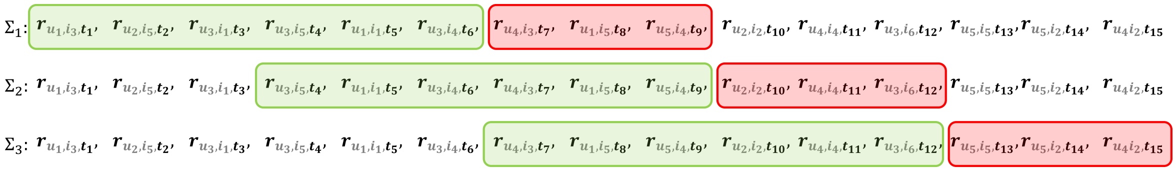

Σ_1: r_{u_1,i_3,t_1}, r_{u_2,i_5,t_2}, r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_1,i_1,t_5}, r_{u_3,i_4,t_6}, r_{u_4,i_3,t_7}, r_{u_1,i_5,t_8}, r_{u_5,i_4,t_9}, $r_{u_2,i_2,t_{10}}$, $r_{u_4,i_4,t_{11}}$, $r_{u_3,i_6,t_{12}}$, $r_{u_5,i_5,t_{13}}$, $r_{u_5,i_2,t_{14}}$, $r_{u_4 i_2,t_{15}}$
Σ_2: r_{u_1,i_3,t_1}, r_{u_2,i_5,t_2}, r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_1,i_1,t_5}, r_{u_3,i_4,t_6}, r_{u_4,i_3,t_7}, r_{u_1,i_5,t_8}, r_{u_5,i_4,t_9}, $r_{u_2,i_2,t_{10}}$, $r_{u_4,i_4,t_{11}}$, $r_{u_3,i_6,t_{12}}$, $r_{u_5,i_5,t_{13}}$, $r_{u_5,i_2,t_{14}}$, $r_{u_4 i_2,t_{15}}$
Σ_3: r_{u_1,i_3,t_1}, r_{u_2,i_5,t_2}, r_{u_3,i_1,t_3}, r_{u_3,i_5,t_4}, r_{u_1,i_1,t_5}, r_{u_3,i_4,t_6}, r_{u_4,i_3,t_7}, r_{u_1,i_5,t_8}, r_{u_5,i_4,t_9}, $r_{u_2,i_2,t_{10}}$, $r_{u_4,i_4,t_{11}}$, $r_{u_3,i_6,t_{12}}$, $r_{u_5,i_5,t_{13}}$, $r_{u_5,i_2,t_{14}}$, $r_{u_4 i_2,t_{15}}$

Figure 4.10. Example of fixed-time window splits, using $q_{time\_window\_Tr} = 6$ and $q_{time\_window\_Te} = 3$. Green shadowed ratings correspond to training data, and red shadowed ratings correspond to test data.

We note that this method leads to faster training and evaluation processes compared with those of the increasing-time window method, since the training set sizes do not increase. However, it has the disadvantage of losing part of the training data, which may be valuable for some TARS. This method has been used in (Pradel et al., 2011).

Time-dependent cross-validation methods require (as well as the time-dependent rating ordering condition) the availability of rating timestamps. Moreover, they suffer the same problems of applying a time-based rating set size condition, since some users may have all of their ratings within the training (or test) timespan. However, they let maintain training-test temporal dependencies, which, as discussed before, is a more realistic scenario for TARS performance evaluation.

The cross-validation techniques described in this section cover most of the methods used in TARS evaluation. By using one of these methods, the variability of evaluation results is diminished. Having defined the training-test splitting and the cross-validation conditions to use, it is possible to perform the evaluation of TARS on a rating prediction task. In contrast, to evaluate a top-N recommendations task, it is necessary to define some additional specific conditions, as we describe in the next section.

4.5 Specific evaluation conditions for top-N recommendations

These conditions are specific for the evaluation of a top-N recommendations task. They include conditions for the selection of the target items, and for the identification of the items considered as relevant. These conditions state which items are ranked in order to select the top items for recommendation, and which items are considered as relevant for each user, respectively.

4.5.1 Target item conditions

These conditions select the target items $Target_u$ to be ranked by the evaluated TARS. We recall that the need to take these conditions into consideration arises from the different nature of rating prediction and top-N recommendations tasks. In rating prediction, a RS is requested to predict the rating a target user would give to a target item. In top-N recommendations, there is no target item, but only a target user; the RS is then requested to estimate the set of top items the target user would prefer.

We note that a broad –and close to a real world setting– target item condition would be ranking all the items except the target user u's training items, which are already known by u, i.e., setting $Target_u = I \backslash I_{Tr_u}$. In the revised literature, nonetheless, smaller item sets have been used as $Target_u$, letting a faster evaluation, as fewer items have to be ranked by the assessed TARS.

The impact of using different target item sets on recommendation performance assessment has been studied by Bellogín et al. (2011). In the following we define conditions describing the target item sets found in the revised TARS papers.

User-based target items. The items in the target user's test set Te_u are ranked, i.e., $Target_u = I_{Te_u}$.

The rationale for this condition is to avoid ranking items for which there is no explicit evidence of user preferences. This type of target item set has been used in (Adomavicius et al., 2005) and (Ma et al., 2007).

Community-based target items. All the items in the test set Te (i.e., the ratings of the whole community of test users) are ranked, i.e., $Target_u = \left(\bigcup_{v \in U} I_{Te_v}\right) \backslash I_{Tr_u}$. A variation that includes more target items consists of ranking all the items in the community training set Tr, i.e., $Target_u = \left(\bigcup_{v \in U} I_{Tr_v}\right) \backslash I_{Tr_u}$.

The rationale of this condition is to include in $Target_u$ items interpreted as non-relevant for user u, in order to assess a recommendation algorithm's ability to better rank relevant items. The underlying assumption is that items rated by u are relevant, and unrated items are presumably non-relevant. This type of target item set has been used in (Pradel et al., 2011) and (Zimdars et al., 2001).

One-plus random target items. In order to describe this condition, we first define the set of highly relevant items for user u as $I_{hrel_u} = \left\{i \in I \mid r_{u,i} \in Te_u, r_{u,i} > \tau_{hrel}\right\}$, where τ_{hrel} is a high-relevance threshold, i.e., items in the ground truth of u with high ratings; and the set of non-relevant items for user u as $I_{\overline{rel}_u} = \{i \in I \mid r_{u,i} = \emptyset\}$, i.e., items that have not been rated by u. Several sets $Target_u^k$ for the target user u are built, each of them consisting of one highly relevant item $i^k \in I_{hrel_u}$, plus a set of non-relevant items $J_{\overline{rel}_u} \subseteq I_{\overline{rel}_u}$: $Target_u^k = i^k \cup J_{\overline{rel}_u}$.

The rationale for this condition is to find out whether a recommendation algorithm is able to consistently rank the selected relevant items above all other non-relevant items (Cremonesi et al., 2010).This type of target item set has been used in (Stormer, 2007).

Other target item conditions. A particular target item set we identified in our review corresponds to a *given list of target items*, where a fixed set of items is ranked. This approach has been used in domains where there is a fixed set of possible items to recommend at a particular moment, as in TV show recommendation, in which there is a set of shows being broadcast at a particular time. In such a case $Target_u$ contains the TV listings at recommendation time, as shown in (Vildjiounaite et al., 2008).

Another target item set we identified corresponds to a *one-item target item*, in which $Target_u$ is composed of just one item at a time. In this case, a TARS has to decide whether or not to recommend a given item. This condition can be used to measure the ability of an algorithm to recommend only relevant items by repeating the evaluation process with all known items. This particular target item set was used with the leave-one-out cross-validation method in (Panniello et al., 2009a).

All these conditions broadly address all the definitions of target item set used in TARS evaluation. From the evaluation design point of view, the usage of different target item conditions lets control the amount of items to be ranked –and consequently, the number of rating predictions to compute–, which has an important effect on the time needed to perform the evaluation. Once defined the items to be ranked, it is necessary to establish which of those items will be interpreted as relevant for each user, which is required by several evaluation metrics used for assessing the top-N recommendations task. The conditions defining relevant items are described in the next subsection.

4.5.2 Relevant item conditions

Relevant item conditions select the items to be interpreted as relevant for the target user. The notion of relevance is central for information retrieval metrics applied to evaluate top-N recommendations. A RS has a set of ratings for some items, and depending on such ratings, the items have to be interpreted as relevant or non-relevant. In this context we define two main conditions, namely the test-based and the threshold-based relevant item conditions.

Test-based relevant items. The set of relevant items for user u, I_{hrel_u}, is formed by the items in u's test set: $I_{rel_u} = I_{Te_u}$. By using this condition, rating/consuming an item is interpreted as indicative of interest for such item. This type of relevant item set has been used in (Liu et al., 2010b).

Threshold-based relevant items. The items in the user's test set rated/consumed above a threshold value τ_{rel} are considered as relevant, i.e., $I_{rel_u} = \{i \in I | r_{u,i} \in Te_u, r_{u,i} \geq \tau_{rel}\}$. Thus, the test set is pruned from low rated items. This type of relevant item set has been used in (Adomavicius et al., 2005) and (Vildjiounaite et al., 2008). Note that the definition of I_{rel_u} is similar to the definition of I_{hrel_u} used in the description of the one-plus random target item condition. The difference between both sets is their threshold value, being $\tau_{hrel} > \tau_{rel}$ in general.

The usage of a threshold-based relevant item condition lets a more detailed control of which items should be interpreted as relevant for the user. As noted by (Parra and Amatriain, 2011), items with low rating or low usage/consumption rates can be interpreted as negative feedback, and thus, it is counter-intuitive to interpret such items as relevant ones –which is the results of using a test-based relevant item condition.

Target item and relevant item conditions let define the specific decisions needed for assessing a top-N recommendations task. With them we conclude the description of evaluation conditions used in TARS evaluation.

4.6 Conclusions

A careful review and comparison of the evaluation protocols followed to assess state-of-the-art TARS showed us that there are several methodological differences on how the evaluation has been conducted among the different research works. Analyzing such differences, we pose a number of methodological questions regarding the design of a TARS evaluation protocol:

- What base rating set is used to perform the training-test splitting?
- What rating ordering is used to assign ratings to the training and test sets?
- How many ratings comprise the training and test sets?
- What cross-validation method is used for increasing the reliability of the evaluation results and of their generalization?

In addition to these questions, we have also observed differences in assessing the top-N recommendations task, posing the following two methodological questions:

- Which items are considered as target items (in a top-N recommendations task)?
- Which items are interpreted as relevant for each user (in a top-N recommendations task)?

These questions are addressed by means of a number of evaluation conditions that we characterized and formalized, related with the evaluation settings found in the review of TARS literature. The conditions express decisions related to the training-test splitting and cross-validation processes in the evaluation of RS, and specific aspects regarding the evaluation of top-N recommendations.

In order to facilitate the comprehension of such decisions, in this chapter we have presented a methodological framework for describing and formalizing evaluation conditions adopted when designing an offline evaluation of TARS. This framework is aimed to make the evaluation process fair and reproducible under different circumstances, by means of facilitating a precise communication of the evaluation conditions that drive an evaluation process.

The formalism of the framework includes the definition of a splitting procedure that, using a set of conditions as input parameter, lets precisely build and reproduce data splits (i.e., training and test sets) for a given evaluation setting.

By using the splitting procedure and different combinations of the formalized conditions, diverse evaluation settings for TARS can be accurately described. We have included examples of replication of data splits used commonly in TARS evaluation to show the usage and capabilities of the framework. In this way, the proposed framework may help researchers and practitioners in conducting fair evaluations of new TARS, and facilitate reproducibility of results and comparisons with other TARS proposals.

We note that the influence of these conditions in evaluation results is still an open research question. In the next chapter, we use the proposed framework to analyze the impact of the conditions in the literature of TARS, and to perform an empirical evaluation and comparison of several TARS using different combinations of conditions, aiming to shed light and better understand the effect of changing evaluation conditions in TARS performance assessment.

Chapter 5

Analysis of evaluation conditions for time-aware recommender systems

The methodological framework proposed in Chapter 4 lets provide a precise statement and reproducibility of the conditions in which a TARS is evaluated. To the best of our knowledge, the impact that using some or others of such conditions has on the assessment of recommendation results has not been studied in the literature. Given that using different combinations of the conditions lets define distinct training and test sets –which are the basic input for building recommendation models/heuristics and computing evaluation metrics–, we hypothesize that differences on the used evaluation conditions may have an important impact on the measured and reported results.

In this chapter we conduct an empirical analysis on the conditions established in the proposed methodological framework. In Section 5.1 we classify state-of-the-art TARS according to the conditions used in their evaluation. In Section 5.2 we report a study analyzing the impact that some key conditions have on the performance of well-known TARS, in the movie and music recommendation domains. In Section 5.3 we provide a number of general guidelines to select proper conditions for evaluating particular TARS, drawn from the analysis of the findings presented in Sections 5.1 and 5.2. Finally, in Section 5.4 we end the chapter with some conclusions of the conducted analysis.

5.1 Evaluation conditions in state-of-the-art time aware recommender systems

The methodological framework defined in Chapter 4 serves as a descriptive tool that enables the accurate communication of the decisions taken during the design of a protocol to evaluate TARS. Although the evaluation conditions that characterize the framework were identified from the review of TARS literature, we only have an approximation of the volume of TARS studies applying each of these conditions. In order to identify the most common evaluation settings, in this section we provide an exhaustive classification of state-of-the-art TARS according to the evaluation conditions defined in our methodological description framework. This classification lets an easy identification of the evaluation conditions used in the reviewed work, and thus facilitate the replication of the setting used for evaluating a particular TARS approach.

Table 5.1 provides the summary of the revised papers on time-aware recommendation approaches that make use of an offline evaluation, by showing the used conditions, as defined in Chapter 4. In the table, each row represents a particular combination of conditions, and each column is associated to an evaluation condition; some papers include more than one evaluation and/or condition combination –hence, some papers appear in more than one row.

In the table, we first observe that despite the fact that the revised papers deal with time-aware recommendation, in 24.6% (14 out of 57) of them, a time-independent ordering of ratings is used in the evaluation protocols. On the other hand, a combination of a community-centered base rating set and a time-dependent rating ordering –which provides the evaluation scenario most similar to a real-world setting, maintaining temporal dependencies between training and test ratings, $\forall r_{Te}, r_{Tr}, t(r_{Te}) > t(r_{Tr})$– is used in only 38.6% of the papers (22). Additionally, in less than the half of the papers (24) a time-based size condition is utilized.

Regarding cross-validation methods, the basic hold-out procedure (i.e., no cross-validation) is used in 70.2% of the papers (40). Only in 10 of the 19 papers in which cross-validation was used (17.5% of the whole list of papers) a time-based cross-validation method was used.

With respect to specific evaluation conditions of the top-N recommendations task, we first note that in 25 papers these conditions are not applicable, since the above task was not evaluated. We observe that in the majority of TARS-related papers addressing the top-N recommendations task (21 out of 32), a test-based relevant item condition was used, but we find an even distribution on the use of different target item conditions.

Table 5.1. List of revised papers about TARS, and their used offline evaluation conditions. "X" denotes an evaluation condition (at the corresponding column) that is used in a paper (at the corresponding row). Some papers include more than one evaluation and/or condition combination (one at each row). "-" denotes an evaluation condition that is not applicable for a paper. "?" indicates that we could not identify whether an evaluation condition was used or not in a paper. CV stands for cross-validation.

Paper	General evaluation conditions										Top-N recommendations task evaluation conditions					
	Base rating set		Rating ordering		Rating set size			Cross-validation			Target items				Relevant items	
	Community-centered	User-centered	Time-independent	Time-dependent	Proportion-based	Fixed	Time-based	None (Hold-out)	Time-independent CV	Time-dependent CV	User-based	Community-based	One-plus random	Other	Test-based	Threshold-based
(Adomavicius et al., 2005)	X		X		X				X		X					X
(Ardissono et al., 2004)	X			X			X	X			?	?	?	?	X	
(Baltrunas and Amatriain, 2009)	X		X		X				X		-	-	-	-	-	-
(Bell et al., 2007)		X		X		X		X			-	-	-	-	-	-
(Bell et al., 2008)		X		X		X		X			-	-	-	-	-	-
(Brenner et al., 2010)	X			X			X	X				X				X
(Campos et al., 2010)	X			X			X	X				X				X
(Campos et al., 2011b)	X			X			X			X		X				X
(Cao et al., 2009)		X		X	X			X				X			X	
(Chen et al., 2012)		X		X		X		X			-	-	-	-	-	-
(Cremonesi and Turrin, 2009)		X	X			X			X				X		X	
(Cremonesi and Turrin, 2010)	X			X			X			X			X		X	
(Ding and Li, 2005)		X		X		X		X			-	-	-	-	-	-
		X	?	?		X		X			-	-	-	-	-	-
(Ding et al., 2006)		X		X		X		X			-	-	-	-	-	-
(Gantner et al., 2010)	X			X			X	X				X				X
(Gordea and Zanker, 2007)		X	X			X			X				X		X	
		X		X		X				X			X		X	
(Gorgoglione and Panniello, 2009)	X			X			X	X						X		X
(Hermann, 2010)	X			X			X			X	?	?	?	?	X	
(Iofciu and Demartini, 2009)	X			X			X	X						X	X	
(Jahrer and Töscher, 2012)		X		X		X		X			-	-	-	-	-	-
(Jahrer et al., 2010)		X		X		X		X			-	-	-	-	-	-
(Karatzoglou, 2011)	X			X	X			X			-	-	-	-	-	-
(Karatzoglou et al., 2010)	X		X		X				X		-	-	-	-	-	-
(Koenigstein et al., 2011)		X		X		X		X			-	-	-	-	-	-
(Koren, 2009b)		X		X		X		X			-	-	-	-	-	-
(Koren, 2009a)		X		X		X		X			-	-	-	-	-	-
(Lathia et al., 2009a)	X			X			X			X	-	-	-	-	-	-
(Lathia et al., 2010)	X			X			X			X		X			X	
(Lee et al., 2010)	X		X		X			X					X		X	
(Lee et al., 2008)	?	?	?	?	X			X			?	?	?	?	X	

Paper	General evaluation conditions										Top-N recommendations task evaluation conditions					
	Base rating set		Rating ordering		Rating set size			Cross-validation			Target items				Relevant items	
	Community-centered	User-centered	Time-independent	Time-dependent	Proportion-based	Fixed	Time-based	None (Hold-out)	Time-independent CV	Time-dependent CV	User-based	Community-based	One-plus random	Other	Test-based	Threshold-based
(Lee et al., 2009)	?	?	?	?	X			X			?	?	?	?	X	
(Li et al., 2011)	X		X		X				X		-	-	-	-	-	-
(Lipczak et al., 2009)	X			X			X	X						X	X	
(Liu et al., 2010a)	X			X			X	X				X				X
(Liu et al., 2010b)	X			X			X			X	?	?	?	?	X	
(Lu et al., 2009)	X			X			X	X			-	-	-	-	-	-
(Ma et al., 2007)		X	X		X			X			X				?	?
(Min and Han, 2005)	?	?	?	?	?	?	?	X			-	-	-	-	-	-
(Montanés et al., 2009)		X		X			X	X						X	X	
(Panniello et al., 2009a)	X		X			X			X					X	X	
	X			X			X	X						X	X	
(Panniello et al., 2009b)	X			X			X	X						X		X
(Pradel et al., 2011)	X			X			X			X				X	X	
	X			X			X			X		X			X	
(Rendle, 2011)	X			X			X			X	-	-	-	-	-	-
(Rendle et al., 2011)	X		X		X				X		-	-	-	-	-	-
(Stormer, 2007)	X		X			X			X				X		X	
(Tang et al., 2003)		X	X		X			X			X					X
		X		X		X		X			X					X
(Töscher and Jahrer, 2008)		X		X		X		X			-	-	-	-	-	-
(Töscher et al., 2008)		X		X		X		X			-	-	-	-	-	-
(Vildjiounaite et al., 2008)		X		X			X	X						X		X
(Wu et al., 2011)		X		X		X		X			-	-	-	-	-	-
(Xiang and Yang, 2009)		X		X		X		X			-	-	-	-	-	-
	?	?	?	?	X			X			-	-	-	-	-	-
(Xiang et al., 2010)		X		X		X		X					X		X	
(Xiong et al., 2010)	X			X			X	X			-	-	-	-	-	-
		X		X		X		X			-	-	-	-	-	-
		X	X			X			X		-	-	-	-	-	-
(Zhan et al., 2006)		X		X			X	X			?	?	?	?	X	
(Zheng and Li, 2011)		X		X	X					X		X			X	
(Zheng et al., 2012)		X		X		X		X			-	-	-	-	-	-
(Zimdars et al., 2001)	X			X			X	X				X			X	
	X		X		X			X				X			X	
Number of papers using the condition (total number of papers: 57)	**29**	**26**	**14**	**45**	**15**	**22**	**24**	**40**	**10**	**10**	**3**	**10**	**6**	**8**	**21**	**10**

The presented classification provides valuable information for reproducing the evaluation setting used in assessing each of the reviewed TARS. Moreover, we have examined the usage of the evaluation conditions throughout the revised work, in order to detect the most frequent ones. In the next section we investigate the effect of using different combinations of these conditions in recommendation results assessment.

5.2 Empirical comparison of evaluation conditions

The methodological framework presented in Chapter 4 introduces a number of conditions that drive the evaluation process for TARS. Different combinations of such conditions result in a wide and diverse set of evaluation methodologies followed in the TARS literature. Our framework lets state the particular evaluation setting used for assessing a given TARS approach. The framework thus facilitates the reproducibility of reported evaluation results, and moreover, as discussed in Chapter 3, the framework also lets verify if using different framework conditions influences obtained evaluation results. In this context, we hypothesize that methodological differences between studies cause divergences on the recommendation results from certain TARS reported by different researchers.

In order to analyze the impact that changes in conditions have on evaluation results, we conducted an empirical study comparing the recommendation performance of a number of algorithms when different combinations of the above conditions are used. In this comparison we include two categorical TARS –as they are instances of the most widely used approach for context-aware recommendation– and a well-known heuristic-based continuous TARS. We consider both the rating prediction and the top-N recommendations tasks, and analyze several rating prediction accuracy and ranking precision metrics, as well as novelty and diversity metrics. Moreover, we use three publicly available datasets that belong to different domains –movies and music[18]–, and have distinct types of ratings –explicit and implicit ratings. In the next subsections we present the recommendation algorithms evaluated, the datasets used, and the evaluation methodologies and metrics that were compared. We present and discuss the results obtained, taking advantage of our framework to fairly state the conditions in which the experiments were conducted.

5.2.1 Datasets

In our study we used four datasets with timestamp information obtained from MovieLens[19] (Herlocker et al., 1999), Netflix[20] (Bennett and Lanning, 2007), and Last.fm[21] (Celma,

[18] In this work we use a pure collaborative filtering approach for the music recommendation domain. Content-based approaches –exploiting special characteristics of music, such as chord, melody, lyrics, musical genre, and composer– could be used instead, but they fall out of the scope of this study.
[19] MovieLens movie recommender system, http://movielens.umn.edu
[20] Netflix on-demand video streaming, http://www.netflix.com
[21] Last.fm Internet radio, http://www.lastfm.es

2008) systems. The MovieLens and Netflix datasets have explicit ratings for movies, and the Last.fm contains implicit ratings (listening to records) for music artists. Some basic statistics about the datasets are shown in Table 5.2. The MovieLensR dataset was built similarly as done in (Ding and Li, 2005), that is, by selecting the ratings of the first 60 users in the dataset (according to their identifiers). We used this dataset to replicate the results reported in that work, where the Time Decay algorithm obtained significant improvements over the *k*NN algorithm.

Table 5.2. Statistics of the used datasets.

	MovieLens	**MovieLensR**	**Netflix**	**Last.fm**
Number of users	6,040	60	480,189	992
Number of items	3,706	2,056	17,770	174,091
Number of events	1,000,209	8,979	100,480,507	19,150,868 (898,073 user-item pairs)
Timespan	~3 years (2000/04/26 – 2003/02/28)	~2 years (2000/12/27 – 2003/01/07)	~6 years (1999/11/11 – 2005/12/31)	~4.5 years (2005/02/14 – 2009/06/19)
Sparsity	0.0447	0.0834	0.0118	0.0052

To make the Netflix dataset more manageable, we divided it into 5 different sub-datasets on which we performed the evaluations. Specifically, we binned the original set of users into 50 equally sized bins, maintaining an increasing size of the user profiles in subsequent bins. Similarly to (Lathia et al., 2009a), we built each sub-dataset with the ratings of 1,000 randomly sampled users from each bin, plus those ratings generated during the first 500 days in the original dataset (from all users). Each of the new datasets had around 60,000 users, 17,765 items, and 11.7 million ratings, ranging the same timespan as the original dataset.

5.2.2 Recommendation algorithms

For our study, we evaluated two categorical heuristic-based TARS, namely contextual pre-filtering and contextual post-filtering algorithms. These algorithms have been widely used in the CARS-related research literature, and enable an easy incorporation of time context information. We also evaluated a Time Decay algorithm, as an example of continuous heuristic-based TARS.

As baseline algorithm we considered a context-unaware *weighted user-based kNN* algorithm (Herlocker et al., 1999):

$$F(u,i) = \bar{r}_u + \frac{\sum_{vN(u)}\left(r_{v,i} - \bar{r}_v\right) \cdot sim(u,v)}{\sum_{v\in N(u)} sim(u,v)}$$

where $sim(u,v)$ denotes a user similarity function based on the type of ratings used, including weights to penalize user similarities based on little information (understood as a low number of data points). For explicit ratings, the similarity function is the *weighted* Pearson's correlation coefficient, defined as:

$$sim(u,v) = \frac{n}{w} \cdot \frac{\sum_{i\in I_v\cap I_u}\left(r_{v,i} - \bar{r}_v\right) \cdot \left(r_{u,i} - \bar{r}_u\right)}{\sum_{i\in I_v\cap I_u}\left(r_{v,i} - \bar{r}_{\mathrm{v}}\right)^2 \cdot \sum_{i\in I_v\cap I_u}\left(r_{u,i} - \bar{r}_u\right)^2}$$

where n is the number of items rated by both users u and v, and w is a constant. In case that $n \geq w$, no penalty is applied and the above similarity turns into the standard Pearson's correlation coefficient (Eq. 2.3). For implicit ratings, the used similarity function is the weighted cosine similarity, defined as:

$$sim(u,v) = \frac{n}{w} \cdot \frac{\sum_{i\in I_v\cap I_u} r_{v,i} \cdot r_{u,i}}{\sqrt{\sum_{i\in I_v\cap I_u}\left(r_{v,i}\right)^2} \cdot \sqrt{\sum_{i\in I_v\cap I_u}\left(r_{u,i}\right)^2}}$$

where $r_{u,i}$ denotes the number of times the user u consumed item i. We set $w = 50$ and $k = 200$ in all our experiments, as they provided good results and tendencies similar to other tested values. In cases where *k*NN was unable to compute a recommendation (e.g. because the target user/item did not appear in a training set), we used the user's/item's/global mean rating as the default prediction value. In case of implicit ratings, the default prediction value was set to 0, as there is not a meaningful mean rating value, but rather a long-tailed item consumption rate.

The first evaluated recommendation algorithm is an implementation of the *contextual pre-filtering (PRF)* approach presented in (Adomavicius and Tuzhilin, 2011). This algorithm selects ratings relevant to the target context, and, using the selected ratings, it computes rating predictions with a context-unaware recommendation strategy. Specifically, we used the $timeOfTheWeek = \{workday, weekend\}$ categorical variable as time context, and the *k*NN approach described above as the underlying rating prediction strategy.

The second evaluated recommendation algorithm is an implementation of the *contextual post-filtering (POF)* approach presented in (Adomavicius and Tuzhilin, 2011). This algorithm first computes rating predictions, which can be generated by a context-unaware strategy, and then rating predictions are contextualized according to the target context. We used the categorical time variable and *k*NN rating prediction strategy used in the PRF approach. The contextualization of rating predictions was performed by a filtering strategy presented in (Panniello et al., 2009b), which penalizes the recommendation of items that are not relevant in the target context as follows. The relevance of an item i for the target user u in a particular context c is approximated by the probability $P_c(u,i) = \frac{|U_{u,i,c}|}{k}$, where $U_{u,i,c} = \{v \in N(u) | r_{v,i,c} \neq \emptyset\}$, that is, the user's neighbors v who have rated/consumed item i in context c. The item relevance is determined by a threshold value τ_{P_c} (set to 0.1 in our experiments) that is used to contextualize the ratings as:

$$F(u,i,c) = \begin{cases} F(u,i) & if \quad P_c(u,i) \geq \tau_{P_c} \\ \min(R) & if \quad P_c(u,i) < \tau_{P_c} \end{cases}$$

where $\min(R)$ returns the minimum rating value for the domain at hand.

We note that this particular implementation of POF is better suited for a top-N recommendations task, as rating predictions may be heavily penalized in some cases (due to replacement of the predicted rating value by $\min(R)$ when $P_c(u,i) < \tau_{P_c}$), thus affecting rating prediction accuracy metrics.

The third evaluated recommendation algorithm is an implementation of the *Time Decay (TD)* approach, proposed in (Ding and Li, 2005):

$$F(u,i,t) = \bar{r}_u + \frac{\sum_{v \in N(u)} \left(r_{v,i} - \bar{r}_v\right) \cdot sim(u,v) \cdot e^{-\lambda \cdot \left(t - \mathcal{T}(r_{v,i})\right)}}{\sum_{v \in N(u)} sim(u,v)}$$

with $\lambda = 1/200$ and t being a continuous time variable. In this implementation we use time values with day granularity as done in (Ding and Li, 2005).

5.2.3 Evaluation metrics and methodologies

Aiming to adequately cover the spectrum of mostly used recommendation quality metrics in offline evaluations, and to obtain an overview of distinct properties of recommendations generated by the tested algorithms under the selected evaluation methodologies, in our experiments, we considered both the rating prediction and the top-N recommendations tasks. We assessed accuracy for the two recommendation tasks, and novelty and diversity for the top-N recommendations task. Specifically, we used *Root Mean Squared Error* (RMSE) (Eq. 2.6) to assess accuracy in rating prediction, and *Precision* (P) (Eq. 2.7), *Recall* (R) (Eq. 2.8), and *normalized Discounted Cumulative Gain* (nDCG) (Eq. 2.10) to assess accuracy (ranking precision) of top-N recommendations. We computed novelty and diversity by means of *Self-Information* (SI) (Eq. 2.12) and *Intra-list Similarity* (ILS) (Eq. 2.13) respectively. We computed the *P*, *R*, *I* and *ILS* metrics at cut-off 10, and *nDCG* on the whole lists of recommended items.

Regarding the methodologies, aiming to find and analyze differences in recommendation quality results obtained with distinct combinations of evaluation conditions, we selected four different evaluation methodologies –three of them used a time-dependent order condition, and the other one used a time-independent order condition.

Having all the previous issues into account, the first used methodology (denoted as $\mathscr{b}_{uc}\mathscr{o}_{ti}\mathscr{s}_{prop}$) consists of a combination of a user-centered base rating set ($\mathscr{b}_{uc}$), a time-independent rating order ($\mathscr{o}_{ti}$), and a proportion-based size ($\mathscr{s}_{prop}$ with $q_{prop} = 0.2$) condition, which is used to generate a splitting $\Sigma_{\mathscr{b}_{uc}\mathscr{o}_{ti}\mathscr{s}_{prop}}$. According to the framework introduced in Chapter 4, this splitting is represented as:

$$\Sigma_{\mathscr{b}_{uc}\mathscr{o}_{ti}\mathscr{s}_{prop}} = \langle M, \mathscr{b}_{uc}, \mathscr{o}_{ti}, \mathscr{s}_{prop}\left(q_{prop} = 0.2\right)\rangle$$

The second methodology (denoted as $\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{prop}$) consists of the $\mathscr{b}_{uc}\mathscr{o}_{ti}\mathscr{s}_{prop}$ evaluation conditions, but using a time-dependent rating order. The generated splitting is:

$$\Sigma_{\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{prop}} = \langle M, \mathscr{b}_{uc}, \mathscr{o}_{td}, \mathscr{s}_{prop}\left(q_{prop} = 0.2\right)\rangle$$

The third methodology (denoted as $\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{prop}$) is equivalent to the $\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{prop}$ methodology with a community-centered base rating set condition. The generated splitting is:

$$\Sigma_{\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{prop}} = \langle M, \mathscr{b}_{cc}, \mathscr{o}_{td}, \mathscr{s}_{prop}\left(q_{prop} = 0.2\right)\rangle$$

Finally, the fourth methodology (denoted as $\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{fix}$) consists of a combination of a user-centered base rating set, a time-dependent rating order, and a fixed size ($q_f = 9$) condition. The generated splitting is:

$$\Sigma_{\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{fix}} = \langle M, \mathscr{b}_{uc}, \mathscr{o}_{td}, \mathscr{s}_{fix}\left(q_f = 9\right)\rangle$$

In case a user has less than 10 ratings, the size condition is switched to the proportion based size condition $\mathscr{s}_{prop}$ with $q_{prop} = 0.5$ in order to maintain such user in the training and test sets.

All the combinations use a hold-out procedure, a community-based target item condition$Target_u = \left(\bigcup_{v \in U} I_{Te_v}\right) \backslash I_{Tr_u}$, and a test-based relevant item condition $I_{rel_u} = I_{Te_u}$.

5.2.4 Experimental results

Tables 5.3, 5.4, 5.5 and 5.6 respectively show the average recommendation performance results obtained on the MovieLens, MovieLensR, Netflix and Last.fm datasets.

Table 5.3. Performance results on the MovieLens dataset, grouped by evaluation methodology. For each methodology, green-up, yellow-diagonal-up, yellow-diagonal-down and red-down arrows indicate the first, the second, the third and the fourth performing algorithm on the corresponding metric, respectively. Statistical significant differences (Wilcoxon p < 0.05) of TARS algorithms are indicated with respect to kNN (*).

Methodology	Algorithm	Metric					
		RMSE	P@10	R@10	nDCG	I@10	ILS@10
$\mathcal{B}_{uc}\,\mathcal{O}_{ti}\,\mathcal{S}_{prop}$	KNN	↑ 0.9066	↘ 0.0206	↘ 0.0090	↗ 0.3424	↘ 7.8234	↑ 0.0536
	TD	↘ 0.9396*	↑ 0.0123*	↑ 0.0049*	↑ 0.3294*	↗ 7.5165*	↗ 0.0632*
	PRF	↗ 0.9136*	↗ 0.0167*	↗ 0.0077*	↘ 0.3468*	↓ 8.4496*	↘ 0.0980*
	POF	↓ 1.4350*	↓ 0.1020*	↓ 0.0382*	↓ 0.4163*	↑ 2.7039*	↓ 0.1800*
$\mathcal{B}_{uc}\,\mathcal{O}_{td}\,\mathcal{S}_{prop}$	KNN	↑ 0.9246	↑ 0.0067	↗ 0.0031	↗ 0.3227	↓ 9.6791	↑ 0.0340
	TD	↘ 0.9448*	↗ 0.0070	↑ 0.0030	↑ 0.3196*	↘ 9.2671*	↗ 0.0390*
	PRF	↗ 0.9389*	↘ 0.0071	↘ 0.0033	↘ 0.3249*	↗ 8.8760*	↘ 0.0607*
	POF	↓ 1.6062*	↓ 0.0585*	↓ 0.0215*	↓ 0.3815*	↑ 2.7090*	↓ 0.1754*
$\mathcal{B}_{cc}\,\mathcal{O}_{td}\,\mathcal{S}_{prop}$	KNN	↑ 0.9631	↘ 0.0322	↘ 0.0054	↘ 0.4642	↘ 8.4924	↑ 0.0346
	TD	↗ 0.9637*	↗ 0.0317	↘ 0.0054	↗ 0.4640*	↗ 8.4813	↗ 0.0350
	PRF	↘ 0.9709*	↑ 0.0196*	↑ 0.0035*	↑ 0.4570*	↓ 8.9337*	↘ 0.0633*
	POF	↓ 1.9169*	↓ 0.1988*	↓ 0.0252*	↓ 0.5255*	↑ 2.5657*	↓ 0.1714*
$\mathcal{B}_{uc}\,\mathcal{O}_{td}\,\mathcal{S}_{fix}$	KNN	↑ 0.9531	↘ 0.0081	↘ 0.0091	↗ 0.2413	↗ 5.7619	↗ 0.1073
	TD	↘ 0.9804*	↑ 0.0043*	↑ 0.0047*	↑ 0.2338*	↘ 5.9759*	↑ 0.1052*
	PRF	↗ 0.9628*	↗ 0.0068*	↗ 0.0075*	↘ 0.2447*	↓ 6.7210*	↘ 0.1617*
	POF	↓ 1.4048*	↓ 0.0209*	↓ 0.0232*	↓ 0.2782*	↑ 2.6309*	↓ 0.1963*

Table 5.4. Performance results on MovieLensR dataset, grouped by evaluation methodology. For each methodology, green-up, yellow-diagonal-up, yellow-diagonal-down and red-down arrows indicate the first, the second, the third and the fourth performing algorithm on the corresponding metric, respectively. Statistical significant differences (Wilcoxon p < 0.05) of TARS algorithms are indicated with respect to kNN (*).

