Centro de Informática U·F·P·E

Pós-Graduação em Ciência da Computação

Douglas Véras e Silva

CD-CARS: CROSS-DOMAIN CONTEXT-AWARE RECOMMENDER SYSTEMS

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

Recife
2016

Douglas Véras e Silva

“CD-CARS: Cross-Domain Context-Aware Recommender Systems”

A Ph.D. Thesis presented to the Centro de Informática of Universidade Federal de Pernambuco in partial fulfillment of the requirements for the degree of Philosophy Doctor in Ciência da Computação.

Advisor: Prof. Dr. Carlos André Guimarães Ferraz
Co-Advisor: Prof. Dr. Ricardo Bastos Cavalcante Prudêncio

Recife
2016

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

S586c Silva, Douglas Véras e

CD-cars: cross domain context-aware recomender systems / Douglas Véras e Silva. – 2016.

240 f.: il., fig., tab.

Orientador: Carlos André Guimarães Ferraz.

Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2016.

Inclui referências.

1. Inteligência artificial. 2. Sistemas de recomendação. 3. Filtragem colaborativa. I. Ferraz, Carlos André Guimarães (orientador). II. Título.

006.3 CDD (23. ed.) UFPE- MEI 2016-139

Douglas Véras e Silva

CD-CARS: CROSS-DOMAIN CONTEXT-AWARE RECOMMENDER SYSTEMS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de **Doutor** em Ciência da Computação.

Aprovado em: 21/07/2016.

____

Prof. Carlos André Guimarães Ferraz
Orientador do Trabalho de Tese

BANCA EXAMINADORA

____

Prof. Dra. Patricia Cabral de Azevedo Restelli Tedesco
Centro de Informática/UFPE

____

Prof. Dr. Kiev Santos da Gama
Centro de Informática/UFPE

____

Prof. Dr. Sérgio Ricardo de Melo Queiroz
Centro de Informática/UFPE

____

Prof. Dr. Byron Leite Dantas Bezerra
Escola Politécnica/UPE

____

Prof. Dr. Evandro de Barros Costa
Instituto de Computação/UFAL

I dedicate this thesis to my parents and fiancée.

Acknowledgements

First of all, I thank God for giving me health and strength necessary to conclude this work. Without Him the conclusion of this work would not be possible.

I thank my parents, Maria Aparecida and Jailson Antônio, and brother, Matheus Véras, who always give me love and support in all my life. Also, I would like to thank my fiancée, Laura Regina, her brothers (Diego Dermeval and Amauri Junior) and her parents, Laura Maria and Amauri Campos, for their patience, comprehension, love and support.

My sincere gratitude to professors Carlos Ferraz and Ricardo Prudêncio, respectively, my advisor and co-advisor, for their incentive, trust, support, knowledge transmitted and friendship. I also thank Alysson Bispo, Thiago Prota and Rafael Ferreira, my friends and colleagues at Universidade Federal de Pernambuco (UFPE) and Universidade Federal Rural de Pernambuco (UFRPE), for the incentive and help in the development of this thesis.

I am very grateful to UFPE and UFRPE for the opportunity and support to develop my research in their facilities. Also, my sincere gratitude to Fundação de Amparo a Ciência e Tecnologia de Pernambuco (FACEPE) for the financial support to the development of this research.

I thank to professors Byron Leite, Patricia Tedesco, Kiev Gama, Evandro Costa and Sérgio Queiroz for providing constructive reviews, corrections and suggestions in order to improve my thesis.

Finally, I would like to thank all my family members and friends for the friendship and incentive as well as to all the people who contributed directly or indirectly to this research.

“And if I have a prophet’s power,
and have knowledge of all secret things;
and if I have all faith,
by which mountains may be moved from their place,
but have not love,
I am nothing.”
(I Corinthians 13:2 - Holy Bible)

Resumo

Tradicionalmente, “sistemas de recomendação de domínio único” (SDRS) têm alcançado bons resultados na recomendação de itens relevantes para usuários, a fim de resolver o *problema da sobrecarga de informação.* Entretanto, “sistemas de recomendação de domínio cruzado” (CDRS) têm surgido visando melhorar os SDRS ao atingir alguns objetivos, tais como: “melhoria de precisão”, “melhor diversidade”, abordar os problemas de “novo usuário” e “novo item”, dentre outros. Ao invés de tratar cada domínio independentemente, CDRS usam conhecimento adquirido em um domínio *fonte* (e.g. livros) a fim de melhorar a recomendação em um domínio *alvo* (e.g. filmes). Assim como acontece na área de pesquisa sobre SDRS, a filtragem colaborativa (CF) é considerada a técnica mais popular e amplamente utilizada em CDRS, pois sua implementação para qualquer domínio é relativamente simples. Além disso, sua qualidade de recomendação é geralmente maior do que a dos algoritmos baseados em filtragem de conteúdo (CBF). De fato, a maioria dos “sistemas de recomendação de domínio cruzado” baseados em filtragem colaborativa (CD-CFRS) podem oferecer melhores recomendações em comparação a “sistemas de recomendação de domínio único” baseados em filtragem colaborativa (SD-CFRS), aumentando o nível de satisfação dos usuários e abordando problemas tais como: “início frio”, “esparsidade” e “diversidade”. Entretanto, os CD-CFRS podem não ser mais precisos do que os SD-CFRS. Por outro lado, “sistemas de recomendação sensíveis à contexto” (CARS) tratam de outro tópico relevante na área de pesquisa de sistemas de recomendação, também visando melhorar a qualidade das recomendações. Diferentes informações contextuais (e.g. localização, tempo, humor, etc.) podem ser utilizados a fim de prover recomendações que são mais adequadas e precisas para um usuário dependendo de seu contexto. Desta forma, nós acreditamos que a integração de técnicas desenvolvidas separadamente (de “domínio cruzado” e “sensíveis a contexto”) podem ser úteis em uma variedade de situações, nas quais as recomendações podem ser melhoradas a partir de informações obtidas em diferentes fontes além de refinadas considerando informações contextuais específicas. Nesta tese, nós definimos uma nova formulação do problema de recomendação, considerando tanto a disponibilidade de informações de diferentes domínios (*fonte* e *alvo*) quanto o uso de informações contextuais. Baseado nessa formulação, nós propomos a integração de abordagens de “domínio cruzado” e “sensíveis a contexto” para um novo sistema de recomendação (CD-CARS). Para avaliar o CD-CARS proposto, nós realizamos avaliações experimentais através de dois “conjuntos de dados” com três diferentes dimensões contextuais e três domínios distintos. Os resultados dessas avaliações mostraram que o uso de técnicas sensíveis a contexto pode ser considerado como uma boa abordagem a fim de melhorar a qualidade de recomendações de “domínio cruzado” em comparação às recomendações de CD-CFRS tradicionais.

Palavras-Chave: Recomendação de Domínio Cruzado. Recomendação Sensível a Contexto. Filtragem Colaborativa. Recomendação de Domínio Cruzado Sensível a Contexto.

Abstract

Traditionally, single-domain recommender systems (SDRS) have achieved good results in recommending relevant items for users in order to solve the *information overload problem*. However, cross-domain recommender systems (CDRS) have emerged aiming to enhance SDRS by achieving some goals such as accuracy improvement, diversity, addressing new user and new item problems, among others. Instead of treating each domain independently, CDRS use knowledge acquired in a source domain (e.g. books) to improve the recommendation in a target domain (e.g. movies). Likewise SDRS research, collaborative filtering (CF) is considered the most popular and widely adopted approach in CDRS, because its implementation for any domain is relatively simple. In addition, its quality of recommendation is usually higher than that of content-based filtering (CBF) algorithms. In fact, the majority of the cross-domain collaborative filtering RS (CD-CFRS) can give better recommendations in comparison to single-domain collaborative filtering recommender systems (SD-CFRS), leading to a higher users' satisfaction and addressing cold-start, sparsity, and diversity problems. However, CD-CFRS may not necessarily be more accurate than SD-CFRS. On the other hand, context-aware recommender systems (CARS) deal with another relevant topic of research in the recommender systems area, aiming to improve the quality of recommendations too. Different contextual information (e.g., location, time, mood, etc.) can be leveraged in order to provide recommendations that are more suitable and accurate for a user depending on his/her context. In this way, we believe that the integration of techniques developed in isolation (cross-domain and context-aware) can be useful in a variety of situations, in which recommendations can be improved by information from different sources as well as they can be refined by considering specific contextual information. In this thesis, we define a novel formulation of the recommendation problem, considering both the availability of information from different domains (source and target) and the use of contextual information. Based on this formulation, we propose the integration of cross-domain and context-aware approaches for a novel recommender system (CD-CARS). To evaluate the proposed CD-CARS, we performed experimental evaluations through two real datasets with three different contextual dimensions and three distinct domains. The results of these evaluations have showed that the use of context-aware techniques can be considered as a good approach in order to improve the cross-domain recommendation quality in comparison to traditional CD-CFRS.

Keywords: Cross-domain Recommendation. Context-Aware Recommendation. Collaborative Filtering Recommendation. Cross-Domain Context-Aware Recommendation.

List of Figures

List of Tables

Contents

1 Introduction

In this chapter, we contextualize and motivate the problem addressed in this thesis, respectively, in Section 1.1 and Section 1.2. The problem statement is described in Section 1.3. In Section 1.4, we define the objectives of this thesis and in Section 1.5 we describe an overview of the proposal. At last, we highlight the expected contributions and describe the structure of this thesis (Section 1.7).