Methodology	Algorithm	Metric					
		RMSE	P@10	R@10	nDCG	I@10	ILS@10
$\mathcal{B}_{uc}\,\mathcal{O}_{ti}\,\mathcal{S}_{prop}$	KNN	↑ 1.1320	↓ 0.0367	↘ 0.0125	↓ 0.3784	↑ 4.6233	↗ 0.0022
	TD	↗ 1.1474	↓ 0.0367	↓ 0.0115	↘ 0.3793	↗ 4.5967	↑ 0.0017
	PRF	↘ 1.1756	↘ 0.0450	↗ 0.0208*	↗ 0.3836	↘ 4.3867*	↘ 0.0292*
	POF	↓ 2.2777*	↑ 0.0833*	↑ 0.0293*	↑ 0.4176*	↓ 2.5559*	↓ 0.2896*
$\mathcal{B}_{uc}\,\mathcal{O}_{td}\,\mathcal{S}_{prop}$	KNN	↗ 1.1767	↓ 0.0317	↓ 0.0107	↓ 0.3651	↑ 4.4579	↑ 0.0042
	TD	↑ 1.1759	↓ 0.0317	↓ 0.0107	↘ 0.3659*	↗ 4.4330	↑ 0.0042
	PRF	↘ 1.2247	↘ 0.0400	↘ 0.0157	↗ 0.3706	↘ 4.3593	↘ 0.0934*
	POF	↓ 2.1047*	↑ 0.0500	↑ 0.0238*	↑ 0.3869*	↓ 2.8733*	↓ 0.2511*
$\mathcal{B}_{cc}\,\mathcal{O}_{td}\,\mathcal{S}_{prop}$	KNN	↗ 1.2370	↓ 0.0958	↓ 0.0071	↘ 0.4535	↑ 4.8243	↑ 0.0022
	TD	↑ 1.2369	↓ 0.0958	↓ 0.0071	↓ 0.4534	↗ 4.8094	↗ 0.0023
	PRF	↘ 1.2387	↘ 0.1000	↑ 0.0273	↗ 0.4668	↘ 4.4043*	↘ 0.0675*
	POF	↓ 2.0521*	↑ 0.1292	↘ 0.0178*	↑ 0.4929*	↓ 2.8091*	↓ 0.2579*
$\mathcal{B}_{uc}\,\mathcal{O}_{td}\,\mathcal{S}_{fix}$	KNN	↗ 1.2596	↓ 0.0233	↓ 0.0259	↓ 0.3222	↗ 4.0093	↑ 0.0048
	TD	↑ 1.2463	↘ 0.0283	↘ 0.0315	↘ 0.3270*	↑ 4.0130	↗ 0.0052
	PRF	↘ 1.3133*	↑ 0.0450*	↑ 0.0500*	↑ 0.3434*	↘ 3.8496*	↘ 0.1600*
	POF	↓ 1.9723*	↗ 0.0317	↗ 0.0352	↗ 0.3282*	↓ 2.5812*	↓ 0.2375*

Table 5.5. Average performance results on the 5 sub-datasets generated from the Netflix dataset, grouped by evaluation methodology. For each methodology, green-up, yellow-diagonal-up, yellow-diagonal-down and red-down arrows indicate the first, the second, the third and the fourth performing algorithm on the corresponding metric, respectively. Statistical significant differences (Wilcoxon p < 0.05) of TARS algorithms are indicated with respect to kNN (*).

Methodology	Algorithm	Metric					
		RMSE	P@10	R@10	nDCG	I@10	ILS@10
$\mathcal{b}_{uc}\,\sigma_{ti}\,\mathcal{s}_{prop}$	kNN	↑ 0.9208	↗ 0.0060	↘ 0.0012	↗ 0.2694	↘ 9.8411	↗ 0.0683
	TD	↘ 0.9445*	↓ 0.0046*	↓ 0.0008*	↓ 0.2648*	↗ 10.1921*	↑ 0.0551*
	PRF	↗ 0.9210*	↘ 0.0055*	↗ 0.0021*	↘ 0.2682*	↑ 10.9453*	↘ 0.1756*
	POF	↓ 1.7514*	↑ 0.0668*	↑ 0.0133*	↑ 0.3497*	↓ 3.8757*	↓ 0.2452*
$\mathcal{b}_{uc}\,\sigma_{td}\,\mathcal{s}_{prop}$	kNN	↑ 0.9379	↗ 0.0023	↗ 0.0005	↗ 0.2655	↑ 13.1286	↗ 0.0259
	TD	↘ 0.9587*	↘ 0.0018	↓ 0.0003*	↓ 0.2619*	↗ 12.9199*	↑ 0.0200*
	PRF	↗ 0.9454*	↓ 0.0017*	↘ 0.0004*	↘ 0.2624*	↘ 12.6859*	↘ 0.0729*
	POF	↓ 2.0637*	↑ 0.0632*	↑ 0.0144*	↑ 0.3351*	↓ 3.8053*	↓ 0.2448*
$\mathcal{b}_{cc}\,\sigma_{td}\,\mathcal{s}_{prop}$	kNN	↑ 1.0276	↘ 0.0032	↘ 0.0007	↗ 0.2955	↑ 13.8497	↗ 0.0123
	TD	↘ 1.0356*	↗ 0.0033*	↓ 0.0006*	↘ 0.2944*	↗ 13.6752*	↑ 0.0116*
	PRF	↗ 1.0333*	↓ 0.0030*	↓ 0.0006*	↓ 0.2923*	↘ 13.1698*	↘ 0.0419*
	POF	↓ 2.2643*	↑ 0.0830*	↑ 0.0134*	↑ 0.3536*	↓ 3.4888*	↓ 0.2348*
$\mathcal{b}_{uc}\,\sigma_{td}\,\mathcal{s}_{fix}$	kNN	↑ 0.9722	↘ 0.0009	↘ 0.0010	↗ 0.1899	↗ 10.2081	↑ 0.0702
	TD	↘ 0.9838*	↓ 0.0007*	↓ 0.0009*	↓ 0.1867*	↘ 8.7635*	↗ 0.0909*
	PRF	↗ 0.9837*	↗ 0.0012*	↗ 0.0013*	↘ 0.1892*	↑ 10.9743*	↘ 0.1478*
	POF	↓ 1.7705*	↑ 0.0116*	↑ 0.0131*	↑ 0.2346*	↓ 3.7158*	↓ 0.2464*

Table 5.6. Performance results on the Last.fm dataset, grouped by evaluation methodology. For each methodology, green-up, yellow-diagonal-up, yellow-diagonal-down and red-down arrows indicate the first, the second, the third and the fourth performing algorithm on the corresponding metric, respectively. Statistical significant differences (Wilcoxon p < 0.05) of TARS algorithms are indicated with respect to kNN (*).

Methodology	Algorithm	Metric				
		P@10	R@10	nDCG	I@10	ILS@10
$\mathcal{b}_{uc}\,\sigma_{ti}\,\mathcal{s}_{prop}$	kNN	↓ 0.0013	↓ 0.0001	↓ 0.4084	↑ 5.4079	↓ 0.5886
	PRF	↘ 0.0044*	↘ 0.0004	↘ 0.4143	↘ 4.4504	↘ 0.5856
	POF	↑ 0.1254*	↑ 0.0070	↑ 0.4354	↓ 2.2878	↑ 0.3700
$\mathcal{b}_{uc}\,\sigma_{td}\,\mathcal{s}_{prop}$	kNN	↓ 0.0005	↓ 0.0000	↓ 0.3856	↑ 5.4116	↓ 0.5662
	PRF	↘ 0.0022*	↘ 0.0001*	↘ 0.3910*	↘ 4.4670*	↘ 0.5402*
	POF	↑ 0.0874*	↑ 0.0051*	↑ 0.4132*	↓ 2.3687*	↑ 0.2796*
$\mathcal{b}_{cc}\,\sigma_{td}\,\mathcal{s}_{prop}$	kNN	↓ 0.0031	↓ 0.0002	↓ 0.3422	↑ 5.1873	↓ 0.5540
	PRF	↘ 0.0057*	↘ 0.0006*	↘ 0.3497*	↘ 4.3868*	↘ 0.5084*
	POF	↑ 0.0546*	↑ 0.0059*	↑ 0.3571*	↓ 2.2138*	↑ 0.2555*
$\mathcal{b}_{uc}\,\sigma_{td}\,\mathcal{s}_{fix}$	kNN	↓ 0.0002	↓ 0.0003	↓ 0.1839	↑ 4.8600	↓ 0.5267
	PRF	↓ 0.0002	↓ 0.0003	↘ 0.1846	↘ 4.0791*	↘ 0.4560*
	POF	↑ 0.0032*	↑ 0.0061*	↑ 0.2108*	↓ 2.3819*	↑ 0.2474*

In these tables results are grouped by evaluation methodology, facilitating the identification of absolute and relative differences of algorithms within and between evaluation conditions. On the Last.fm dataset, we only tested the top-N recommendations task, since this dataset does not have explicit ratings with which rating prediction comparisons could be done. Thus, RMSE cannot be computed for such dataset. Also, in that dataset, the TD algorithm was not assessed because of multiple events (i.e., listening records) and timestamps related to the same user-item pair, which do not let set a unique timestamp to apply the time decay weight.

In the tables, we observe that the performance results provided by each of the assessed metrics for a particular algorithm are very dissimilar when different evaluation methodologies are used. For instance, POF values of P@10 range from 0.0032 up to 0.1254 on Last.fm dataset. Figures 5.1 and 5.2 show the differences of RMSE and nDCG metrics across methodologies for the four evaluated algorithms, on the MovieLens and MovieLensR datasets respectively.

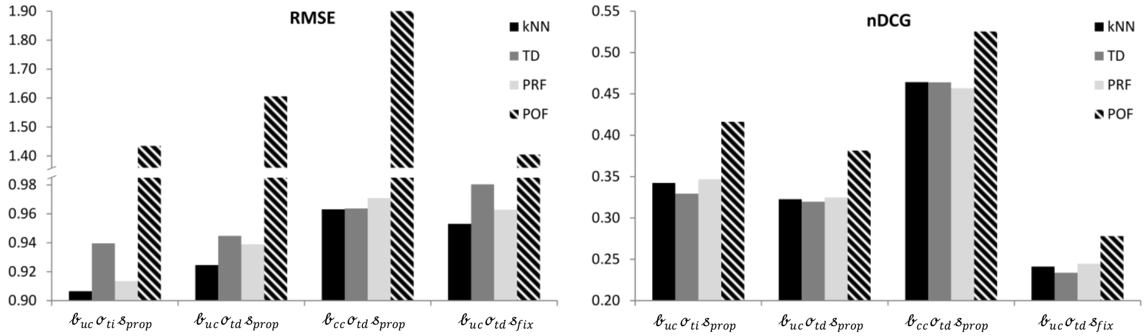


Figure 5.1. RMSE and nDCG values of different algorithms across evaluation methodologies, on the MovieLens dataset.

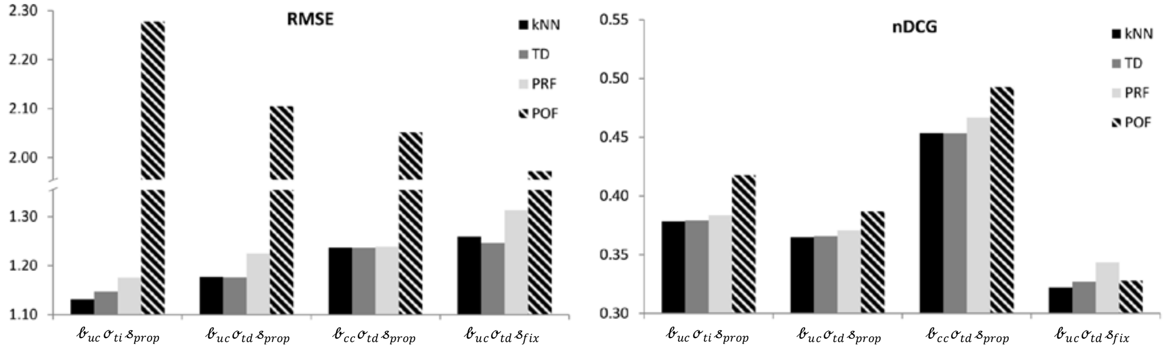


Figure 5.2. RMSE and nDCG values of different algorithms across evaluation methodologies, on the MovieLensR dataset.

We observe that using the $\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{prop}$ and $\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{fix}$ methodologies, kNN, TD and PRF values of RMSE are larger than when using $\mathscr{b}_{uc}\mathscr{o}_{ti}\mathscr{s}_{prop}$ and $\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{prop}$, on both datasets. Conversely, the $\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{fix}$ methodology leads to the lowest RMSE values of POF. Moreover, the $\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{fix}$ methodology leads to the lowest values of nDCG for all the tested algorithms –statistical significant differences in regard to the values obtained with the other methodologies.

The obtained results show that recommendation assessment under different evaluation protocols (metrics and methodologies) is an issue that has to be carefully taken into consideration in order to derive well-founded conclusions about relative performance of recommendation algorithms.

The conducted experiments also reveal that *dissimilar relative rankings of the tested algorithms are obtained, depending on the analyzed dataset, metric, and methodology*. For instance, regarding the rating prediction accuracy measured with the RMSE metric on the MovieLens dataset, when the $\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{prop}$ methodology was used TD outperformed PRF, differently to what was obtained when using the other methodologies[22]. A more notorious example can be observed in the MovieLensR dataset, where according to RMSE and using the $\mathscr{b}_{uc}\mathscr{o}_{ti}\mathscr{s}_{prop}$ methodology, the best performance is achieved by kNN. For the same metric using any of the other methodologies, the best performance is achieved by TD, although differences were not statistically significant. In the case of the Netflix dataset, TD was not able to outperform kNN in any of the used methodologies. On the other hand, PRF and POF showed worse performance than kNN in terms of RMSE, regardless of the methodology used.

The relative performance rankings of the algorithms according to the ranking precision metrics also show differences across datasets, metrics, and methodologies. One example is observed when comparing the algorithms' rankings using P@10 as performance metric. With a user-centered base rating set ($\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{prop}$), the algorithms are ranked as POF, PRF, TD and kNN on MovieLens, observing little difference between PRF, TD and kNN results. Changing into a community-centered base rating set ($\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{prop}$), and using the same dataset, the ranking is POF, kNN, TD and PRF, and the difference between kNN and TD becomes statistically significant.

We observe similar switches on the algorithms' rankings when changing the rating order condition (e.g. by comparing R@10 results on Netflix, using the $\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{prop}$ methodology instead of $\mathscr{b}_{uc}\mathscr{o}_{ti}\mathscr{s}_{prop}$) and the size condition (e.g. comparing P@10 results on Netflix, using the $\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{fix}$ methodology instead of $\mathscr{b}_{uc}\mathscr{o}_{td}\mathscr{s}_{prop}$). Moreover, it is notable the contrast in performance between rating prediction accuracy and ranking

[22] These differences are statistically significant (Wilcoxon $p < 0.05$).

precision metrics. POF consistently showed a superior performance in terms of P@10, R@10 and nDCG across datasets and methodologies, and an inferior performance according to RMSE. We also remark that the magnitude of metric values may vary considerably from one methodology to another on the same dataset.

Regarding novelty and diversity, we observe even more variations on the relative rankings depending on the datasets, methodology, and metric, compared with the ones observed in rating prediction accuracy and ranking precision metrics. It is interesting to note, anyhow, that the relative rankings on Last.fm dataset are stable across methodologies. Additionally, we also observe that in general there is a trade-off between precision-ranking accuracy, and diversity and novelty of TARS. This trade-off was also observed in (Panniello et al., 2013) when exploiting other contextual dimensions.

Supported by the fact that they were obtained on several datasets, in different recommendation domains and tasks, and with various types of ratings (explicit and implicit), *the above results show the importance of clearly stating the conditions under which TARS are evaluated*. Differences of absolute metric values obtained by the same algorithm across methodologies confirm the difficulty of comparing results reported by other authors when evaluation conditions are not described precisely. And more importantly, differences on relative rankings of the algorithms across datasets, metrics, and methodologies show the need of selecting a proper evaluation protocol for identifying the improvement capabilities of new TARS correctly.

The conducted evaluation let us detect divergences on recommendation results due to the usage of different evaluation protocols, even when the experiments were not exhaustive. We did not test all the described methodologies, and, moreover, alternative cross-validation methods, target items, and relevant item conditions may be used. However, based on the reported results, we can confirm our hypothesis that *different evaluation conditions lead to differences on recommendation results obtained*, and thus, in order to compare TARS approaches, we claim that it is necessary to do it with the same well defined evaluation setting.

The above findings remark the importance of knowing under which conditions a given TARS approach was evaluated, in order to be able to compare its performance to other approaches. In the next section, we sum up the results of the descriptive percentages obtained in Section 5.1 and the empirical comparison presented in this section, and provide a set of methodological guidelines for selecting appropriate conditions for TARS evaluation.

Taking advantage of our methodological framework, in the next section we analyze and classify state-of-the-art TARS literature based on the conditions used in their evaluation.

5.3 Analysis of key findings

In this section we summarize the key findings of the research presented in this chapter. These findings are presented as an initial set of methodological guidelines covering the selection of evaluation conditions for TARS, formulated from the analysis of the results obtained in our experiments and the usage of these conditions in TARS literature. These guidelines are aimed to help researchers and practitioners interested in TARS evaluation to select proper combinations of evaluation conditions.

We must note that these guidelines are based on the insights derived from the works analyzed in Section 5.1, and the experiments we conducted. They do not cover all possible combinations of dataset characteristics, and evaluation conditions, metrics, and methodologies. Hence, we also identify additional research required for a deeper understanding of the evaluation conditions that comprise the methodological framework presented in Chapter 4.

5.3.1 Guidelines for TARS Evaluation

From the summary given in Table 5.1 we observe that a considerable number of studies has used a time-independent rating ordering condition ($\mathscr{o}_{ti}$). In the empirical comparison, however, we found that an evaluation methodology with that condition was unable to detect the performance improvements obtained by continuous TARS on some of the used datasets. Thus, our first guideline is:

Guideline 1: *Use a time-dependent rating order condition ($\mathscr{o}_{td}$) for TARS evaluation*.

The use of this condition avoids ignoring variations on performance induced by the exploitation of time information by an evaluation methodology.

A second finding from Table 5.1, is that there is a similar amount of papers using community-centered ($\mathscr{b}_{cc}$) and user-centered ($\mathscr{b}_{uc}$) base rating set conditions. As discussed before, the combination $\mathscr{b}_{cc},\ \mathscr{o}_{td}$ provides a real world-like evaluation scenario. However, a problem of this combination is that many users may not be evaluated due to a lack of ratings in their training or test sets. Thus, considering the application of Guideline 1, we state a second guideline:

Guideline 2: *If the dataset has an even distribution of data among users, use the community-centered base condition. Otherwise, use a user-centered base condition in order to avoid biases towards profiles with long-term ratings*.

Note that in this guideline we refer to an imprecise notion of "even distribution of data among users". We do not have a specific characterization (e.g. in terms of profile sizes, timespans, and sparsity levels) as to precisely define it. Additional research on this issue is required to provide a more precise guideline.

With respect to the size condition, we noted that when using a $\mathscr{b}_{cc}$, $\mathscr{o}_{td}$ combination, proportion-based and time-based size conditions can be equivalent. This is due to the possibility of finding a proportion value that defines a splitting point equal to that from a time threshold. On the other hand, when using a $\mathscr{b}_{uc}$, $\mathscr{o}_{td}$ combination, a time-based size definition may suffer from the same general problems of $\mathscr{b}_{cc}$, $\mathscr{o}_{td}$ combination (leaving some users without training or test data). Likewise, the use of a fixed size with a $\mathscr{b}_{uc}$ condition implies that users with small profiles will have a greater proportion of their profiles held out as test data, which may lead to a cold start situation for such users. In this way, our third guideline is:

Guideline 3: *Use a proportion-based size condition.*

This guideline ensures the appropriate control of the proportion of user profiles held for test purposes in case of using a $\mathscr{b}_{uc}$ condition, and provides an adequate control of training/test proportions when using either a $\mathscr{b}_{cc}$ or a $\mathscr{b}_{uc}$ condition. Our experiments did not cover the effect of using an ending time to limit the size of the test data; this effect may have particular importance on domains with seasonal changes, and further research on this topic is required to assess its impact on measured performance values of TARS.

These first three guidelines have been derived from the empirical results reported in Section 5.1 and the classification of state-of-the art TARS literature presented in Section 5.1, and encompass the methodological questions MQ1, MQ2 and MQ3 stated in Chapter 4. Regarding MQ4, MQ5, and MQ6, although we did not perform experiments for assessing their impact in metric results, insights from the analysis of the surveyed work let us formulate two additional guidelines.

Guideline 4: *Apply a cross-validation method consistent with the conditions derived from guidelines 1, 2 and 3.*

Despite we did not test empirically the effect of different cross-validation methods on evaluation metrics, it is known that the use of more than one data split can diminish the variability of results (Dietterich, 1998; Arlot and Celisse, 2010). In this way, it is highly advisable to use a cross-validation method. Moreover, the selection of the method has to be consistent with the application of guidelines 1-3, i.e., the cross-validation method to use must apply the same rating order, base rating set, and rating set size condition advised from the guidelines.

In the case of the target item and relevant item conditions, it is important to note that they are required only when assessing a top-N recommendations task. These conditions let state which items have to be ranked to select the top-ranked items for recommendation, and which items in the test set have to be considered as relevant for the user, respectively.

Guideline 5: *For a top-N recommendations evaluation, use a community-based target item and a threshold-based relevant item condition*.

In the case of target item conditions, the closest condition to a real world setting would be to rank all available items unknown by the user. A slight variation, which may let perform faster offline evaluations, consists of considering all items selected for the test set, that is, a community-based target item condition applied over the test set. Bellogín et al. (2011) found that it makes no differences in algorithm relative ranking to apply this condition on the training or the test sets, although they did not test time-aware algorithms. Despite there are several works that apply a user-based target item condition, relative ranking of algorithms may be different from that obtained with a community-based condition (Bellogín et al., 2011). Thus, following the idea of mimicking a real world setting, it is advisable to use a community-based target item condition.

With respect to relevant item conditions, as Parra and Amatriain (2011) noted, low ratings and consumption rates could be treated as evidence of negative feedback about the items' relevance. Hence, interpreting low rated/consumed items as relevant results may be counterintuitive. In this context, using a threshold-based condition leads to a more fair evaluation of performance.

By following these guidelines we believe that TARS performance can be assessed more objectively and realistically. Moreover, the guidelines enable an easier and fairer comparison of evaluation results between approaches from different authors, which would ease the development of better TARS.

In a more general perspective, we note that the results of the experimental comparison reported in Section 5.1 show important divergences on the performance of algorithms across measured recommendation properties. Divergences are particularly remarkable between rating prediction accuracy and ranking precision metrics. In fact, the best performing algorithms on RMSE show poor results on ranking precision metrics, and vice versa. From this, we note that relying only on rating prediction accuracy metrics for assessing the performance of recommender systems is not advisable, especially when the most valuable recommendation task is distinct from rating prediction. This is an important consideration, given that most work on TARS has been commonly evaluated in terms of rating prediction accuracy, without taking into account other metrics, recommendation properties and tasks. More importantly, we stress the value of providing clear and detailed specifications of the evaluation protocols used. Having such specifications at their disposal will let other researchers and practitioners in the field to compare results fairly, and test whether a new algorithm is able to outperform existing TARS. The methodological description framework introduced in Chapter 4, which lets provide rigorous descriptions of evaluation conditions used, may help in this task.

5.3.2 Open questions

Despite the remarkable findings of this thesis –finally reflected on the proposed evaluation guidelines–, a number of issues requires further research in order to fully understand and take advantage of the different evaluation conditions identified so far.

First, more experimentation is required to properly analyze the impact of combinations of conditions not covered in this work. In particular, we did not consider cross-validation conditions, given the combinatorial explosion of conditions that should be tested. We leave the empirical study of those conditions and specific conditions regarding the top-N recommendations task as an interesting and important line of future research. For this purpose, we believe that the proposed evaluation framework provides an important conceptual structure to guide the research.

Another important pending issue is related to the analysis of the relation between characteristics of datasets (and individual user profiles), and evaluation conditions. For instance, the notion of "even distribution of data," stated in guideline 2, is imprecise, and requires further experiments in order to obtain a specific definition. Beyond that, the appropriateness of certain evaluation conditions for specific rating distributions through time/users/items, types of events (item ratings, purchases, and consumption), domains, etc. has to be investigated.

The relation between accuracy and novelty/diversity metrics also remains as an open evaluation issue. Given the increasing importance of the latter metrics in the RS field, additional analysis and explanations are required in order to provide time-aware recommendations with adequate levels of those properties.

A final question is whether improvements of TARS performance measured by offline evaluation results are effectively perceivable for real users. As noted e.g. by Knijnenburg et al. (2012), accuracy improvements are not necessarily observable by users. The lack of online evaluation studies on TARS is a major limitation to address the above question

5.4 Conclusions

In this chapter we analyzed the evaluation conditions that comprise the methodological description framework introduced in Chapter 4. We conducted an empirical comparison of the impact of several evaluation protocols on measuring relative performances of three widely used TARS approaches and one well-known non-contextual recommendation approach. Moreover, based on the methodological framework, we provided a comprehensive classification of the evaluation conditions used in state-of-the-art TARS literature.

From our analysis and experiments, we reported key methodological issues that a robust evaluation of TARS should take into consideration in order to perform a fair evaluation of approaches, and facilitate comparisons among published experiments. In particular, the obtained results showed that the use of different evaluation conditions not only yields remarkable differences between metrics measuring distinct recommendation properties –namely accuracy, precision, novelty, and diversity. They also may affect the relative ranking of approaches for a particular metric. From the results obtained in our experiments, and the analysis of the evaluation protocols used in the TARS literature, we concluded a set of general guidelines aimed to facilitate the selection of conditions for a proper TARS evaluation. These guidelines recommend making training-test splitting based on a time-dependent rating order over the whole set of ratings in a dataset, applying a proportion-based size definition for training and test sets, using a compatible cross-validation method. In the case of top-N recommendations evaluation, using a real-world like set of items to rank, and a more confident interpretation of item relevance is advised.

With the presented study we confirmed our hypothesis that the use of different evaluation conditions leads to differences on recommendation results. Nonetheless, we believe that this investigation still raises interesting additional research questions regarding TARS evaluation. We consider of key importance studying the specific impact of each identified evaluation condition on the assessment of recommendation performance. Moreover, we propose as future research to deepen the analysis of existing relations between dataset characteristics and evaluation conditions, and general effects on less studied novelty and diversity metrics. By having a robust understanding of these effects, it would be possible to select the most appropriate and fair protocol for a given recommendation evaluation task.

Finally, we highlight the need of clearly stating the conditions in which offline experiments are conducted to evaluate RS in general, and TARS in particular. By having fair and consensual evaluation conditions, we will enable the reproducibility of experiments, and ease the comparison of recommendation approaches. In the hope to contribute to such purpose, we developed the methodological description framework presented in this thesis.

Part III

Exploiting time context information in recommendation tasks

Chapter 6

Evaluating the performance of time-aware recommender systems

Exploiting time context information has been proved to be an effective approach to improve recommendation performance, as explained in the literature review presented in Chapter 3. However, despite individual improvements, little work has been done in comparing different approaches to determine which of them outperforms the others, and under what circumstances; and a number of different protocols have been used for evaluating time-aware recommender systems without consensus on the evaluation methodologies and metrics used. As shown in Chapter 5, the use of distinct evaluation conditions may lead not only to significant differences in absolute performance values, but also to distinct relative ranking of recommendation approaches, making it difficult to fairly compare TARS assessed under different evaluation settings.

In this chapter we assess the improvements on recommendation results obtained from the use of several time-aware recommendation approaches under a common, clear and reproducible evaluation setting. With this purpose, we adapt some existing approaches, propose new heuristics for some general context-aware RS, and study their performance when only exploiting time-context information. Moreover, we develop a novel methodology specifically targeted to evaluate contextualized top-N recommendations, and aimed to provide a realistic setting for the evaluation of such recommendation task. Finally, we perform a user study in order to obtain and exploit explicit and reliable time context information in the movie domain.

In Section 6.1 we present the user study where we collected user ratings for movies, together with information about the time context in which users prefer to watch the rated movies. In Section 6.2 we describe the evaluated recommendation approaches, including the new heuristics proposed. In Section 6.3 we report the results of an empirical comparison of the approaches in the rating prediction task. In Section 6.4 we describe the proposed evaluation methodology for top-N recommendations, and report the results of a comparison of the above approaches and methodologies for this task. The conclusions in Section 6.5 end the chapter.

6.1 Time context and user preferences: A user study

In general, obtaining contextual information imposes an extra effort from the user –who has to explicitly state or describe the item usage/consumption context–, and some system/device requirements to automatically infer such item usage/consumption context, e.g. by capturing time and location signals, or by analyzing the user's interactions with the system. Due to these issues, there is a lack of publicly available context-enriched datasets.

To overcome this limitation, many TARS have been evaluated using datasets with time stamped ratings. In these cases, however, it is important to note that if a RS collects ratings instead of usage/consumption data, the collected timestamps do not necessarily correspond to item usage/consumption time, and thus could not be considered as the context in which the user prefers to use/consume the items[23].

In order to count with confident context signals related to user preferences, we collected a movie rating dataset including time context information explicitly requested to users. Since we were interested in the effect of time context on user interests, we built our own Web application, and asked users (recruited via social networks) to provide personal ratings for movies they had watched. Specifically, participants rated freely chosen sets of movies using a rating scale from 1 to 5 (1 representing no user interest, and 5 a maximum user interest). The built dataset consisted of 481 ratings from 67 users given to 174 movies.

In addition to ratings, participants stated in which period of the day (*morning*, *afternoon*, *night*, and *indifferent*) and period of the week (*working day*, *weekend*, and *indifferent*) they would prefer to watch the rated movies. These categorical time context variables had been previously found as significant for time-aware recommendation (Adomavicius and Tuzhilin, 2005; Baltrunas and Amatriain, 2009). As it would not have been practical to ask users for the exact date/time in which they watched movies, we did not consider continuous time context variables.

Aimed to obtain first insights about the context influence on user preferences, we analyzed the rating differences between movie genres and contexts. Figure 6.1 shows the average movie rating value computed over the considered time contexts, globally and per movie genre. As shown in the figure, there are notable differences in the average rating values between different contexts. These results show that time context information has an impact on user preferences in the movie domain, and thus, can be useful in the rating prediction task, as analyzed in Section 6.3.

[23] Some studies, e.g. (Said et al., 2011), have found that users tend to rate items nearly after their usage/consumption. However, this is not necessarily true for all users. Furthermore, this can affect the time context information reliability, particularly of those time contexts involving short timespans, e.g. period of the day.

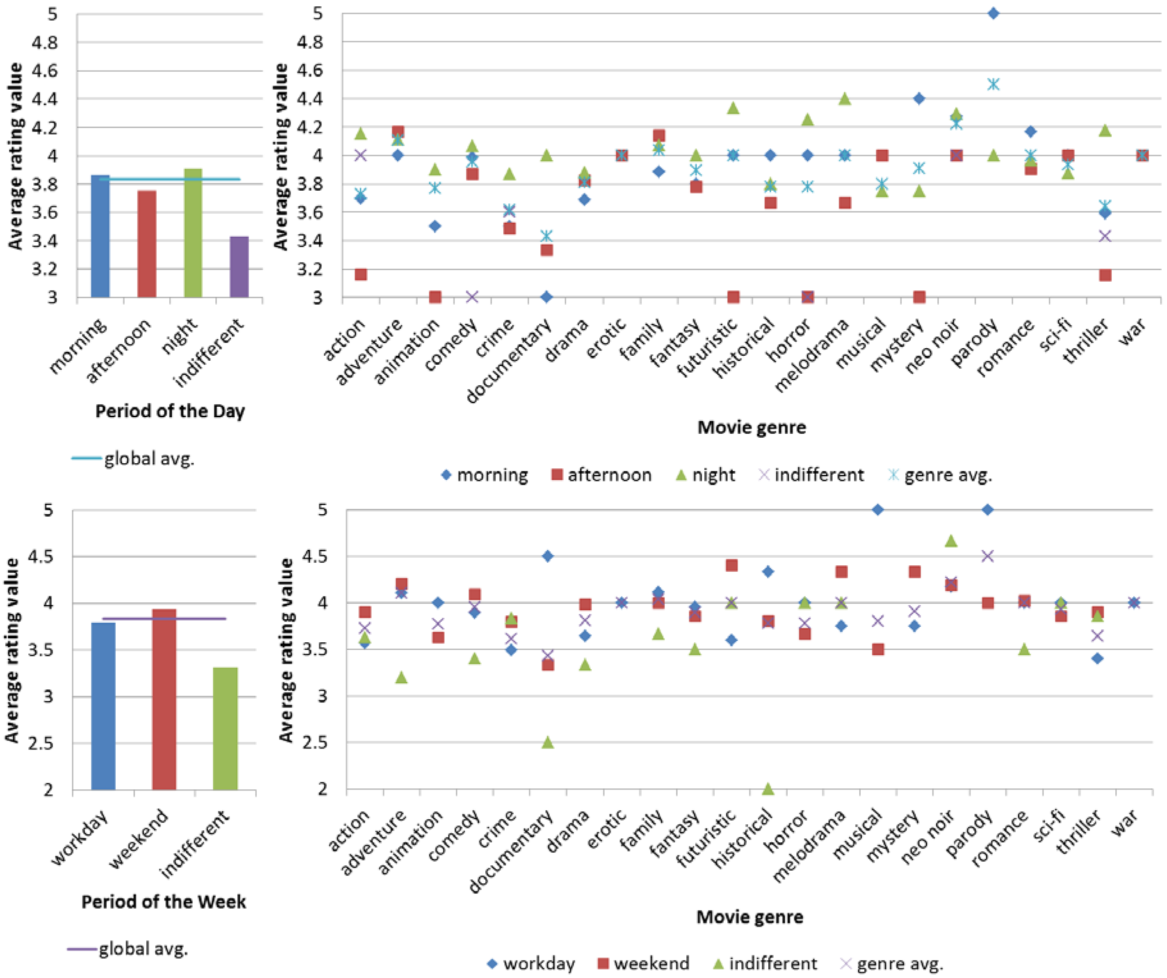


Figure 6.1. Average movie rating values computed over different time contexts and movie genres on the collected context-enriched dataset.