1.1 Contextualization

In the last years, the growth of the Internet has increased the amount of information available for users. Consequently, the task of finding relevant information among a myriad of options became a problem. This problem is traditionally known as *the information overload problem* (RESNICK et al., 1994)(HILL et al., 1995)(SHARDANAND; MAES, 1995)(ADOMAVICIUS; TUZHILIN, 2005)(RICCI; ROKACH; SHAPIRA, 2011). Taking into account the variety of information provided by applications on the Internet, we can consider *information* as any *item* capable of being consumed and rated by a user (e.g. movies, books, music, and so on). Given that, the information overload problem makes the process of finding relevant items in an extensive amount of options difficult for a user.

Fortunately, this problem has received notable attention of researchers in the Artificial Intelligence area. In this way, recommender systems (RSs) have been designed in order to solve this problem (ADOMAVICIUS; TUZHILIN, 2005)(RICCI; ROKACH; SHAPIRA, 2011). For example, a recommender system (RS) can be used for suggesting to users interesting movies to watch, books to read, music to listen to, etc. The suggestions (recommendations) are provided according the users' profile, which could be inferred from the consumption log, for instance.

Recently, a large number of Web sites and applications have adopted recommender systems to provide their users with more relevant items, such as Amazon[1], Netflix[2], Youtube[3], Last.fm [4], BookCrossing[5], Buscape[6], among many others. However, most of these systems are developed to recommend items in a specific domain such as movies, videos, music, books, and so on. Thus, they are known as *single-domain* RS, since they just consider the user profile from a unique domain to recommend relevant items in the same domain. For example, a single-domain RS could recommend a movie based on the

[1] Amazon e-commerce site, http://www.amazon.com
[2] Netflix on-demand streaming media site, http://www.netflix.com
[3] YouTube video sharing site, https://www.youtube.com
[4] Last.fm on-line radio, http://www.last.fm
[5] BookCrossing on-line book exchange site, http://bookcrossing.com
[6] Buscape helps users to find best prices for products and services, http://www.buscape.com.br

previous movies watched or recommend a book similar to other ones read by a user. It is important to mention that despite there are several definitions of “domain”, our notion of domain can be expressed as a “kind of item” (e.g. movies, music, books, news, among others) (FERNÁNDEZ-TOBÍAS et al., 2012).

Although single-domain RSs have achieved a good quality in suggesting relevant items for users, some issues still are significant for the *information overload problem* (RICCI; ROKACH; SHAPIRA, 2011), such as:

- *cold-start*: it is related to situations in which a RS is unable to generate recommendations due to an initial lack of user preferences;
- *sparsity*: when the average number of ratings per user and item is low, which may negatively affect the quality of the recommendations, a sparsity problem might exist;
- *diversity*: when very similar or redundant items are recommended, which may not satisfy the users;
- *accuracy*: even when the issues mentioned above are resolved, the RS may still not be accurate, which means that incorrect predictions of ratings may be made as well as a list of recommended items may not satisfy a user; among others.

In order to alleviate these problems, *cross-domain* recommender systems (WINOTO; TANG, 2008)(CREMONESI; TRIPODI; TURRIN, 2011) have arisen, aiming to improve the quality of single-domain recommendations (FERNÁNDEZ-TOBÍAS et al., 2012). Instead of treating each domain independently, cross-domain RSs use knowledge acquired in a source domain (e.g. books) to improve the recommendation of a target domain (e.g. movies). To illustrate this, consider a user with no information about his/her favorite movie genres. This lack of information can be inferred either from his/her favorite book genres or from his/her similarity with users across different domains, for example. One of the first studies on this emerging research topic was that presented in (WINOTO; TANG, 2008), which investigated if the consumption behaviors on related items from different domains could be useful to make recommendations in a target domain. As shown by Winoto e Tang (2008), joint recommendations of items from multiple domains may be less accurate, but more diverse than recommendations of items in a single domain.

Since the first cross-domain recommender systems have arisen, several approaches have been proposed to deal with different goals (FERNÁNDEZ-TOBÍAS et al., 2012). For instance, knowledge-based recommender systems try to exploit knowledge about users and items besides the relationship between them in order to produce recommendations (TREWIN, 2000). In this way, knowledge-based recommender systems demand a great amount of knowledge about users and items (and their domains), which should be stored and organized in a way that enables inferring and reasoning. However, that knowledge acquisition is a very difficult process and a knowledge engineer is required to construct the knowledge-base which causes a bottle-neck for the knowledge-based recommender systems (AZAK, 2010).

As described above, the knowledge-based recommender systems rely on an “ad-hoc” approach, which may be difficult to customize to new situations, once that they are usually designed for a specific domain (TREWIN, 2000). However, other approaches have been successfully adopted for cross-domain RS and, in general, require little domain knowledge (FERNÁNDEZ-TOBÍAS et al., 2012), since they are based on simple information obtained from user ratings.

Fernández-Tobías et al. (2012) stated that domains can be explicitly or implicitly linked by means of content-based (CBF) or collaborative filtering (CF) characteristics associated with users and/or items, such as ratings, social tags, and latent factors. Cremonesi, Tripodi e Turrin (2011) surveyed and categorized *cross-domain collaborative filtering recommender systems* (CD-CFRS), which recommend items from target domain exploring the similarities between users considering ratings from source and target domains, as illustrated in Figure 1. Likewise single-domain RS research, collaborative filtering is considered the most popular and widely implemented approach in cross-domain RS, because its implementation and integration in existing domains is relatively easy as well as its quality is generally higher than other approaches (ADOMAVICIUS; TUZHILIN, 2005)(FERNÁNDEZ-TOBÍAS et al., 2012).

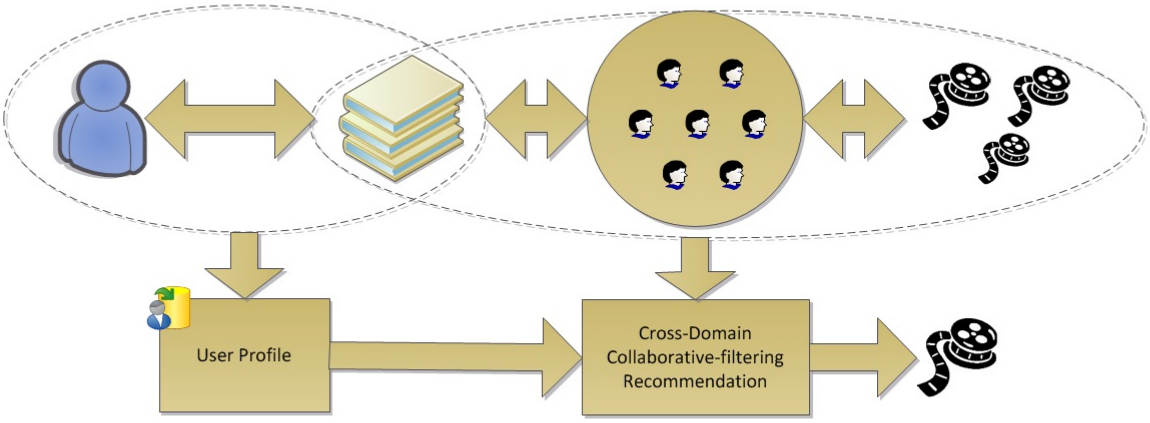


Figure 1 – Cross-domain collaborative filtering recommendation (based on (CREMONESI; TRIPODI; TURRIN, 2011)(SANTOS et al., 2012)).

1.2 Motivation

In fact, the majority of the CD-CFRS can give better recommendations in comparison to single-domain RS, leading to a higher user satisfaction and addressing cold-start, sparsity, and diversity problems, but not necessarily, they can be more accurate than single-domain collaborative filtering RS (WINOTO; TANG, 2008)(FERNÁNDEZ-TOBÍAS et al., 2012). Meanwhile, context-aware recommender systems (CARS) is another relevant topic of research in RS and has been used in order to enhance the quality of recommendations Adomavicius e Tuzhilin (2015), especially for providing accurate recommendations taking into account the user's context.

The context-aware approach uses different contextual information (e.g., location, time, mood, etc.) to improve the accuracy of recommendations (ADOMAVICIUS; TUZHILIN, 2015), as illustrated in Figure 2. In many applications, such as recommending a vacation package, recommending a TV program, among others, it may not be sufficient to consider only users and items, it is also important to incorporate contextual information into the recommendation process in order to recommend items to users under certain circumstances (ADOMAVICIUS; TUZHILIN, 2015). For example, using the temporal context, a travel recommender system could provide a recommendation in the winter that can be very different from the one in the summer. More specifically, on weekdays a user might prefer to watch news programs when he/she turns on his/her TV in the morning, or to watch soccer games at night, and on weekends to watch comedy movies.

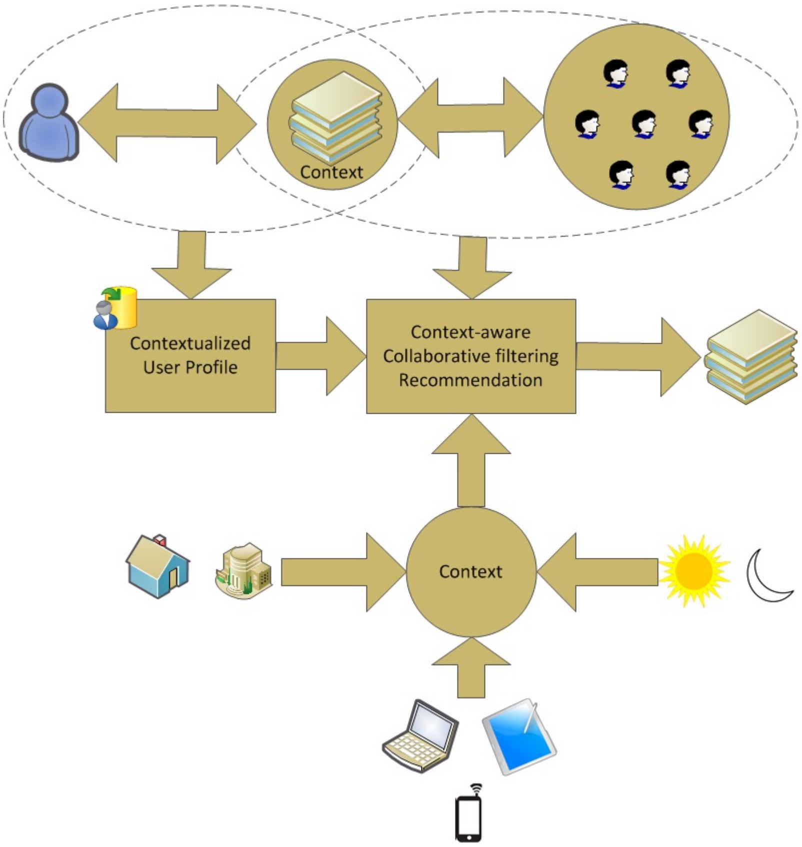

Context
Contextualized User Profile
Context-aware Collaborative filtering Recommendation
Context

Figure 2 – Context-aware collaborative filtering recommendation.

Researchers in the recommender systems area have recognized the importance of contextual information (ADOMAVICIUS; TUZHILIN, 2015). In addition, in the cross-domain RS field, Fernández-Tobías et al. (2012) and Cantador et al. (2015) highlighted that context can be treated as a bridge between different domains and just a few works have considered context-aware techniques in cross-domain recommender systems. The use of context-aware techniques in cross-domain RS is an interesting open research direction, since the majority of the works about cross-domain recommender systems only adopt CBF and CF approaches, both considering only users and considering items attributes, without taking into account any additional contextual information.

Accurate prediction of user preferences undoubtedly depends upon the degree to which the recommender system has incorporated the relevant contextual information into a recommendation method (ADOMAVICIUS et al., 2005)(ADOMAVICIUS; TUZHILIN, 2015). In this way, using a context-aware approach in cross-domain recommender systems may be useful to make suggestions under difficult situations such as in the cold-start, sparsity and diversity problems while improving the accuracy of recommendations, in comparison to traditional CD-CFRS, by using context and knowledge from different domains.

We believe that the integration of techniques developed in isolation to cross-domain and context-aware RS can be useful in a variety of situations, in which recommendations can be improved by information from different sources and can be refined by considering specific contextual information.

1.3 Problem Statement

In this thesis, we address the cross-domain recommendation problem under the CF and context-awareness perspectives. Thus, besides considering user ratings, we also consider the context of these ratings in the recommendations, which implies one more dimension (context) in the user-rating matrices, or in this case, user-rating-context tensors (User x Item x Context). In our cross-domain context-aware recommendation problem, there exists a user-rating-context tensor for each domain (source and target). In both source and target tensors, there is no item in common (no item overlap), but the same set of contexts is observed (context overlap) and at least a user must have ratings in both tensors (user overlap).

Therefore, our problem is to explore the user-rating-context tensors from source and target domains to improve recommendations in the target domain, i.e., to improve the estimation of unknown ratings for items in a target domain by exploiting the user-rating-context tensors from these domains.

Based on the above problem, we state the hypothesis of this thesis:

The application of context-aware techniques can improve the accuracy of cross-domain collaborative filtering recommendations.

It is important to mention that *accuracy*, in that hypothesis, is one *quality* aspect of recommender systems. However, other aspects can also be addressed in this thesis such as alleviating the *cold-start* and *sparsity* issues, or improving the recommendation *diversity* or *coverage*, but they are not our main goals.

1.4 Objectives

The main objective of this thesis is to improve the accuracy of cross-domain collaborative filtering recommendations by adding context-aware techniques. In order to achieve this, we aim to build a *cross-domain context-aware recommender system* (CD-CARS) through the proposal of novel algorithms for cross-domain recommendations.

Specific objectives of this thesis are:

- To allow that traditional CF-based algorithms can be used in combination with the proposed algorithms.
- To allow that the proposed algorithms can be extended and configured.
- To allow that cross-domain context-aware recommendations can be made either to more related domains (e.g. *Book* and *Television*) or less related ones (e.g. *Book* and *Music*) among themselves. We consider this relation among distinct domains according to the set of item genres of them. As more the domains have item genres in common the more related they are considered (e.g. *Book* and *Television* have several item genres in common such as "romance", "educational", "religion", etc.).
- To provide a solution for identifying the most relevant contextual features that can be used for a specific domain.
- To build a method responsible for generating contextual user profiles on different domains.
- To develop a solution for extracting relevant information (e.g. contextual information, user profiles in different domains, etc.) for the datasets used in the CD-CARS evaluation.

1.5 Proposal Overview

As mentioned previously, this thesis aims to improve the quality of cross-domain collaborative filtering recommendations by adding context-aware techniques. In order to achieve this, it is necessary to understand current approaches available in the literature, especially, about two emerging research areas: *context-aware recommender systems* (CARS) and *cross-domain recommender systems* (CDRS).

After searching and exploring several approaches in these areas, we propose CD-CARS algorithms based on distinct context-aware paradigms (*pre-filtering, post-filtering, and modelling*) (ADOMAVICIUS; TUZHILIN, 2015) combined with a CD-CFRS aiming to improve its quality (accuracy of the recommendations). Thus, this CD-CFRS is transformed into a CD-CARS which recommends items from target domain exploring the similarities between users considering ratings, and also their contexts, from source and target domains, as illustrated in Figure 3. The CD-CFRS algorithms adopted in the proposal belong to two traditional CF-based categories: neighborhood-based and matrix factorization.

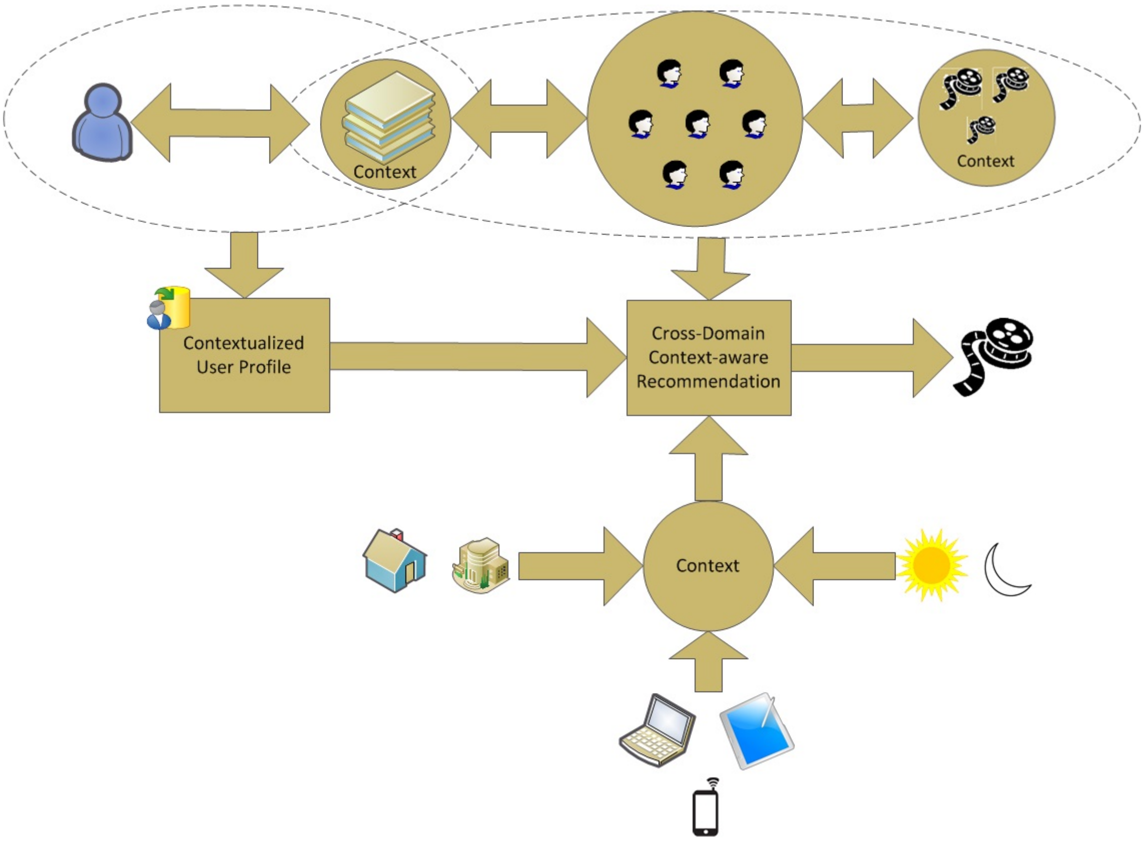

Context
Context
Contextualized User Profile
Cross-Domain Context-aware Recommendation
Context

Figure 3 – Cross-domain context-aware recommendation.