6.2 Evaluated context-aware recommendation approaches

In this section we describe the evaluated recommendation algorithms. Since the collected dataset only contains categorical time context variables, we focused on categorical TARS. As described in Chapter 3, most categorical TARS are special cases of the more general pre-filtering, post-filtering, and contextual modeling strategies for context-aware recommendation. Hence, we evaluated algorithms implementing each of these approaches. Furthermore, we proposed novel heuristics aimed to improve the way the studied approaches exploit context information.

6.2.1 Pre-filtering approaches

In the pre-filtering case, we used the context-aware strategy suggested by Adomavicius et al. (2005), and the Item Splitting technique proposed by Baltrunas and Ricci (2009a, 2009b, 2013).

As explained in Section 3.3.2, the pre-filtering approach (PRF) uses only ratings relevant to the target context to compute rating predictions with a context-unaware recommendation technique (Adomavicius and Tuzhilin, 2011). In our study, we used the k-nearest neighbor, kNN (Herlocker et al., 1999), and the Matrix Factorization, MF (Koren, 2009a) algorithms as base recommendation techniques.

Item Splitting (IS) is a variant of context pre-filtering. This method splits user preference data for items according to the context in which such data were generated, in cases where there are significant differences in the user preferences received by items among contexts. In order to determine whether such differences are significant an impurity criterion is used. When an item i is split, two new (artificial) items are created, i_{c_a} and i_{c_b}, and each of them are assigned to a subset of the users' preferences from the original item, according to the associated context value. Thus, one of these new items corresponds with the preferences generated on one contextual condition, that is, $i_{c_a} = \{r_{\cdot,i,c_a} | r_{\cdot,i,c_a} \in M\}$, and the other item corresponds with the remainder preferences for the original item, i.e., $i_{c_b} = \{r_{\cdot,i,\cdot} | r_{\cdot,i,\cdot} \in M\} \backslash i_{c_a}$. The original item is removed from the dataset, and afterwards, any non-contextualized recommendation technique is performed on the modified dataset.

In order to decide whether or not to split the set of ratings given to an item i, we utilized several impurity criteria, based on Baltrunas and Ricci's findings (Baltrunas and Ricci, 2013). Additionally, we proposed a new impurity criterion – $ic_F(i,s)$– based on the Fisher's exact test (Fisher, 1922). An impurity criterion $ic(i,s)$ returns a score of the differences between the ratings given to an item i in a split $s \in S$, where S represents the set of possible contextual splits. For instance, if there are three contextual values c_a, c_b and c_c, then $S = \{(c_a, c_b \cup c_c), (c_b, c_a \cup c_c), (c_c, c_a \cup c_b)\}$.

More specifically, we consider the three commonly used $ic_{IG}(i,s)$, $ic_M(i,s)$, $ic_P(i,s)$ criteria, and propose a new criterion $ic_F(i,s)$, which are defined as follows.

- $ic_{IG}(i,s)$ impurity criterion is based in the measurement of the information gain – also referred to as Kullback-Leibler divergence (Kullback and Leibler, 1951)– given by s to the knowledge of item i rating:

  $$ic_{IG}(i,s) = H(i) - H(i_{c_a})P_{i_{c_a}} + H(i_{c_b})P_{i_{c_b}}$$

  where $H(i)$ is the entropy of the item i rating value distribution and $P(i_c)$ is the proportion of ratings that i_c receives from item i.

- $ic_M(i,s)$ impurity criterion estimates the statistical significance of the difference in the means of ratings associated to each context in s using the t-test:

$$ic_M(i,s) = \left| \frac{\mu_{i_{c_a}} - \mu_{i_{c_b}}}{\sqrt{\sigma^2_{i_{c_a}}/n_{i_{c_a}} + \sigma^2_{i_{c_b}}/n_{i_{c_b}}}} \right|$$

where μ_{i_c} is the mean rating value of item i_c, $\sigma^2_{i_c}$ is the rating value variance of item i_c and n_{i_c} is the number of ratings given to item i_c.

- $ic_P(i,s)$ impurity criterion estimates the statistical significance of the difference between the proportion of high and low ratings in each context of s using the two-proportion z-test (in the case of the used dataset, ratings 4 and 5 are considered high):

$$ic_P(i,s) = \frac{P^h_{i_{c_a}} - P^h_{i_{c_b}}}{\sqrt{P(1-P)\left(1/n_{i_{c_a}} + 1/n_{i_{c_b}}\right)}}$$

where $P = \left(P^h_{i_{c_a}} n_{i_{c_a}} + P^h_{i_{c_b}} n_{i_{c_b}}\right) / \left(n_{i_{c_a}} + n_{i_{c_b}}\right)$ and $P^h_{i_c}$ is the proportion of high ratings in i_c.

- $ic_F(i,s)$ impurity criterion estimates the statistical significance of the difference between the proportion of low and high ratings in each context of s using the Fisher's exact test:

$$\begin{aligned} ic_F(i,s) \quad &= \quad \frac{\binom{P^h_{i_{c_a}} + P^h_{i_{c_b}}}{P^h_{i_{c_a}}}\binom{P^l_{i_{c_a}} + P^l_{i_{c_b}}}{P^l_{i_{c_a}}}}{\binom{n}{P^h_{i_{c_a}} + P^l_{i_{c_a}}}} \\ &= \quad \frac{\left(P^h_{i_{c_a}} + P^h_{i_{c_b}}\right)! + \left(P^l_{i_{c_a}} + P^l_{i_{c_b}}\right)! + \left(P^h_{i_{c_a}} + P^l_{i_{c_a}}\right)! + \left(P^h_{i_{c_b}} + P^l_{i_{c_b}}\right)}{\left(P^h_{i_{c_a}}\right)! + \left(P^h_{i_{c_b}}\right)! + \left(P^l_{i_{c_a}}\right)! + \left(P^l_{i_{c_b}}\right)! + n!} \end{aligned}$$

where $P^l_{i_c}$ is the proportion of low ratings in i_c and n is the total number of ratings given to i.

A set of item ratings is split if the corresponding criterion returns a score above certain threshold. If several splits obtain a score above the threshold, the split with the highest score is chosen. Note that by using this heuristic, when more than one context variable is used for splitting (e.g. *time of the day* and *period of the week*), the impurity score lets dynamically select the best context variable for performing the split of a given item, i.e., the one that maximizes the differences in item rating patterns among contextual conditions. As in PRF, we used the kNN and MF algorithms separately as base recommendation techniques to be applied after IS.

6.2.2 Post-filtering approaches

In the post-filtering case, rating predictions are first generated by a context-unaware algorithm, and then the predictions are contextualized according to the target context. We used the kNN and MF rating prediction algorithms used with pre-filtering approaches. In order to contextualize the rating predictions generated by kNN, we performed the filtering heuristic presented by Panniello et al. (2009b), denoted as POF. In order to contextualize recommendations generated by MF, we proposed a novel heuristic based in the probability of rating the target item in the target context, denoted as POF-MF.

The contextualization of kNN rating predictions was performed by the POF filtering strategy, which penalizes the recommendation of items that are not relevant in the target context as follows. The relevance of an item i for the target user u in a particular context c is approximated by the probability $p_c(u,i,c) = \frac{|U_{u,i,c}|}{k}$, where k is the number of neighbors used by kNN, and $U_{u,i,c} = \{v \in N(u) | r_{v,i,c} \neq \emptyset\}$ is the users in the neighborhood of u, $N(u)$, who have rated/consumed the item i in the context c. The item relevance is determined by a threshold value τ_{p_c} set to 0.1 in our experiments –based on findings of Panniello et al. (2009b)– that is used to contextualize the ratings as:

$$F(u,i,c) = \begin{cases} F(u,i) & \text{if} \quad p_c(u,i,c) \geq \tau_{p_c} \\ F(u,i) - \Upsilon & \text{if} \quad p_c(u,i,c) < \tau_{p_c} \end{cases}$$

where $F(u,i)$ denotes the context-unaware rating prediction given by a RS, $F(u,i,c)$ denotes the context-aware rating prediction, and Υ is a penalty value. We defined a penalty value of 0.5 instead of assigning the minimum rating value, in order to avoid the introduction of excessive error in the rating prediction task.

In MF the notion of neighbors does not exist. Hence, we proposed a novel heuristic for contextualizing the rating predictions generated by such recommendation algorithm. The heuristic POF-MF is based on the a-priori probability $p_c(i)$ of rating the target item i in the target context c, according to the observed (training) data: $p_c(i) = \frac{n_{i_c}}{n_i}$, where n_{i_c} is the number of ratings given to i in context c, and n_i is the total number of ratings given to item i. We used the same threshold τ_{p_c} and penalty Υ defined above, and MF rating predictions were contextualized by:

$$F(u,i,c) = \begin{cases} F(u,i) & \text{if} \quad p_c(i) \geq \tau_{p_c} \\ F(u,i) - \Upsilon & \text{if} \quad p_c(i) < \tau_{p_c} \end{cases}$$

6.2.3 Contextual modeling approaches

In the contextual modeling case, we adapted and implemented the contextual-neighbors algorithm presented in (Panniello and Gorgoglione, 2012). This algorithm, denoted as CM, is based on the definition of contextualized user profiles $profile(u, c_j)$, which contains user preferences associated to each context value c_i. For instance, if we consider the context *period of the week*, we would have two contextual profiles for each user, one for *workday* and the other for *weekend*.

As noted by Panniello and Gorgoglione (2012), these contextual profiles can be defined in many different ways. In our case, we built the profiles with the ratings associated to each contextual value, that is, $profile(u, c_j) = \left\{r_{u,i,c_j} | r_{u,i,c_j} \in M\right\}$.

Once the contextualized profiles were built, we used all the contextualized profiles in a joint model. In this way, each contextualized profile is exploited as a new user profile. In the original formulation of Panniello and Gorgoglione (2012), a kNN algorithm is used afterwards to select a number of nearest neighbors among these contextualized profiles, by means of different strategies to constraint the profiles eligible as neighbors. In our case, we treated each contextualized profile as an independent user profile without constraints, letting different recommendation algorithms exploit such profiles. We used kNN and MF as underlying recommendation techniques.

We note that CM can be viewed as a type of pre-filtering. Nonetheless, we note that no ratings are discarded for rating computation (as it is the case of pre-filtering), and no contextualization of computed ratings is required (as it is the case of post-filtering)[24].

6.3 Evaluating rating prediction

In this section we describe the empirical results obtained on the rating prediction task. We begin by describing the experimental setting, and then present the results in two subsections. First, we present an analysis on the impact of threshold values in applying the item splitting method. And then, we present a comparison across all the implemented methods. In the comparison we aim to i) determine the best performing approaches, and detect whether there is an overall best contextualization approach; ii) identify the most informative time context signal in terms of performance; and iii) observe if the increased sparsity of the data, due to the additional dimension of context information, affects the approaches capacity to generate recommendations. The latter is done by measuring the proportion of predictions computed by an approach from the total number of test ratings.

[24] Following this reasoning, Item Splitting can also be classified as a type of contextual modeling because no rating data are discarded previous to recommendation computation. Nonetheless, here we follow the classification given in (Baltrunas and Ricci, 2009a, 2009b, 2013).

6.3.1 Experimental setting

The approaches evaluated in this chapter require an underlying recommendation algorithm for producing rating predictions. We used kNN and MF implementations provided by the Apache Mahout project[25], with $k = 30$ and the Pearson's Correlation for kNN, and 60 factors for the MF algorithm. Best parameter values could be obtained for particular tasks and time context signals, but we used the same settings across experiments to avoid differences not due to the contextualization approach. To obtain full coverage, in cases where an algorithm was unable to compute a rating prediction, the global average training rating value was provided as default prediction.

The implemented approaches were evaluated using the data collected in the user study presented in Section 6.1. Aiming to ensure a rigorous and reproducible evaluation setting, we used the methodological framework introduced in Chapter 4 for describing the evaluation conditions, and applied the guidelines proposed in Chapter 5. Since the used dataset does not count with timestamps, we only were able to employ a time-independent rating order condition $\mathscr{o}_{ti}$. We used a community-centered base rating set $\mathscr{b}_{cc}$, and a proportion-based size condition $\mathscr{s}_{prop}$. Given the small size of the dataset, to avoid biases in the results, we used a cross-validation method compatible with the above conditions. We thus performed 10-fold cross validation in all the experiments.

We computed the accuracy of the evaluated approaches in terms of the error on rating prediction, by means of the Mean Average Error, MAE (see Eq. 2.5), and the Root Mean Squared Error, RMSE (see Eq. 2.6). As we noted before, in some cases certain algorithms were unable to generate a rating prediction, due e.g. to lack of knowledge about user preferences in a given context. In order to provide a more complete perspective of the performance of the considered approaches, we also computed the proportion of predictions effectively computed (denoted as PPEC), that is, the proportion of predictions computed by each algorithm from the total predictions required (the remaining correspond to default value predictions). Moreover, we report the MAE and RMSE values obtained in effectively computed predictions, which we denote as EC-MAE and EC-RMSE respectively.

6.3.2 Selecting thresholds for Item Splitting

As described in Section 6.2.1, Item Splitting requires an impurity criterion $ic(i, s)$ that returns a score of the differences between the ratings given to an item i in a split s. The item i is split if the used criterion returns a score above certain threshold τ. In order to determine the best threshold values for the different criteria used in this analysis, we computed the RMSE values obtained by using different thresholds on each contextual condition. Figure 6.2 shows the obtained results.

[25] Apache Mahout machine learning library, http://mahout.apache.org/

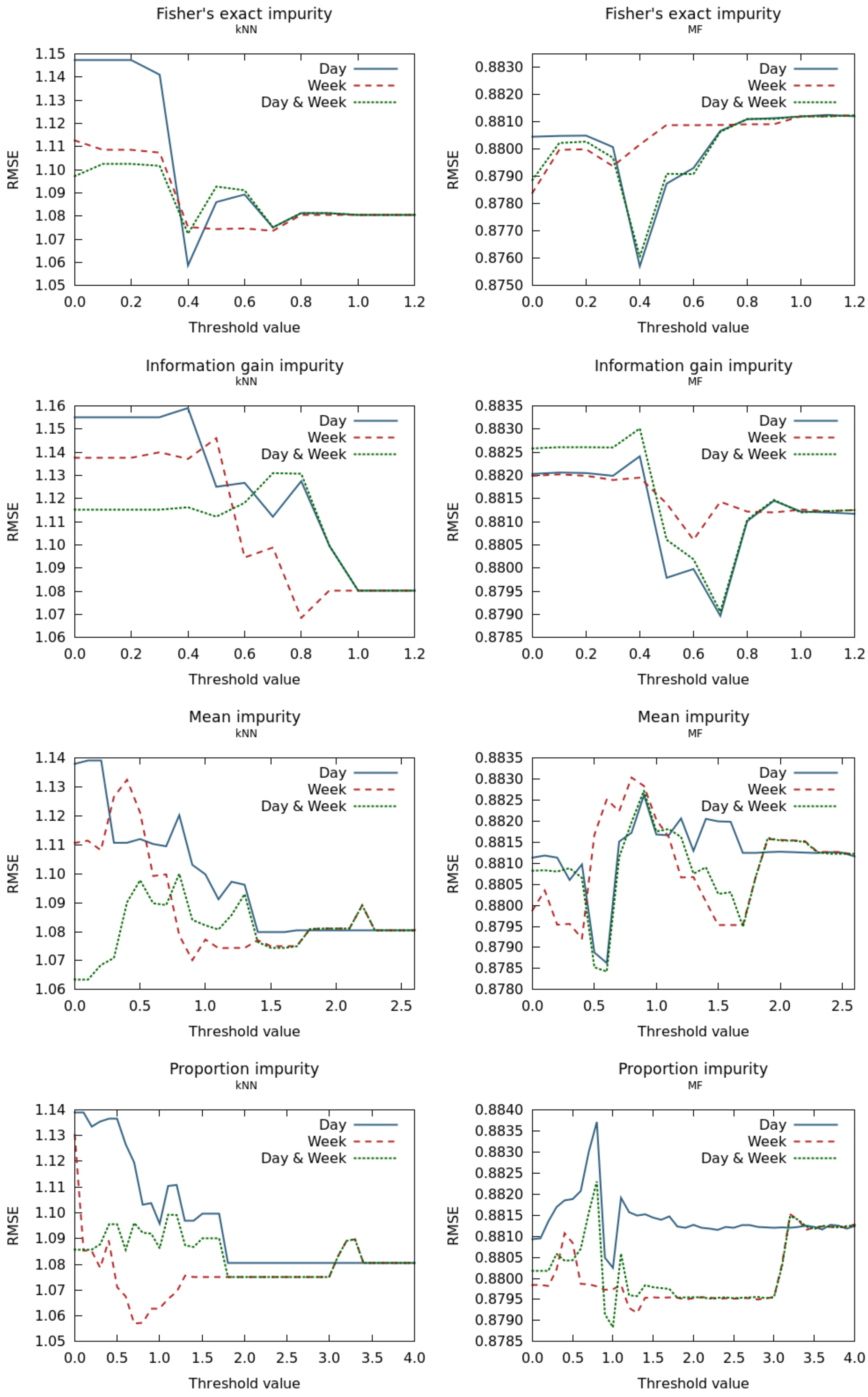


Figure 6.2. Threshold value vs. RMSE obtained by the different impurity criteria and recommendation approach tested.

For all the impurity criteria, we tested thresholds in the range [0,4.5] with increments of 0.1. We note that, as the threshold value becomes higher, a criterion becomes less sensitive, and finally no item is split. We cut the graphs in the figure at the threshold for which no item was split.

The figure shows that each impurity criterion meets the lowest RMSE at a different threshold, depending on the time context information used to split the items, and the underlying recommendation approach (kNN or MF). These results reveal that Item Splitting is able to exploit differences in global item preferences across time contexts, but a careful selection of the threshold is required in order to obtain performance improvements.

6.3.3 Experimental results

Tables 6.1 and 6.2 show the results obtained by each of the tested CARS approaches on our context-enriched dataset, using kNN and MF as underlying recommendation algorithm respectively. The results are grouped according to the time context information provided to each recommendation approach. In the case of IS approaches, we report the results obtained with the best performing thresholds for each combination of impurity criterion and time context. These thresholds are reported in the tables.

In order to put in perspective the performance of the contextualization approaches, we included as baselines the basic recommendation algorithms, namely kNN and MF without contextualization, using the same parameter values as in the other approaches. The results of the baseline algorithms are in accordance with those reported in previous studies: MF has a superior performance compared to kNN on all the analyzed metrics, and MF provides a higher proportion of personalized rating predictions. This is due to the structure of MF, which builds a model of latent factors for all the users and items in the training set at the same time. The non-personalized predictions correspond to target users or items not present in the training set. In the case of kNN, it is required to find some similar users (to the target user) who have rated the target item, which is not possible in many cases.

These important differences motivated us to analyze results separately for implementations using kNN and MF. It let us study the improvements due to the contextualization approaches. Moreover, it let us observe if some approaches are able to improve performance independently of the underlying recommendation algorithm.

In Table 6.1 we observe that, being kNN the underlying recommendation algorithm, the best performing approaches according to MAE and RMSE are PRF and CM. In fact, the best global MAE and RMSE values are obtained by PRF exploiting the *period of the day* time context, individually or in conjunction with *period of the week*. CM is also able to improve considerably MAE and RMSE values when *period of the day* context is available.

In the case of IS and POF, only slight performance improvements were obtained, regardless of the time context signal exploited. Moreover, MAE and RMSE values from POF were worse than those of the baseline when exploiting *period of the week* context.

Observing the PPEC, we note that PRF and CM heavily penalize the ability of algorithms to compute rating predictions. This is due to the interaction of these filtering techniques and kNN requisites. In the case of PRF, many ratings were discarded previous to rating prediction computation, which makes it harder to find neighbors having rated the target item. In the case of CM, no rating was discarded, but user profiles were partitioned into contextualized profiles, leaving less rated items in each contextualized profile, and making it more difficult to find neighboring contextualized profiles with the target item.

Table 6.1. Performance values in the rating prediction task obtained by pre-filtering, post-filtering, and contextual modeling-based recommender systems using kNN as underlying recommendation algorithm. Global top values of each column are in bold, and best values for each context are underlined. Green-up arrow heads, yellow lines and red-down arrow heads indicate better, equal, and worse values of the metric in the column with respect to the baseline, respectively.

Context	*Approach*	*RMSE*	*MAE*	*PPEC*	*EC-RMSE*	*EC-MAE*
	Baseline(kNN)	1.0804	0.8038	0.3028	1.3768	1.0396
Period of the day	PRF	▲ **<u>0.9781</u>**	▲ **<u>0.7510</u>**	▼ 0.0147	▲ **<u>0.7083</u>**	▲ **<u>0.7017</u>**
	IS(ic_F, τ = 0.4)	▲ 1.0587	▲ 0.7934	▼ 0.2612	▲ 1.3141	▲ 0.9941
	IS(ic_{IG}, τ = 1.0)	▬ 1.0804	▬ 0.8038	▬ **<u>0.3028</u>**	▬ 1.3768	▬ 1.0396
	IS(ic_M, τ = 1.5)	▲ 1.0799	▲ 0.8027	▬ **<u>0.3028</u>**	▲ 1.3752	▲ 1.0355
	IS(ic_P, τ = 2.0)	▬ 1.0804	▬ 0.8038	▬ **<u>0.3028</u>**	▬ 1.3768	▬ 1.0396
	POF	▲ 1.0782	▲ 0.8006	▼ 0.3028	▲ 1.3730	▲ 1.0291
	CM	▲ 1.0106	▲ 0.7751	▼ 0.0458	▲ 1.2337	▼ 1.0802
Period of the week	PRF	▲ <u>0.9963</u>	▲ <u>0.7513</u>	▼ 0.1095	▲ <u>1.2050</u>	▲ <u>0.9325</u>
	IS(ic_F, τ = 0.7)	▲ 1.0736	▲ 0.8001	▼ 0.3008	▲ 1.3615	▲ 1.0252
	IS(ic_{IG}, τ = 0.8)	▲ 1.0685	▲ 0.7954	▼ 0.2967	▲ 1.3522	▲ 1.0165
	IS(ic_M, τ = 0.9)	▲ 1.0702	▲ 0.7927	▼ 0.2823	▲ 1.3695	▲ 1.0225
	IS(ic_P, τ = 0.7)	▲ 1.0570	▲ 0.7862	▼ 0.2885	▲ 1.3306	▲ 0.9956
	POF	▼ 1.0990	▼ 0.8181	▬ **<u>0.3028</u>**	▼ 1.4215	▼ 1.0848
	CM	▲ 1.0667	▲ 0.8018	▼ 0.1834	▼ 1.4628	▼ 1.2164
Period of the day & period of the week	PRF	▲ **<u>0.9781</u>**	▲ **<u>0.7510</u>**	▼ 0.0147	▲ **<u>0.7083</u>**	▲ **<u>0.7017</u>**
	IS(ic_F, τ = 0.4)	▲ 1.0724	▼ 0.8042	▼ 0.2346	▼ 1.4255	▼ 1.0952
	IS(ic_{IG}, τ = 1.0)	▬ 1.0804	▬ 0.8038	▬ **<u>0.3028</u>**	▬ 1.3768	▬ 1.0396
	IS(ic_M, τ = 0.0)	▲ 1.0636	▲ 0.7910	▼ 0.1823	▲ 1.3170	▲ 0.9953
	IS(ic_P, τ = 2.0)	▲ 1.0750	▲ 0.7938	▬ **<u>0.3028</u>**	▲ 1.3594	▲ 1.0044
	POF	▲ 1.0721	▲ 0.7971	▼ 0.3028	▲ 1.3549	▲ 1.0157
	CM	▲ 1.0119	▲ 0.7771	▼ 0.0413	▲ 1.3170	▼ 1.2098

We note that EC-MAE and EC-RMSE show similar trends to those observed on MAE and RMSE. This is probably due to the fact that the baseline also has a low PPEC, and also shows that using the average rating value as default rating prediction does not harm importantly MAE and RMSE of kNN-based approaches.

Table 6.2. Performance values in the rating prediction task obtained by pre-filtering, post-filtering, and contextual modeling-based recommender systems using MF as underlying recommendation algorithm. Global top values of each column are in bold, and best values for each context are underlined. Green-up arrow heads, yellow lines and red-down arrow heads indicate better, equal, and worse values of the metric in the column with respect to the baseline, respectively.

Context	*Approach*	*RMSE*	*MAE*	*PPEC*	*EC-RMSE*	*EC-MAE*
	Baseline(MF)	0.8813	0.6946	0.7508	0.8534	0.6855
Period of the day	PRF	▼ 0.8916	▲ 0.6938	▼ 0.5457	▲ **0.7921**	▲ 0.6483
	IS($ic_F, \tau = 0.4$)	▲ **0.8757**	▲ 0.6905	▬ **0.7508**	▲ 0.8458	▲ 0.6800
	IS($ic_{IG}, \tau = 0.7$)	▲ 0.8789	▲ 0.6927	▬ **0.7508**	▲ 0.8499	▲ 0.6826
	IS($ic_M, \tau = 0.6$)	▲ 0.8786	▼ 0.6946	▬ **0.7508**	▲ 0.8497	▲ 0.6853
	IS($ic_P, \tau = 1.0$)	▲ 0.8802	▼ 0.6947	▬ **0.7508**	▲ 0.8515	▲ 0.6851
	POF-MF	▼ 0.8977	▼ 0.7105	▬ **0.7508**	▼ 0.8740	▼ 0.7063
	CM	▼ 0.8841	▲ 0.6901	▼ 0.6879	▲ 0.8374	▲ 0.6663
Period of the week	PRF	▼ 0.8965	▼ 0.6958	▼ 0.5979	▼ 0.8711	▼ 0.6876
	IS($ic_F, \tau = 0.0$)	▲ 0.8784	▲ 0.6906	▬ **0.7508**	▲ 0.8497	▲ 0.6803
	IS($ic_{IG}, \tau = 0.6$)	▲ 0.8806	▲ 0.6936	▬ **0.7508**	▲ 0.8523	▲ 0.6841
	IS($ic_M, \tau = 0.4$)	▲ 0.8792	▲ 0.6923	▬ **0.7508**	▲ 0.8508	▲ 0.6826
	IS($ic_P, \tau = 1.3$)	▲ 0.8792	▲ 0.6919	▬ **0.7508**	▲ 0.8506	▲ 0.6818
	POF-MF	▼ 0.8993	▼ 0.7109	▬ **0.7508**	▼ 0.8775	▼ 0.7071
	CM	▼ 0.8925	▼ 0.7020	▼ 0.7074	▼ 0.8568	▲ 0.6825
Period of the day & period of the week	PRF	▼ 0.9130	▼ 0.6969	▼ 0.4378	▲ 0.8000	▲ **0.6399**
	IS($ic_F, \tau = 0.4$)	▲ 0.8761	▲ **0.6893**	▬ **0.7508**	▲ 0.8463	▲ 0.6785
	IS($ic_{IG}, \tau = 0.7$)	▲ 0.8790	▲ 0.6923	▬ **0.7508**	▲ 0.8500	▲ 0.6821
	IS($ic_M, \tau = 0.6$)	▲ 0.8784	▲ 0.6926	▬ **0.7508**	▲ 0.8495	▲ 0.6827
	IS($ic_P, \tau = 1.0$)	▲ 0.8789	▲ 0.6924	▬ **0.7508**	▲ 0.8498	▲ 0.6820
	POF-MF	▼ 0.9107	▼ 0.7194	▬ **0.7508**	▼ 0.8907	▼ 0.7177
	CM	▼ 0.9016	▼ 0.6996	▼ 0.6167	▲ 0.8225	▲ 0.6519

In the case of the approaches that use MF as underlying recommendation algorithm (Table 6.2), the best performing approach according to MAE and RMSE was IS. The best global RMSE value was obtained by IS using the proposed ic_F impurity criterion and exploiting the *period of the day* time context, while the best global MAE was obtained by the same algorithm exploiting *period of the day* and *period of the week* time contexts.

Regarding the remaining algorithms, none of them was able to improve either MAE or RMSE values of the baseline, regardless of the time context signal exploited.

Observing the PPEC and the EC-MAE and EC-RMSE values of MF-based approaches, it is possible to better understand the difference of their MAE and RMSE compared with those of the kNN-based approaches. For instance, we note that the best EC-MAE and EC-RMSE values were obtained by PRF exploiting *period of day* alone, and both *period of day* and *period of week* contexts, respectively. These results are very similar to those observed in Table 6.1. However, in both cases, the PPEC is very low compared with the other approaches. In this case, the low number of predictions effectively computed by PRF and CM is worsening the results on MAE and RMSE.

The above results let us observe important clues regarding the application of time-aware approaches to recommendation in the rating prediction task. First, **we did not observe a unique superior TARS approach for improving rating prediction performance**. We observed that performance improvements have a strong dependency with the used recommendation algorithm. In general PRF provided the best performance values when using kNN, while IS had most of the improvements when using MF as underlying recommendation algorithm, particularly using the proposed ic_F impurity criterion. Second, the **period of the day** context, used individually or in conjunction with other time context, was **the most informative time context** in terms of rating prediction error, particularly in the case of RMSE. Finally, we note that **the final rating prediction performance also depends on the proportion of predictions effectively computed** by each recommendation algorithm and contextualization approach, and the default rating value used.

6.4 Evaluating top-N recommendations

In this section we focus on the evaluation of top-N recommendations task. We first analyze the additional evaluation conditions required for the assessment of such task. Then, from this analysis we propose a new methodology in terms of a new target item condition within our methodological framework, aiming to provide a more realistic evaluation setting for the task. Finally, we describe the used experimental setting, and report and discuss the results obtained when comparing the different recommendation approaches. In this comparison – similarly to the analysis performed in Section 6.3–, we focus on i) determining the best performing approaches, and detecting whether there is an overall best contextualization approach; and ii) identifying the most informative time context signal in terms of recommendation performance.

6.4.1 Evaluation conditions for time-aware top-N recommendations

To compute top-N recommendations metrics, we first require defining the target item and relevant item conditions, which indicate the items that will be ranked, and the items in the test set that will be considered as relevant (and non-relevant) for computing relevant-based metrics, respectively.

The selection of these conditions is not trivial. As indicated by Cremonesi et al. (2010), a careful construction of the test set is required. Moreover, the evaluation of time-aware recommendations requires special care, as the rating prediction is computed not only for a target user and item, but also for a target time context that must be taken into account by the system to generate contextualized recommendations.

In the case of the recommendation approaches evaluated in this chapter, the target *period of the day* and *period of the week* categorical contexts are known by a RS prior to its rating prediction computation. To evaluate categorical TARS –and generic CARS exploiting categorical context variables–, the most common approach is to utilize a user-based target item condition and a threshold-based relevant item condition. That is, to rank the items in the user's test set –for which the user's ratings are known–, by considering as relevant those items rated above some predefined threshold. By using such target item condition, a recommender can receive the target time context for which compute a given rating prediction in the test set.

However, as discussed in (Koren, 2008; Cremonesi et al., 2010; Bellogín et al., 2011), this evaluation setting completely miss any assessment on unrated items, a situation far away of the reality, where all items in the system's catalog should be eligible for recommendation. Hence, a more realistic setting should include unrated items in the set of target items. This motivated our suggestion (guideline number 5 in Chapter 5) of using a community-based target item condition for evaluating TARS. The community-based target item condition forces to rank all items in the test set for each target user (of course, with the exception of those items rated by the target user in the training set).

A problem that arises from the community-based target item condition is the selection of the target time context in which compute rating predictions for unrated items. The simplest way to address that problem is to randomly assign time contexts to each unrated item. Such strategy, nonetheless, forces to combine rating predictions targeted to different time contexts into a unique ranking. It is likely (and expectable) that an item may get different rating predictions for different contexts, and thus, combining items targeted to different contexts may make it difficult to estimate the ranking position a relevant item should have.

Alternatively, we aim to use a target item condition that lets rank relevant and non-relevant items targeted to the same context, and include unrated items at the same time.

Revising the different target item conditions included in the methodological framework presented in Chapter 4, we observe that none of them achieves both goals. However, we note that a one-plus random (OPR) target item condition can be easily adapted for this purpose.

The original formulation of OPR states that for each highly relevant item in the test set (among those rated by the user), a number k of unrated items is randomly selected. Hence, a ranking is computed for each of these sets, which are composed of one relevant item and k non-relevant items. We propose a simple modification on this condition, called *contextual one-plus random target item condition*, in which the context of the relevant item is used as target context for all the items in the corresponding set. In this way, all items in each ranking are targeted to the same context. The formal definition of this condition is as follows:

Contextual one-plus random target items. Let I_{hrel_u} be the set of highly relevant items for user u defined as $I_{hrel_u} = \{i \in I \mid r_{u,i} \in Te_u, r_{u,i} > \tau_{hrel}\}$, where τ_{hrel} is a high-relevance threshold, i.e., the items in the test set of u, Te_u, that have high ratings; and let $I_{\overline{rel}_u}$ be the set of non-relevant items for user u defined as $I_{\overline{rel}_u} = \{i \in I \mid r_{u,i} = \emptyset\}$, i.e., the items that have not been rated by u. For each item $i^k \in I_{hrel_u}$, a set $Target_u^k$ is built as the union of i^k and a number N of non-relevant items randomly selected from $I_{\overline{rel}_u}$. All items in $Target_u^k$ are assigned with the time context of i^k.

The usage of this target item condition implicitly forces to use a threshold-based relevant item condition because the value τ_{hrel} establishes the threshold for interpreting an item as relevant or not.

6.4.2 Experimental setting

We used the same recommendation algorithms and contextualization approaches evaluated in Section 6.3, and the thresholds that lead to the best RMSE values for the impurity criteria of Item Splitting (see Tables 6.1 and 6.2), in order to compare the same recommendation models in both tasks.

Moreover, we applied the same evaluation conditions for training-test split, that is, we performed 10-fold cross validation in all the experiments, corresponding to the use of a time-independent rating order condition $\mathcal{o}_{ti}$, a community-centered base rating set condition $\mathcal{b}_{cc}$, and a proportion-based size condition $\mathcal{s}_{prop}$.

Regarding the specific top-N recommendations evaluation conditions, we performed the proposed contextual one-plus random target item condition, and a threshold-based relevant condition with a threshold $\tau_{hrel} = 5.0$. The number n of randomly selected non-relevant items to rank with each relevant item was set to $n = 10$. We did not use larger values of n due to the small sample size of the dataset.

We computed the accuracy of the evaluated approaches using Precision (see Eq. 2.7), Recall (see Eq. 2.8), and F-measure (see Eq. 2.9) metrics at level 5 (denoted as P@5, R@5 and F@5, respectively), in order to assess the ability of the approaches to rank the relevant items among their 5 top recommendations. We used the normalized Discounted Cumulative Gain metric (nDCG, see Eq. 2.10) and the Area Under the Curve (AUC, see Eq. 2.11) on the full list of recommendations, in order to assess the whole rankings generated.

6.4.3 Experimental results

Tables 6.3 and 6.4 show the results obtained by the tested recommendation approaches using kNN and MF as underlying recommendation algorithms, respectively. The results are grouped according to the time context information provided to each approach. The specific thresholds for Item Splitting are reported in the tables.

In the tables we also show the results obtained by kNN and MF algorithms as baselines. We observe that MF has a superior performance compared to kNN –which is in accordance with results on rating prediction, and with other studies. We also observe that P@5 –and consequently F@5– values may be considered low. This is due to the followed evaluation methodology: only one relevant item is included in each recommendation list. Thus, the maximum achievable P@5 is 0.2. These low values are not inconvenient for evaluation, as the metric results are used to rank algorithms in terms of performance, and thus, the absolute numeric metric values are not informative by themselves.

In Table 6.3 we observe that, using kNN as underlying recommendation algorithm, the best P@5, R@5 and F@5 values are obtained by PRF. The best global value of P@5 is obtained by PRF exploiting the *period of the day* time context, individually or in conjunction with *period of the week*. In the case of R@5, the best global value is obtained by PRF exploiting *period of the week* time context. In the case of F@5, the best global value is obtained by PRF exploiting either of the time context variables individually or simultaneously.

CM is also able to improve these metrics with respect to the baseline, in particular when exploiting both time contexts in conjunction. POF is only able to improve the baseline's performance on these metrics when exploiting *period of the day* time context individually. Contrarily, IS provides a superior performance with respect to the baseline when exploiting *time of the week* context, individually or in conjunction with *time of the day*. We note that the best results from IS in P@5, R@5 and F@5 are obtained when exploiting both time contexts in conjunction, using the proposed ic_F impurity criterion.

Regarding nDCG and AUC metrics, the best global values are obtained by IS using ic_P and PRF respectively, when exploiting *period of the week* time context. In general, most contextualization approaches are unable to improve the baseline's performance when exploiting the *period of the day* time context alone.

Table 6.3. Performance values in the top-N recommendations task obtained by pre-filtering, post-filtering, and contextual modeling-based recommender systems using kNN as underlying recommendation algorithm. Global top values of each column are in bold, and best values for each context are underlined. Green-up arrow heads, yellow lines and red-down arrow heads indicate better, equal, and worse values of the metric in the column with respect to the baseline, respectively.

Context	*Approach*	*P@05*	*R@05*	*F@05*	*nDCG*	*AUC*
	Baseline(kNN)	0.0805	0.2897	0.1215	0.3665	0.4877
Period of the day	PRF	▲ **0.0933**	▲ 0.3261	▲ 0.1390	▼ 0.3463	▲ 0.5206
	IS($ic_F, \tau = 0.4$)	▼ 0.0764	▼ 0.2684	▼ 0.1145	▼ 0.3566	▼ 0.4752
	IS($ic_{IG}, \tau = 1.0$)	▬ 0.0805	▬ 0.2897	▬ 0.1215	▬ 0.3665	▬ 0.4877
	IS($ic_M, \tau = 1.5$)	▬ 0.0805	▬ 0.2897	▬ 0.1215	▼ 0.3613	▼ 0.4852
	IS($ic_P, \tau = 2.0$)	▬ 0.0805	▬ 0.2897	▬ 0.1215	▬ 0.3665	▬ 0.4877
	POF	▲ 0.0839	▲ 0.3064	▲ 0.1271	▼ 0.3653	▲ 0.4897
	CM	▲ 0.0899	▲ 0.3125	▲ 0.1336	▼ 0.3479	▲ 0.5154
Period of the week	PRF	▲ 0.0917	▲ **0.3383**	▲ **0.1390**	▼ 0.3598	▲ **0.5294**
	IS($ic_F, \tau = 0.7$)	▲ 0.0841	▲ 0.3033	▲ 0.1271	▲ 0.3680	▲ 0.4959
	IS($ic_{IG}, \tau = 0.8$)	▲ 0.0820	▲ 0.3023	▲ 0.1245	▲ 0.3697	▲ 0.4898
	IS($ic_M, \tau = 0.9$)	▲ 0.0838	▲ 0.2933	▲ 0.1257	▲ 0.3745	▲ 0.5055
	IS($ic_P, \tau = 0.7$)	▲ 0.0865	▲ 0.3061	▲ 0.1297	▲ **0.3764**	▲ 0.4993
	POF	▼ 0.0789	▲ 0.2914	▼ 0.1199	▼ 0.3646	▼ 0.4780
	CM	▲ 0.0876	▲ 0.3140	▲ 0.1316	▼ 0.3449	▲ 0.5016
Period of the day & period of the week	PRF	▲ **0.0933**	▲ 0.3261	▲ 0.1390	▼ 0.3463	▲ 0.5206
	IS($ic_F, \tau = 0.4$)	▲ 0.0909	▲ 0.3260	▲ 0.1374	▼ 0.3656	▲ 0.5182
	IS($ic_{IG}, \tau = 1.0$)	▬ 0.0805	▬ 0.2897	▬ 0.1215	▬ 0.3665	▬ 0.4877
	IS($ic_M, \tau = 0.0$)	▲ 0.0906	▲ 0.3093	▲ 0.1345	▲ 0.3725	▲ 0.5201
	IS($ic_P, \tau = 2.0$)	▲ 0.0837	▲ 0.2927	▲ 0.1255	▲ 0.3747	▲ 0.4975
	POF	▬ 0.0805	▬ 0.2897	▬ 0.1215	▼ 0.3664	▲ 0.4935
	CM	▲ 0.0919	▲ 0.3225	▲ 0.1370	▼ 0.3461	▲ 0.5134

In the case of approaches that use MF as underlying recommendation algorithm (Table 6.4), we observe that the best performing approach in terms of P@5, R@5 and F@5 is our proposed POF-MF approach, on all the context signals. The best global values of these metrics are obtained by the approach when exploiting *period of the day* and *period of the week* time context in conjunction. The remaining approaches are unable to improve the baseline performance on all these metrics simultaneously.

Regarding nDCG, the best results are obtained by PRF, particularly when exploiting *period of the day* time context. In the case of AUC, the best results are obtained by POF-MF. The best global values on nDCG and AUC are obtained by PRF and POF-MF respectively, when exploiting *period of the day* and *period of the week* time contexts simultaneously. The remainder algorithms show improvements over the baseline performance on these metrics when exploiting *period of the week* time context, with the exception of CM.

Table 6.4. Performance values in the top-N recommendations task obtained by pre-filtering, post-filtering, and contextual modeling-based recommender systems using MF as underlying recommendation algorithm. Global top values of each column are in bold, and best values for each context are underlined. Green-up arrow heads, yellow lines and red-down arrow heads indicate better, equal, and worse values of the metric in the column with respect to the baseline, respectively.

Context	*Approach*	*P@05*	*R@05*	*F@05*	*nDCG*	*AUC*
	Baseline(MF)	0.1238	0.4748	0.1898	0.4245	0.6227
Period of the day	PRF	▲ 0.1247	▼ 0.4549	▼ 0.1886	▲ 0.4453	▲ 0.6404
	IS($ic_F, \tau = 0.4$)	▼ 0.1236	▼ 0.4694	▼ 0.1890	▼ 0.4217	▼ 0.6180
	IS($ic_{IG}, \tau = 0.7$)	▼ 0.1216	▼ 0.4493	▼ 0.1846	▼ 0.4180	▼ 0.6139
	IS($ic_M, \tau = 0.6$)	▼ 0.1187	▼ 0.4467	▼ 0.1810	▼ 0.4174	▼ 0.6050
	IS($ic_P, \tau = 1.0$)	▼ 0.1226	▼ 0.4590	▼ 0.1868	▼ 0.4198	▼ 0.6194
	POF-MF	▲ 0.1360	▲ 0.5201	▲ 0.2084	▲ 0.4452	▲ 0.6722
	CM	▼ 0.1232	▲ 0.4771	▼ 0.1895	▲ 0.4295	▲ 0.6231
Period of the week	PRF	▼ 0.1199	▼ 0.4460	▼ 0.1824	▲ 0.4285	▼ 0.6039
	IS($ic_F, \tau = 0.0$)	▼ 0.1201	▼ 0.4623	▼ 0.1851	▲ 0.4367	▲ 0.6313
	IS($ic_{IG}, \tau = 0.6$)	▼ 0.1227	▼ 0.4634	▼ 0.1873	▲ 0.4361	▲ 0.6277
	IS($ic_M, \tau = 0.4$)	▲ 0.1245	▼ 0.4667	▼ 0.1897	▲ 0.4396	▲ 0.6349
	IS($ic_P, \tau = 1.3$)	▲ 0.1276	▲ 0.4794	▲ 0.1947	▲ 0.4285	▲ 0.6356
	POF-MF	▲ 0.1286	▲ 0.4937	▲ 0.1971	▲ 0.4323	▲ 0.6532
	CM	▼ 0.1161	▼ 0.4462	▼ 0.1783	▼ 0.4109	▼ 0.6070
Period of the day & period of the week	PRF	▲ 0.1281	▼ 0.4652	▲ 0.1937	▲ **0.4603**	▲ 0.6581
	IS($ic_F, \tau = 0.4$)	▼ 0.1196	▼ 0.4512	▼ 0.1825	▼ 0.4136	▼ 0.6083
	IS($ic_{IG}, \tau = 0.7$)	▼ 0.1236	▼ 0.4593	▼ 0.1879	▲ 0.4282	▲ 0.6287
	IS($ic_M, \tau = 0.6$)	▼ 0.1119	▼ 0.4211	▼ 0.1712	▼ 0.4203	▼ 0.6204
	IS($ic_P, \tau = 1.0$)	▲ 0.1246	▼ 0.4690	▲ 0.1901	▲ 0.4402	▲ 0.6415
	POF-MF	▲ **0.1394**	▲ **0.5373**	▲ **0.2141**	▲ 0.4516	▲ **0.6774**
	CM	▼ 0.1118	▼ 0.4207	▼ 0.1708	▼ 0.4131	▼ 0.5986

From these results we obtain some important insights regarding the contextualization of top-N recommendations. First, we observe that there is **no dominant TARS approach for improving top-N recommendations performance**. We found a strong dependency between the performance of the contextualization approaches and the used base recommendation algorithm. This finding is similar to the observed in the case of the rating prediction task. In general, PRF provided the best performance values when using kNN, while the proposed POF-MF had most of the improvements when using MF as underlying recommendation algorithm. From these, we note that the contextualization approach to use should be selected depending on the recommendation task and the underlying recommendation algorithm. And second, regarding the use of time context information, ***period of the day* was less informative than *period of the week* time context**, as most recommendation approaches were not able to improve performance when exploiting the former time context. This is contradictory with the results observed in the rating prediction task, where *period of the day* was the most informative time context. These results indicate that the time context to exploit has to be selected carefully depending on the recommendation task and the contextualization approach at hand.

6.5 Conclusions

In this chapter we have evaluated the performance of various time-aware recommendation approaches in two important recommendation tasks, namely rating prediction and top-N recommendations. We used a dataset of movie ratings with explicit time context information, obtained through a user study specifically designed for such aim. We adapted several context-aware recommender systems able to exploit the collected time context information, and proposed new heuristics for different CARS approaches. All these approaches were assessed using a common and reproducible evaluation setting, which was precisely described by means of the methodological framework presented in Chapter 4. Moreover, starting from the guidelines and evaluation conditions for top-N recommendations task presented in Chapter 5, we proposed and used a new methodology for evaluating that task. This methodology let built ranked list of items targeted for the same time context, including unrated items in the list, thus providing a more realistic evaluation setting than those from other methodologies in the literature.

Based on the results reported in this chapter, we may conclude that **there is no unique dominant TARS in either the rating prediction or top-N recommendations task**. Moreover, we observed that performance improvements achieved by the tested contextualization approaches depend on the underlying recommendation algorithm and the exploited time context. This conclusion is in line with findings of previous research comparing CARS on e-commerce applications, e.g. (Panniello et al., 2009b). The identification of the best performing approach, thus, requires a time-consuming evaluation and comparison of candidate TARS implementations on the target data. Furthermore, some contextualization approaches may require an intensive testing of parameters, as is the case of IS.

Regarding the heuristics proposed in the chapter, we remark the performance of IS obtained with the ic_F impurity criterion in the rating prediction task, and the performance of POF-MF in the top-N recommendations task. These two new heuristics are able to **effectively contextualize recommendations** generated by the high-performing MF recommendation algorithm, thus leading to the **best global values** on the majority of metrics in the rating prediction and top-N recommendations tasks, respectively. In this way, the use of the proposed heuristics in conjunction with a Matrix Factorization recommendation algorithm can be considered a good approach to contextualize recommendations when time context information about user preferences is available.

One important remark that may help on the search of better TARS is the fact that **the performance may be considerably affected by the proportion of predictions that are assigned a default value**, particularly in the prediction of ratings. In this chapter we used the average rating value in the dataset as default value, in order to obtain full coverage and avoid biases in the assessment of metrics due to the use of a more sophisticated method. Nonetheless, the usage of a more accurate default rating method may help improve considerably the performance of some approaches, such as PRF.

We note that the conclusions obtained in this analysis are not necessarily generalizable, due to the small size of the used dataset, and the fact that only one recommendation domain was evaluated. Nonetheless, we remark that our analysis provides an objective comparison of approaches based on the utilization of a common and precisely defined evaluation setting. Moreover, we highlight the extensibility of the methodological evaluation framework proposed in Chapter 4, as we could easily incorporate a new target item condition that integrated seamlessly with other conditions, and furthermore, could follow a more realistic evaluation methodology for the evaluation of TARS approaches in the top-N recommendations task.

Chapter 7

Identification of active users in shared user accounts

Popular online rental services such as Netflix and MoviePilot are usually accessed via household accounts. A household account is typically shared by various users who live in the same house, but in general does not provide a mechanism by which current active users are identified, and thus leads to considerable difficulties for making effective personalized recommendations. The identification of the active household members, defined as the discrimination of the users from a given household who are interacting with a system (e.g. an on-demand video service), is thus an interesting challenge for the recommender systems research community. In this chapter, we formulate the above task as a classification problem, and address it by means of methods that only exploit time context information from the users' past activity logs. Moreover, we extend the methodological framework introduced in Chapter 4 for the evaluation of this task. This lets assess the proposed methods' performance using different evaluation methodologies, and properly take into account the evolution of user preferences and behavior through time, from an evaluation point of view.

In Section 7.1 we provide a general definition of the task, and a brief review of related work. In Section 7.2 we present an empirical analysis of the temporal behavior of users in households, performed on a movie rating dataset with household data, and show the suitability of a time-based approach for active user identification. In Section 7.3 we detail the methods used for the task, and report their performance on a publicly available test dataset. In Section 7.4 we present a comparison of the methods by using diverse evaluation methodologies, in order to determine their robustness when using different evaluation settings. Finally, in Section 6.4 we end with partial conclusions from our analysis.

7.1 Identifying users in shared accounts

Many online services providers offer access to their services via user accounts. These accounts can be seen as a mechanism to identify the active user, and track her behavior, letting e.g. build a personalized profile. The user profile can be used afterwards to provide personalized services, e.g. recommendation. User accounts, however, can be shared by multiple users. An example of shared account is a *household account*, that is, an account shared by several users who usually live in the same house. In general, it is hard to detect whether a household account is being accessed by more than one user, and this raises difficulties and limitations for providing an effective personalized assistance (Kabutoya et al., 2010; Berkovsky et al., 2011).

Users sharing a household do not necessarily access the service together, and consume offered items at the same time. Consider for instance a four members family (formed e.g. by a father, a mother, a son, and a daughter), sharing a household account of a video-on-demand service. Each member of the family has distinct viewing interests and habits, and thus each of them watches video differently regarding gender, time, and many other contextual variables. If one member of the family asks for video recommendations, it is likely that those recommendations do not fit the user’s interests, because the account profile contains a mixture of preferences from the four family members.

Two main strategies can be adopted in order to overcome such problem (Campos et al., 2012). The first strategy is to increase the diversity of delivered recommendations (Zhang and Hurley, 2008), aiming to cover the heterogeneous range of preferences of the different members in the household. The second strategy is to identify the active household members for which recommendations have to be delivered. In this thesis, we focus on the second strategy since it lets make more accurate recommendations, by only using preferences of active members, and discarding preferences of other, non-present members (Kabutoya et al., 2010).

The identification of active household members, defined as the discrimination of which users from a given household are interacting with a system (e.g. an on-demand video service), is thus an interesting challenge for the recommender systems research community. In fact, the convenience of identifying users in households for recommendation purposes has been addressed in the RS literature. Several proposed recommendation approaches on the TV domain consider the knowledge of which users are receiving the recommendations by means of explicit identification of users. For instance, Ardissono et al. (2001) propose a personalized Electronic Programming Guide for TV shows, requiring the user to log in the system for receiving personalization. Vildjiounaite et al. (2008) propose a method to learn a joint model of user subsets in households, and use individual remote control devices for identifying users. The methods considered in this chapter, in contrast, aim to identify the user who is currently interacting with the system, by analyzing temporal patterns of individual users, without requiring to log in or to use special devices at recommendation time.

Specific methods for the identification of users from household accounts have been proposed in the RS research field. Goren-Bar and Glinansky (2004) predict which users are watching TV based on a temporal profile manually stated. In (Goren-Bar and Glinansky, 2004) users indicate the time lapses in which they would probably be in front of the TV. Oh et al. (2012) derive time-based profiles from household TV watching logs, which model preferences of time lapses instead of individual users. In this way, the target profile corresponds to the time lapse at which recommendations are requested. These methods assume that users have a fixed temporal behavior through time.

Other works have also dealt with problems raised by the use of shared user accounts. In the context of the Netflix Prize competition, Koren (2009) discusses some difficulties for RS that could emerge from the use of household accounts –note that the well-known Netflix Prize competition dataset in fact contains household identifiers, not user identifiers. In that work, Koren proposes a temporal recommendation model that assumes the existence of a drifting *meta-user* associated with each household account. Similarly, Kabutoya et al. (2010) aim to identify *latent users* sharing an account, by using a probabilistic topic model.

The above approaches use household-level training data, i.e., data where it is unknown which users compose a household, and which household members really provided particular (training) ratings; and aim to improve recommendation accuracy over withheld test ratings. With this respect, these works differs from the research presented herein, in the sense that they focus on detecting latent preference patterns within account user profiles, in order to improve the final performance of certain recommender system. We propose, on the contrary, to model knowledge about such patterns independently from the recommender system used. Thus, in our approach, once the active members of household accounts are identified, any recommendation algorithm could be performed. In this way, we believe that recommended items would better fit the active users' preferences.

7.2 Discrimination of active user based on time context information

The 2011 edition of the Context-Aware Movie Recommendation (CAMRa) Challenge (Berkovsky et al., 2011) requested participants to identify which members of particular households were responsible for a number of events –interactions with the system in the form of ratings. The contest provided a training dataset with information about ratings in a movie RS, including the users who provided the ratings, and their associated timestamps. It also provided information about a number of users utilizing household accounts. The challenge's goal was to identify the users of household accounts who had been responsible for certain events (ratings), and whose household and timestamp were given in a randomly sampled test dataset.

Figure 7.1 shows a schematic view of the challenge's task. In the matrix, each row represents a household (hh) and each column a represents a movie. Each cell of the matrix contains the known ratings given by household members to the corresponding movie (training data). The question marks (?) indicate cases to identify which member of the corresponding household performed the given rating (test data). Given the availability of rating timestamps, the contest's task can be assumed equivalent to that of identifying active users requesting recommendations at a particular time, and thus the contest's data can be used for testing methods aimed to discriminate active users in household accounts.

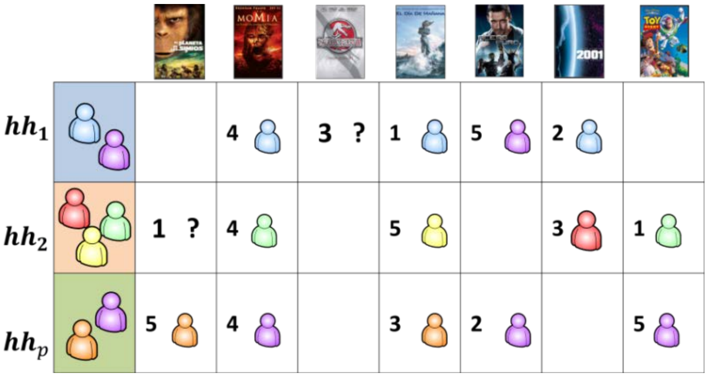


Figure 7.1. Schematic view of CAMRa 2011 Challenge household member identification task.

CAMRa 2011's MoviePilot Dataset is a movie rating dataset from the German movie recommender MoviePilot[26], consisting of a training set of 4,536,891 time stamped ratings from 171,670 users on 23,974 items in a timespan from July 11, 2009 up to July 12, 2010, and two test sets –there were two challenge tracks: track #1 corresponding to a household rating prediction task, and track #2 corresponding to the household member identification task. The test set of track #1 contains 4,482 ratings from 594 users on 811 items in a timespan from July 15, 2009 up to July 10, 2010 and the test set of track #2 contains 5,450 ratings from 592 users on 1706 items in a timespan from July 13, 2009 up to July 11, 2010. Additionally, the dataset contains information about 602 users that belong to one of 290 household accounts.

Figure 7.2 shows the rating, community, and catalog growth of training data (upper side) and testing data for the track #2 (lower side) through time. It may be seen that data growth follows a similar proportion on both rating sets. Table 7.1 shows the size distribution of households in the dataset. 2-sized households represent the 93.8% of all the

[26] MoviePilot movie recommendations, http://www.moviepilot.de

households, while 3-sized and 4-sized households represent the 4.8% and 1.4% respectively.

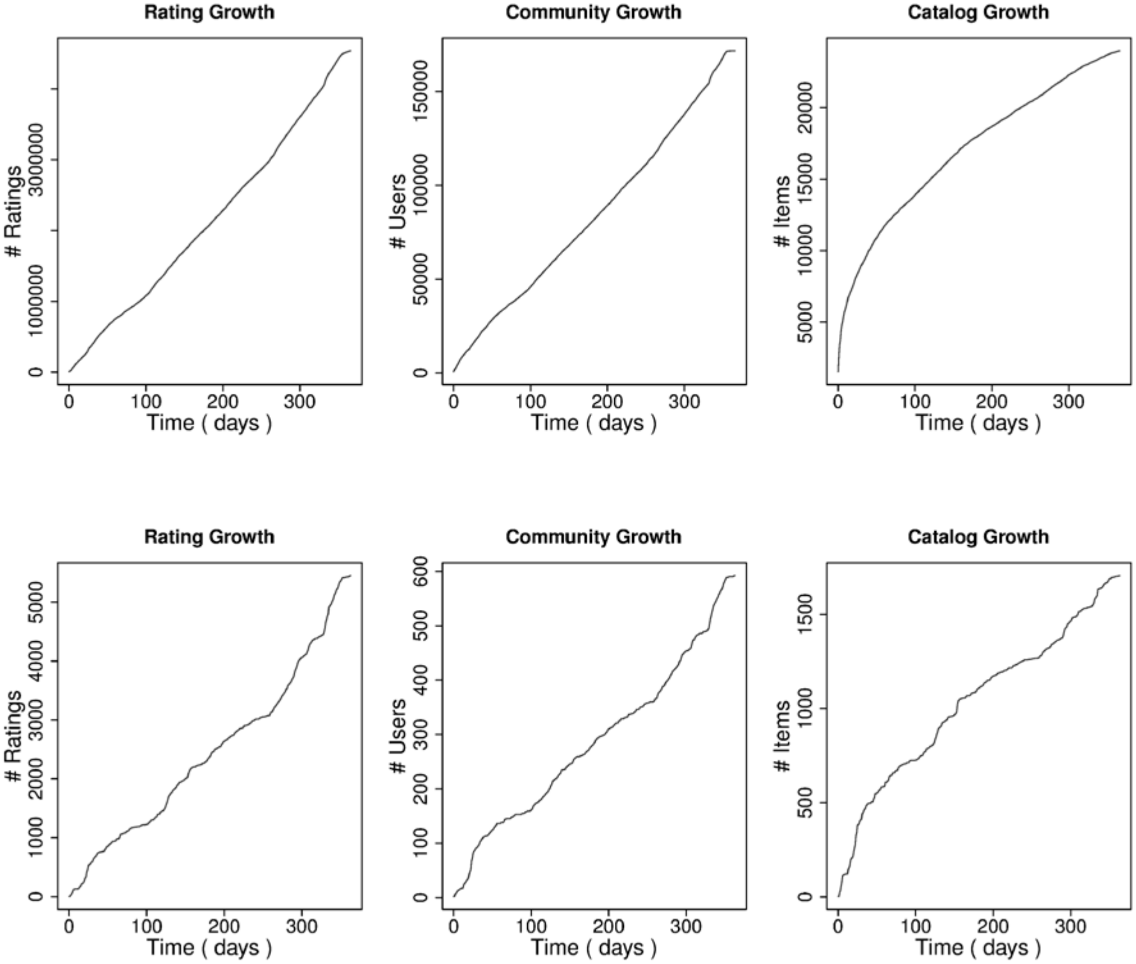


Figure 7.2. Training (upper side) and testing (lower side) CAMRa 2011's MoviePilot dataset growth through time.

Table 7.1. Distribution of household account sizes.

Number of members per account	Number of accounts
2	272
3	14
4	4

Taking advantage of timestamp information, we are able to derive several categorical time context variables. Using these variables, we observe that user temporal behavior within a household is not uniform. For instance, Figure 7.3 shows the rating hour *probability mass function* (PMF) of the two users in household account #1. We observe that there is a clear disparity between the hours employed by each of the household's members for rating movies. The user u40246 has a probability close to 1 (0.93) of rating movies in the period from 18:00 to 19:00. On the contrary, the user u311738 rates movies starting at 20:00 and later on, that is, mostly by night. Similar patterns are repeated along the data set, suggesting that time-based strategies might be useful for the household member identification task.

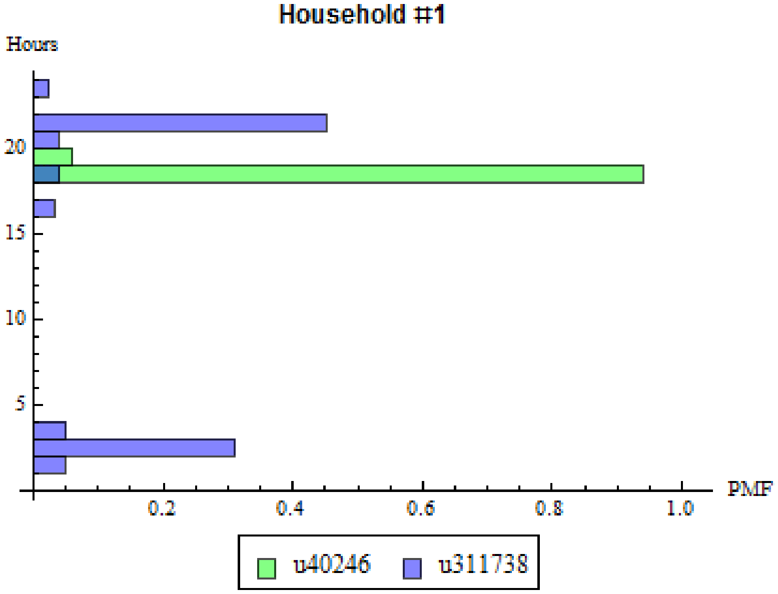


Figure 7.3. Probability mass function (PMF) of rating hours of users in household account #1.

When analyzing the rating date from each user, it is also possible to detect some interesting facts. Figure 7.4 shows how many ratings are made by users through time. The left frame shows that the mean rating window size (i.e., the timespan at which users perform ratings) is very small, –just a few days. The center and right frames also show that the vast majority of ratings are incorporated during the first days after the users start providing ratings. Considering that users start their participation in different days, this information can be helpful for the task at hand. We also note that there are differences on the day of the week each user rates movies.

These findings motivated us to use probability-based models in order to classify users in a given household, by exploiting time context information with a good discrimination power. Our approach can be formalized as follows. Let us consider a set of events $E = \{e_1, e_2, \dots, e_m\}$, and a set of users $U_h = \{u_{1,h}, u_{2,h}, \dots, u_{n,h}\}$ in a household h, such that event e_i is associated to one, and only one, user $u_{j,h}$. Also, let us consider that each of these events is described by means of a feature vector, called X_{e_i}. The question to address is whether it is possible to determine which user is associated to an event e_i once the values $x_{e_i,k}$ of (some) components $X_{e_i,k}$ of its feature vector X_{e_i} are already known. In the following, events correspond to instances of user ratings, and feature vectors correspond to time context information associated to the events. Each time context variable corresponds to a feature describing the time at which the event was produced. As we focus on time context information, we use interchangeably the terms time context variable and feature in the remainder of this chapter.

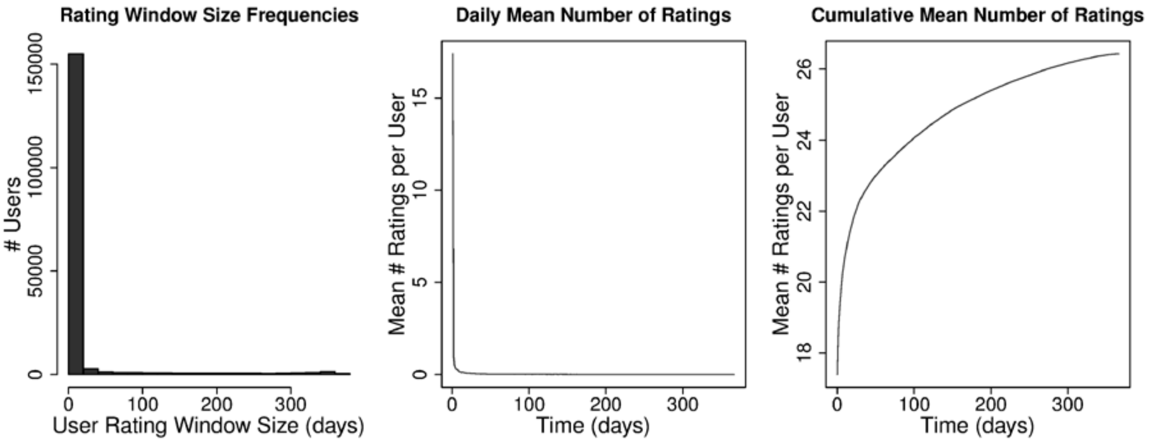


Figure 7.4. Time-based rating frequencies: from left to right, rating window size, daily and cumulative number of ratings through time.

Table 7.2 shows the time context features analyzed in this study. Aiming to estimate the discrimination power of such features, we used the well-known *Kullback-Leibler Divergence* (KLD) (Kullback and Leibler, 1951), which lets us to measure the divergence between pairs of users in a household, regarding the probability distribution of the features for the users in each household:

$$KLD\left(p_{u_1,T^k}, p_{u_2,T^k}\right) = \sum\nolimits_t \ln\left(\frac{p_{u_1,T^k}(t)}{p_{u_2,T^k}(t)}\right) p_{u_1,T^k}(t)$$

where p_{u,T^k} is the probability mass function of user u for time feature T^k, and $p_{u,T^k}(t)$ is the value of p_{u,T^k} at time t. Higher values of KLD correspond to more divergent probability distributions, and can be interpreted as having users in households with differentiated habits with respect to the corresponding time features.

In the table, the features are sorted in descending order by the average KLD value computed over all pairs of users in each household. We note that, to avoid biases due to the order of computations, we computed for each user pair the average $\overline{KLD}\left(p_{u_1,T^k}, p_{u_2,T^k}\right) = \left(KLD\left(p_{u_1,T^k}, p_{u_2,T^k}\right) + KLD\left(p_{u_2,T^k}, p_{u_1,T^k}\right)\right)/2$. The best discriminant features according to KLD were the **absolute date** (D), the **day of the week** (T_w), and the **hour of the day** (H).

Table 7.2. Analyzed time features.

Time feature	Domain	KLD
Absolute date ($\boldsymbol{D}$)	$1,2, \dots$, # of days in training set	5.79
Day of the week ($\boldsymbol{W}$)	$1,2, \dots 7$	4.56
Hour of the day ($\boldsymbol{H}$)	$0,1, \dots ,23$	4.53
Time of the day ($\boldsymbol{T_d}$)	*morning*, *noon*, *afternoon*, *evening*, *night*	2.28
Time of the week ($\boldsymbol{T_w}$)	*workday*, *weekday*	1.79
Meridian ($\boldsymbol{M}$)	*AM*, *PM*	1.47
Minute of the hour ($\boldsymbol{M_h}$)	$0,1, \dots ,59$	0.97
Quarter of the hour ($\boldsymbol{Q_h}$)	1,2,3,4	0.70
Month of the year ($\boldsymbol{M_y}$)	$1,2, \dots ,12$	0.36

The use of KLD as a predictor of the discrimination power of a time feature in the household member identification task requires to be confirmed experimentally. Furthermore, the use of feature vectors including different combination of time features may have diverse impact on the performance in the task. In order to test the discrimination power of the analyzed time features, in the next section we use distinct classification methods to identify the user associated to an event in a given household.

7.3 Classification accuracy of active user identification methods

In this section, we present and evaluate several methods that use time context information for the classification of users as (currently) active or inactive within a household at a given time. In Section 7.3.1 we present the used methods. In Section 7.3.2 we describe the experimental setting followed for the evaluation of such methods. In Section 7.3.3 we report and discuss obtained evaluation results.

7.3.1 Methods for active user identification

The first considered method is the *A priori* model described in (Campos et al., 2011a). This method computes probability distribution functions, which represent the probabilities that users are associated to particular events, and uses computed probabilities to assign a score to each user in a household, given a new event. More specifically, we compute the PMF of each feature X given a particular user, restricted to the information related with that user's household, that is, $\{p(X = x|u_j)\}_{u_j \in U_h}$, where U_h is the set of users in the household h.

Then, for each new event e, we obtain its representation as a feature vector $\hat{X}_e$, and identify the user who maximizes the PMF, that is, $u_j^*(e) = \arg\max_{\mathrm{u_j} \in U_h} p(\hat{X}_e|u_j)$. When more than one feature is used, we assume independence and use the joint probability function, i.e., the product of the features' PMFs.

We also evaluate Machine Learning (ML) algorithms described in (Campos et al., 2012), that are able to deal with heterogeneous attributes. Specifically, we consider the following methods: Bayesian Networks (BN), Decision Trees (DT), and Logistic Regression (LR) (Bishop, 2006). These methods provide a score $\{s(\hat{X}_e, u_j)\}_{u_j \in U_h}$ based on different statistics from the training data, and select the user with highest scores. The above methods use a fixed set of time features in the classification task, and thus they use the same set of features over all the households. It is important to note, however, that data from only one household is used to classify events of that household, i.e., the methods do not use data from other households to identify members of a given household.

Additionally, we considered two baselines for comparison purposes, namely a *Random* classifier, and a *Frequency-based* classifier, which for a given test event, selects the household member who has the largest number of previous events in the training set, and no rating for the event's item.

7.3.2 Experimental setting

For the evaluation of the methods, we used the CAMRa 2011 Challenge proposed test set (track #2 test set). We computed the accuracy of the methods in terms of the correct classification rate by household ($acc_{\mathbb{H}}$), i.e., the number of correct active member predictions divided by the total number of predictions, averaged by household, as proposed by the CAMRa 2011 Challenge organizers. Formally, let $\mathbb{H}$ be the entire set of households in the dataset, and let $g(\cdot)$ be a method under evaluation. The metric is expressed as follows:

$$acc_{\mathbb{H}} = \frac{1}{|\mathbb{H}|} \sum_{h \in \mathbb{H}} \frac{1}{|h|} \sum_{(e_i, u_i) \in h} L(u_i, g(e_i))$$

where $g(e_i) = \hat{u}$ is the user predicted by $g(\cdot)$ as associated to e_i, $L(u, \hat{u}) = 1$ if $u = \hat{u}$, and 0 otherwise, and (e_i, u_i) is a pair event-user of household h in the test set.

7.3.3 Experimental results

We first study whether time features alone are a valuable source of information to properly discriminate users for identifying active household members. Table 7.3 shows the $acc_{\mathbb{H}}$ values obtained by the *A priori* method when using each of the proposed time features (see Table 7.2), and using different combinations of such features. Note that in the table, the diagonal cells contain the $acc_{\mathbb{H}}$ values obtained from the use of a single feature, and the remainder cells contain the $acc_{\mathbb{H}}$ values obtained from the use of feature combinations.

We observe that the best single performing features are H, D and W, which is in accordance with the KLD-based feature ranking reported in Table 7.2, confirming the predictive power of KLD. In cases where two features were used, combinations including any of H, D and W features obtained better results. Furthermore, we evaluated all the possible combinations of features, and found that combinations including H, D and W achieved the best results. In particular, the best $acc_{\mathbb{H}}$ value of 0.9737 was achieved by combining the features H, D, W and Q_h.

Table 7.3. Accuracy of the *A priori* method using different time feature combinations. Darker grey cells indicate worse values of the metric. Global best value is in bold.