To illustrate the CD-CARS, suppose that a user X, who enjoys to read romance books on weekdays and does not have any preference known about movies, is very similar to another user Y that also enjoys romance books on weekdays and likes to watch action movies on weekdays and comedy movies on weekends, so, a CD-CARS could prioritize movies enjoyed by the user Y on the top of the recommended item list for the user X in those particular contexts (comedy movies on weekends and action movies on weekdays), just by knowing the book preferences from user X without his/her movie preferences.

Although many cross-domain RS have used contextual information in an ad-hoc way as part of their knowledge-based approaches (see Section 2.4.2) (BLANCO-FERNÁNDEZ et al., 2011)(MOE; AUNG, 2014b)(KAMINSKAS et al., 2014), to the best of our knowledge, there are not works in the literature addressing the cross-domain recommendation task by means of contextual features using systematic paradigms (*pre-filtering, post-filtering, and modelling*) (FERNÁNDEZ-TOBÍAS et al., 2012)(CANTADOR et al., 2015). Those knowledge-based cross-domain RSs may be difficult to customize to new situations (domains) and to compare their performances with other approaches. For instance, the knowledge-based framework proposed in (KAMINSKAS et al., 2014) is specific for the considered domains (points of interest and music). Our CD-CARS, in turn, relies on the use of systematic context-aware approaches (ADOMAVICIUS; TUZHILIN, 2015), which have been successfully adopted for single domain recommendation and in general require just a little knowledge about the domain (e.g user ratings) and can be customized to new domains in a simpler way.

1.6 Contributions

The contributions of this thesis are multiple, mainly in the recommender systems area. The main ones are listed below:

1. The formalization of the cross-domain context-aware recommendation problem from the survey of two emergent research fields: cross-domain and context-aware RS;

2. Improving the quality of CD-CFRS, through the realization of a CD-CARS by the proposal of novel algorithms based on three distinct and systematic paradigms of context-aware recommendation, which were chosen rather than ad-hoc context-aware approaches;

3. Provision of real datasets for evaluating CD-CARS, taking into account different domains and contextual information;

4. Providing a CD-CARS that can be useful to recommend items from any domain (e.g. books, music, movies, etc.), which allows generating cross selling or bundle recommendations for items from multiple domains (e.g. the recommendation of a music accompanied of a movie to watch or a book to read).

Through the findings of this thesis, we expect to contribute for the cross-domain RS area towards future research and challenges in cross-domain context-aware recommendations.

1.7 Thesis Outline

This thesis is structured as follows:

- Chapter 1 introduces and states the cross-domain context-aware problem, describes the proposal of this thesis as well as its objectives and contributions.
- Chapter 2 reviews the literature about recommender systems focusing on cross-domain RS and context-aware RS. Also in this chapter, we compare related work to our thesis.
- Chapter 3 presents the proposed CD-CARS. For that, we describe the formalization of the CD-CARS problem, how the contextual information is modelled, the proposed recommendation algorithms, and cross-domain CF-based algorithms adopted in combination with the CD-CARS algorithms.
- Chapter 4 describes particular details of an implementation of the CD-CARS proposal.
- Chapter 5 presents the results of an experimental evaluation of the implemented CD-CARS as well as a discussion about the findings of this research. Besides, we describe details about the experiments' settings and evaluation metrics.
- Chapter 6 presents the conclusions, limitations and future works of this thesis.

2 Background and Related Work

In this chapter, we provide an explanation of concepts related to this thesis. Initially, we describe the main concepts of recommender systems (Section 2.1) such as approaches, algorithms, user profiling, and evaluation metrics. Following, we introduce concepts of cross-domain recommender systems (Section 2.2) and their most common approaches. Finally, we describe the foundations of context-aware recommender systems (Section 2.3).

These concepts are necessary as background for understanding the proposed CD-CARS and for positioning the proposal of this thesis in the state-of-the-art research. However, since the union of cross-domain and context-aware fields has not been deeply explored, we also describe and classify cross-domain RS and CD-CARS, both related to the proposal of this thesis (Section 2.4).

2.1 Recommender Systems

In recent years, recommender systems (RSs) have been crucial for dealing with the *information overload problem* (ADOMAVICIUS; TUZHILIN, 2005). This issue is related to the explosive growth and variety of information available on the Web, which frequently has overwhelmed users with a large myriad of options.

RSs are software tools and techniques providing suggestions of relevant items to users (RESNICK; VARIAN, 1997)(BURKE, 2002)(BURKE, 2007). The suggestions relate to various decision-making processes, such as what products to buy, what music to listen to, or what TV programs to watch. Therefore, recommender systems can help people to identify contents of their interest among a large set of options available. These systems became an important research area since the publication of landmark papers in the 1990's, when the term "collaborative filtering" was coined (RESNICK; VARIAN, 1997).

Since then, the number of research papers published has increased significantly in many application fields (books, documents, images, movies, music, shopping, TV programs, and others) (PARK et al., 2012), as well as the amount of commercial applications of recommender systems by large companies such as *Amazon.com* (LINDEN; SMITH; YORK, 2003), *Google* (DAS et al., 2007), *Last.fm* (EYKE, 2009), *Netflix* (BENNETT; LANNING, 2007), among others.

For a better understanding of RSs, we describe some perspectives of them in the following subsections.

2.1.1 Strategies

There are several RS strategies (or approaches) in the literature. A strategy can be seen as a type or category of RS, and it may vary according to the paradigm of its recommendation algorithm, i.e., how the recommendation is made. This variation of strategies leads to different classifications of RS. Below, we describe eight categories of RS, which are based on (BURKE, 2007), (RICCI; ROKACH; SHAPIRA, 2011) and (VÉRAS et al., 2015) classifications.

1. *Non-personalized*: Non-personalized recommender algorithms present any user a predefined list of items. Such algorithms usually serve as a baseline for more advanced personalized algorithms. For example, one non-personalized algorithm, called Top Popular (TopPop), recommends the top-N items (e.g. movies) with the highest popularity (largest number of ratings) (CREMONESI; GARZOTTO; TURRIN, 2012).

2. *Content-based filtering (CBF)*: Content-based recommendation systems try to recommend similar items to those that a user has liked or consumed in the past (PAZZANI, 1999). Indeed, the basic process performed by a content-based recommender consists in matching up the attributes of a user profile, in which preferences and interests are stored, with the attributes of a item's content, in order to recommend to that user new interesting items. For example, TV contents that are similar (based on their genres, actors, and so on) to those the user preferred in the past are recommended. Since such RSs tend to recommend items with the same characteristics as the ones that a user liked in the past, the recommended items typically lack novelty, meaning the RS proposes a limited variety of unexpected (but relevant) recommendations (ADOMAVICIUS; TUZHILIN, 2005).

3. *Collaborative filtering (CF)*: Collaborative recommender systems ignore content and exploit collective preferences of the crowd, i.e., they generate recommendations using different users' rating profiles and suggest items that other users with similar tastes liked in the past (PAZZANI, 1999). The degree to which two users' preferences are considered similar is based on a similarity measure of their rating histories. This approach can be illustrated by the expression: “people who watched this TV program also watched...”. In addition, CF is considered the most popular and widely implemented approach in RS, because its implementation and integration in existing domains are relatively easy, and its quality is usually higher than that of CBF algorithms. A common criticism of CF recommenders is that they tend to be biased toward popularity, constraining the degree of diversity, so, they are not able to recommend unrated items (related to cold-start and sparsity issues) (ADOMAVICIUS; TUZHILIN, 2005). The “community-filtering” strategy can be considered a

specialization of the collaborative filtering one (KAMAHARA et al., 2005). The "community-filtering" strategy recommends items based on the preferences of the user's friends (or friend of a friend)(KAMAHARA et al., 2005)(BOURKE; MCCARTHY; SMYTH, 2011)(HAN et al., 2015). Evidence suggests that people tend to rely more on recommendations from their friends than on recommendations from similar but anonymous individuals (KAMAHARA et al., 2005). This observation, combined with the growing popularity of social networks, is generating a rising interest in community-based systems or, as or as they usually referred to, social recommender systems (KAMAHARA et al., 2005).

4. *Data mining*: Significantly, many researchers have used data mining techniques to improve the recommender systems performance (PARK et al., 2012). Data mining techniques are defined as extracting or mining knowledge from data. These techniques are used for the exploration and analysis of a large amount of data in order to discover meaningful patterns and rules. They can be used to lead decision making and to predict the effect of decisions. For example, TV programs can be classified into two classes: "watched" and "not watched" and a user profile is then a collection of attributes together with the number of times they occur in positive and negative examples. Hence, the RS computes prior probabilities that a TV program has to belong to one of those two classes and the conditional probability that a feature is present if a TV program is classified into either positive or negative class. It must be noted that features are, in this case, related to content (e.g. genre) or not (e.g. time of the day).

5. *Context-awareness*: Context-awareness aims to give to its applications advantages in the use of contextual information, such as the user's location, to offer proactive services to the user without any explicit request (ABOWD et al., 1999). In this way, a personalization system based on context-awareness could adapt its functionality or behavior so that it reacts differently depending on the user's context (location, friends, family, time, among others) and the resources available at that moment, in accordance with his/her personal preferences (RICCI; ROKACH; SHAPIRA, 2011). For example, a RS could recommend a TV program that suits the user and his current situation: staying at home or on the train, in the noon or the evening, being in front of his TV or smartphone. For each situation (context), the user's preference may be different, so, some contextual patterns could be found and exploited by recommendation algorithms, e.g. knowing that a user particularly likes watching sports in the evening time, at home (MOON et al., 2009).

6. *Semantic-based*: Semantic Web is based on describing Web resources by semantic annotations (meta-data), formalizing these annotations in an ontology and applying reasoning processes aimed at discovering new knowledge (BERNERS-LEE; HENDLER,

2001). The synergy between recommender systems and Semantic Web has already been explored in many domains (including the TV domain), showing significant increases in the recommendation accuracy (BLANCO-FERNÁNDEZ; PAZOS-ARIAS, 2008). Instead of employing traditional syntactic approaches, Semantic-based strategy discovers semantic relationships between the users' preferences and the items available in the domain ontology through semantic similarity metrics. For example, using the semantic approach, a RS could recommend a place to visit (e.g. offering a tourist package) according to the places showed in a movie or sports game that a user liked (BLANCO-FERNÁNDEZ et al., 2011).

The approaches described above focused on making recommendations for individual users and do not consider the problem of group recommendation. The problem of group recommendation has also been investigated recently (QUEIROZ; CARVALHO, 2004)(RICCI; ROKACH; SHAPIRA, 2011). Various techniques have been proposed, targeting different types of recommendation items (e.g., movie, TV program, music) and different groups (e.g., family, friends, dynamic social groups). Most group recommendation techniques consider the preferences of individual users and propose various strategies to either combine the individual user profiles into a single group profile (a pseudo user) and make recommendations for that pseudo user (BRUSILOVSKY; KOBSA; NEJDL, 2007), or generate recommendation lists for individual group members and merge the lists for group recommendation(MARILLY et al., 2011). This kind of approach is usually adopted in the TV domain, because watching TV activity is, traditionally, performed by a group of people (e.g. a family). For example, a TV program could be recommended according to the average rating of a group of users (based on their individual preferences) (BRUSILOVSKY; KOBSA; NEJDL, 2007).

2.1.2 User Profiling

The user profile, which usually is composed of preferences and personal characteristics, is one of the most important aspects of the recommendation process. Recommender systems necessarily make use of user profiles in order to recommend items related to those profiles. However, there are different approaches for creating a user profile. Each approach has its benefits and its limitations (UBERALL; MUTTUKRISHNAN, 2009). Therefore, this is an important perspective of any recommender system, which can be categorized into three categories as follows (VÉRAS et al., 2015):

1. *Explicit Profiling*: An explicit profile can be created by a user in the first time that he/she logs in the recommendation system. In this case, users set their preferences (interests) such as favorite TV shows or genre of TV shows, favorite actors of movies, favorite channels, ratings for movies (e.g. rating "four" for a TV show on a scale of

zero to five), among others. Furthermore, users can modify any information of their profiles at any moment through the system (UBERALL; MUTTUKRISHNAN, 2009). However, the explicit profiling approach could bother and tire users (REICHLING; WULF, 2009) in order to fulfill their profiles every time that they find something interesting on TV, for example.

2. *Implicit Profiling*: On other hand, an implicit profile can be created automatically by the recommender system. In this case, the RS logs and saves the viewing behaviour of a user (UBERALL; MUTTUKRISHNAN, 2009). Through the user's log, like watched programs (watching time, watching duration, genre, etc.), his/her preferences (interests) are inferred (instead of explicitly set). This inference may result in user's favorite TV shows (or genre of TV shows), favorite actors of movies, favorite channels, and even a rating for a movie (e.g. calculated using the ratio between the watching duration and the TV show duration), among others. Sometimes, this approach could have the problem of incorrectly expressing the user profile (HU; KOREN; VOLINSKY, 2008), because the user could be sleeping or doing something else while TV is on, for example.

3. *Contextual Profiling*: The "contextual profile" usually is generated by "Context-Aware Recommendation Systems"(ABBAR; BOUZEGHOUB; LOPEZ, 2009). In this approach, the user's profile is created through the relationship between contexts and "common" user profiles (explicit or implicit profiling). Thus, the recommendation process is based on the contextual profile, which contains contextual information besides user preferences (MUKHERJEE et al., 2011). Users' contextual information can be obtained explicitly or implicitly (most common), such as location, friends, family members, watching day/time, activity, and so on. Therefore, according to the manner that the contextual information is obtained - explicitly or implicitly, the same issues from these approaches are applied for the contextual profile.

2.1.3 Evaluation

Evaluation of recommender systems is fundamental in assessing the quality of their recommendations. However, many different measures have been defined in the literature with the aim of making better choices in general or for a specific application area. Likewise the RS algorithms, we describe evaluation metrics in a top-level way, as follows (VÉRAS et al., 2015):

1. *Qualitative measures*: these measures are used when we want a model to minimize the number of errors. Hence, these metrics are usual in many direct applications of recommenders. Inside this category, some of these measures are more appropriate

for some kinds of recommenders, predictors or information retrieval tasks. Examples of these measures are Accuracy (LEE; YANG, 2003), F-measure (LEKAKOS; GIAGLIS, 2004), Coverage (LEKAKOS; CARAVELAS, 2008), Diversity (CREMONESI; TURRIN, 2010), among others (RICCI; ROKACH; SHAPIRA, 2011). During the evaluation, items are commonly labeled as relevant or irrelevant for a user and a metric is adopted to measure the quality of the items once classified by the RSs. The recommendation problem is treated as a classification task.

2. *Probabilistic (Predictive) measures*: these measures are especially useful when we want an assessment of the reliability of the predictions returned by a RS, whether they have recommended a non-relevant item with high or low probability. The main examples of these measures are Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) (SHANI; GUNAWARDANA, 2011)(HERLOCKER et al., 2004). The recommendation problem, in this case, is usually treated as a regression task when the actual rate of an item is compared to a rate or score predicted by the RS.

3. *Ranking (Classification) measures*: these measures are very common in the RSs area because they are based on "how well" recommender systems rank the recommended items. Thus, there are many examples of evaluation metrics in this category, such as Precision and Recall curves (ZHIWEN; XINGSHE, 2003), Normalized Discounted Cumulative Gain (NDCG) (BALTRUNAS; MAKCINSKAS; RICCI, 2010), Mean Average Precision (MAP) (HOPFGARTNER; JOSE, 2010), hit rate (HR) (O'SULLIVAN; SMYTH; WILSON, 2004), Fall-out (JOJIC; SHUKLA; BHOSAREKAR, 2011), Area under the ROC Curve (AUC) (ZHANG; ZHENG, 2005), Breese score (BREESE; HECKERMAN; KADIE, 1998), among others (RICCI; ROKACH; SHAPIRA, 2011). Differently from the previous category, these metrics assess the quality of a ranking of items returned by the RS instead of the average quality of the raw scores returned by the RS. The recommendation problem is treated in this case as a ranking task.

4. *User satisfaction*: in this category, there are empirical experiments with users in order to verify their satisfaction about the RS (BLANCO-FERNáNDEZ et al., 2008). Although this method is used in many RSs and collects personal feedback of users (LóPEZ-NORES et al., 2009), this kind of evaluation may have some problems, such as biases, lack of an objective measure for RS quality assessment, comparison between different systems, and so on (RICCI; ROKACH; SHAPIRA, 2011).

2.2 Cross-Domain Recommender Systems

Nowadays, the majority of recommender systems provide recommendations for items belonging to a single domain. For instance, *Netflix* recommends movies (BENNETT; LANNING, 2007), *Last.fm* recommends songs (EYKE, 2009), among others. These single-domain recommender systems have been successfully adopted by several websites, however, some of them such as *Amazon*[1] and *eBay*[2] usually maintain user preferences for items from multiple domains. In addition, it is common that users of social networks provide their preferences and interests for a variety of items from distinct domains (e.g. music, books, movies, etc.) (SHAPIRA; ROKACH; FREILIKHMAN, 2013).

Leveraging all the user preferences available in several systems or domains may be useful for generating better recommendations, e.g., by alleviating the *cold-start* and *sparsity* problems in a target domain, or by providing recommendations for items from multiple domains. Thus, cross-domain recommender systems aim to generate or improve recommendations in a target domain (e.g. music, etc.) by exploiting knowledge from a source domain (e.g. books, etc.) (FERNÁNDEZ-TOBÍAS et al., 2012).

The cross-domain approach is a challenging and emergent field of recommender systems (CANTADOR et al., 2015). Although it has been addressed from distinct perspectives, there are several distinct definitions of the cross-domain recommendation task. According to surveys about cross-domain RS (FERNÁNDEZ-TOBÍAS et al., 2012)(CANTADOR et al., 2015), we describe some of its perspectives in the following subsections.