	D	W	H	T_d	T_w	M	M_h	Q_h	M_y
D	0.9413								
W	0.9426	0.9310							
H	**0.9727**	0.9652	0.9457						
T_d	0.9557	0.9467	0.9391	0.8260					
T_w	0.9430	0.9298	0.9531	0.8885	0.7991				
M	0.9553	0.9435	0.9402	0.8544	0.8614	0.7832			
M_h	0.9509	0.9424	0.9511	0.8944	0.8942	0.8793	0.8396		
Q_h	0.9517	0.9409	0.9532	0.8786	0.8770	0.8642	0.8404	0.8081	
M_y	0.9420	0.9372	0.9538	0.8472	0.8332	0.8077	0.8657	0.8351	0.7190

We also evaluated BN, LR and DT Machine Learning methods. We used Weka [27] implementations of BN, LR and J48 DT algorithms, with default parameter values. Their accuracy values are shown in Table 7.4 for combinations of the best individual performing time features.

We observe that, in general, these methods outperform the *A priori* model for a small margin. We also note that as more features are used, the higher accuracy is obtained, although combining only H and D features achieves high accuracy values as well. The

[27] Waikato Environment for Knowledge Analysis, a suite of machine learning software developed at the University of Waikato, New Zealand. Available at http://www.cs.waikato.ac.nz/ml/weka/

highest accuracy was obtained by the DT method when using all the considered time features.

Table 7.4. Accuracy of Machine Learning methods using different time feature combinations. Darker grey cells indicate worse values of the metric. Global best value is in bold.

	BN	**LR**	**DT**	**A priori**
D	0.9538	0.9515	0.9472	0.9413
W	0.9438	0.9405	0.9435	0.9310
H	0.9442	0.9432	0.9459	0.9457
DW	0.9484	0.9564	0.9470	0.9426
DH	0.9740	0.9769	0.9709	0.9727
WH	0.9690	0.9701	0.9750	0.9652
DWH	0.9744	0.9759	0.9752	0.9720
All	0.9722	0.9785	**0.9787**	0.9663

From these results we conclude that the identification of active household members within the evaluation setting proposed by the CAMRa 2011 Challenge can be effectively addressed by exploiting only time context information, regardless of the classification method used.

To conclude our study on the classification accuracy of the methods, we compare them with the proposed baselines. Moreover, we use the two available CAMRa 2011 Challenge's test sets. The purpose of this is to observe the performance of proposed methods on independent test sets, in order to avoid unintentional overfitting on the data of test set #2.

Table 7.5 shows the obtained accuracy results. Random and frequency-based baselines had a poor performance on test set #1, and a better performance on test set #2. This may be due to the differences on the rating data distributions in such test sets, which were built with distinct purposes. We observe that in test set #1, every test item assigned to a household had not been previously rated by a member of the household. This fact turns the frequency-based classifier into a random classifier, since it was not able to decrease its uncertainty by getting rid of some of the users in the household (who previously rated the test event's item).

We note that the tendency of results is similar on the two test sets, although better results were obtained on test set #2. The best result of the *A priori* model was obtained with the combination of DH features on both test sets, while the best result among ML models was obtained by LR using all features for test set #1. On test set #2, the best value was obtained by DT using all features.

Table 7.5. Accuracy of the evaluated methods on test sets #1 and #2. Darker grey indicate worse values of the metric in each column. Best value in each column is in bold.

Method	Test set #1	Test set #2
Random	0.4988	0.4890
Frequency	0.4906	0.8100
A priori (All features)	0.9384	0.9663
A priori (DH)	0.9504	0.9727
BN (All features)	0.9482	0.9722
LR (All features)	**0.9552**	0.9785
DT (All features)	0.9528	**0.9787**

These results show that the correct classification rate is prone to minor differences depending on the utilized household member identification method. In any case, the use of adequate time features brings the most significant improvements, achieving much higher accuracy values than the random- and frequency-based classifiers.

The obtained results indicate that simple algorithms are able to achieve high accuracy values on this task when certain time context variables are used, using the provided evaluation setting –the CAMRa 2011 Challenge’s test sets. However, considering the discussion on evaluation methodologies presented in Chapter 3, we question whether the evaluation methodology used for building the CAMRa 2011 Challenge’s test sets is fair enough for evaluating time-based predictive models. In the next section, we take advantage of the methodological framework proposed in Chapter 4 for assessing the above household member identification methods on different evaluation settings. These include some settings that take into account the time dependences between training and test data.

7.4 Robust evaluation of active user identification methods

Results reported in Section 7.3, as well as in other works exploiting time context signals for active household user identification (Bento et al., 2011; Campos et al., 2011a, 2012), show that the analysis of temporal patterns on historical data of household accounts provides important information for the discrimination of users, letting accurately identify active members at a given time.

Nonetheless, it is important to note that proposed methods have been assessed using evaluation methodologies based on the random selection of test cases. As reported in Chapter 5, however, it has been shown that using randomly selected test data may not be fair enough for evaluation, particularly when temporal trends are being considered by the evaluated methods. We question whether this is also applicable for the task at hand, and in such case, which accuracy for active user identification would be achieved by using other evaluation methodologies.

In this section we perform an empirical comparison of the methods introduced in Section 7.3 using different evaluation methodologies, including some that take into account the temporal order of data for building the training and test sets. To do so, we make use of the methodological evaluation framework proposed in Chapter 4 in order to select proper evaluation conditions, and specify the methodologies followed. In Section 7.4.1 we discuss the applicability and extensions of our methodological evaluation framework for the task at hand. In Section 7.4.2 we describe the experimental setting for this comparison, and in Section 7.4.3 we report and discuss the obtained results.

7.4.1 Using the TARS methodological evaluation framework for assessing active user identification methods

The methodological framework introduced in Chapter 4 provides a conceptual support for selecting different conditions in the evaluation of time-aware recommender systems, thus facilitating the specification of diverse methodologies for assessing the performance of such time-based predictive models. Moreover, many conditions that comprise the framework (base rating set, rating ordering, and rating set size conditions) are related with the formation of adequate training and test sets, constituted by tuples in the form $\langle u, i, r, t \rangle$, where u and i are pairs of entities (user and item), and r is a value associated to the pair (u, i) at time t. In the case of the task at hand, similar pieces of information must be handled, incorporating relations between user and households. Given this, and the fact that the methods under evaluation exploit time information, the proposed framework seems to fit well for defining more robust evaluation methodologies for the task.

We note that the evaluation conditions regarding the training-test splitting procedure (Algorithm 4.1) in the framework can be easily extrapolated to the task at hand. In the case of base rating set conditions, the community-centered $\mathscr{b}_{cc}$ condition can be applied because there is individual rating data for each user-item pair in the dataset, and the user-centered $\mathscr{b}_{uc}$ condition can also be applied because each rating is associated to a user. In the case of rating order conditions, the time-independent (random) $\mathscr{o}_{ti}$ condition can be applied, as it does not require any type of specific information in the data, and the time-dependent $\mathscr{o}_{td}$ condition can also be applied because each rating has a timestamp. In the case of rating set size conditions, the proportion-based $\mathscr{s}_{prop}$, fixed-based $\mathscr{s}_{fix}$ and time-based $\mathscr{s}_{time}$ conditions can be applied, since the dataset contains individual rating data.

From the above, we observe that we can use the proposed framework for generating different training-test splits in order to test the reliability of the methods used for household member identification. These conditions, however, do not take into account the relation between users and households. Moreover, we note that it may be desirable to count with evaluation data focused on the household level –as opposed to the community or the user level. This additional requirement can be accomplished by defining a new base rating set condition, the *household-centered* base rating set ($\mathscr{b}_{hc}$) as follows:

Household-centered base rating set condition. A base dataset M_h is built with the ratings of all users u^h that belongs to the household h:

$$M^{\mathscr{b}_{hc}} = \{M_h \mid h \in \mathbb{H}\}, M_h = \left\{r_{u,\cdot} \mid r_{u,\cdot} \in M, u \in h\right\}$$

This base rating set condition forces the application of rating ordering and rating set size conditions on each household's data. Furthermore, it lets define and describe several methodologies particularly suited for evaluation of the active user identification task.

Regarding the remaining conditions, we note that conditions related with top-N recommendations evaluation (target item and relevant item) do not apply in this case, because we do not evaluate recommendation performance. In the case of cross-validation conditions, the conditions can be used following the guideline 4 stated in Chapter 5, that is, applying a cross-validation method consistent with the selected base rating set, rating ordering, and rating set size conditions. We note, however, that the household-centered base rating set condition is not included in the defined set of cross-validation methods.

7.4.2 Experimental setting

In this evaluation we compared the same methods presented in Section 7.3 (*A priori*, BN, LR and DT), using again the CAMRa 2011 Challenge MoviePilot dataset. Based on the results reported in the previous section, the time features considered were the **absolute date** (D), the **day of the week** (W), and the **hour of the day** (H), as they were the best performing features for the task.

Aiming to analyze differences on the accuracy of the methods, we selected three evaluation methodologies, which are described in the following. Two of them use a time-dependent rating order condition, and the other one use a time-independent order condition.

The first methodology (denoted as $\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{fix}$) consists of combining a community-centered base rating set ($\mathscr{b}_{cc}$), a time-dependent rating order ($\mathscr{o}_{td}$), and a fixed size ($\mathscr{s}_{fix,q=5450}$) condition. Specifically, all ratings in the dataset were sorted according to their timestamp, and the last 5,450 ratings were assigned to the test set (the first 149,551 were assigned to the training set). In this way, we built a test set of similar size to that of test set #2.

The second methodology (denoted as $\mathscr{b}_{hc}\mathscr{o}_{td}\mathscr{s}_{fix}$) is equivalent to $\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{fix}$ with a household-centered base rating set condition ($\mathscr{b}_{hc}$). Specifically, the ratings of each household were sorted according to timestamp, and the last 19 ratings from each household were assigned to the test set. We chose 19 ratings aiming to build a test set of similar total size to that of the one built with $\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{fix}$.

The third methodology (denoted as $\mathscr{b}_{hc}\mathscr{o}_{ti}\mathscr{s}_{fix}$) is similar to $\mathscr{b}_{hc}\mathscr{o}_{td}\mathscr{s}_{fix}$ with a time-independent rating order condition ($\mathscr{o}_{ti}$). That is, 19 ratings were randomly selected from each household, and assigned to the test set.

As in the previous experiments, we computed the accuracy of the evaluated methods in terms of the correct classification rate by household ($acc_{\mathbb{H}}$).

7.4.3 Experimental results

Table 7.6 shows the $acc_{\mathbb{H}}$ results obtained by the evaluated methods using the three methodologies described above. The table also shows the results obtained on the test set #2, proposed by CAMRa organizers for the task (column titled CAMRa), for comparison purposes. The table shows the results obtained by using individual time features, grouped by method.

Table 7.6. Accuracy of the evaluated methods using different time features and evaluation methodologies. Darker grey cells indicate worse values of the metric in each column. Global top values in each column are in bold, and the best values for each method are underlined.

Method	Time Feature	$\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{fix}$	$\mathscr{b}_{hc}\mathscr{o}_{td}\mathscr{s}_{fix}$	$\mathscr{b}_{hc}\mathscr{o}_{ti}\mathscr{s}_{fix}$	CAMRa
A priori	*H*	0.6087	0.8163	0.9468	0.9457
	W	0.6167	0.8069	0.9299	0.931
	D	0.4947	0.8152	0.9461	0.9413
BN	*H*	0.6533	0.8232	0.9539	0.9442
	W	0.6907	0.8189	0.9412	0.9438
	D	0.6506	**0.8575**	**0.9574**	**0.9538**
DT	*H*	0.6637	0.8229	0.9541	0.9459
	W	**0.6963**	0.8223	0.9417	0.9435
	D	0.6506	0.8544	0.9535	0.9472
LR	*H*	0.6674	0.8256	0.9537	0.9432
	W	0.6908	0.8132	0.9381	0.9405
	D	0.6147	0.8307	0.9555	0.9515

In the table, we observe similar results when using methodologies based on a time-independent (random) rating order condition (CAMRa and $\mathscr{b}_{hc}\mathscr{o}_{ti}\mathscr{s}_{fix}$). Much worse results are observed when using methodologies employing a time-dependent rating order condition ($\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{fix}$ and $\mathscr{b}_{hc}\mathscr{o}_{td}\mathscr{s}_{fix}$). Particularly lower accuracies are achieved when using $\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{fix}$. We note that this latter methodology provides the evaluation scenario most similar to a real-world situation: data up to a certain point in time is available for training purposes, and data after that (unknown at that time) is then used as ground truth. In our case, this methodology provides a small number of training events for some households, which affects the methods' ability to detect temporal patterns of users. In fact, for some households, there is no training data at all. In this way, $\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{fix}$ represents a hard, but realistic evaluation methodology for the task.

On the contrary, methodologies using a time-independent rating order condition provide easy, but unrealistic evaluation scenarios, because they let the methods use training data that would not be available in a real-world setting. The $\mathscr{b}_{hc}\mathscr{o}_{td}\mathscr{s}_{fix}$ methodology provides an intermediate scenario, in which an important part of data is available for learning temporal patterns of each household's members.

In the table we also observe that the discrimination power of the different time features varies among methodologies. In the case of the *A priori* method, the best results on time-independent methodologies and $\mathscr{b}_{hc}\mathscr{o}_{td}\mathscr{s}_{fix}$ are obtained with the **hour of the day** (H) feature, while the **absolute date** (D) achieves the best results among ML methods –we note that results show small differences across features. However, when using the stricter $\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{fix}$, the best results among methods are obtained with the **day of the week** (W) feature, nearly followed by the **hour of the day** feature. On the contrary, the **absolute date** feature performs the worst consistently.

The above highlights how unrealistic the less strict methodologies are for the task, because they let the methods exploit a temporal behavior (the exact date of interaction) that in a real situation would be impossible to learn. This also shows that the **hour of the day** and more strongly the **day of the week** features describe a consistent temporal pattern of users through time.

Table 7.7 shows the $acc_{\mathbb{H}}$ results obtained by the evaluated methods using combinations of time features, and the same methodologies reported in Table 7.6. The results show that using less strict methodologies, combinations including the **absolute date** feature perform better. On the contrary, using the realistic $\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{fix}$ methodology, the best results across methods are achieved by the combination of **hour of the day** and **day of week**. These results are in accordance with those observed in Table 7.6.

All these results show that a correct classification rate is prone to major differences depending on the followed evaluation methodology. The discrimination power of time features varies considerably when assessed by different methodologies. Moreover, the accuracy achieved by the methods is much lower when using the more realistic $\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{fix}$ methodology.

Table 7.7. Accuracy of the evaluated methods using combinations of time features, on different evaluation methodologies. Darker grey cells indicate worse values of the metric in each column. Global top values in each column are in bold, and the best values for each method are underlined.

Method	Time feature	$\mathscr{b}_{cc}\mathscr{o}_{td}\mathscr{s}_{fix}$	$\mathscr{b}_{hc}\mathscr{o}_{td}\mathscr{s}_{fix}$	$\mathscr{b}_{hc}\mathscr{o}_{ti}\mathscr{s}_{fix}$	CAMRa
A priori	*HW*	$\underline{0.6496}$	$\underline{0.8421}$	0.9688	0.9652
	HD	0.4947	0.8205	0.9739	$\underline{0.9727}$
	WD	0.4947	0.8152	0.947	0.9426
	HWD	0.4947	0.8205	$\underline{0.9746}$	0.972
BN	*HW*	$\underline{0.6876}$	0.8325	0.9721	0.969
	HD	0.6262	0.8287	$\underline{0.9773}$	0.974
	WD	0.6529	0.8127	0.9534	0.9484
	HWD	0.6809	$\underline{0.8401}$	0.977	$\underline{0.9744}$
DT	*HW*	**$\underline{0.7188}$**	0.8644	0.9773	0.975
	HD	0.6389	$\underline{0.8648}$	0.9753	0.9709
	WD	0.6932	0.8417	0.9526	0.947
	HWD	0.695	0.8599	$\underline{0.9777}$	$\underline{0.9752}$
LR	*HW*	$\underline{0.6635}$	0.8652	0.9768	0.9701
	HD	0.6515	0.865	**$\underline{0.9824}$**	**$\underline{0.9769}$**
	WD	$\underline{0.6636}$	**$\underline{0.8697}$**	0.9553	0.9564
	HWD	0.6591	0.867	0.9808	0.9759

7.5 Conclusions

In this chapter we have presented and evaluated a number of methods to effectively identify which user of a shared household account is currently interacting with an online recommender system at a particular time, by only exploiting knowledge about past user interactions with the system. We focused this study on two main axes: (i) we analyzed existing differences in temporal rating habits, described in terms of various time features. These features were used to discriminate between users in a household by means of a classification algorithm; and (ii) we made an empirical comparison of these methods with different methodologies previously applied on time-aware recommender systems evaluation. Given that the methods are based on exploiting temporal patterns, we used a time-based rating order evaluation condition, taking advantage of the methodological framework introduced in Chapter 4, and following the guideline 1 for evaluation stated in Chapter 5.

Regarding (i), we found that simple algorithms are able to achieve good accuracy values when certain time features are used, showing that isolated time features are valuable sources of information for discriminating users in a shared household account.

Concerning (ii), we found that the discrimination power of time features, alone and combined, varies considerably when assessed by different methodologies. We observed that less strict methodologies provide unreliable results, due to the exploitation of temporal information that is hard to obtain in a realistic evaluation scenario. Moreover, the accuracy achieved by all the methods was much worse when using a strict time-aware evaluation methodology.

These findings show that, despite the described methods have good accuracy rates, additional improvements are required to provide accurate identification of active household members in real-world applications. More importantly, the presented study remarks the importance of assessing the performance of time-aware algorithms using a robust evaluation protocol that properly takes the evolution of data through time into account.

We finally highlight the flexibility and extensibility of the methodological evaluation framework proposed in Chapter 4. In particular, the conditions regarding the training-test split of data were directly applicable for generating the training and test sets required for a more robust evaluation of the developed methods. Moreover, we could easily extend the framework by incorporating an additional condition specific for the task at hand –the household-centered base rating set condition. The structure of the framework lets an easy incorporation of this new condition, and a seamlessly integration with the rest of evaluation conditions.

Chapter 8

Conclusions

The work presented in this thesis was motivated by the need of understanding and improving the exploitation of time context information by recommender systems. As initial steps towards such objective, in this thesis we have focused on the definition, formulation and use of robust evaluation protocols for comprehensive and fair assessment of different time-aware recommendation models. We then have adapted and proposed time-aware methods for different recommendation tasks, based on the experience derived from the more reliable measurement of the performance improvements obtained. More specifically, we have addressed the following research goals:

- The characterization of conditions involved in the evaluation of time-aware recommender systems.
- The analysis of the effect of different evaluation conditions on the assessment of time-aware recommendation performance.
- The adaptation of existing recommendation approaches to make better use of available time context information.
- The exploitation of time context information in a non well-established recommendation task.

In the first part of the thesis, we have reviewed existing approaches to recommendation computation and evaluation, putting particular emphasis and detail on time-aware recommendation approaches. Starting from such comprehensive review, in the second part of the thesis, we have formalized, analyzed, and empirically compared the conditions that drive the evaluation process of TARS, and have proposed a methodological description framework that lets precisely state the conditions used in the evaluation of a particular TARS. Furthermore, we have proposed a set of guidelines aimed to help selecting appropriate conditions for a reliable evaluation of TARS. Finally, in the third part of the thesis, we have presented different applications of time-aware approaches to recommendation tasks, proposing new heuristics and adaptations of existing methods in the case of the well-established rating prediction and top-N recommendations tasks, and developing novel methods in the case of the recently proposed task of identifying active users in shared accounts. We have utilized the proposed framework to evaluate the performance of the proposed adaptation and methods.

In this chapter, we present the main conclusions of our work. In Section 8.1 we summarize the contributions of the thesis. In Section 8.2 we detail the validation of the stated hypotheses, and in Section 8.3 we describe potential future research directions.

8.1 Summary and discussion of contributions

In the next subsections we summarize and discuss the main contributions of this thesis, regarding the research goals and hypotheses stated in Chapter 1. These contributions are organized according to the addressed research goals. First, we investigated the conditions that drive the evaluation process of TARS. Second, we analyzed the differences between recommendation performance assessments due to the change of the used evaluation conditions, in order to establish a set of conditions leading to a robust evaluation protocol. Third, we proposed new heuristics and adaptations to existing recommendation approaches in order to enhance the exploitation of time context information. And fourth, we proposed novel time-aware methods for the less studied task of identifying active users in shared accounts.

8.1.1 Characterization of conditions involved in the evaluation of TARS

From a comprehensive survey of the research literature on time-aware recommender systems, we observed that reported results and conclusions about how to incorporate and exploit time information within the recommendation process seem to be contradictory in some cases. We hypothesized that existing discrepancies could be caused by meaningful divergences in the used evaluation protocols, –metrics and methodologies. A careful review of such evaluation protocols showed several methodological differences on the evaluations conducted among works. With more detail, in Section 4.1 we observed that the training-test splitting process is an important source of methodological divergence, particularly when rating timestamps are available. We identified several design decisions to be taken when defining an evaluation setting that lead to methodological differences. Analyzing such differences, we posed a number of key **methodological questions** regarding the design of a TARS evaluation protocol:

- MQ1: What base rating set is used to perform the training-test splitting?
- MQ2: What rating ordering is used to assign ratings to the training and test sets?
- MQ3: How many ratings comprise the training and test sets?
- MQ4: What cross-validation method is used for increasing the generalization of the evaluation results?

In addition to these questions, we also covered specific conditions for the evaluation design of the top-N recommendations task, posing the following two methodological questions:

- MQ5: Which items are considered as target items (in a top-N recommendations task)?

- MQ6: Which items are considered relevant for each user (in a top-N recommendations task)?

We addressed these questions by means of a number of **evaluation conditions** that we stated from the review of evaluation settings found in the TARS literature. These conditions express decisions related to the training-test splitting and cross-validation processes in the evaluation of RS, and specific aspects regarding the evaluation of top-N recommendations. Specifically, in Chapter 4 we characterized and formalized the following evaluation conditions:

- *Community-centered* and *user-centered base rating set conditions*, which indicate if next conditions have to be applied on the full set of ratings, or independently on each user's ratings set, addressing MQ1.
- *Time-dependent* and *time-independent rating ordering conditions*, which indicate whether or not to sort the set of ratings by time, addressing MQ2.
- *Proportion-based, fixed* and *time-based rating set size conditions*, which state the criterion used to define the sizes of training and test sets, addressing MQ3.
- *Time-dependent* and *time-independent cross-validation conditions*, which establish the cross-validation methods applicable depending on the compatibility with the ratings' time-sort restrictions, addressing MQ4.
- *User-based*, *community-based*, *one-plus random* and *other target item conditions,* which indicate the criterion used to determine the set of items to be ranked in a top-N recommendations task, addressing MQ5.
- *Test-based* and *threshold-based relevant item conditions*, which set the criterion used to determine the relevance of items, addressing MQ6.

These conditions cover the wide range of alternative design decisions used in the evaluation process of approaches in the TARS literature.

Based on the defined conditions, we developed a **methodological description framework** aimed to facilitate the comprehension of such conditions. The framework is intended to make the evaluation process fair and reproducible under different circumstances, by letting state clearly and meticulously the settings used in the evaluation of TARS. The formalism of the framework includes the definition of a **splitting procedure** proposed by us, and described in Section 4.2, which, taking as input a set of evaluation conditions, lets precisely build and reproduce data splits (i.e., training and test sets) for a given evaluation setting. Using the splitting procedure and different combinations of the conditions included in the framework, the diverse evaluation settings for TARS can be accurately defined, as was shown by means of the examples given in Section 4.3.4.

Moreover, in Section 5.1, we conducted a comprehensive **classification of state-of-the-art TARS** in terms of the characterized evaluation conditions, mapping such conditions to the evaluation settings used in the time-aware recommender system literature, and providing a general overview of the conditions and methodologies more commonly used in the evaluation of such systems. We found that almost a 25% of the revised studies used a time-independent rating ordering –despite the fact that the reviewed papers deal with time-aware approaches–, and that approximately 40% of the studies use a combination of a community-centered rating base set and a time-dependent ordering of ratings –which provides the evaluation scenario most similar to a real-world setting. We also found an even distribution on the use of rating set size criterions and a low usage of cross-validation methods. Regarding the conditions specific for evaluating top-N recommendations, we observed that most TARS-related papers addressing this task use a test-based criterion for defining the relevance of items, and a more even distribution of the criteria used for selecting target item sets.

8.1.2 Analysis of the effect of different evaluation conditions on the assessment of TARS performance

Alongside the characterization of evaluation conditions presented in Chapter 4, we **discussed the effect** of using alternative conditions on addressing each posed key methodological question involved in the evaluation of TARS (MQ1 – MQ6). From this discussion, we observed the important differences of applying data splitting conditions on the full set of ratings in a dataset –that is, using a community-centered base rating set condition– vs. applying such conditions independently on each user data –i.e., using a user-centered base rating set condition. Moreover, we noted the differences in the generated data splits induced by applying time-independent, or alternatively a time-dependent rating ordering condition. In order to study the influence of using different combinations of evaluation conditions on the assessment of recommendation performance, we performed an **empirical comparison** of several TARS following different evaluation protocols. In particular, in Chapter 5 we reported the evaluation of three widely used TARS approaches, and one well-known non-contextual recommendation approach using four different evaluation methodologies. The obtained results showed that the use of distinct evaluation conditions not only yields remarkable differences between metrics measuring distinct recommendation properties –namely accuracy, precision, novelty and diversity, but also may affect the relative ranking of approaches for a particular metric.

From our analysis and experiments, we reported key methodological issues that a robust evaluation of TARS should take into consideration in order to perform a fair assessment of recommendation approaches performance, and facilitate comparisons among published experiments. From this, in Section 5.3 we concluded a set of **methodological guidelines** aimed to facilitate the selection of conditions for a proper TARS evaluation. In the evaluation of the rating prediction task, our guidelines suggest making training-test splitting based on a time-dependent rating ordering over the full set of ratings in a dataset, applying a proportion-based size criterion for training and test sets, and using a cross-validation method compatible with the suggested data splitting conditions. In the evaluation of the top-N recommendations task, our guidelines suggest to rank a mixed set of items including some for which user relevance is completely unknown, as this is the common setting in real-world applications, and using a threshold-based relevant item condition, which discards low rated/consumed items from the set of relevant items, thus providing a more confident interpretation of item relevance.

8.1.3 Adaptation of existing recommendation approaches to make better use of available time context information

The evaluation guidelines proposed in this thesis let us establish a **fair and common evaluation setting** for assessing performance results from different recommendation approaches exploiting time context information. Starting from the analysis of such results and the characteristics of the recommendation approaches, in Chapter 6 we proposed **new heuristics and adaptations** for some of the approaches, in order to make better use of available time context information.

In particular, in Section 6.2.1 we proposed a **new impurity criterion based on the Fisher's exact test**, to be used in Item Splitting (Baltrunas and Ricci, 2009a, 2009b) –a general pre-filtering contextualization approach. Moreover, in Section 6.3.2, we adjusted several impurity criteria used by Item Splitting by finding the best thresholds for diminishing rating prediction error when exploiting different time contexts. In Section 6.2.2, we also developed a **new post-filtering strategy based on the probability of rating an item in the target recommendation context**. This heuristic let perform a post-filtering contextualization of recommendations generated by the Matrix Factorization recommendation algorithm (Takács et al., 2008; Koren et al., 2009). Moreover, in Section 6.2.3, we adapted the contextual neighbors method (Panniello and Gorgoglione, 2012) –a general contextual modeling approach– by eliminating constraints originally considered to control the used type of contextualization, in order to be able to utilize different recommendation algorithms together with the method.

From the analysis of the suggestions given in the methodological guidelines proposed in Chapter 5 for the top-N recommendations task, and the particularities of the studied approaches –that are able to handle categorical representations of time context–, we proposed and used a **new methodology for assessing contextualized top-N recommendations**. This novel methodology, described in Section 6.4.1, let build ranked lists of items targeted for the same time context independently of the used time representation –an issue not previously addressed in the TARS literature–, while including unrated items in the list. By doing so, the proposed methodology provides an evaluation setting more similar to that deployed TARS shall confront –correctly rank unrated items for a given target context in order to recommend the relevant ones– than those from other methodologies previously used in the literature.

We evaluated the proposed adaptations together with other approaches able to exploit time context information –namely exact pre-filtering (Adomavicius and Tuzhilin, 2011) and post-filtering (Panniello et al., 2009a)– on a context-enriched dataset of movie preferences from real users. The obtained results, discussed in Sections 6.3.3 and 6.4.3, showed the importance of **selecting a proper threshold** for each combination of impurity criterion and time context signal in the case of Item Splitting to obtain the best achievable recommendation performance.

Furthermore, these results also revealed that **there is no unique dominant TARS** in either the rating prediction or the top-N recommendations task, and that the performance improvements achieved by the tested approaches depend on the underlying recommendation algorithm and the exploited time context. This finding is in line with conclusions from previous research comparing context-aware RS in e-commerce applications, e.g. (Panniello et al., 2009a). The identification of the best performing approach, thus, requires a time-consuming evaluation and comparison of candidate TARS implementations on the target data. Furthermore, some contextualization approaches may require an intensive testing of parameters, as in the case of Item Splitting.

Despite the above mentioned, we note that the new heuristics proposed in Chapter 6 –the new impurity criterion for Item Splitting and post-filtering strategy for Matrix Factorization– are able to **effectively contextualize recommendations** generated by the high-performing Matrix Factorization recommendation algorithm. Furthermore, they showed the **best global values** on the majority of metrics of rating prediction and top-N recommendations task, respectively, on the performed comparison of approaches. Thus, the use of the proposed heuristics in conjunction with a Matrix Factorization recommendation algorithm can be considered a good approach to contextualize recommendations when time context information about user preferences is available.

8.1.4 Exploitation of time context information on a non well-established recommendation task

The exploitation of different time contexts associated to user ratings let us address a recommendation task out of the scope of the well-established rating prediction and top-N recommendations tasks: the identification of active users in shared accounts (households) (Berkovsky et al., 2011). In Chapter 7 we **proposed and evaluated a number of methods** to effectively identify which user of a shared household account is interacting at a given time with a recommender system, by only exploiting knowledge about past user interactions with the system. The methods are based on the identification of differences in

user temporal rating habits, described in terms of various time context signals or features including the *absolute date* (e.g. November 1st, 2013), the *day of the week* (e.g. Monday) and the *hour of the day* (e.g. 4:00 p.m.) at which users interact with the system. We formulated the task as a **classification problem**, and the time features were used to discriminate between users in a household by means of diverse classification algorithms.

Moreover, we **adapted methodologies used in TARS evaluation** in order to reliably assess the performance of the proposed methods on the identification of active users in shared accounts task. This required the formalization of a new condition, specific for the task –the household-centered base rating set condition–, which was incorporated into the proposed methodological framework, as described in Section 7.4. By utilizing the conceptual structure of the framework, we were able to specify methodologies based on the guidelines for TARS evaluation proposed in Chapter 5 that were used in the evaluation of the task. The above also showed the extensibility and ease of integration of new evaluation conditions of the proposed framework.

The results obtained in the experiments showed that some of the most elementary algorithms proposed were able to achieve good accuracy values when certain time contexts were exploited. Nonetheless, we observed that the discrimination power of time contexts, alone and combined, **varied considerably when they were evaluated with different methodologies**. We found that methodologies less strict from a temporal viewpoint –that is, methodologies that do not avoid a temporal overlap of training and test data– provided less reliable results. From these results, we noted the importance of following **robust evaluation protocols** such the ones suggested by the methodological guidelines proposed in the thesis.

8.2 Validation of stated hypotheses

In this section we detail on the validation of the hypotheses stated at the beginning of this thesis. Their validity has been tested by means of the experimental results obtained in the thesis.

Hypothesis 1: Variations in the evaluation protocol lead to differences on recommendation results assessment.

The results obtained in the empirical comparison of TARS evaluation methodologies, presented in Chapter 5 and further discussed in Section 8.1.2, let us prove this hypothesis. In particular, these results showed differences in the absolute value of metrics, and more importantly, on the relative ranking of approaches for a particular metric, when using different evaluation settings. From the above, we remark that the comparison of TARS approaches under distinct evaluation protocols may yield **completely different results**. All of this emphasizes the importance of counting with reliable protocols for the evaluation of TARS.

The validity of the above hypothesis has important implications for the reproducibility and comparability of reported performance results in the TARS literature. By using the same evaluation metrics and methodologies it is possible to reproduce and fairly compare results from different works on TARS. From this, we highlight the need of **clearly stating the conditions** in which offline experiments are conducted to evaluate RS in general, and TARS in particular. Furthermore, by following consensual and fair evaluation conditions –i.e., a robust evaluation protocol–, we will enable the reproducibility of experiments, and will ease the comparison of recommendation approaches. In the hope to contribute to such purpose, we developed the methodological description framework presented in this thesis.

Hypothesis 2: The appropriate exploitation of time context information leads to improvements on assessed recommendation results.

The results obtained in the evaluation of methods for well-established recommendation tasks –namely rating prediction and top-N recommendations–, presented in Chapter 6 and further discussed in Section 8.1.3, let us prove this hypothesis. In particular, these results showed that by **appropriately selecting** the time context signal, the underlying recommendation algorithm, and the specific parameters required by some approaches, it is possible to improve the recommendations generated by methods not exploiting time context. The assessed performance of the approaches in our experiments depended to a great extent on the underlying recommendation algorithm used. However, for instance, we note that a proper selection of threshold values for the impurity criteria used by Item Splitting let improve the results of high performance algorithms such as Matrix Factorization.

Proving this hypothesis is in accordance with reported results from previous research in the area. Nonetheless, we stress that not all the methods that exploit time context information are able to obtain better results than those that do not exploit such information. Moreover, the fact of observing improvements **depends on the evaluation protocol followed**, as showed in the results reported in Section 5.2.4. Our results indicate that a **careful selection** of the methods' parameters, the time context signals, and the underlying recommendation algorithms is required in order to effectively leverage recommendation performance when following a robust evaluation protocol.

Hypothesis 3: From a temporal viewpoint, a robust evaluation protocol of recommendation models and techniques exploiting time context information, leads to a decrease on performance with respect to a less robust evaluation protocol.

This hypothesis is proved on the basis of the results obtained in the experimental comparison of methods exploiting time context information for the identification of active users in shared accounts. This included methodologies with both time-independent and time-dependent rating ordering conditions, presented in Chapter 7 and further discussed in Section 8.1.4. In particular, these results showed an impressive performance of the proposed methods on the task –over 95% of accuracy on the identification of active user– when using methodologies based on a time-independent rating ordering condition. However, the same methods showed an **important decrease on performance** when assessed with methodologies based on a time-dependent rating ordering condition. Moreover, the lowest performance of the methods –rounding 65% of accuracy– was measured when using the evaluation conditions suggested by our methodological guidelines.

The validity of this hypothesis highlights the importance of defining and utilizing a robust evaluation protocol to accurately assess the degree of performance improvement obtained from the exploitation of time context information by TARS approaches. In this context, the methodological guidelines proposed in Chapter 5 are a powerful tool for increasing the reliability of performance assessments.

8.3 Future work

In this thesis we have presented a comprehensive review and analysis of protocols used in TARS evaluation, which have led us define a methodological framework, and a set of guidelines to provide robust evaluation settings for TARS. Moreover, he have proposed and evaluated adaptations and new methods for exploiting time context information. Despite these important contributions and findings, the research conducted in this thesis raises interesting additional research questions regarding TARS development and evaluation. In the following subsections we discuss a number of issues that call for further research, and depict possible work lines to address such issues.