2.2.1 Definition of Domain

In the literature, researchers have considered different definitions of "domain". For instance, some of them have considered items like movies and books as belonging to distinct domains, while others have considered different item genres as different item domains (e.g. "action movies" and "comedy movies"). Cantador et al. (2015) define the "domain" concept regarding the attributes and types of recommended items. They consider that domain may be defined at four levels (see Figure 4):

- (Item) *Attribute level.* Recommended items have the same type and the same attributes, but they differ in the value of a certain attribute. For instance, two movies of different genres (e.g. "action movies" and "comedy movies") belong to distinct domains (CAO; LIU; YANG, 2010).
- (Item) *Type level.* In this level, recommended items have similar types and have some attributes in common. For example, movies and TV programs belong to distinct

[1] http://www.amazon.com
[2] http://www.ebay.com

domains, since they have some attributes in common (title, genre, etc.), but they also have different ones (e.g., airtime, channel, etc.) (HU et al., 2013)(LONI et al., 2014).

- *Item level.* Recommended items have different types and attributes (or the majority of them). For instance, movies and books belong to distinct domains even with some attributes in common (title, release/publication year, etc.) (GAO et al., 2013)(ENRICH; BRAUNHOFER; RICCI, 2013).

- *System level.* In this level, recommended items are from different systems, which are considered as distinct domains. For example, a user could rate a movie in MovieLens[3] as well as in Netflix[4] (PAN; XIANG; YANG, 2012)(PAN; YANG, 2013).

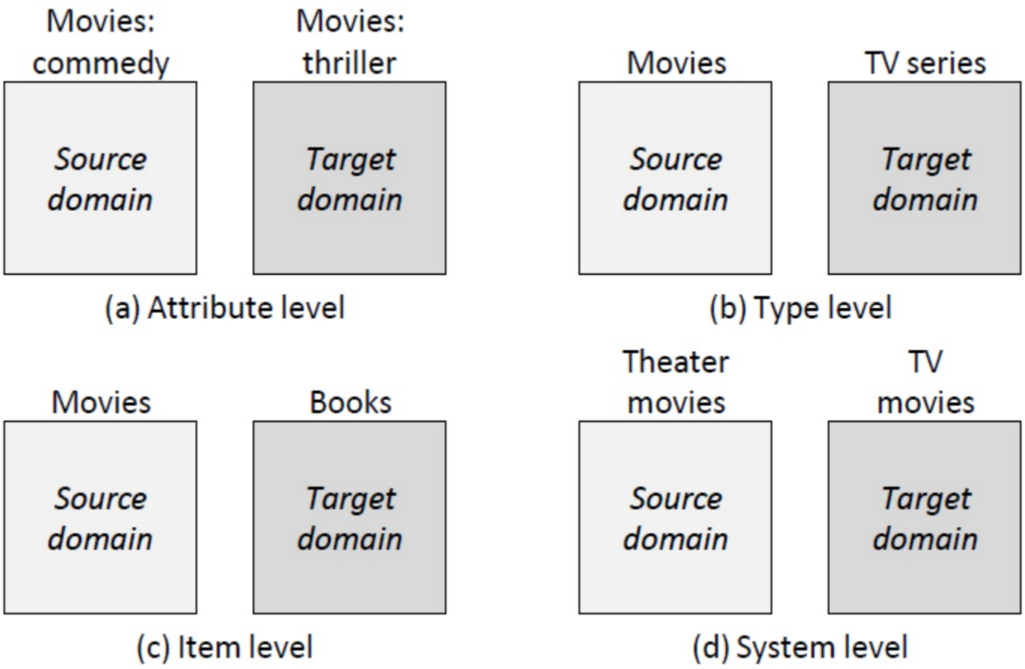


Figure 4 – “Domain” definitions according to attributes and types of recommended items (CANTADOR et al., 2015).

It is important to mention that the notion of domain adopted in this thesis is based on the *Item level*, considering that movies and books belong to different domains, for example.

2.2.2 Cross-Domain Recommendation Tasks

In the literature about cross-domain RSs, the works usually aim to exploit knowledge from a source domain to generate better recommendations in a target domain. Despite

[3] https://movielens.org/
[4] https://www.netflix.com

there is not a unified definition of cross-domain recommender systems, Cantador et al. (2015) identified three recommendation tasks of them:

- *Multi-domain recommendation.* The task is to recommend items in both source and target domains by exploiting knowledge from both domains. In this case, a significant user overlap may be necessary (CARMAGNOLA; CENA, 2009). This recommendation task is becoming feasible since users maintain profiles in several interconnected social networks or websites (CARMAGNOLA; CENA; GENA, 2011).
- *Linked-domain recommendation.* In this task, items are recommended only in the target domain by exploiting knowledge from the source and target domains. This recommendation task has been mainly explored to improve the recommendations in a target domain where there is a lack of user preferences either caused by *cold-start* or *sparsity* problems (LOW; AGARWAL; SMOLA, 2011). A minimal user overlap may be necessary to perform this task (VERAS et al., 2015), and some approaches aim to establish knowledge-based links between source and target domains (MORENO et al., 2012).
- *Cross-domain recommendation.* This task aims to recommend items only in the target domain by exploiting knowledge only from the source domain. In this case, the task is to provide recommendations in a target domain where there is no information about the users in that domain. Therefore, there is not user overlap among domains, and approaches intend to establish knowledge-based links between domains (TIROSHI; KUFLIK, 2012) or to transfer knowledge from the source domain to the target domain (STEWART et al., 2009).

Figure 5 illustrates the three cross-domain recommendation tasks identified by Cantador et al. (2015). In the figure, I_S and I_T are sets of items from source (D_S) and target (D_T) domains, respectively. U_S and U_T are sets of users from source and target domain, respectively. Grey filled areas represent the target users and recommended items, and hatched areas represent the exploited data for generating recommendations.

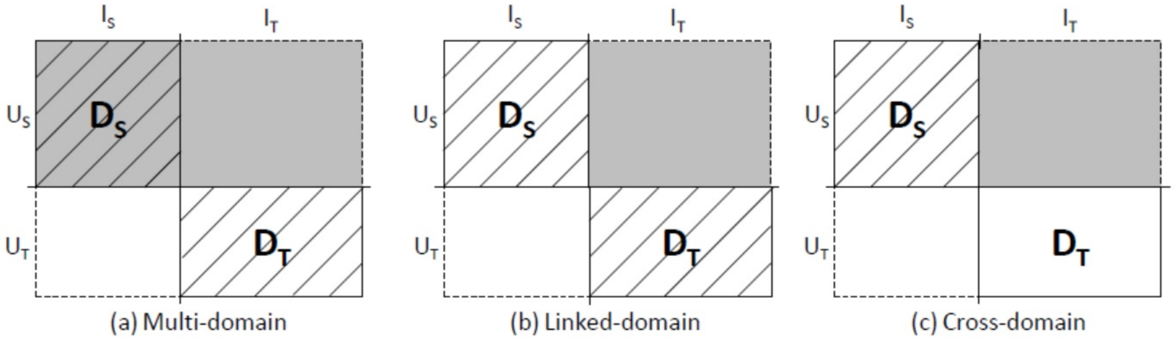


Figure 5 – Cross-domain recommendation tasks (CANTADOR et al., 2015).

As mentioned before in Section 1.3, the problem of this thesis is to explore the user-rating-context tensors (also called “multidimensional matrices” or, informally, “cubes”) from source and target domains to improve recommendations in the target domain. According to this issue and the definitions of a cross-domain RS task mentioned in this section, we can consider that problem as: “how to improve the quality of the *Linked-domain recommendation* task?”.

However, despite the task that we aim to perform is classified as a *Linked-domain recommendation*, we refer the recommender system proposed in this thesis as a *Cross-domain recommender system*. This referral is made as a matter of simplicity and based on the literature of cross-domain RS (CANTADOR et al., 2015), in which the majority of the papers that perform *Linked-domain* and *Multi-domain* recommendation tasks refers themselves as *Cross-domain* RS.

2.2.3 Cross-Domain Recommendation Goals

Likewise the cross-domain recommendation tasks, the cross-domain recommendation goals can vary. According to (CANTADOR et al., 2015), we present some of the most common goals addressed by cross-domain RSs:

- Alleviating the cold-start problem. This issue may occur when a RS is unable to generate recommendations due to an initial lack of user preferences. A possible solution is to obtain the user preferences from other domain (source) in order to enrich the user preferences in the target domain (SHAPIRA; ROKACH; FREILIKHMAN, 2013).

- Alleviating the new user problem. This issue may happen when a user begins using a RS that initially has no knowledge about his/her preferences. In this case, the RS cannot make recommendations. This issue may be alleviated by exploiting the user’s preferences from a different domain (source) (CREMONESI; TRIPODI; TURRIN, 2011)(WINOTO; TANG, 2008)(SANTOS et al., 2012).

- Improving accuracy. Recommender systems may have to deal with a low average number of ratings per user or item, which may negatively affect the quality of the recommendations. Ratings obtained from other domain (source) could increase the rating density in the target domain, which may improve the recommendation quality (STEWART et al., 2009)(MORENO et al., 2012).

- Increasing diversity. Recommender systems may provide similar or redundant items for users. Thus, their satisfaction may be compromised. In this case, the diversity of recommendations could be increased by considering item preferences from multiple domains (WINOTO; TANG, 2008).

Again, as mentioned in Section 1.3, the problem in this thesis is to improve the quality of cross-domain collaborative filtering recommender systems (CD-CFRS). This quality refers to *accuracy improvement* by addition of context-aware techniques while maintaining the advantages of CD-CFRS in relation to cold-start and sparsity issues.