8.3.1 Evaluation of time-aware recommendation approaches

The methodological framework proposed in this thesis is composed of a set of conditions that let define the setting in which a recommendation approach is evaluated, covering the evaluation methodologies used in TARS literature. The evaluation reported in Chapter 5 provided us evidence about the effect on the assessment of recommendation performance due to changes in the used evaluation conditions, letting us to propose a set of guidelines for selecting conditions for a robust evaluation of TARS. Nonetheless, more experimentation is required to properly analyze the impact of combinations of conditions not addressed in our study. Also, new conditions should be defined leading to apply the proposed guidelines in other recommendation tasks where time context could be exploited, and that were not studied in this thesis, such as the recommend sequence task (Herlocker et al., 2004). For these purposes, we believe that the proposed framework provides an important conceptual structure to guide such research.

Another important pending issue is related to the analysis of the relation between different characteristics of datasets (e.g. user profile sizes, timespans, and sparsity levels) and the effect on performance from using dissimilar evaluation protocols. Beyond this, the appropriateness of using certain evaluation conditions when using datasets with particular rating distributions through time/users/items, types of feedback, domains, etc. could to be investigated.

The relation between accuracy and novelty/diversity metrics also remains as an open evaluation issue. Given the increasing importance of the latter metrics in the RS field, additional analysis and explanations are required in order to provide time-aware recommendations with adequate levels of such performance properties. For instance, as noted by Lathia et al. (2010), from a temporal viewpoint, recommendation diversity is an important facet a recommender system should have.

An additional interesting question is whether improvements of TARS performance measured by offline evaluation are effectively perceivable for real users. As noted e.g. by Knijnenburg et al. (2012), accuracy improvements are not necessarily observable by users. The lack of online evaluation studies on TARS is a major limitation to address the above question.

8.3.2 Development of new and better time-aware recommendation approaches

Using a context-enriched dataset of movie preferences from real users, the experiments reported in Chapter 6 let us derive important insights regarding the circumstances in which certain recommendation approaches outperform others. Nonetheless, the obtained conclusions are not necessarily general, due to the small size of the used dataset, and the fact that only one recommendation domain was evaluated. Repeating the evaluation on different datasets, from diverse domains, and types of user feedback, would let establish more general conclusions regarding the applicability of specific TARS and time contexts. Indeed, as stated by Adomavicius and Tuzhilin (2011), one of the main challenges on context-aware recommendation is the investigation of which contextualization approaches perform better, and under which circumstances. In such evaluation, it is also important to consider recommendation properties beyond accuracy and precision. An interesting example of this in the more general field of context-aware recommender systems is the work from Panniello et al. (2013), where different CARS approaches are compared in terms of accuracy and diversity. For such purpose, we remark the importance of using a common evaluation protocol for the reproducibility and comparability of results.

We note that our survey of the literature showed the existence of two main types of TARS, according to how time context information is represented, namely in continuous and discrete representations. However, in the evaluation of TARS approaches reported in Chapter 6, we focused on TARS based on the latter representation, due to the characteristics of the used datasets. The comparison of such approaches with other recommendation approaches based on a continuous time representation is thus an issue to be investigated.

An additional research line is the joint exploitation of both types of time representation. One way to accomplish this would be building hybrid approaches (Burke, 2007) that combine recommendations from several TARS. Other possible way to address such issue may be to develop and improve model-based approaches able to handle both types of time representations, such as the *timeSVD++* technique proposed by Koren (2009a), and tensor factorization-based models like the *Bayesian Probabilistic TF* proposed by Xiong et al. (2010).

Appendix A

Introducción

En este capítulo presentamos una visión general de la tesis doctoral. Describimos los principales temas de investigación abordados y las limitaciones que motivaron la realización de la misma, proporcionando un resumen del trabajo llevado a cabo, y presentando y discutiendo los resultados obtenidos.

En la Sección A.1 reseñamos los temas de investigación que motivaron esta tesis. En la Sección A.2 definimos el alcance de este trabajo, estableciendo el problema general y los objetivos de investigación abordados. A continuación, en las Secciones A.3 y A.4 detallamos las principales contribuciones y listamos las publicaciones originadas a partir de la investigación realizada. Finalmente, en la Sección 1.5 describimos la estructura de este documento.

A.1 Motivación: Recomendación, contexto y tiempo

Los Sistemas de Recomendación (SR) son aplicaciones de software cuyo propósito es ayudar a los usuarios en tareas de acceso y recuperación de información en grandes colecciones de ítems (productos o servicios), sugiriendo ítems, de modo general, de acuerdo a las preferencias personales mostradas en el pasado por los usuarios.

La última década ha sido fértil para la investigación en el campo de los SR. Se han investigado, entre otros, diferentes problemas y tareas de recomendación (Adomavicius y Tuzhilin, 2005), aproximaciones algorítmicas (Herlocker et al., 1999), o métricas y metodologías de evaluación (Shani y Gunawardana, 2011), dando lugar a importantes avances en los SR en operación y aumentando el interés en construir más y mejores SR. Por un lado, los usuarios de SR obtienen sugerencias personalizadas sobre ítems en los que pueden estar interesados y que pueden ser difíciles de encontrar. Por otro lado, las empresas que utilizan SR obtienen mayores beneficios gracias al incremento del consumo de los ítems sugeridos. Estos factores han llevado a la creación y expansión de importantes servicios personalizados apoyados por tecnologías de SR en Internet, tales como Amazon[28], Netflix[29], y Last.FM[30], por nombrar algunos.

La explotación de un SR permite recolectar grandes registros de preferencias de usuarios –*ratings* (valoraciones) o registros de consumo–, los cuales pueden incluir información sobre el **contexto** en el cual las preferencias de usuario fueron expresadas (Adomavicius y Tuzhilin, 2011). Por ejemplo, junto con las preferencias de un usuario particular, un SR puede registrar el tipo de dispositivo utilizado por el usuario (p. ej. un ordenador o un teléfono móvil), su localización (p. ej. en el hogar o en la oficina), el estado de humor del usuario (p. ej. feliz o triste), la compañía del usuario (p. ej. solo, con familiares o con amigos) o el instante en el que el usuario expresa su preferencia (p. ej. por la mañana o por la tarde). Explotando esta información, los SR conscientes del contexto (SRCC) pueden sugerir ítems que se ajusten de mejor forma a los intereses del usuario en ciertas circunstancias o situaciones (contextos), constituyéndose en valiosas herramientas para incrementar la eficacia de las recomendaciones proporcionadas (Koren, 2009a; Adomavicius y Tuzhilin, 2011; Panniello et al., 2013).

Entre las dimensiones contextuales existentes, el **contexto temporal** puede considerarse como uno de los más útiles. Éste contexto facilita el seguimiento de la evolución de las preferencias de usuario (Xiang et al., 2010), permitiendo por ejemplo identificar periodicidad en las preferencias de usuario (Baltrunas y Amatriain, 2009). También puede llevar a mejoras significativas en la exactitud de las recomendaciones, como fue el caso del equipo ganador de la competición Netflix Prize (Koren, 2009a). Más

[28] Tienda en línea Amazon.com, http://www.amazon.com

[29] Servicio de transmisión de vídeo bajo demanda Netflix.com, http://www.netflix.com

[30] Radio vía internet Last.FM, http://www.last.fm

aún, la información de contexto temporal es, en general, fácil de recolectar, sin esfuerzo adicional del usuario ni requisitos estrictos en los dispositivos usados.

Debido a estos beneficios, los años recientes han sido prolíficos en la investigación y desarrollo de **SR conscientes del tiempo** (SRCT), esto es, SRCC que explotan la dimensión temporal para estrategias tanto de modelado como de recomendación. Es posible encontrar diferentes propuestas de SRCT en la literatura que muestran mejoras sobre SR tradicionales en la eficacia de las recomendaciones. Sin embargo, cabe destacar que algunos estudios han mostrado **divergencias en las suposiciones sobre las que se construyen los modelos de SRCT**, generando dudas sobre la generalización de las capacidades de las recomendaciones conscientes del tiempo. De hecho, por ejemplo, algunas aproximaciones de SRCT penalizan los datos de preferencias antiguas, asumiendo que los datos recientes reflejan de mejor forma los gustos actuales de los usuarios, en comparación con los datos más antiguos (Ding y Li, 2005; Ma et al., 2007; Lee et al., 2008). Por el contrario, algunos autores, como por ejemplo Koren (2009a), han encontrado que este tipo de penalización lleva a una disminución en la calidad de las recomendaciones.

Aunque esta inconsistencia podría ser explicada por diversas razones, p. ej. diferencias en las características de usuario e ítem, y peculiaridades de los dominios de aplicación, nosotros creemos que la evaluación juega un rol fundamental. La existencia de **metodologías de evaluación diferentes** facilita encontrar un protocolo de evaluación idóneo para una aproximación algorítmica particular, pero no usable o inadecuado para otras aproximaciones. En efecto, algunos autores tales como Lathia et al. (2009a, 2009b) han mostrado discrepancias importantes en la calidad de la recomendación dependiendo de cómo se eligen los datos de entrenamiento y prueba para la evaluación de las recomendaciones. Los problemas que surgen a partir de esta situación representan un impedimento creciente para comparar, de forma ecuánime, resultados y conclusiones de diferentes investigaciones (Bellogín et al., 2011), haciendo más difícil la selección de la mejor solución de recomendación para una tarea dada (Gunawardana y Shani, 2009). Por tanto es una preocupación fundamental de nuestra investigación el estudio de las cuestiones metodológicas que una evaluación robusta de SRCT debería tener en cuenta, con el fin de aumentar la confiabilidad de las mejoras de calidad atribuidas a SRCT así como a facilitar la comparación de distintos planteamientos.

El **descubrimiento de resultados inesperados en diferentes estudios sobre SRCT** demuestra que aún se requiere de más investigación para comprender cabalmente la relación entre la información de contexto temporal y los resultados de recomendación. Baltrunas y Amatriain (2009) proveen un ilustrativo ejemplo de esto. Realizando experimentos para aumentar la eficacia de un SRCC sobre diversas particiones de datos de preferencias de usuario dependientes del tiempo, encontraron que la partición temporal poco común {*horas pares*, *horas impares*} mejora las recomendaciones en mayor cuantía con respecto a otras particiones tales como {*mañana*, *tarde*} y {*día de semana*, *día de fin* *de semana*}. En palabras de Baltrunas y Amatrian, la partición de *horas* corresponde a una partición “sin sentido”, y por lo tanto llaman a incrementar la investigación al respecto. Más aún, la falta de estudios comparativos de la eficacia de SRCT mantiene sin explorar las circunstancias bajo las cuales algunas aproximaciones de SRCT –y las señales o condiciones de contexto temporal explotadas, p.ej. el momento del día o el período de la semana– son capaces de superar a otras aproximaciones. Esto también impide ajustar los SRCT para explotar de mejor forma la información temporal disponible en situaciones particulares.

Adicionalmente a las cuestiones mencionadas anteriormente, una revisión de los trabajos publicados en esta área pone de manifiesto que la mayor parte de los SRCT han sido desarrollados para la **tarea de predicción de rating**. A pesar de ello, hoy en día el foco de recomendación está cambiando desde la disminución del error en las predicciones de rating hacia la búsqueda de (listas de) ítems relevantes/atractivos para el usuario destino de los mismos, i.e. la **tarea de recomendación de los N-mejores**. Más aún, el uso extendido de sistemas de recomendación en diversas tareas de usuario posibilita encontrar nuevas aplicaciones donde la información de contexto temporal puede contribuir de manera distintiva. Considerando todo lo anterior, la comprensión de cómo la información temporal puede ser explotada para mejorar las tareas de recomendación, más allá de (pero también incluyendo) la predicción de rating, constituye otra meta principal de nuestra investigación.

En resumen, tomando como punto de partida el estado del arte sobre aproximaciones a SRCT para la generación y evaluación de recomendaciones contextualizadas, esta tesis estudia, sintetiza y analiza cómo la información de contexto temporal ha sido explotada en la literatura de sistemas de recomendación, con el fin de a) caracterizar una metodología de evaluación robusta que permita realizar evaluaciones ecuánimes de nuevos SRCT, así como facilitar la comparación de resultados entre SRCT; y b) mejorar la explotación de información de contexto temporal en diferentes tareas de recomendación, llevando a nuevas y mejores aplicaciones de las tecnologías de recomendación conscientes del tiempo.

A.2 Planteamiento del problema, objetivos de investigación e hipótesis

Desde un punto de vista general, el problema de recomendación consiste en sugerir ítems que deberían ser los más atractivos para un usuario de acuerdo a sus preferencias. Tradicionalmente, la mayor parte de las propuestas de sistemas de recomendación no toman en cuenta ninguna información de carácter contextual, esto es, sólo consideran dos tipos de entidades para generar recomendaciones: usuarios e ítems (Adomavicius y Tuzhilin, 2011). En muchas aplicaciones, sin embargo, la información contextual puede ser una valiosa fuente de mejora de las recomendaciones, bajo la suposición de que circunstancias (contextos) similares se relacionan con preferencias de usuarios afines.

En esta tesis nos centramos específicamente en problemas que incorporan el **tiempo** como fuente de información contextual para estrategias tanto de modelado como de recomendación. El objetivo final de la tesis es abordar el problema de recomendación desde una perspectiva consciente del tiempo, basándonos en dos líneas de acción principales. Por un lado, estableciendo un protocolo de evaluación robusto que tome en cuenta las dependencias temporales de los datos, de forma tal que permita una evaluación objetiva y rigurosa de los resultados de recomendación de SRCT; y por otro lado, abordando diferentes tareas de recomendación desde una perspectiva consciente del tiempo, de forma que se obtengan ventajas del uso de información de contexto temporal para mejorar la eficacia de los métodos actuales en dichas tareas. Por medio del uso de un protocolo de evaluación robusto tratamos de disponer de una medida fiable de las mejoras obtenidas. Para afrontar estas líneas de acción hemos definido los siguientes objetivos de investigación:

OI1: Caracterización de las condiciones involucradas en la evaluación de SRCT. En este objetivo debemos realizar una profunda revisión y análisis de los protocolos empleados para la evaluación de la actual generación de SRCT, con el propósito de distinguir y formalizar las condiciones clave que conducen las evaluaciones realizadas. Abordamos este objetivo de investigación en el Capítulo 4.

Cabe destacar que en todo protocolo de evaluación existen dos componentes fundamentales que definen el escenario en el cual se mide la eficacia de un sistema: las *métricas de evaluación*, que definen qué se debe medir, y las *metodologías de evaluación*, que definen cómo medir. En el campo de los sistemas de recomendación, existen métricas aceptadas de uso habitual (Herlocker et al., 2004; Gunawardana y Shani, 2009). Sin embargo, no existe consenso respecto de las metodologías a usar (Bellogín et al., 2011). Más aún, es práctica común informar de las métricas aplicadas para medir la eficacia de los sistemas de recomendación desarrollados, pero es menos común encontrar descripciones claras sobre las metodologías de evaluación utilizadas. Debido a esto focalizaremos nuestro estudio en las divergencias metodológicas en la evaluación de SRCT.

OI2: Análisis del efecto del uso de diferentes condiciones de evaluación en la medición de la eficacia de SRCT. Mediante este objetivo queremos determinar si la aplicación de diferentes condiciones de evaluación lleva a diferencias en la medición de resultados de recomendación de SRCT. A partir de esto debemos definir el conjunto de condiciones que permitan efectuar evaluaciones ecuánimes y reproducibles de SRCT, con el fin de realizar mediciones rigurosas de la eficacia de SRCT. Abordamos este objetivo de investigación en el Capítulo 5.

Tal como se mencionó en la Sección A.1, planteamos como hipótesis que la evaluación juega un rol preponderante en la explicación de las discrepancias encontradas en la literatura de SRCT. Sin embargo, hasta donde conocemos, no se ha estudiado el impacto que tiene el uso de diferentes escenarios de evaluación en la medición de resultados. A partir del análisis de tal efecto, y de las características de las condiciones de evaluación, pretendemos establecer un conjunto de condiciones que proporcionen escenarios confiables para la evaluación de SRCT. Este conjunto de condiciones deberá usarse a lo largo de todo el trabajo experimental de esta tesis, para medir de forma apropiada las mejoras logradas por el uso de la información de contexto temporal asociada a los datos de preferencia de usuario.

OI3: Adaptación de propuestas de recomendación existentes para hacer un mejor uso de la información de contexto temporal disponible. Debemos investigar la relación entre la información de contexto temporal y las preferencias de usuario, al objeto de mejorar los resultados de recomendación de una o más propuestas de recomendación basadas en el conocimiento del contexto temporal. Este conocimiento permitirá ajustar o adaptar propuestas de recomendación existentes para mejorar la forma en que el contexto temporal es explotado. Las mejoras obtenidas serán medidas utilizando un conjunto de condiciones que aseguren una evaluación ecuánime y la comparabilidad con otras propuestas. Abordamos este objetivo de investigación en el capítulo 6.

Se ha comprobado que la explotación de la información de contexto temporal es una aproximación efectiva para mejorar la calidad de la recomendación, como lo demostró, por ejemplo, el equipo ganador de la conocida competición Netflix Prize (Koren, 2009b). En la literatura es posible encontrar múltiples propuestas de sistemas capaces de explotar información de contexto temporal. No obstante, el cambio de enfoque que va desde la disminución del error de predicción hasta la búsqueda de ítems relevantes, unido a la falta de protocolos de evaluación estandarizados, hace difícil establecer qué propuestas utilizan de mejor forma la información de contexto temporal. Por tanto, contando con un escenario de evaluación ecuánime, sería posible determinar las circunstancias en las cuales algunos algoritmos superan a los demás. A partir de esto, seríamos capaces de ajustar o adaptar el funcionamiento de algunas propuestas de recomendación con el objeto de mejorar su eficacia.

OI4: Explotación de la información de contexto temporal en una tarea de recomendación novedosa. En este objetivo pretendemos obtener ventaja de la experiencia y conocimiento sobre la utilización y evaluación de modelos de recomendación conscientes del tiempo, por medio del desarrollo de aplicaciones novedosas de estas técnicas. Con este objetivo en mente, consideraremos tareas relacionadas con la recomendación –más allá de la predicción de rating y la recomendación de los N-mejores (N mejores ítems o top-N)– donde la información de contexto temporal disponible pueda ser una fuente importante de mejoras. Desarrollaremos nuevas propuestas basadas en la explotación del contexto temporal para abordar una de estas tareas y usaremos un escenario de evaluación que asegure una evaluación ecuánime y robusta. Abordamos este objetivo de investigación en el Capítulo 7.

El desarrollo de los objetivos de investigación antes mencionados se basa en las siguientes hipótesis:

Hipótesis 1: Variaciones en el protocolo de evaluación llevan a diferencias en la medición de resultados de recomendación. Esta hipótesis está relacionada con OI1 y OI2.

Hipótesis 2: La explotación adecuada de la información de contexto temporal lleva a mejoras en los resultados de recomendación medidos. Esta hipótesis está relacionada con OI3 y OI4.

Hipótesis 3: Desde un punto de vista temporal, el uso de un protocolo de evaluación robusto para modelos y técnicas de recomendación que explotan información de contexto temporal provoca un descenso de la eficacia medida con respecto a un protocolo de evaluación menos robusto. Esta hipótesis está relacionada con OI2, OI3 y OI4.

A.3 Contribuciones

La investigación llevada a cabo en esta tesis busca contribuir a mejorar la confiabilidad en la medición de resultados de sistemas de recomendación conscientes del tiempo, permitiendo una mejor explotación de la información de contexto temporal en los sistemas de recomendación. Por ello, las principales contribuciones de nuestra investigación son:

- **La caracterización de las condiciones que conducen el proceso de evaluación de los sistemas de recomendación conscientes del tiempo**. Realizamos una revisión exhaustiva de la literatura sobre SRCT, identificando las cuestiones metodológicas clave que se deben afrontar durante el diseño experimental de una evaluación *offline* de SRCT. A partir de esto, formalizamos un conjunto de condiciones usadas en la evaluación de SRCT que abordan las cuestiones metodológicas analizadas. Las condiciones definidas se encuentran relacionadas principalmente con el proceso de partición de datos en conjuntos de entrenamiento y prueba, el cual puede realizarse de diferentes formas debido a la existencia de información de contexto temporal asociada a los datos. Incluimos también condiciones requeridas para evaluar tareas de recomendación específicas, tal y como se detalla en el Capítulo 4.

- **El desarrollo de un marco de trabajo metodológico para describir las condiciones usadas en la evaluación de SRCT**. Proponemos un marco de trabajo de descripción metodológica que incorpora las condiciones de evaluación caracterizadas en la tesis, con el propósito de facilitar la descripción y adopción de protocolos de evaluación, y hacer el proceso de evaluación ecuánime y reproducible. Este marco de trabajo, introducido en el Capítulo 4, incluye la definición de un nuevo algoritmo de partición de repositorios de datos para generar conjuntos de datos de entrenamiento y prueba, usando las condiciones de evaluación formalizadas. El uso de este marco de trabajo facilita la comparación de resultados de diferentes propuestas de SRCT, ya que permite difundir de manera simple y formal las distintas condiciones de evaluación utilizadas para medir la eficacia de los SRCT.

- **El análisis de aspectos metodológicos que una evaluación *offline* robusta de SRCT en particular, y de SR en general, debería tener en cuenta**. Sintetizamos y discutimos el efecto de usar diferentes condiciones que abordan las cuestiones metodológicas clave involucradas en la evaluación de SRCT a través del Capítulo 4. Adicionalmente, en el Capítulo 5 clasificamos la literatura de SRCT revisada en términos de las condiciones de evaluación definidas, analizando y mapeando el uso de tales condiciones en un amplio número de trabajos de investigación sobre sistemas de recomendación conscientes del contexto y del tiempo. Más aún, realizamos una rigurosa comparación experimental de resultados obtenidos de diferentes protocolos de evaluación de SRCT, la cual es presentada en dicho Capítulo 5. Evaluamos un conjunto de conocidos SRCT en los dominios de recomendación de películas y música, usando diferentes tipos de datos de preferencias de usuario, a saber, ratings explícitos e implícitos. El propósito de esta comparación es valorar la influencia de las condiciones de evaluación en los resultados de eficacia medidos, por medio de métricas de precisión y *ranking*.

- **La propuesta de un conjunto de guías metodológicas cuyo propósito es facilitar la selección apropiada de condiciones para la evaluación *offline* de SRCT**. A partir de los resultados obtenidos en nuestros experimentos y del análisis de los protocolos de evaluación utilizados en la literatura de SRCT, en el Capítulo 5 concluimos un conjunto de guías generales destinadas a facilitar la selección de condiciones para una evaluación de SRCT apropiada. Estas guías incluyen la elección de condiciones para realizar la partición de datos en conjuntos de entrenamiento y prueba, necesaria para calcular las métricas de evaluación y para la aplicación de un método de validación cruzada adecuado. También incluimos guías para seleccionar condiciones específicas requeridas para evaluar recomendaciones de los N-mejores.

- **La propuesta de nuevas heurísticas y adaptaciones para algunos sistemas de recomendación conscientes del contexto para hacer un mejor uso de la información de contexto temporal**. Implementamos SRCC del estado del arte y proponemos nuevas heurísticas con el fin de mejorar su eficacia al explotar información de contexto temporal. Específicamente, en el Capítulo 6 proponemos un nuevo criterio de impureza para ser utilizado por el algoritmo *Item Splitting* (Baltrunas and Ricci, 2009a, 2009b), y desarrollamos una estrategia de post-filtrado que permite contextualizar las recomendaciones generadas por el destacado algoritmo de recomendación de factorización de matrices (Takács et al., 2008; Koren et al., 2009). Adicionalmente, ajustamos otros criterios de impureza utilizados por *Item Splitting* y adaptamos una propuesta de modelado contextual de Panniello y Gorgoglione (2012). Las heurísticas y adaptaciones propuestas se basan en la medición de resultados obtenidos a partir de datos contextualizados de usuarios reales, utilizando un protocolo de evaluación común y definido de manera precisa.

- **La propuesta de una nueva metodología para evaluar resultados de recomendación de los N-mejores**. Proponemos y utilizamos una nueva metodología para evaluar la tarea de recomendación de los N-mejores ítems en el estudio presentado en el Capítulo 6, la cual permite construir listas ordenadas de ítems destinadas al mismo contexto temporal, incluyendo ítems no valorados en la lista, proporcionando así un escenario de evaluación más realista que aquellos resultantes de otras metodologías descritas en la literatura.

- **El desarrollo de nuevas propuestas conscientes del tiempo para abordar la tarea de identificación de usuarios activos en cuentas de usuario compartidas**. En el Capítulo 7 proponemos y desarrollamos nuevos métodos que explotan la información de contexto temporal para abordar esta tarea de recomendación recientemente definida, que consiste en identificar de forma automática al usuario activo (en un instante concreto) en una cuenta de usuario compartida (por ejemplo en el hogar). Formulamos esta tarea como un problema de clasificación y evaluamos diferentes clasificadores que explotan atributos temporales de registros de consumo de ítems de los usuarios de un hogar. El análisis de los atributos temporales obtenidos muestra la existencia de diferentes hábitos temporales de valoración por parte de los usuarios de cuentas compartidas, los cuales permiten diferenciar qué usuario se encuentra activo en un momento determinado.

- **La adaptación de metodologías de evaluación de SRCT para medir la eficacia de diferentes métodos en la tarea de identificación de usuarios activos en cuentas de usuario compartidas**. En el Capítulo 7 describimos una extensión al marco de trabajo metodológico propuesto para la evaluación de SRCT, por medio de la definición de una condición adicional, específica para esta reciente tarea. Concluimos que la estructura del marco de trabajo permite incorporar fácilmente la nueva condición. Basados en esto, utilizamos el marco de trabajo para adaptar las metodologías recomendadas por nuestras guías a la evaluación de esta tarea, con el fin de valorar los métodos propuestos para la misma.

A.4 Publicaciones

Las contribuciones de esta tesis han originado un conjunto de publicaciones, las cuales se detallan a continuación. Las hemos agrupado de acuerdo al capítulo y tema de investigación con el cual se relacionan.

Capítulo 4

Metodologías de evaluación y SRCT

Una propuesta inicial orientada a establecer un marco de trabajo para la evaluación de sistemas de recomendación conscientes del tiempo fue presentada en:

- Campos, P. G., Díez, F. (2010). **La Temporalidad en los Sistemas de Recomendación: Una Revisión Actualizada de Propuestas Teóricas**. *I Congreso Español de Recuperación de Información* (CERI 2010), pp. 65-76. Madrid, España.

En este trabajo describimos una revisión del estado del arte en SRCT, a partir de la cual advertimos la necesidad de mejorar los protocolos de evaluación utilizados en la valoración de la eficacia de SRCT. Esta observación motivó el propósito principal de esta tesis –la necesidad de proporcionar una evaluación de la eficacia de los SRCT más fiable. Con la finalidad de cumplir tal propósito, desarrollamos un marco de trabajo metodológico para seleccionar y describir las condiciones utilizadas para evaluar y comparar SRCT. Las condiciones de evaluación que constituyen el marco de trabajo metodológico presentado en el capítulo son estudiadas en:

- Campos, P.G., Díez, F., Cantador, I. (2013). **Time-Aware Recommender Systems: A Comprehensive Survey and Analysis of Existing Evaluation Protocols**. *User Modeling and User-Adapted Interaction*, Special Issue on Context-Aware Recommender Systems. En prensa, pendiente de publicación (publicación *online*: 2013).

En este trabajo, formalizamos un conjunto de condiciones utilizadas para evaluar SRCT, a partir del análisis de protocolos de evaluación encontrados en una revisión exhaustiva de la literatura sobre SRCT. Estas condiciones permiten describir de forma precisa las metodologías empleadas en la medición de la eficacia de SRCT, facilitando la reproducibilidad de escenarios de evaluación y la comparación de diversas propuestas de SRCT.

Capítulo 5

Escenarios de evaluación y eficacia de recomendación

Una vez identificada la importancia del escenario utilizado para evaluar SRCT, estudiamos la eficacia de propuestas de SRCT conocidos bajo diferentes protocolos de evaluación. Este estudio fue presentado en:

- Campos, P.G., Díez, F., Sánchez-Montañés, M. (2011). **Towards a More Realistic Evaluation: Testing the Ability to Predict Future Tastes of Matrix Factorization-based Recommenders**. *5th ACM Conference on Recommender Systems* (RecSys 2011), pp. 309-312, Chicago, IL, USA.

En este trabajo, comparamos la eficacia del algoritmo de factorización de matrices (FM) –el cual no es consciente del tiempo– frente a la aproximación de FM con dinámicas temporales (Koren, 2009a), bajo dos protocolos de evaluación: aquel utilizado en la competición Netflix Prize, y un escenario que utiliza una separación temporal estricta de los datos de entrenamiento y prueba. Del análisis llevado a cabo encontramos diferencias importantes en el ordenamiento relativo de las propuestas evaluadas al cambiar el escenario de evaluación, mostrando así claramente la necesidad de una evaluación de propuestas de SRCT más robusta. Los protocolos de evaluación probados en este trabajo sirvieron como base para definir las condiciones de evaluación utilizadas en la comparación empírica de SRCT presentada en el capítulo.

Capítulo 6

Evaluación de la eficacia de recomendaciones conscientes del tiempo

Una vez que observamos que la variabilidad de diferentes SRCT en la literatura se debe principalmente al uso de diferentes escenarios de evaluación, decidimos implementar y comparar diferentes propuestas de SRCT bajo un protocolo de evaluación claro y común. De esta forma, es posible identificar qué propuestas superan a otras, y bajo qué circunstancias. Un primer estudio comparativo fue presentado en:

- Campos, P.G., Díez, F., Cantador, I. (2012). **A Performance Comparison of Time-Aware Recommendation Models**. *Proceedings of the 2nd Spanish Conference in Information Retrieval* (CERI 2012), Valencia, España.

En este trabajo comparamos SRCT que explotan información de contexto temporal continua, utilizando una metodología de evaluación que toma en cuenta el orden temporal de los ratings. Sin embargo, estuvimos limitados a usar un conjunto de datos de ratings con marcas de tiempo, sin contar con información sobre el contexto temporal en el cual los ítems fueron consumidos y/o utilizados efectivamente. En un trabajo posterior, realizamos un estudio de usuario con el fin de obtener información de contexto temporal confiable, para comparar diferentes propuestas de recomendación que explotan información de contexto. Este último estudio es descrito en:

- Campos, P.G., Fernández-Tobías, I., Cantador, I., Díez, F. (2013). **Context-Aware Movie Recommendations: An Empirical Comparison of Pre-Filtering, Post-Filtering and Contextual Modeling Approaches**, *Proceedings of the 14th International Conference on Electronic Commerce and Web Technologies* (EC-Web 2013), pp 137-149, Prague, Czech Republic.

Este trabajo se enfoca en la comparación de propuestas generales de SRCC que son capaces de explotar información de contexto temporal en la forma de variables categóricas. Más aún, comparamos información de contexto temporal y social, de manera de estudiar cuál proporciona más información a las propuestas evaluadas, en términos de mejoras en la tarea de predicción de rating. El marco de trabajo metodológico propuesto sirvió de base para definir el escenario de evaluación en este estudio.

Sistemas de recomendación conscientes del contexto e información de contexto temporal

Estudiamos la capacidad de SR conscientes del contexto de mejorar la eficacia de las recomendaciones, a partir de la explotación de señales de contexto temporal modeladas como variables categóricas, derivadas de información de contexto temporal continua (en la forma de marcas de tiempo) asociadas a los ratings. Evaluamos una propuesta de pre-filtrado del estado del arte en:

- Campos, P.G., Cantador, I., Díez, F. (2013). **Exploiting Time Contexts in Collaborative Filtering: An Item Splitting Approach**, *3rd workshop on Context-Awareness in Retrieval and Recommendation* (CaRR 2013) desarrollado conjuntamente con 6th ACM International Conference on Web Search and Data Mining (WSDM 2013), pp. 3-6, Rome, Italy.

Este trabajo se enfoca en el análisis del algoritmo de pre-filtrado *Item Splitting*, buscando las mejores combinaciones de señales de contexto temporal tales como *periodo del día* y *periodo de la semana*, así como de diferentes parámetros utilizados por dicho algoritmo, con el fin de obtener mejoras en predicciones de rating, así como en la tarea de recomendación de los N-mejores.

Capítulo 7

Estudio de los hábitos temporales de los usuarios en valoración de items

El análisis de la información de contexto temporal asociada a los ratings de usuario nos permitió abordar una tarea relacionada con los sistemas de recomendación que ha sido menos estudiada: la identificación de usuarios en cuentas de usuario compartidas. Esta tarea fue propuesta como una competición en el marco del segundo Taller sobre Recomendación de Películas consciente del Contexto (CAMRa 2011, por sus siglas en inglés). El análisis inicial de los datos proporcionados, y nuestras primeras propuestas para la tarea, fueron presentados en:

- Campos, P.G., Díez, F., Bellogín, A. (2011). **Temporal Rating Habits: A Valuable Tool for Rating Discrimination**. *Proceedings of the 2nd Workshop on Contex-aware Movie Recommendation* (CAMRa 2011), desarrollado conjuntamente con 5th ACM Conference on Recommender Systems (RecSys 2011), pp. 29-35, Chicago, IL, USA.

En este trabajo examinamos diferentes variables de contexto temporal derivadas de marcas de tiempo, así como información adicional asociada a ratings de usuario, encontrando diferencias importantes en el comportamiento de usuarios distintos para realizar valoraciones dentro una misma cuenta compartida (en el hogar). Más aún, en este trabajo propusimos una aproximación basada en un modelo probabilístico para la identificación del usuario activo en un momento dado.

Identificación de usuarios activos en cuentas compartidas basada en información de contexto temporal

Motivados por el buen desempeño de las aproximaciones propuestas, implementamos y evaluamos diversos métodos para la tarea antes mencionada, basados exclusivamente en la explotación de información de contexto temporal. Estos métodos y su eficacia en la tarea se describen en:

- Campos, P.G., Bellogín, A., Díez, F., Cantador, I. (2012). **Time feature selection for identifying active household members**. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management* (CIKM’12), pp. 2311-2314 Maui, HI, USA.

Los métodos presentados en este trabajo son capaces de abordar la tarea con gran exactitud (sobre un 95%) utilizando el protocolo de evaluación establecido por los organizadores de la competición de CAMRa 2011, la cual se basa en la selección aleatoria de datos de prueba.

Evaluación robusta de métodos para la identificación de usuarios activos en cuentas de usuario compartidas

Con el fin de probar la confiabilidad de los métodos propuestos, decidimos adaptar y utilizar el marco de trabajo metodológico propuesto en esta tesis para valorar la eficacia de los métodos bajo diferentes protocolos de evaluación. Esta evaluación es presentada en:

- Campos, P.G., Bellogín, A., Cantador, I., Díez, F. (2013). **Time-Aware Evaluation of Methods for Identifying Active Household Members in**

Recommender Systems, *Proceedings of the 15th Spanish Conference on Artificial Intelligence* (CAEPIA 2013), Madrid, España. Pendiente de publicación.

La contribución de este estudio fue doble. Por un lado, mostramos que el poder de discriminación de los métodos propuestos varía considerablemente al ser medidos con diferentes metodologías. Por otro lado, mostramos la flexibilidad y extensibilidad del marco de trabajo metodológico propuesto en esta tesis, empleándolo para la evaluación de modelos predictivos conscientes del tiempo destinados a una tarea diferente de aquella para la cual el marco de trabajo fue originalmente diseñado.

Contribuciones relacionadas

La observación de las dificultades para comparar la eficacia de diferentes SRCT surgió a partir de un estudio comparativo sobre la eficacia de SRCT en diversas dimensiones de evaluación, realizado en el Trabajo de Fin de Máster del autor, titulado "Temporal Models in Recommender Systems: An Exploratory Study on Different Evaluation Dimensions" (Campos, 2011). La revisión y comparación de resultados publicados, realizada en dicho trabajo, nos mostró la necesidad de contar con un protocolo de evaluación para sistemas de recomendación conscientes del tiempo más fiable. De esta forma, dicho trabajo sirvió de germen para desarrollar las contribuciones de esta tesis.