2.2.4 Cross-Domain Recommendation Scenarios

CD-CFRSs are based on the set of ratings provided by users about items of source and/or target domains. According to the overlap among users and/or items of both domains, Cremonesi, Tripodi e Turrin (2011) identified four different cross-domain scenarios:

- *No overlap.* There is no overlap between users and items in the domains. In other words, each item belongs to only one domain, and each user only has preferences for items of one domain. In this case, traditional single-domain CF-based RS cannot make recommendations due to the lack of common data between the domains (ABEL et al., 2011)(SZOMSZOR et al., 2008).

- *User overlap.* Some users have preferences for items of, at least, two domains (source and target), but each item belongs only to a single domain. For instance, this scenario may happen when a dataset has ratings of the same user for two domains (e.g. movies and books) (SAHEBI; BRUSILOVSKY, 2013)(CREMONESI; TRIPODI; TURRIN, 2011).

- *Item overlap.* In this scenario, there are items belonging to distinct domains (source and target). Users can give different ratings for these items depending on their domains. For example, this scenario may occur when a user rates a TV program in multiple systems (e.g. Movielens and Netflix), which could be considered as domains by the RS (CREMONESI; TRIPODI; TURRIN, 2011). In this case, the domain can be classified according to the *System level* (BERKOVSKY; KUFLIK; RICCI, 2007), as described in Section 2.2.1).

- *User and item overlap.* In this scenario, there is overlap between the users as well as between the items (BERKOVSKY; KUFLIK; RICCI, 2007)(TIROSHI; KUFLIK, 2012).

Figure 6 illustrates possible scenarios of user or item overlap between the source and target domains. In the figure, I_S and I_T are sets of items from source (D_S) and target (D_T) domains, respectively. U_S and U_T are sets of users from source and target domain, respectively. Grey filled areas represent the target users and recommended items, and hatched areas represent the user and/or item overlap.

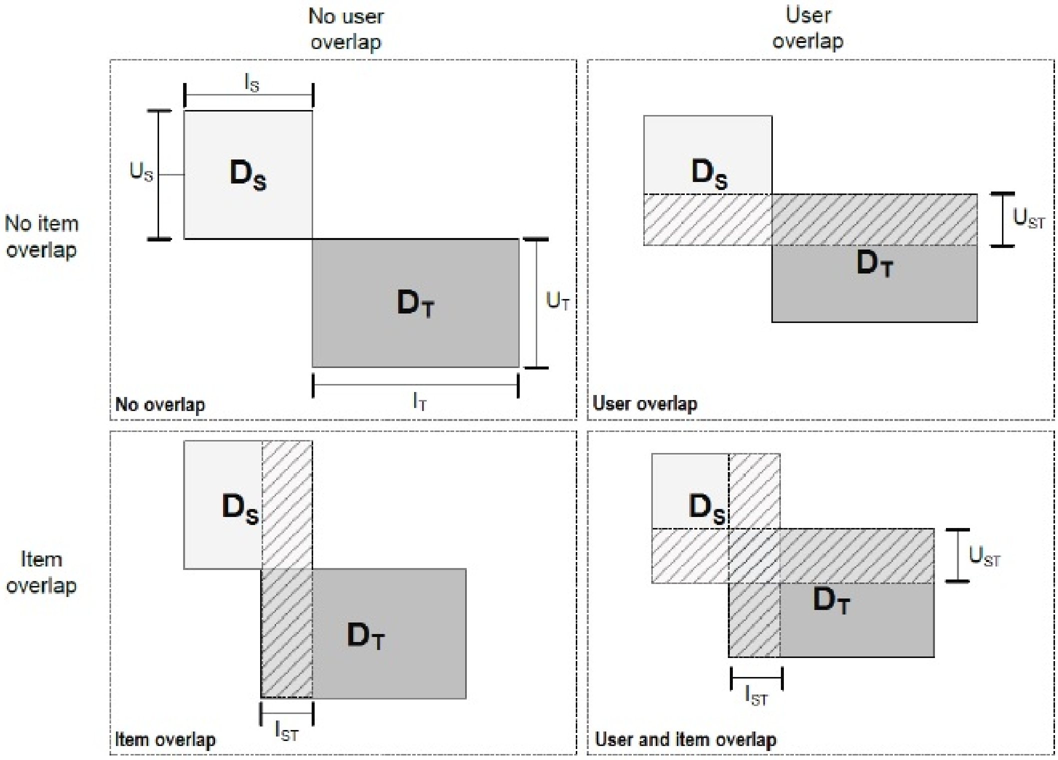


Figure 6 – Possible scenarios of user and/or item overlap between the source and target domains (CREMONESI; TRIPODI; TURRIN, 2011).

As stated in the problem of this thesis (Section 1.3), it is necessary a *User overlap* among source and target domains whereas an *Item overlap* is not. In addition to the literature, we can say that, in our problem, it is also necessary a *Contextual overlap* among such domains, i.e., the same contexts observed in the source domains are observed in the target domain.

2.2.5 Cross-Domain Approaches

As discussed earlier, the cross-domain recommendation has been addressed from various perspectives. This fact led to the development of a variety of recommendation approaches. In many cases, these approaches are difficult to compare, since each one may be based on a different algorithm and adopt different data models about user preferences.

Cantador et al. (2015) analyzed some surveys about cross-domain RS (CREMONESI; TRIPODI; TURRIN, 2011)(FERNÁNDEZ-TOBÍAS et al., 2012), and unified different categorizations from these surveys by proposing a two-level taxonomy of cross-domain RS approaches, focusing on the exploitation of knowledge in cross-domain recommendations. This taxonomy is presented below:

- *Aggregating knowledge.* Knowledge from one or more source domains is aggregated to perform recommendations in a target domain. Three approaches are considered:

1. *Merging user preferences* – user preferences of different forms and scales are aggregated in a single set. These preferences may be ratings, tags, "like/dislike" binary preferences, among others (BERKOVSKY; KUFLIK; RICCI, 2007)(SAHEBI; BRUSILOVSKY, 2013).
2. *Mediating user modeling data* – user modeling data from several recommender systems are aggregated in a single model. For instance, user similarities and user neighborhoods (SHAPIRA; ROKACH; FREILIKHMAN, 2013)(STEWART et al., 2009).
3. *Combining recommendations* – recommendations or predictions of single-domain RS are aggregated in a single RS (ZHUANG et al., 2010)(GIVON; LAVRENKO, 2009).

- *Linking and transferring knowledge.* In this approach, the knowledge is linked or transferred between domains (source and target). Three possible approaches are:

   1. *Linking domains* – source and target domains are linked by a common knowledge, e.g., item attributes, association rules, semantic networks, among others (SHI; LARSON; HANJALIC, 2011)(CHUNG; SUNDARAM; SRINIVASAN, 2007)(AZAK, 2010).
   2. *Sharing latent features* – the source and target domains are related using implicit latent features (ENRICH; BRAUNHOFER; RICCI, 2013)(PAN et al., 2010).
   3. *Transferring rating patterns* – explicit or implicit rating patterns from source domains are exploited in the target domain (LI; YANG; XUE, 2009b)(GAO et al., 2013).

Figure 7 illustrates the taxonomy about cross-domain approaches proposed in (CANTADOR et al., 2015).

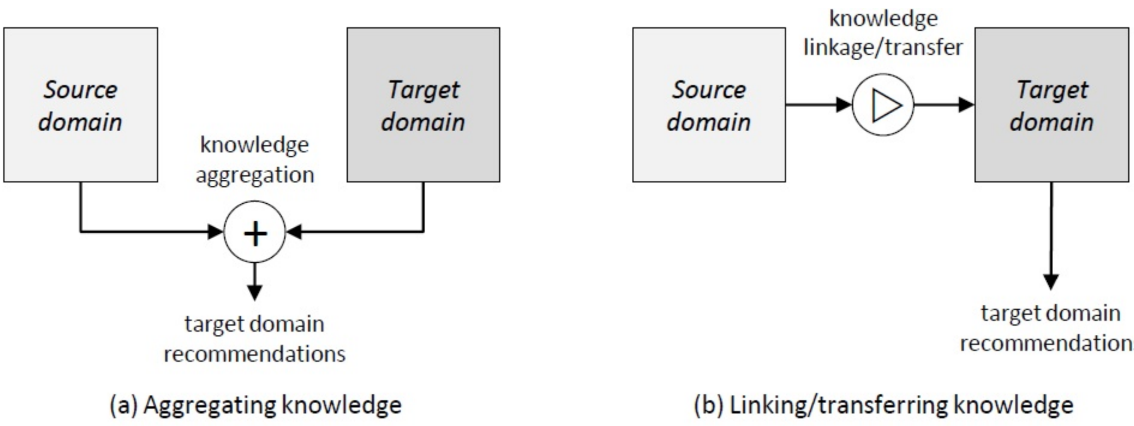

Source domain
Target domain
knowledge aggregation
+
target domain recommendations
(a) Aggregating knowledge
Source domain
knowledge linkage/transfer
Target domain
target domain recommendations
(b) Linking/transferring knowledge

Figure 7 – Cross-domain recommendation approaches taxonomy (CANTADOR et al., 2015).