Durante la realización de la tesis, se publicaron otras contribuciones en temas relacionados con sistemas de recomendación. Específicamente, investigamos 1) heurísticas para recomendación consciente del tiempo, 2) aproximaciones de recomendación capaces de explotar otros tipos de información de contexto, y 3) aproximaciones alternativas para identificar usuarios activos en cuentas compartidas. La primera propuesta sirvió como base para explorar nuevas aproximaciones a SRCT descritas en la Sección 6.2. La segunda corresponde a extensiones de las propuestas presentadas en el capítulo 6, capaces de explotar todo tipo de información de contexto. La tercera corresponde a una nueva aproximación para abordar la tarea descrita en el Capítulo 7.

Heurísticas para recomendación consciente del tiempo

Evaluamos heurísticas para explotar información de contexto temporal en:

- Campos, P.G., Bellogín, A, Díez, F., Chavarriaga, J.E. **Simple Time-Biased KNN-based recommendations**. *Workshop Challenge on Context-aware Movie Recommendation* (CAMRa 2010), desarrollado conjuntamente con 4th ACM Conference on Recommender Systems, pp. 20-23, Barcelona, España.

Las heurísticas estudiadas en este trabajo permiten adaptar recomendaciones basadas en kNN por medio de la explotación exclusiva de ratings en el entorno temporal cercano del momento de recomendación. Así, estas heurísticas ayudan a mejorar los resultados de recomendación proporcionados por el algoritmo kNN, mientras que reducen la cantidad de información requerida para generar recomendaciones.

Recomendación consciente del contexto basada en modelo

También investigamos diferentes aproximaciones de recomendación consciente del contexto basadas en modelo, capaces de explotar diferentes tipos de información de contexto. Una propuesta que explota información de contexto social fue presentada en:

- Díez, F., Chavarriaga, J.E., Campos, P.G., Bellogín, A. (2010) **Movie Recommendations based in explicit and implicit features extracted from the Filmtipset dataset**. *Proceedings of the Workshop Challenge on Context-aware Movie Recommendation* (CAMRa 2010), desarrollado conjuntamente con 4th ACM Conference on Recommender Systems 2010 (RecSys 2010), pp. 45-52, Barcelona, España.

En este trabajo, utilizamos diferentes algoritmos de filtrado colaborativo basados en Caminos Aleatorios para explotar información de contexto social en la forma de relaciones de amistad en un conjunto de datos de ratings de películas. Utilizando un enfoque diferente, probamos SRCC basados en contenido en:

- Fernández-Tobías, I., Campos, P.G., Cantador, I., Díez, F. (2013). **A Contextual Modeling Approach for Model-based Recommender Systems**, *Proceedings of the 15th Spanish Conference on Artificial Intelligence* (CAEPIA 2013), Madrid, España. Pendiente de publicación.

En este trabajo evaluamos diferentes algoritmos de aprendizaje automático que explotan patrones de usuarios que incluyen preferencias de género de películas e información de contexto social en la forma de compañía social, además del contexto temporal y espacial (de localización) en los cuales los usuarios prefieren ver películas y escuchar música. Estos trabajos mostraron la capacidad de las aproximaciones propuestas de mejorar la eficacia de las recomendaciones a partir de la explotación de información de contexto.

Modelado basado en teoría de juegos para identificar usuarios activos en cuentas compartidas

Probamos diferentes aproximaciones de modelado con el fin de abordar la novedosa tarea de identificar usuarios en cuentas compartidas. Una de tales aproximaciones se describe en:

- Díez, F., Campos, P.G. (2012). **Identificación de usuarios en Sistemas de Recomendación mediante un modelo basado en Teoría de Juegos**. *II Congreso Español de Recuperación de Información* (CERI 2012), Valencia, España.

Una de las contribuciones más interesantes de este trabajo, además de la novedad de emplear un esquema de modelado basado en Teoría de Juegos, consiste en el enfoque de seleccionar dinámicamente las mejores fuentes de información de forma independiente para cada cuenta de usuario compartida.

A.5 Estructura de la tesis

Esta tesis se ha dividido en tres partes. La primera parte revisa la literatura sobre sistemas de recomendación en general, y sobre sistemas de recomendación conscientes del tiempo en particular. La segunda parte caracteriza un protocolo de evaluación robusto para sistemas de recomendación conscientes del tiempo, basado en la identificación y análisis de las condiciones que conducen las metodologías de evaluación; y evalúa el efecto de utilizar diferentes condiciones en los resultados de recomendación medidos. Las condiciones identificadas dan forma a un marco de trabajo metodológico para la evaluación de SRCT. La tercera y última parte presenta diferentes aplicaciones que explotan información de contexto temporal, tomando ventaja del marco de trabajo propuesto para proporcionar mediciones más fiables de las mejoras debidas al uso de modelos conscientes del tiempo. Concretamente, los contenidos de esta tesis se distribuyen de la siguiente forma:

Parte I. Estado del arte: Sistemas de recomendación y contexto temporal

- El **Capítulo 2** proporciona una visión general del estado del arte en sistemas de recomendación, considerando tareas de recomendación, tipos de retroalimentación de usuario, técnicas y evaluación de estos sistemas.

- El **Capítulo 3** presenta una revisión exhaustiva del estado del arte en sistemas de recomendación conscientes del tiempo, considerando una clasificación de las principales aproximaciones en la literatura sobre el modelado y la explotación de información de contexto temporal. Adicionalmente, se discuten las metodologías y métricas utilizadas en la evaluación de estos sistemas.

Parte II. Caracterización de un protocolo de evaluación robusto de recomendaciones conscientes del tiempo

- El **Capítulo 4** analiza las cuestiones metodológicas clave involucradas en el diseño de protocolos para evaluar sistemas de recomendación conscientes del tiempo, y formaliza un conjunto de condiciones que abordan estas cuestiones. A partir de las condiciones establecidas, se define un marco de trabajo metodológico cuyo propósito es caracterizar el proceso de evaluación de SRCT.

- El **Capítulo 5** presenta una clasificación del estado del arte en la literatura de SRCT basada en las condiciones clave utilizadas en su evaluación, y describe un análisis empírico de dichas condiciones. A partir del análisis de los resultados obtenidos, se proporciona un conjunto de guías generales para seleccionar condiciones apropiadas para evaluar SRCT particulares.

Parte III. Explotación de información de contexto temporal en tareas de recomendación

- El **Capítulo 6** expone una comparación de diferentes propuestas de SRCT sobre dos tareas de recomendación habituales, a saber, predicción de rating y recomendación de los N-mejores. Se proponen nuevas heurísticas, así como adaptaciones y ajustes a algunas propuestas, que mejoran la explotación de señales de contexto temporal. Tomando ventaja del marco de trabajo metodológico propuesto se proporciona un escenario de evaluación ecuánime y común, con el fin de obtener una medición fiable de las mejoras de eficacia. También se detalla, sobre un estudio de usuario llevado a cabo para recolectar información explícita de contexto temporal de los usuarios, la cual sirve como fuente de entrada para los SRCT evaluados.

- El **Capítulo 7** describe nuevos métodos conscientes del tiempo desarrollados para abordar una tarea relacionada con sistemas de recomendación: la identificación de usuarios activos en cuentas compartidas (en el hogar). Los métodos propuestos, basados en la explotación de información de contexto temporal asociada a eventos de rating, son valorados bajo diferentes escenarios de evaluación proporcionados por la adaptación para la evaluación de esta tarea del marco de trabajo metodológico propuesto anteriormente.

- El **Capítulo 8** concluye la tesis con un resumen de las principales contribuciones y una discusión sobre líneas de trabajo futuro.

Appendix B

Conclusiones

El trabajo presentado en esta tesis estuvo motivado originalmente por la necesidad de comprender y mejorar la explotación de la información de contexto temporal por parte de los sistemas de recomendación en la actualidad. Para cumplir con dicho objetivo, en la tesis nos hemos centrado sobre la definición, formulación y uso de protocolos de evaluación robustos para la evaluación exhaustiva y ecuánime de diferentes modelos de recomendación. A partir de lo anterior adaptamos y propusimos métodos conscientes del tiempo para diferentes tareas de recomendación, basadas en una medición más fiable de las mejoras obtenidas. Más específicamente, hemos abordado los siguientes objetivos de investigación:

- La caracterización de condiciones involucradas en la evaluación de sistemas de recomendación conscientes del tiempo
- El análisis del efecto de diferentes condiciones de evaluación en la medición de la eficacia de las recomendaciones conscientes del tiempo
- La adaptación de propuestas de recomendación existentes para hacer un mejor uso de la información de contexto temporal disponible.
- La explotación de información de contexto temporal en una tarea de recomendación novedosa.

En la primera parte de esta tesis hemos revisado las aproximaciones existentes para generar y evaluar recomendaciones, con particular énfasis y detalle en las aproximaciones de recomendación consciente del tiempo. A partir de dicha revisión exhaustiva, en la segunda parte de esta tesis hemos formalizado, analizado, y comparado empíricamente las condiciones que conducen el proceso de evaluación de SRCT, y hemos propuesto un marco de trabajo de descripción metodológica que permite declarar de manera precisa las condiciones utilizadas en la evaluación de un SRCT particular. Más aún, hemos propuesto un conjunto de guías cuyo propósito es ayudar a seleccionar las condiciones apropiadas para una evaluación fiable de SRCT. Finalmente, en la tercera parte de esta tesis, hemos presentado diferentes aplicaciones de propuestas conscientes del tiempo en tareas de recomendación, proponiendo nuevas heurísticas, así como adaptaciones de métodos existentes en el caso de las bien establecidas tareas de predicción de rating y recomendación de los N-mejores, y desarrollando nuevos métodos en el caso de la recientemente propuesta tarea de identificar usuarios activos en cuentas compartidas. Hemos utilizado el marco de trabajo propuesto para evaluar la eficacia de las heurísticas, adaptaciones y métodos propuestos.

En este capítulo, presentamos las principales conclusiones de nuestro trabajo. En la Sección 8.1 resumimos las contribuciones de esta tesis. En la Sección 8.2 detallamos la validación de las hipótesis planteadas y, en la Sección 8.3, describimos algunas directrices para el potencial trabajo futuro.

B.1 Resumen y discusión de contribuciones

En las siguientes subsecciones resumimos y discutimos las principales contribuciones de esta tesis, con respecto a los objetivos de investigación y las hipótesis planteados en el Capítulo 1. Estas contribuciones se encuentran organizadas de acuerdo a los objetivos de investigación abordados. En primer lugar, investigamos las condiciones que guían el proceso de evaluación de Sistemas de Recomendación Conscientes del Tiempo (SRCT). En segundo lugar, analizamos las diferencias en las mediciones de la eficacia de las recomendaciones a consecuencia de los cambios en las condiciones de evaluación utilizadas, con el fin de establecer un conjunto de condiciones que den lugar a un protocolo de evaluación robusto. En tercer lugar, propusimos nuevas heurísticas, así como adaptaciones a aproximaciones de recomendación existentes, con el objeto de mejorar la explotación de la información de contexto temporal. Y en cuarto lugar, propusimos novedosos métodos conscientes del tiempo para la tarea de identificación de usuarios activos en cuentas compartidas.

B.1.1 Caracterización de las condiciones involucradas en la evaluación de SRCT

A partir de una revisión exhaustiva de la literatura publicada sobre sistemas de recomendación conscientes del tiempo, observamos que los resultados y conclusiones existentes acerca de cómo incorporar y explotar información temporal en el proceso de recomendación, parecen contradictorios en algunos casos. Nosotros planteamos como hipótesis que tales discrepancias pueden ser causadas por diferencias significativas en los protocolos de evaluación utilizados –métricas y metodologías. Una revisión cuidadosa de tales protocolos de evaluación nos mostró múltiples diferencias metodológicas en las evaluaciones llevadas a cabo en diferentes trabajos. Concretamente, en la Sección 4.1 observamos que el proceso de partición de datos en conjuntos de entrenamiento y prueba es una importante fuente de divergencia metodológica, particularmente cuando se encuentran disponibles las marcas de tiempo de los ratings. Identificamos diversas decisiones de diseño que deben tenerse en cuenta al definir un escenario de evaluación, las cuales llevan a diferencias metodológicas. Analizando dichas diferencias, planteamos un conjunto de **preguntas metodológicas** clave respecto del diseño de un protocolo de evaluación para SRCT:

- MQ1: ¿Qué conjunto de ratings base se utiliza para realizar la partición de datos en conjuntos de entrenamiento y prueba?

- MQ2: ¿Qué ordenamiento de los ratings se utiliza para asignar estos a los conjuntos de entrenamiento y prueba?

- MQ3: ¿Cuántos ratings conforman los conjuntos de entrenamiento y prueba?

- MQ4: ¿Qué método de validación cruzada se emplea para incrementar la fiabilidad de la generalización de los resultados de la evaluación?

Adicionalmente a estas preguntas, también establecimos condiciones específicas para el diseño de la evaluación de la tarea de recomendación de los N-mejores, plateando las dos siguientes preguntas metodológicas:

- MQ5: ¿Qué ítems se consideran como ítems objetivo (en la tarea de recomendación de los N-mejores)?
- MQ6: ¿Qué ítems se consideran relevantes para cada usuario (en la tarea de recomendación de los N-mejores)?

Abordamos la respuesta a estas preguntas empleando un conjunto de **condiciones de evaluación** que planteamos a partir de la revisión de los escenarios de evaluación encontrados en la literatura de SRCT. Estas condiciones ayudan a la toma de decisiones relacionadas con los procesos de partición de datos en conjuntos de entrenamiento y prueba y de validación cruzada en la evaluación de SR, y con aspectos específicos con respecto a la evaluación de la recomendación de los N-mejores ítems. Así, específicamente, en el Capítulo 4 caracterizamos y formalizamos las siguientes condiciones de evaluación:

- *Condiciones de conjunto base de ratings centrada en la comunidad* y *centrada en el usuario*, las cuales especifican si se deben aplicar las restantes condiciones bien sobre todo el conjunto de ratings, o bien independientemente sobre el conjunto de ratings de cada usuario, en respuesta a la pregunta MQ1.
- *Condiciones de ordenamiento de ratings dependiente del tiempo* e *independiente del tiempo*, las cuales indican si se debe ordenar temporalmente o no el conjunto de ratings, en respuesta a la pregunta MQ2.
- *Condiciones de tamaño de conjunto de ratings basado en proporción*, *fijo* y *basado en tiempo*, las cuales indican el criterio utilizado para definir los tamaños de los conjuntos de entrenamiento y prueba, en respuesta a la pregunta MQ3.
- *Condiciones de validación cruzada dependiente del tiempo* o *independiente del tiempo*, las cuales indican los métodos de validación cruzada aplicables dependiendo de la compatibilidad con las restricciones de ordenamiento temporal de los ratings, en respuesta a la pregunta MQ4.
- C*ondiciones de ítem objetivo basado en el usuario*, *basado en la comunidad*, *one-plus-random* y *otras condiciones de ítem destino*, las cuales indican el criterio usado para determinar el conjunto de ítems a ser ordenado en la tarea de recomendación de los N-mejores, en respuesta a la pregunta MQ5.

- *Condiciones de ítem relevante basado en el conjunto de prueba* y *basado en umbral*, las cuales indican el criterio utilizado para determinar la relevancia de los ítems, en respuesta a la pregunta MQ6.

Estas condiciones en conjunto cubren el amplio espectro de alternativas de decisiones de diseño usadas en el proceso de evaluación en la literatura de SRCT.

Basándonos en las condiciones definidas, desarrollamos un **marco de trabajo metodológico**, cuyo propósito es facilitar la comprensión de tales condiciones. Más aún, el marco de trabajo pretende hacer el proceso de evaluación ecuánime y reproducible bajo diferentes circunstancias, al declarar de forma clara y meticulosa el escenario utilizado en la evaluación de SRCT. El formalismo empleado en el marco de trabajo incluye la definición de un **procedimiento de partición**, descrito en la Sección 4.2, el cual, tomando como entrada un conjunto de condiciones de evaluación (escenario de evaluación), permite realizar y reproducir particiones de datos en conjuntos de entrenamiento y prueba. Haciendo uso del procedimiento de partición y de las distintas combinaciones de las condiciones incluidas en el marco de trabajo, es posible definir de forma exacta los diversos escenarios de evaluación de SRCT, como se mostró por medio de los ejemplos incluidos en la Sección 4.3.4.

Adicionalmente, en la Sección 5.1 realizamos una exhaustiva **clasificación del estado del arte en SRCT** en términos de las condiciones de evaluación caracterizadas, mapeando tales condiciones hacia los escenarios de evaluación utilizados en la literatura de sistemas de recomendación conscientes del tiempo y proporcionando una visión general de las condiciones y metodologías más comúnmente usadas en la evaluación de tales sistemas. Encontramos que casi un 25% de los estudios revisados utilizan un ordenamiento de ratings independiente del tiempo –a pesar del hecho que los artículos revisados versan sobre propuestas conscientes del tiempo–, y que aproximadamente un 40% de los estudios usan una combinación de conjunto de ratings base centrado en la comunidad y ordenamiento de ratings dependiente del tiempo –condiciones tales que proporcionan el escenario de evaluación más similar al existente en el mundo real–. También encontramos una distribución equitativa en el uso de los criterios sobre el tamaño del conjunto de ratings, así como un uso escaso de métodos de validación cruzada. Respecto de las condiciones específicas para evaluar la recomendación de los N-mejores, observamos que la mayor parte de los artículos relacionados con SRCT que abordan esta tarea utilizan un criterio basado en el conjunto de prueba para definir la relevancia de los ítems, y una distribución equitativa en el uso de los criterios para seleccionar los conjuntos de ítems objetivo.

B.1.2 Análisis del efecto de las diferentes condiciones de evaluación en la medición de la eficacia de los SRCT

Junto con la caracterización de las condiciones de evaluación presentadas en el Capítulo 4, **discutimos el efecto** de utilizar diferentes condiciones para abordar cada pregunta metodológica clave involucrada en la evaluación de SRCT (MQ1 – MQ6). Como conclusión, observamos las importantes diferencias resultantes de aplicar las condiciones de partición de datos sobre el conjunto total de ratings en un conjunto de datos –esto es, usando una condición de conjunto de ratings base centrado en la comunidad– versus la aplicación de tales condiciones de forma independiente sobre los datos de cada usuario –i.e., usando una condición de conjunto de ratings base centrado en el usuario. Más aún, notamos las diferencias en las particiones de datos generadas inducidas por la aplicación de una condición de ordenamiento de ratings independiente del tiempo, o de forma alternativa, dependiente del tiempo. Con el fin de estudiar la influencia del uso de diferentes combinaciones de condiciones de evaluación en la medición la eficacia de las recomendaciones, realizamos una **comparación empírica** de varios SRCT empleando diversos protocolos de evaluación. En particular, en el Capítulo 5 presentamos la evaluación de tres propuestas de SRCT y una de recomendación no contextual ampliamente utilizadas, usando cuatro metodologías de evaluación diferentes. Los resultados obtenidos mostraron que el uso de diferentes condiciones de evaluación da lugar no sólo a diferencias importantes entre las métricas que miden diferentes propiedades de recomendación –a saber exactitud, precisión, novedad y diversidad–, sino que también pueden afectar al orden relativo de los algoritmos respecto de una métrica en particular.

Basándonos en los experimentos y análisis realizados, presentamos las cuestiones metodológicas clave que una evaluación robusta de SRCT debe tener en consideración para realizar una medición ecuánime de la eficacia de la recomendación, y facilitar la comparación entre experimentos publicados. A partir de esto, en la Sección 5.3 concluimos un conjunto de **guías metodológicas** destinadas a facilitar la selección de condiciones para una evaluación apropiada de SRCT. Para la evaluación de la tarea de predicción de rating, las guías propuestas sugieren realizar la partición en conjuntos de entrenamiento y prueba basada en un ordenamiento de ratings dependiente del tiempo sobre todo el conjunto de ratings, aplicando un criterio de tamaño basado en proporción y utilizando un método de validación cruzada compatible con las condiciones de partición de datos sugeridas. Para la evaluación de la tarea de recomendación de los N-mejores, nuestras guías sugieren ordenar un conjunto de ítems que incluya algunos para los cuales la relevancia para el usuario sea completamente desconocida. Este es el escenario más común en las aplicaciones del mundo real. Utilizando una condición de ítem relevante basada en umbral, la cual descarta del conjunto de ítems relevantes aquellos con valoraciones o consumos bajos, proporciona una interpretación más confiable de la relevancia de los ítems.

B.1.3 Adaptación de propuestas de recomendación existentes para hacer un mejor uso de la información de contexto temporal disponible

Las guías de evaluación propuestas en esta tesis nos permitieron establecer un escenario de evaluación ecuánime y común para medir la eficacia de diferentes modelos de recomendación que explotan información de contexto temporal. A partir del análisis de estos resultados y de las características de los modelos, en el Capítulo 6 propusimos **nuevas heurísticas y adaptaciones**, con el fin de hacer un mejor uso de la información de contexto temporal disponible.

En particular, en la Sección 6.2.1 propusimos un **nuevo criterio de impureza basado en la prueba exacta de Fisher**, para ser utilizado por el algoritmo *Item Splitting* (Baltrunas y Ricci, 2009a, 2009b) –una propuesta general de pre-filtrado consciente del contexto. Adicionalmente, en la Sección 6.3.2 ajustamos diferentes criterios de impureza usados por *Item Splitting* para encontrar los umbrales óptimos que disminuyen el error en la predicción de rating al explotar diferentes contextos temporales. También desarrollamos, en la Sección 6.2.2, una **nueva estrategia de post-filtrado basada en la probabilidad de valorar un ítem en el contexto de recomendación objetivo**. Esta heurística permite realizar la contextualización de post-filtrado de las recomendaciones generadas por el algoritmo de recomendación de factorización de matrices (Takács et al., 2008; Koren et al., 2009). Por su parte, en la Sección 6.2.3 adaptamos el método de vecinos contextuales (Panniello y Gorgoglione, 2012) – una aproximación general de modelado contextual– eliminando restricciones consideradas en la propuesta original para controlar el tipo de contextualización utilizada, con el fin de permitir la utilización de diferentes algoritmos de recomendación junto a dicho método.

Adicionalmente, a partir del análisis de las sugerencias dadas en las guías metodológicas propuestas en el Capítulo 5 para la tarea de recomendación de los N-mejores ítems, y las particularidades de las propuestas estudiadas –que son capaces de utilizar representaciones categóricas del contexto temporal–, propusimos y utilizamos una **nueva metodología para medir recomendaciones contextualizadas de los N-mejores**. Esta nueva metodología, descrita en la Sección 6.4.1, permite construir listas ordenadas de ítems destinadas al mismo contexto temporal, independientemente de la representación temporal utilizada –una cuestión no abordada previamente en la literatura de SRCT–, al tiempo que incluye ítems no valorados en la lista. De esta forma, la metodología propuesta proporciona un escenario de evaluación más similar a aquel que los SRCT en funcionamiento deben enfrentar –ordenar correctamente ítems no valorados para un contexto de destino dado, con el fin de recomendar aquellos relevantes– que aquellos resultantes de otras metodologías previamente usadas en la literatura.

Evaluamos las adaptaciones propuestas, junto con otras aproximaciones capaces de explotar información de contexto temporal –a saber pre-filtrado exacto (Adomavicius y

Tuzhilin, 2011) y post-filtrado (Panniello et al., 2009a)– en un conjunto de datos de preferencias de películas de usuarios reales enriquecido con información de contexto. Los resultados obtenidos, discutidos en las Secciones 6.3.3 y 6.4.4, mostraron la importancia de **seleccionar un umbral apropiado** para cada combinación de criterio de impureza y señal de contexto temporal en el caso de *Item Splitting*, para poder obtener la mejor eficacia en la recomendación posible.

Adicionalmente, estos resultados también revelaron que **no existe un único SRCT dominante** tanto en la tarea de predicción de rating como de recomendación de los N-mejores, y que las mejoras logradas por las propuestas evaluadas dependen del algoritmo de recomendación subyacente y el contexto temporal explotado. Este hallazgo es concordante con las conclusiones de investigaciones previas que comparan SR conscientes del contexto en aplicaciones de comercio electrónico, p. ej. (Panniello et al., 2009a). De esta forma, la identificación de la mejor aproximación requiere de una evaluación y comparación exhaustiva de implementaciones de SRCT candidatas sobre el conjunto de datos sobre el que desean utilizarse. Más aún, algunas aproximaciones de contextualización pueden requerir de una prueba intensiva de parámetros, como es el caso de *Item Splitting*.

A pesar de lo antes mencionado, observamos que las heurísticas propuestas en el Capítulo 6 –el nuevo criterio de impureza para *Item Splitting* y la estrategia de post-filtrado para factorización de matrices– son capaces de **contextualizar de manera efectiva las recomendaciones** generadas por el eficiente algoritmo de recomendación de factorización de matrices. Más aún, éstos mostraron los **mejores valores globales** en la mayoría de las métricas de las tareas de predicción de rating y recomendación de los N-mejores, respectivamente, en la comparación de propuestas realizada. De esta forma, el uso de las heurísticas propuestas en conjunto con un algoritmo de recomendación basado en factorización de matrices puede considerarse como una buena aproximación para contextualizar recomendaciones cuando se encuentra disponible información de contexto temporal sobre las preferencias de usuario.

B.1.4 Explotación de información de contexto temporal en una tarea de recomendación novedosa

La explotación de diferentes contextos temporales asociados a ratings de usuario nos permitió abordar una tarea de recomendación fuera del ámbito de las tareas habituales de predicción de rating y recomendación de los N-mejores: la identificación de usuarios activos en cuentas compartidas (en el hogar) (Berkovsky et al., 2011). En el capítulo 7 **propusimos y evaluamos un conjunto de métodos** para identificar de forma efectiva qué usuario de una cuenta compartida en el hogar está interactuando con un sistema de recomendación en un momento dato, explotando sólo conocimiento sobre interacciones pasadas con el sistema. Los métodos propuestos se basan en la identificación de diferencias en los hábitos de valoración temporales de los usuarios, descritos en términos de varias señales de contexto temporal o atributos que incluyen la *fecha absoluta* (p. ej. 1 de Noviembre de 2013), el *día de la semana* (p. ej. Lunes) y la *hora del día* (p. ej. 4:00 p.m.) en la cual los usuarios interactúan con el sistema. Formulamos la tarea como un **problema de clasificación** y usamos los atributos temporales para discriminar entre usuarios en un hogar por medio de diversos algoritmos de clasificación.

Adicionalmente, también **adaptamos metodologías usadas en la evaluación de SRCT** con el fin de medir de forma confiable el rendimiento de los métodos propuestos para la tarea de identificar usuarios activos en cuentas compartidas. Esto requirió de la formalización de una nueva condición, específica para la tarea –la condición de conjunto de ratings base centrada en la cuenta compartida– la cual fue incorporada en el marco de trabajo metodológico propuesto, como fue descrito en la Sección 7.4. Empleando la estructura conceptual del marco de trabajo, fuimos capaces de especificar metodologías basadas en las guías propuestas para la evaluación de SRCT en el Capítulo 5, las cuales fueron usadas en la evaluación de la tarea. Lo anterior también mostró la extensibilidad y facilidad de integración de nuevas condiciones de evaluación del marco de trabajo propuesto.

Los resultados obtenidos en los experimentos mostraron que algunos de los algoritmos propuestos más elementales fueron capaces de obtener elevados valores de exactitud al explotar ciertos contextos temporales. A pesar de ello, observamos que el poder de discriminación de los diferentes contextos temporales, de forma individual y combinados, **variaron considerablemente cuando fueron evaluados con diferentes metodologías**. Encontramos que las metodologías menos estrictas desde un punto de vista temporal –esto es, metodologías que no evitan un solapamiento temporal de los datos de entrenamiento y prueba– proporcionaron resultados menos confiables. A partir de estos resultados se justifica la importancia de seguir **protocolos de evaluación robustos** tales como aquellos sugeridos por las guías metodológicas propuestas en esta tesis.

B.2 Validación de las hipótesis planteadas

En esta sección proporcionamos detalles acerca de la validación de las hipótesis planteadas al comienzo de nuestro trabajo. Su validez fue probada por medio de los resultados experimentales obtenidos en la tesis.

Hipótesis 1: Variaciones en el protocolo de evaluación llevan a diferencias en la medición de resultados de recomendación.

Los resultados obtenidos en la comparación empírica de metodologías para la evaluación de SRCT, presentada en el Capítulo 5 y discutida adicionalmente en la Sección 8.1.2, nos permitió probar esta hipótesis. En particular, estos resultados muestran diferencias en los valores absolutos de las métricas, y de manera más importante, en el ordenamiento relativo de los algoritmos respecto de una métrica en concreto, cuando se utilizan diferentes escenarios de evaluación. A partir de lo anterior, remarcamos que la comparación de aproximaciones de SRCT bajo distintos protocolos de evaluación puede dar lugar a **resultados completamente diferentes**. Todo esto enfatiza la importancia de contar con protocolos fiables para la evaluación de SRCT.

La validez de esta hipótesis tiene importantes implicaciones para la reproducibilidad y comparabilidad de los resultados presentados en la literatura de SRCT. Utilizando las mismas métricas y metodologías de evaluación es posible reproducir y comparar de forma ecuánime los resultados de diferentes trabajos en SRCT. A partir de esto, resaltamos la necesidad de **declarar claramente las condiciones** bajo las cuales se realizan experimentos *offline* para evaluar SR en general y SRCT en particular. Más aún, siguiendo condiciones de evaluación consensuadas y ecuánimes –i.e. un protocolo de evaluación robusto– será posible la reproducibilidad de los experimentos y se facilitará la comparación de los modelos de recomendación propuestos. Para contribuir a este propósito desarrollamos el marco de trabajo de descripción metodológica presentado en esta tesis.

Hipótesis 2: La explotación adecuada de la información de contexto temporal lleva a mejoras en los resultados de recomendación medidos.

Los resultados obtenidos en la evaluación de métodos para tareas de recomendación bien establecidas –a saber predicción de rating y recomendación de los N-mejores– presentada en el Capítulo 6 y discutida adicionalmente en la sección 8.1.3, nos permitió probar esta hipótesis. En particular, estos resultados mostraron que por medio de **la selección apropiada** de la señal de contexto temporal, el algoritmo de recomendación subyacente y los parámetros específicos requeridos por algunas propuestas, es posible mejorar las recomendaciones generadas por métodos que explotan el contexto temporal. La eficacia de los modelos propuestos en los experimentos depende en gran medida del algoritmo de recomendación subyacente. Sin embargo, notamos que por ejemplo una selección apropiada de valores umbral para el criterio de impureza utilizado por *Item Splitting* permite mejorar los resultados de algoritmos muy eficaces tales como es el de factorización de matrices.

La validez de esta hipótesis es concordante con resultados presentados en investigaciones previas en el área. A pesar de esto, remarcamos que no todos los métodos que explotan información de contexto temporal son capaces de obtener mejores resultados que aquellos que no explotan tal información. Más aún, el hecho de observar mejoras **depende del protocolo de evaluación seguido**, tal como mostraron los resultados presentados en la Sección 5.2.4. Nuestros resultados indican que se requiere de una **cuidadosa selección** de los parámetros de los métodos, las señales de contexto temporal y los algoritmos de recomendación subyacentes para mejorar de forma efectiva la eficacia de las recomendaciones, cuando se sigue un protocolo de evaluación robusto.

Hipótesis 3: Desde un punto de vista temporal, el uso de un protocolo de evaluación robusto para modelos y técnicas de recomendación que explotan información de contexto temporal lleva a un descenso en la eficacia medida con respecto a un protocolo de evaluación menos robusto.

Los resultados obtenidos en la comparación experimental de los métodos que explotan información de contexto temporal para la identificación de usuarios activos en cuentas compartidas, que incluyeron metodologías con condiciones de ordenamiento de ratings independientes y dependientes del tiempo, presentados en el Capítulo 7 y discutidos adicionalmente en la Sección 8.1.4, nos permitieron probar esta hipótesis. En particular, estos resultados mostraron un rendimiento excepcional de los métodos propuestos en la tarea –sobre un 95% de exactitud en la identificación del usuario activo– cuando se utilizan metodologías basadas en una condición de ordenamiento de ratings independiente del tiempo. Sin embargo, los mismos métodos mostraron un **importante descenso en la eficacia** cuando se valoraron con metodologías basadas en una condición de ordenamiento de ratings dependiente del tiempo. Más aún, la menor eficacia de los métodos –que estuvo alrededor del 65% de exactitud– fue medida usando las condiciones de evaluación sugeridas por nuestras guías metodológicas.

La validez de esta hipótesis resalta la importancia de definir y utilizar un protocolo de evaluación robusto para medir de forma precisa el grado de mejora obtenida de la explotación de información de contexto temporal por parte de aproximaciones de SRCT. En este contexto, las guías metodológicas propuestas en el Capítulo 5 son una poderosa herramienta para aumentar la confiabilidad de la medición de la eficacia.

B.3 Trabajo futuro

En esta tesis hemos presentado una revisión y análisis exhaustivos de los protocolos utilizados en la evaluación de SRCT, que nos han llevado a definir un marco de trabajo metodológico y un conjunto de guías para proporcionar escenarios de evaluación robustos para SRCT. Más aún, hemos propuesto y evaluado adaptaciones y nuevos métodos para explotar información de contexto temporal. A pesar las importantes contribuciones y hallazgos presentados, la investigación realizada en esta tesis da lugar a interesantes preguntas de investigación adicionales respecto del desarrollo y evaluación de SRCT. En las siguientes subsecciones discutimos un conjunto de cuestiones que requieren de más investigación, e introducimos posibles líneas de trabajo para abordar tales cuestiones.

B.3.1 Evaluación de aproximaciones de recomendación consciente del tiempo

El marco de trabajo metodológico propuesto en esta tesis se compone de un conjunto de condiciones que permiten definir el escenario en el cual se evalúa una propuesta de recomendación, cubriendo las metodologías de evaluación utilizadas en la literatura de SRCT. La evaluación presentada en el Capítulo 5 nos proporcionó evidencias sobre el efecto en la medición de la eficacia de las recomendaciones ante cambios en las condiciones de evaluación utilizadas. Estos hechos nos llevaron a proponer un conjunto de guías para seleccionar las condiciones de una evaluación robusta de SRCT. A pesar de ello, se requiere de experimentación adicional para analizar de forma apropiada el impacto de combinaciones de condiciones no abordadas en nuestro estudio. Además se deberían definir nuevas condiciones que permitan aplicar las guías propuestas en otras tareas de recomendación donde el contexto temporal pueda ser explotado y que no han sido estudiadas en esta tesis, tales como la tarea de recomendación de secuencias (Herlocker et al., 2004). En la consecución de tales propósitos, creemos que el marco de trabajo propuesto proporciona una importante estructura conceptual para guiar esta investigación.

Otra importante cuestión pendiente está relacionada con el análisis de la relación entre diferentes características de los conjuntos de datos (p. ej. tamaño del perfil de usuario, intervalos temporales y niveles de escasez de ratings) y el efecto en la eficacia debido al uso de diferentes protocolos de evaluación. Más allá de esto, lo apropiado de utilizar ciertas condiciones de evaluación al usar conjuntos de datos con distribuciones de ratings a través del tiempo/usuarios/ítems, tipos de retroalimentación, dominios, etc. particulares, debe ser investigado.

La relación entre las métricas de exactitud y novedad/diversidad también permanecen como una cuestión abierta respecto de la evaluación. Dada la importancia creciente de estas últimas métricas en el campo de SR, se requieren análisis y explicaciones adicionales con el fin de proporcionar recomendaciones conscientes del tiempo con niveles adecuados de tales propiedades. Por ejemplo, tal como lo nota Lathia et al. (2010), desde un punto de vista temporal, la diversidad de las recomendaciones es una importante faceta que un sistema de recomendación debe considerar.

Una pregunta adicional es acerca de las mejoras en la eficacia de los SRCT medidas por resultados de evaluaciones offline y cómo son percibidas por parte de usuarios reales. Tal como lo nota por ejemplo Knijnenburg et al. (2012), las mejoras en la exactitud no son necesariamente observables por los usuarios. La falta de estudios de evaluación *online* en SRCT es una limitación mayor para abordar esta pregunta.

B.3.2 Desarrollo de nuevas y mejores aproximaciones de recomendación consciente del tiempo

Utilizando un conjunto de datos de preferencias de películas de usuarios reales enriquecido con información contextual, los experimentos presentados en el Capítulo 6 nos permitieron derivar importantes observaciones respecto de las circunstancias en las que ciertas aproximaciones de recomendación superan a otras. A pesar de esto, las conclusiones obtenidas no son necesariamente generales, debido al tamaño reducido del conjunto de datos usado y al hecho de que sólo se evaluó un dominio de recomendación. La repetición de esta evaluación sobre diferentes conjuntos de datos, de dominios y tipos de retroalimentación diversos, permitirá establecer conclusiones más generales respecto de la aplicabilidad de SRCT y señales contextuales específicos. De hecho, tal como lo indican Adomavicius y Tuzhilin (2011), uno de los principales desafíos en la recomendación consciente del contexto es la investigación de cuales aproximaciones de contextualización son más eficaces y bajo qué circunstancias. En tal evaluación también es importante considerar propiedades de la recomendación más allá de la exactitud y la precisión. Un interesante ejemplo de esto en el ámbito de los sistemas de recomendación conscientes del contexto es el trabajo de Panniello et al. (2013), donde se comparan diferentes aproximaciones de SRCC en términos de exactitud y diversidad. Con dicho propósito remarcamos la importancia de utilizar un protocolo de evaluación común para la reproducibilidad y comparabilidad de resultados.

Advertimos que la revisión de la literatura realizada mostró la existencia de dos tipos principales de SRCT, de acuerdo a cómo se representa la información de contexto temporal, a saber, en una representación continua o discreta. Sin embargo, en la evaluación de propuestas SRCT presentada en el Capítulo 6, nos enfocamos en SRCT basados en esta última representación, debido a las características del conjunto de datos usado. La comparación de tales propuestas con otras basadas en una representación continua del tiempo es por tanto una cuestión que debe ser investigada.

Una línea de investigación adicional es la explotación conjunta de ambos tipos de representación temporal. Una forma de realizar esto sería por medio de la construcción de aproximaciones híbridas (Burke, 2007) que combinen recomendaciones de diferentes SRCT. Otra forma posible de abordar esta cuestión es desarrollar y mejorar aproximaciones basadas en modelo capaces de utilizar ambos tipos de representaciones temporales, tales como la técnica *timeSVD++* propuesta por Koren (2009a), y modelos basados en factorización de tensores tales como *factorización de tensores Probabilística Bayesiana* propuesta por Xiong et al. (2010).

References

Adomavicius, G., Baltrunas, L., de Luca, E.W., Hussein, T., Tuzhilin, A. (2012). 4th Workshop on Context-Aware Recommender Systems (CARS 2012). In: *Proceedings of the Sixth ACM Conference on Recommender Systems* (RecSys'12), pp. 349–350, ACM, New York, NY, USA.

Adomavicius, G., Kwon, Y. (2007). New Recommendation Techniques for Multicriteria Rating Systems. *IEEE Intelligent Systems* 22(3):48–55.

Adomavicius, G., Manouselis, N., Kwon, Y. (2011). Multi-Criteria Recommender Systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Eds.), *Recommender Systems Handbook*, pp. 769–804, Springer US.

Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A. (2005). Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. *ACM Transactions on Information Systems* 23(1):103–145.

Adomavicius, G., Tuzhilin, A. (2001). Multidimensional Recommender Systems: A Data Warehousing Approach. In: *Proceedings of the 2dn International Workshop on Electronic Commerce* (WELCOM'01), pp. 180–192, Springer Berlin Heidelberg, Heidelberg, Germany.

Adomavicius, G., Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering* 17(6):734–749.

Adomavicius, G., Tuzhilin, A. (2011). Context-Aware Recommender Systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Eds.), *Recommender Systems Handbook*, pp. 217–253, Springer US.

Amatriain, X., Jaimes, A., Oliver, N., Pujol, J.M. (2011). Data Mining Methods for Recommender Systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Eds.), *Recommender Systems Handbook*, pp. 39–72.

Amatriain, X., Pujol, J.M., Oliver, N. (2009a). I Like It... I Like It Not: Evaluating User Ratings Noise in Recommender Systems. In: *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization* (UMAP'09), pp. 247–258, Springer-Verlag.

Amatriain, X., Pujol, J.M., Tintarev, N., Oliver, N. (2009b). Rate It Again: Increasing Recommendation Accuracy by User Re-rating. In: *Proceedings of the Third ACM Conference on Recommender Systems* (RecSys'09), pp. 173–180.

Ansari, A., Essegaier, S., Kohli, R. (2000). Internet Recommendation Systems. *Journal of Marketing Research* 37(3):363–375.

Ardissono, L., Gena, C., Torasso, P., Bellifemine, F., Difino, A., Negro, B. (2004). User Modeling and Recommendation Techniques for Personalized Electronic Program Guides. In: Ardissono, L., Kobsa, A., Maybury, M. (Eds.), *Personalized Digital Television*, pp. 3–26, Springer Netherlands.

Ardissono, L., Portis, F., Torasso, P., Bellifemine, F., Chiarotto, A., Difino, A. (2001). Architecture of a System for the Generation of Personalized Electronic Program Guides. In: *UM'01 Workshop on Personalization in Future TV*.

Arlot, S., Celisse, A. (2010). A Survey of Cross-Validation Procedures for Model Selection. *Statistics Survey* 4:40–79.

Baeza-Yates, R.A., Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Balabanović, M., Shoham, Y. (1997). Fab: Content-based, Collaborative Recommendation. *Communications of the ACM* 40(3):66–72.

Baltrunas, L. (2011). *Context-Aware Collaborative Filtering Recommender Systems*. PhD thesis, University of Bozen-Bolzano.

Baltrunas, L., Amatriain, X. (2009). Towards Time-dependant Recommendation Based on Implicit Feedback. In: *Proceedings of the 2009 Workshop on Context-Aware Recommender Systems*.

Baltrunas, L., Ricci, F. (2009a). Context-based Splitting of Item Ratings in Collaborative Filtering. In: *Proceedings of the Third ACM Conference on Recommender Systems* (RecSys'09), pp. 245–248, ACM, New York, NY, USA.

Baltrunas, L., Ricci, F. (2009b). Context-dependent Items Generation in Collaborative Filtering. In: *Proceedings of the 2009 Workshop on Context-Aware Recommender Systems*.

Baltrunas, L., Ricci, F. (2013). Experimental Evaluation of Context-dependent Collaborative Filtering Using Item Splitting. *User Modeling and User-Adapted Interaction* (Special Issue on Context-Aware Recommender Systems).

Bazire, M., Brézillon, P. (2005). Understanding Context before Using It. In: *Proceedings of the 5th International Conference on Modeling and Using Context*, pp. 29–40, Springer-Verlag, Berlin, Heidelberg.

Bell, R.M., Koren, Y., Park, F. (2007). Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights. In: *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining* (ICDM'07), pp. 43–52, IEEE Computer Society, Washington, DC, USA.

Bell, R.M., Koren, Y., Volinsky, C. (2008). *The Bellkor 2008 Solution to the Netflix Prize*. Available from http://www.netflixprize.com.

Bellogín, A., Castells, P., Cantador, I. (2011). Precision-oriented Evaluation of Recommender Systems: An Algorithmic Comparison. In: *Proceedings of the Fifth ACM Conference on Recommender Systems* (RecSys'11), pp. 333–336.

Bennett, J., Lanning, S. (2007). The Netflix Prize. In: *Proceedings of KDD Cup and Workshop 2007*.

Bento, J., Fawaz, N., Montanari, A., Ioannidis, S. (2011). Identifying Users from Their Rating Patterns. In: *Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation* (CAMRa'11), pp. 39–46, ACM Press, New York, NY, USA.

Berkovsky, S., Luca, E.W. De, Said, A. (2011). Challenge on Context-Aware Movie Recommendation: CAMRa2011. In: *Proceedings of the 5th ACM Conference on Recommender Systems* (RecSys'11), pp. 385–386.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Böhmer, M., De Luca, E.W., Said, A., Teevan, J. (2013). 3rd Workshop on Context-awareness in Retrieval and Recommendation. In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining* (WSDM'13), pp. 789–790, ACM, New York, NY, USA.

Breese, J., Heckerman, D., Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (UAI'98), pp. 43–52, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Brenner, A., Pradel, B., Usunier, N., Gallinari, P. (2010). Predicting Most Rated Items in Weekly Recommendation with Temporal Regression. In: *Proceedings of the Workshop on Context-Aware Movie Recommendation* (CAMRa'10), pp. 24–27, ACM, New York, NY, USA.

Burke, R. (2000). Knowledge-based Recommender Systems. In: Kent, A. (Ed.), *Encyclopedia of Library and Information Systems*.

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12(4):331–370.

Burke, R. (2007). Hybrid Web Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (Eds.), *The Adaptive Web*, pp. 377–408, Springer-Verlag, Berlin, Heidelberg.

Campos, P.G., Bellogin, A., Díez, F., Cantador, I. (2012). Time Feature Selection for Identifying Active Household Members. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management* (CIKM'12), pp. 2311–2314, ACM, New York, NY, USA.

Campos, P.G., Bellogín, A., Díez, F., Chavarriaga, J.E. (2010). Simple Time-Biased KNN-based Recommendations. In: *Proceedings of the Workshop on Context-Aware Movie Recommendation* (CAMRa'10), pp. 20–23.

Campos, P.G., Díez, F., Bellogín, A. (2011a). Temporal Rating Habits: A Valuable Tool for Rater Differentiation. In: *Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation* (CAMRa'11), pp. 29–35.

Campos, P.G., Díez, F., Sánchez-Montañés, M. (2011b). Towards a More Realistic Evaluation: Testing the Ability to Predict Future Tastes of Matrix Factorization-based Recommenders. In: *Proceedings of the 5th ACM Conference on Recommender Systems* (RecSys'11), pp. 309–312, ACM, New York, NY, USA.

Cantador, I., Brusilovsky, P., Kuflik, T. (2011). Second Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec2011). In: *Fifth ACM Conference on Recommender Systems* (RecSys'11), pp. 387–388, ACM, New York, NY, USA.

Cao, H., Chen, E., Yang, J., Xiong, H. (2009). Enhancing Recommender Systems Under Volatile User Interest Drifts. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (CIKM'09), pp. 1257–1266, ACM, New York, NY, USA.

Castells, P., Wang, J., Lara, R., Zhang, D. (2011). Workshop on Novelty and Diversity in Recommender Systems - DiveRS 2011. In: *Proceedings of the Fifth ACM Conference on Recommender Systems* (RecSys'11), pp. 393–394, ACM, New York, NY, USA.

Celma, Ò. (2008). *Music Recommendation and Discovery in the Long Tail*. PhD thesis, Universitat Pompeu Fabra.

Chen, P.-L., Tsai, C.-T., Chen, Y.-N., Chou, K.-C., Li, C.-L., Tsai, C.-H., Wu, K.-W., Chou, Y.-C., Li, C.-Y., Lin, W.-S., Yu, S.-H., Chiu, R.-B., Lin, C.-Y., Wang, C.-C., Wang, P.-W., Su, W.-L., Wu, C.-H., Kuo, T.-T., McKenzie, T., Chang, Y.-H., Ferng, C.-S., Ni, C.-M., Lin, H.-T., Lin, C.-J., Lin, S.-D. (2012). A Linear Ensemble of Individual and Blended Models for Music Rating Prediction. *Journal of Machine Learning Research - Proceedings Track* 18:21–60.

Chien, Y.H., George, E.I. (1999). A Bayesian Model for Collaborative Filtering. In: *Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics*.

Cosley, D., Lam, S.K., Albert, I., Konstan, J.A., Riedl, J. (2003). Is Seeing Believing? How Recommender Interfaces Affect Users' Opinions. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 585–592.

Cremonesi, P., Garzotto, F., Negro, S., Papadopoulos, A.V., Turrin, R. (2011). Looking for "Good" Recommendations: A Comparative Evaluation of Recommender Systems. In: *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer*

Interaction - Volume Part III (INTERACT'11), pp. 152–168, Springer-Verlag Berlin, Heidelberg, Germany.

Cremonesi, P., Koren, Y., Turrin, R. (2010). Performance of Recommender Algorithms on Top-N recommendations Tasks. In: *Proceedings of the Fourth ACM Conference on Recommender Systems* (RecSys'10), p. 39, ACM, New York, New York, USA.

Cremonesi, P., Turrin, R. (2009). Analysis of Cold-start Recommendations in IPTV Systems. In: *Proceedings of the Third ACM Conference on Recommender Systems* (RecSys'09), p. 233, ACM, New York, New York, USA.

Cremonesi, P., Turrin, R. (2010). Time-Evolution of IPTV Recommender Systems. In: *Proceedings of the 8th International Interactive Conference on Interactive TV&Video* (EuroITV'10), pp. 105–114, ACM, New York, NY, USA.

Cunningham, P., Hurley, N., Guy, I., Anad, S.S. (2012). Proceedings of the Sixth ACM Conference on Recommender Systems, ACM, New York, NY, USA.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R., Methods, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41(6):391–407.

Desrosiers, C., Karypis, G. (2011). A Comprehensice Survey of Neighborhood-based Recommendation Methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Eds.), *Recommender Systems Handbook*, pp. 107–144.

Dey, A.K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing* 5(1):4–7.

Dietterich, T.G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation* 10(7):1895–1923.

Ding, Y., Li, X. (2005). Time Weight Collaborative Filtering. In: *Proceedings of the 14th ACM International Conference on Information and Knowledge Management* (CIKM'05), pp. 485–492.

Ding, Y., Li, X., Orlowska, M.E. (2006). Recency-based Collaborative Filtering. In: *Proceedings of the 17th Australasian Database Conference - Volume 49* (ADC'06), pp. 99–107, Australian Computer Society, Inc., Darlinghurst, Australia, Australia.

Dourish, P. (2004). What We Talk About When We Talk About Context. *Personal Ubiquitous Computing* 8(1):19–30.

Duda, R.O., Hart, P.E., Stork, D.G. (2001). *Pattern Classification*. John Wiley & Sons.

Ekstrand, M.D., Riedl, J.T., Konstan, J.A. (2011). Collaborative Filtering Recommender Systems. *Foundations and Trends in Human-Computer Interaction* 4(2):81–173.

Felfernig, A., Burke, R., Pu, P. (2012). Preface to the Special Issue on User Interfaces for Recommender Systems. *User Modeling and User-Adapted Interaction* 22(4-5):313–316.

Felfernig, A., Friedrich, G., Jannach, D., Zanker, M. (2011). Developing Contraint-based Recommenders. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Eds.), *Recommender Systems Handbook*, pp. 187–215.

Fisher, R.A. (1922). On the Interpretation of χ 2 from Contingency Tables, and the Calculation of P. *Journal of the Royal Statistical Society* 85(1):87–94.

Gantner, Z., Rendle, S., Schmidt-Thieme, L. (2010). Factorization Models for Context-/time-aware Movie Recommendations. In: *Proceedings of the Workshop on Context-Aware Movie Recommendation* (CAMRa'10), pp. 14–19, ACM, New York, New York, USA.

Goldberg, D., Nichols, D., Oki, B.M., Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM* 35(12):61–70.

Gordea, S., Zanker, M. (2007). Time Filtering for Better Recommendations with Small and Sparse Rating Matrices. In: *Proceedings of the 8th International Conference on Web Information Systems Engineering* (WISE'07), pp. 273–284, Springer-Verlag Berlin, Heidelberg, Germany.

Goren-Bar, D., Glinansky, O. (2004). FIT-recommending TV Programs to Family Members. *Computers & Graphics* 24:149–156.

Gorgoglione, M., Panniello, U. (2009). Including Context in a Transactional Recommender System Using a Pre-filtering Approach: Two Real E-commerce Applications. In: *Proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops* (WAINA'09), pp. 667–672, IEEE Computer Society, Washington, DC, USA.

Gorgoglione, M., Panniello, U., Tuzhilin, A. (2011). The Effect of Context-aware Recommendations on Customer Purchasing Behavior and Trust. In: *Proc. of the Fifth ACM Conference Recommender Systems*, pp. 85–92.

Gunawardana, A., Shani, G. (2009). A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *The Journal of Machine Learning Research* 10:2935–2962.

Herlocker, J.L., Konstan, J.A. (2001). Content-Independent Task-Focused Recommendation. *IEEE Internet Computing* 5(6):40–47.

Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J. (1999). An Algorithmic Framework for Performing Collaborative Filtering. In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR'99), pp. 230–237, ACM, New York, NY, USA.

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems* 22(1):5–53.

Hermann, C. (2010). Time Based Recommendations for Lecture Materials. In: *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2010* (EDMEDIA’10), pp. 1028–1033, Association for the Advancement of Computing in Education, Toronto, Canada.

Hofmann, T. (2003). Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 259–266, ACM, New York, NY, USA.

Hofmann, T. (2004). Latent Semantic Models for Collaborative Filtering. *ACM Transactions on Information Systems* 22(1):89–115.

Hu, Y., Koren, Y., Volinsky, C. (2008). Collaborative Filtering for Implicit Feedback Datasets. In: *Proceedings of the 2008 8th IEEE International Conference on Data Mining* (ICDM’08), pp. 263–272, IEEE Computer Society, Washington, DC, USA.

Iofciu, T., Demartini, G. (2009). Time Based Tag Recommendation Using Direct and Extended Users Sets. In: Eisterlehner, F., Hotho, A., Jäschke, R. (Eds.), *ECML PKDD Discovery Challenge 2009* (DC’09), pp. 99–107, CEUR Workshop Proceedings, Bled, Slovenia.

Jahrer, M., Töscher, A. (2012). Collaborative Filtering Ensemble. *Journal of Machine Learning Research - Proceedings Track* 18:61–74.

Jahrer, M., Töscher, A., Legenstein, R. (2010). Combining Predictions for Accurate Recommender Systems. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD’10), pp. 693–702, ACM, New York, NY, USA.

Jannach, D., Zanker, M., Felfernig, A., Friedrich, G. (2010). *Recommender Systems: An Introduction*1st ed. Cambridge University Press, New York, NY, USA.

Jannach, D., Zanker, M., Konstan, J.A. (2008). Special Issue on Recommender Systems. *AI Commununications* 21(2-3):95–96.

Jawaheer, G., Szomszor, M., Kostkova, P. (2010). Characterisation of Explicit Feedback in an Online Music Recommendation Service. In: *Proceedings of the Fourth ACM Conference on Recommender Systems* (RecSys’10), pp. 317–320, ACM, New York, NY, USA.

Kabutoya, Y., Iwata, T., Fujimura, K. (2010). Modeling Multiple Users’ Purchase over a Single Account for Collaborative Filtering. In: *Proceedings of the 11th International*

Conference on Web Information Systems Engineering (WISE'10), pp. 328–341, Springer-Verlag Berlin, Heidelberg, Germany.

Karatzoglou, A. (2011). Collaborative Temporal Order Modeling. In: *Proceedings of the 5th ACM Conference on Recommender Systems* (RecSys'11), pp. 313–316, ACM, New York, NY, USA.

Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N. (2010). Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering. In: *Proceedings of the 4th ACM Conference on Recommender Systems* (RecSys'10), pp. 79–86, ACM, New York, NY, USA.

Kelly, D., Teevan, J. (2003). Implicit Feedback for Inferring User Preference: a Bibliography. *SIGIR Forum* 37(2):18–28.

Knijnenburg, B.P., Willemsen, M.C., Gantner, Z., Soncu, H., Newell, C. (2012). Explaining the User Experience of Recommender Systems. *User Modeling and User-Adapted Interaction* 22(4-5):441–504.

Koenigstein, N., Dror, G., Koren, Y. (2011). Yahoo! Music Recommendations: Modeling Music Ratings with Temporal Dynamics and Item Taxonomy. In: *Proceedings of the 5th ACM Conference on Recommender Systems* (RecSys'11), pp. 165–172, ACM, New York, NY, USA.

Kohavi, R., Longbotham, R., Sommerfield, D., Henne, R.M. (2009). Controlled Experiments on the Web: Survey and Practical Guide. *Data Mining and Knowledge Discovery* 18(1):140–181.

Konstan, J.A., Riedl, J. (2012). Recommender Systems: From Algorithms to User Experience. *User Modeling and User-Adapted Interaction* 22(1-2):101–123.

Koren, Y. (2008). Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD'08), pp. 426–434, ACM, New York, NY, USA.

Koren, Y. (2009a). Collaborative Filtering with Temporal Dynamics. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD'09), pp. 447–456, ACM, New York, NY, USA.

Koren, Y. (2009b). *The BellKor Solution to the Netflix Grand Prize*. Available from http://www.netflixprize.com.

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

Kullback, S., Leibler, R.A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics* 22(1):79–86.

Lang, K. (1995). Newsweeder: Learning to Filter Netnews. In: *Proceedings of the 12th International Conference on Machine Learning* (ICML'95), pp. 331–339, International Machine Learning Society, Tahoe City, CA, USA.

Lathauwer, L. De, Moor, B. De, Vandewalle, J. (2000). A Multilinear Singular Value Decomposition. *SIAM Journal on Matrix Analalysis and Applications* 21(4):1253–1278.

Lathia, N., Hailes, S., Capra, L. (2009a). Temporal Collaborative Filtering with Adaptive Neighbourhoods. In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR'09), pp. 796–797, ACM, New York, NY, USA.

Lathia, N., Hailes, S., Capra, L. (2009b). Evaluating Collaborative Filtering Over Time. In: *Proceedings of the SIGIR 09 Workshop on the Future of IR Evaluation*, pp. 41–42.

Lathia, N., Hailes, S., Capra, L., Amatriain, X. (2010). Temporal Diversity in Recommender Systems. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 210–217.

Lee, D., Park, S.E., Kahng, M., Lee, S., Lee, S. (2010). Exploiting Contextual Information from Event Logs for Personalized Recommendation. In: Lee, R. (Ed.), *Computer and Information Science 2010*, pp. 121–139.

Lee, T.Q., Park, Y., Park, Y. (2008). A Time-based Approach to Effective Recommender Systems Using Implicit Feedback. *Expert Systems with Applications* 34(4):3055–3062.

Lee, T.Q., Park, Y., Park, Y. (2009). An Empirical Study on Effectiveness of Temporal Information as Implicit Ratings. *Expert Systems With Applications* 36(2):1315–1321.

Li, R., Li, B., Jin, C., Xue, X., Zhu, X. (2011). Tracking User-Preference Varying Speed in Collaborative Filtering. In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence* (AAAI'11), pp. 133–138, AAAI, San Francisco, CA, USA.

Linden, G., Smith, B., York, J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing* 7(1):76–80.

Ling, C.X., Huang, J., Zhang, H. (2003). AUC: A Better Measure Than Accuracy in Comparing Learning Algorithms. In: *Proceedings of the 16th Canadian Society for Computational Studies of Intelligence Conference on Advances in Artificial Intelligence* (AI'03), pp. 329–341, Springer-Verlag Berlin, Heidelberg, Germany.

Lipczak, M., Hu, Y., Kollet, Y., Milios, E. (2009). Tag Sources for Recommendation in Collaborative Tagging Systems. In: Eisterlehner, F., Hotho, A., Jäschke, R. (Eds.), *ECML PKDD Discovery Challenge 2009* (DC'09), pp. 157–172.

Liu, N.N., Cao, B., Zhao, M., Yang, Q. (2010a). Adapting Neighborhood and Matrix Factorization Models for Context Aware Recommendation. In: *Proceedings of the*

Workshop on Context-Aware Movie Recommendation (CAMRa), pp. 7–13, ACM, New York, NY, USA.

Liu, N.N., Zhao, M., Xiang, E., Yang, Q. (2010b). Online Evolutionary Collaborative Filtering. In: *Proceedings of the 4th ACM Conference on Recommender Systems* (RecSys'10), pp. 95–102, ACM, New York, NY, USA.

Lops, P., de Gemmis, M., Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Eds.), *Recommender Systems Handbook*, pp. 73–106.

Lu, Z., Agarwal, D., Dhillon, I.S. (2009). A Spatio-Temporal Approach to Collaborative Filtering. In: *Proceedings of the 3rd ACM Conference on Recommender Systems* (RecSys'09), pp. 13–20, ACM, New York, NY, USA.

Ma, S., Li, X., Ding, Y., Orlowska, M.E. (2007). A Recommender System with Interest-Drifting. In: *Proceedings of the 8th International Conference on Web Information Systems Engineering* (WISE'07), pp. 633–642, Springer-Verlag, Berlin, Heidelberg, Germany.

Manouselis, N., Costopoulou, C. (2007). Analysis and Classification of Multi-Criteria Recommender Systems. *World Wide Web* 10(4):415–441.

Min, S., Han, I. (2005). Detection of the Customer Time-variant Pattern for Improving Recommender Systems. *Expert Systems with Applications* 28(2):189–199.

Mobasher, B., Jannach, D., Geyer, W., Hotho, A. (2012). 4th ACM RecSys Workshop on Recommender Systems and the Social Web. In: *Proceedings of the Sixth ACM Conference on Recommender Systems* (RecSys'12), pp. 345–346, ACM, New York, NY, USA.

Montanés, E., Quevedo, J.R., Díaz, I., Ranilla, J. (2009). Collaborative Tag Recommendation System Based on Logistic Regression. In: *ECML PKDD Discovery Challenge 2009* (DC'09), pp. 173–178.

Oh, J., Sung, Y., Kim, J., Humayoun, M., Park, Y.-H., Yu, H. (2012). Time-Dependent User Profiling for TV Recommendation. In: *Proceedings of the 2nd International Conference on Cloud and Green Computing* (CGC'12), pp. 783–787, IEEE Computer Society, Washington, DC, USA.

Oku, K., Nakajima, S., Miyazaki, J., Uemura, S. (2006). Context-Aware SVM for Context-Dependent Information Recommendation. In: *Proceedings of the 7th International Conference on Mobile Data Management* (MDM'06), pp. 109–109, IEEE Computer Society, Washington, DC, USA.

Palmisano, C., Tuzhilin, A., Gorgoglione, M. (2008). Using Context to Improve Predictive Modeling of Customers in Personalization Applications. *IEEE Transactions on Knowledge and Data Engineering* 20(11):1535–1549.

Panniello, U., Gorgoglione, M. (2012). Incorporating Context into Recommender Systems: An Empirical Comparison of Context-based Approaches. *Electronic Commerce Research* 12(1):1–30.

Panniello, U., Gorgoglione, M., Palmisano, C. (2009a). Comparing Pre-filtering and Post-filtering Approach in a Collaborative Contextual Recommender System: An Application to E-commerce. In: *Proceedings of the 10th International Conference on E-Commerce and Web Technologies* (EC-Web'09), pp. 348–359.

Panniello, U., Tuzhilin, A., Gorgoglione, M. (2013). Comparing Context-Aware Recommender Systems in Terms of Accuracy and Diversity. *User Modeling and User-Adapted Interaction* (Special Issue on Context-Aware Recommender Systems).

Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., Pedone, A. (2009b). Experimental Comparison of Pre-vs. Post-filtering Approaches in Context-aware Recommender Systems. In: *Proceedings of the 3rd ACM Conference on Recommender Systems* (RecSys'09), pp. 265–268, ACM, New York, NY, USA.

Parra, D., Amatriain, X. (2011). Walk the Talk: Analyzing the Relation Between Implicit and Explicit Feedback for Preference Elicitation. In: *Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization* (UMAP'11), pp. 255–268, Springer-Verlag, Berlin, Heidelberg, Germany.

Pazzani, M.J. (1999). A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review* 13(5-6):393–408.

Pazzani, M.J., Billsus, D. (1997). Learning and Revising User Profiles: The Identification ofInteresting Web Sites. *Machine Learning* 27(3):313–331.

Pazzani, M.J., Billsus, D. (2007). Content-based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (Eds.), *The Adaptive Web*, pp. 325–341, Springer-Verlag, Berlin, Heidelberg.

Piotte, M., Chabbert, M. (2009). *The Pragmatic Theory Solution to the Netflix Grand Prize*. Available from http://www.netflixprize.com.

Pradel, B., Savaneary, S., Delporte, J., Guérif, S., Rouveirol, C., Usunier, N., Fogelman-Soulié, F., Dufau-Joel, F. (2011). A Case Study in a Recommender System Based on Purchase Data. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD'11), pp. 377–385, ACM, New York, NY, USA.

Pu, P., Chen, L., Hu, R. (2012). Evaluating Recommender Systems from the User’s Perspective: Survey of the State of the Art. *User Modeling and User-Adapted Interaction* 22(4-5):317–355.

Rashid, A.M., Lam, S.K., Lapitz, A., Karypis, G., Riedl, J. (2007). Towards a Scalable kNN CF Algorithm: Exploring Effective Applications of Clustering. In: *Proceedings of the 8th Knowledge Discovery on the Web International Conference on Advances in Web Mining and Web Usage Analysis* (WebKDD’06), pp. 147–166, Springer-Verlag, Berlin, Heidelberg, Germany.

Rendle, S. (2011). Time-variant Factorization Models. In: Rendle, S. (Ed.), *Context-Aware Ranking with Factorization Models*, pp. 137–153, Springer-Verlag, Berlin, Heidelberg, Germany.

Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L. (2011). Fast Context-aware Recommendations with Factorization Machines. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information* (SIGIR’11), pp. 635–644, ACM, New York, New York, USA.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (CSCW’94), pp. 175–186, ACM, New York, NY, USA.

Ricci, F., Rokach, L., Shapira, B. (2011). Introduction to Recommender Systems Handbook. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Eds.), *Recommender Systems Handbook*, pp. 1–35, Springer.

Ricci, F., Werthner, H. (2006). Introduction to the Special Issue: Recommender Systems. *International Journal of Electronic Commerce* 11(2):5–9.

Riedl, J., Smyth, B. (2011). Introduction to Special Issue on Recommender Systems. *ACM Transactions on the Web* 5(1):1–1.

Said, A., De Luca, E.W., Albayrak, S. (2011). Inferring Contextual User Profiles - Improving Recommender Performance. In: *Proceedings of the 3rd RecSys Workshop on Context-Aware Recommender Systems* (CARS’11), Chicago, IL, USA.

Said, A., Jain, B.J., Narr, S., Plumbaum, T. (2012a). Users and Noise: The Magic Barrier of Recommender Systems. In: *Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization* (UMAP’12), pp. 237–248, Springer-Verlag, Berlin, Heidelberg, Germany.

Said, A., Jain, B.J., Narr, S., Plumbaum, T., Albayrak, S., Scheel, C. (2012b). Estimating the Magic Barrier of Recommender Systems: a User Study. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR’12), pp. 1061–1062, ACM, New York, NY, USA.

Salakhutdinov, R., Mnih, A., Hinton, G. (2007). Restricted Boltzmann Machines for Collaborative Filtering. In: *Proceedings of the 24th International Conference on Machine Learning* (ICML'07), pp. 791–798, ACM, New York, NY, USA.

Salton, G. (1989). *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. In: *Proceedings of the 10th International Conference on World Wide Web* (WWW'01), pp. 285–295, ACM, New York, NY, USA.

Shani, G., Gunawardana, A. (2011). Evaluating Recommendation Systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Eds.), *Recommender Systems Handbook*, pp. 257–297, Springer US, Boston, MA.

Sinha, R., Swearing, K. (2001). Comparing Recommendations Made by Online Systems and Friends. In: *Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*, Dublin, Ireland.

Soboroff, I., Nicholas, C. (1999). Combining Content and Collaboration in Text Filtering. In: *Proceedings of the IJCAI'99 Workshop on Machining Learning in Information Filtering*, pp. 86–91.

Stone, M. (1974). Cross-validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society. Series B (Methodological)* 36(2):111–147.

Stormer, H. (2007). Improving E-Commerce Recommender Systems by the Identification of Seasonal Products. In: *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI), Intelligent Techniques for Web Personalization and Recommender Systems in E-Commerce*, pp. 92–99, AAI.

Su, X., Khoshgoftaar, T.M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence* 2009.

Takács, G., Pilászy, I., Nemeth, B., Tikk, D. (2008). Investigation of Various Matrix Factorization Methods for Large Recommender Systems. In: *Proceedings of the 2008 IEEE International Conference on Data Mining Workshops* (ICDMW'08), pp. 553–562, IEEE Computer Society.

Tang, T.Y., Winoto, P., Chan, K.C.C. (2003). On the Temporal Analysis for Improved Hybrid Recommendations. In: *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence* (WI'03), pp. 214–220, IEEE Computer Society, Washington, DC, USA.

Töscher, A., Jahrer, M. (2008). *The BigChaos Solution to the Netflix Prize 2008*. Available from http://www.netflixprize.com.

Töscher, A., Jahrer, M., Bell, R.M. (2009). *The BigChaos Solution to the Netflix Grand Prize*. Available from http://www.netflixprize.com.

Töscher, A., Jahrer, M., Legenstein, R. (2008). Improved Neighborhood-based Algorithms for Large-scale Recommender Systems. In: *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, pp. 4:1–4:6, ACM, New York, NY, USA.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Vargas, S., Castells, P. (2011). Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems. In: *Proceedings of the Fifth ACM Conference on Recommender Systems* (RecSys'11), pp. 109–116, ACM Press, New York, New York, USA.

Vildjiounaite, E., Hannula, T., Alahuhta, P. (2008). Unobtrusive Dynamic Modelling of TV Program Preferences in a Household. In: *Proceedings of the 6th European Conference on Changing Television Environments* (EuroITV'08), pp. 82–91, Springer-Verlag, Berlin, Heidelberg, Germany.

Whitrow, G.J. (1988). *Time in History*. Oxford University Press.

Wu, Y., Yan, Q., Bickson, D., Low, Y., Yang, Q. (2011). Efficient Multicore Collaborative Filtering. In: *Proceedings of the ACM KDD Cup Workshop*.

Xiang, L., Yang, Q. (2009). Time-Dependent Models in Collaborative Filtering Based Recommender System. In: *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01* (WI-IAT'09), pp. 450–457, IEEE Computer Society, Washington, DC, USA.

Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., Sun, J. (2010). Temporal Recommendation on Graphs via Long- and Short-term Preference Fusion. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD'10), pp. 723–732, ACM, New York, NY, USA.

Xiong, L., Chen, X., Huang, T. (2010). Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization. *Proceedings of SIAM Data Mining* 2010:211–222.

Zhan, S., Gao, F., Xing, C., Zhou, L. (2006). Addressing Concept Drift Problem in Collaborative Filtering Systems. In: *ECAI 2006 Workshop on Recommender Systems,* pp. 34–39.

Zhang, M., Hurley, N. (2008). Avoiding Monotony: Improving the Diversity of Recommendation Lists. In: *Proceedings of the 2nd ACM Conference on Recommender Systems* (RecSys), pp. 123–130, ACM, New York, NY, USA.

Zheng, N., Li, Q. (2011). A Recommender System Based on Tag and Time Information for Social Tagging Systems. *Expert Systems With Applications* 38(4):4575–4587.

Zheng, Z., Chen, T., Liu, N.N., Yang, Q., Yu, Y. (2012). Rating Prediction with Informative Ensemble of Multi-Resolution Dynamic Models. *Journal of Machine Learning Research - Proceedings Track* 18:75–97.

Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J.R., Zhang, Y.-C. (2010). Solving the Apparent Diversity-accuracy Dilemma of Recommender Systems. *Proceedings of the National Academy of Sciences of the United States of America* 107(10):4511–5.

Ziegler, C.-N., McNee, S.M., Konstan, J.A., Lausen, G. (2005). Improving Recommendation Lists through Topic Diversification. In: *Proceedings of the 14th International Conference on World Wide Web* (WWW’05), pp. 22–32, ACM Press, New York, New York, USA.

Zimdars, A., Chickering, D.M., Meek, C. (2001). Using Temporal Data for Making Recommendations. In: *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence* (UAI’01), pp. 580–588, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.