2.2.6 Cross-Domain Evaluation

Basically, two types of evaluation can be used to compare recommender systems in general (FREYNE; BERKOVSKY, 2013). *Offline experiments* evaluate a RS by analyzing past user preferences. They are typically the easiest evaluation type to perform since they do not require interaction with real users. *Online experiments* demands that a group of real users use the RS in a controlled environment and give feedback about their experience with it.

Cantador et al. (2015) compared the corresponding evaluation methods based on the cross-domain recommendation goals (see Section 2.2.3) and verified that the most of the works about cross-domain RS adopt the *offline experiments*. In this way, we only describe the aspects of *offline experiments*, according to (CANTADOR et al., 2015), in the following subsections.

2.2.6.1 Evaluation Data Partitioning

In the *offline* evaluation of traditional RSs, different data partitions can be adopted (e.g. *Hold-out*, *Leave-some-users-out* and *Leave-all-users-out*)(CANTADOR et al., 2015). Thus, the dataset is usually divided into three subsets of ratings:

- *Training profiles*: contain the set of ratings from users for items that are used to train the algorithms under evaluation;
- *Test profiles*: contain the set of users and their known ratings for items that are used as input by the trained algorithms under evaluation; and
- *Test ratings*: contain the set of users and their hidden ratings for items that are used by the algorithms under evaluation for that they can estimate their actual (or known) ratings.

Regarding the *offline* evaluation of cross-domain RS, the same partitions used for evaluation traditional RSs can be adopted. However, for evaluation cross-domain RS is necessary to consider the use of data from the source and target domains. Depending on that use and the cross-domain RS goal, a certain partition may be more suitable, as described in (CANTADOR et al., 2015) and mentioned below:

- *Hold-out* (see Figure 8-left) can be used when the *test profiles* set is a subset of the *training profiles* set and contains ratings from source and target domains. The *test profiles* set is sampled and hidden from the original dataset, taking into account both domains, without partitioning the users. This kind of partitioning may be suitable to evaluate *linked- and multi-domain* RS with the accuracy goal (SAHEBI; BRUSILOVSKY, 2013)(PAN; YANG, 2013).

- *Leave-some-users-out* (see Figure 8-middle) can be adopted when there is not an intersection between the *training profiles* set and the *test profiles* set. Note that both sets contain ratings from source and target domains as well as the *test profiles* set. This partition type may be suitable to evaluate a cross-domain RS with the new user goal (ABEL et al., 2013)(LI; YANG; XUE, 2009b).

- *Leave-all-users-out* (see Figure 8-right) can be adopted when there is not an intersection between the *training profiles* set and the *test profiles* set, but in this case there is also not intersection between the *training profiles* set and the entire target domain data set. Besides, the *test profiles* and *test ratings* sets have only data from the target domain. Thus, this partition may be suitable to evaluate a cross-domain RS with the cold-start and new item goals (JAIN; KUMARAGURU; JOSHI, 2013)(GOGA et al., 2013).

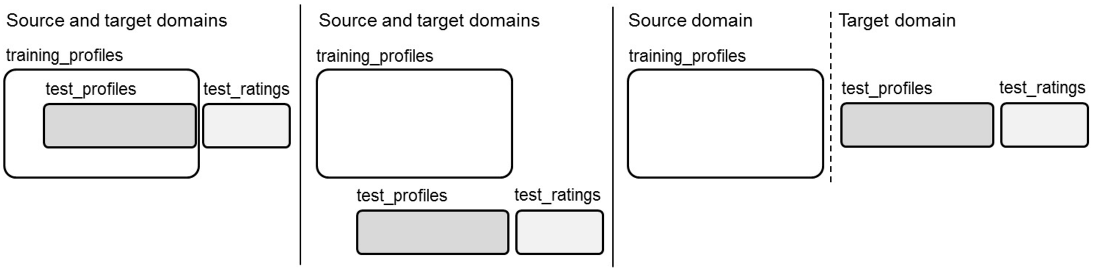


Figure 8 – Partitioning of data: (left) *hold-out*; (middle) *leave-some-users-out*; and (right) *leaveall* (CANTADOR et al., 2015).

2.2.6.2 Evaluation Metrics

As described in Section 2.1.3, there are several metrics for evaluating recommender systems in general. All these metrics can be used in the cross-domain context depending on the cross-domain recommendation goals and tasks (CANTADOR et al., 2015). For instance, *Probabilistic measures* are preferred when the goal is to reduce the sparsity of the target domain; *Ranking measures* are adopted for testing user models, especially in cold-start situations; and *Qualitative measures* are best-suited for the top-N recommendation task.

Finally, the majority of works about cross-domain recommendations adopts prediction metrics (CANTADOR et al., 2015). This is motivated by the fact that the addressed goal is to reduce sparsity and increase accuracy, and the algorithms designed for this are often based on error-metric optimization techniques, which are naturally evaluated using the category of predictive metrics.

2.2.6.3 Sensitivity Analysis

The performance of a cross-domain recommender is mainly affected by three parameters (CANTADOR et al., 2015): the overlap between the source and target domains (SHI; LARSON; HANJALIC, 2011)(CREMONESI; TRIPODI; TURRIN, 2011)(ZHAO et al., 2013)(ABEL et al., 2013), the density of the target domain data (CREMONESI; TRIPODI; TURRIN, 2011)(SHAPIRA; ROKACH; FREILIKHMAN, 2013)(CAO; LIU; YANG, 2010)(PAN et al., 2010), and the size of the target user's profile (SAHEBI; BRUSILOVSKY, 2013)(BERKOVSKY; KUFLIK; RICCI, 2008)(SHI; LARSON; HANJALIC, 2011)(LI; YANG; XUE, 2009b). Thus, it is important to consider the sensitivity of the cross-domain algorithms regarding these three parameters.

According to (CANTADOR et al., 2015), the majority of the works have assumed a full overlap of users between the source and target domains whereas only a few ones have been evaluated by varying the percentage level of user overlap, e.g., in the range 0%-50% (CREMONESI; TRIPODI; TURRIN, 2011) or in the range 0%-100% (ZHAO et al., 2013).

2.3 Context-Aware Recommender Systems

As mentioned before, the context-aware approach uses different contextual information (e.g., location, time, mood, etc.) to improve the accuracy of recommendations (SETTEN; POKRAEV; KOOLWAAIJ, 2004)(ADOMAVICIUS; TUZHILIN, 2015). For some applications it may not be sufficient to consider only users and items.For example, using the temporal context, a travel recommender system could provide a recommendation in the winter that can be very different from the one in the summer. In another example, a user could prefer to watch news programs in the morning and to watch soccer games at night. Therefore, accurate prediction of user preferences might depend on the use of relevant contextual information by recommender systems (ADOMAVICIUS et al., 2005).

In recent years, researchers and companies have developed context-aware recommender systems (CARS) and applied them in a variety of different domains such as movie (SHEPSTONE; TAN; JENSEN, 2014), restaurant (PESSEMIER; DOOMS; MARTENS, 2014), tourism (MAHMOOD; RICCI; VENTURINI, 2009), music (KAMINSKAS; RICCI, 2012)(BALTRUNAS et al., 2011), mobile information (CHURCH et al., 2007), news (LEE; PARK, 2007), among others.

Likewise the cross-domain approach, CARS is a challenging and emergent field of recommender systems (ADOMAVICIUS; TUZHILIN, 2015). In this way, we describe some of its perspectives in the following subsections.

2.3.1 Definition of Context

The definition of "context" varies among different research areas, including Computer Science. Since context has been studied in multiple disciplines, there is not a standard definition of "context". In Computer Science, one of the most known definitions is given by Dey, Abowd e Salber (2001). They refer to "context" as:

> (...) any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and application, including the user and the application themselves.

(BAZIRE; BRÉZILLON, 2005) identified 150 different definitions of context from different fields and made the following observation:

> ... it is difficult to find a relevant definition satisfying in any discipline. Is context a frame for a given object? Is it the set of elements that have any influence on the object? Is it possible to define context a priori or just state the effects a posteriori? Is it something static or dynamic? Some approaches emerge now in Artificial Intelligence [...]. In Psychology, we generally study a person doing a task in a given situation. Which context is relevant for our study? The context of the person? The context of the task? The context of the interaction? The context of the situation? When does a context begin and where does it stop? What are the real relationships between context and cognition?

In the recommender systems area, there is also not a standard definition of "context". However, some authors (PALMISANO; TUZHILIN; GORGOGLIONE, 2008)(ADOMAVICIUS; TUZHILIN, 2015) have a similar point of view about "context" for recommender systems, which is the focus of this thesis. These authors consider "context" as dimensions (e.g. location, time, mood, etc.) and their attributes (e.g. country, city, year, day, sadness, happiness, etc.), which can be used to adapt the recommendations. Based on this definition, we model contextual information in our CD-CARS (Section 3.2). In the next section, we describe how the contextual information can be modelled.

2.3.2 Modelling Contextual Information

Contextual models represent which contextual information is considered in a domain or application, and how this information affects the system's behavior (VIEIRA; TEDESCO; SALGADO, 2009). In general, contextual models define the elements of a particular domain that are considered as context (e.g. location context, temporal context, etc.). They structure entities of a domain and indicate features of these entities, which are managed by the system. However, only this definition of elements does not provide the notion of the context's dynamic. Production rules are usually adopted for this purpose (VIEIRA; TEDESCO; SALGADO, 2009).

Generic contextual models aim to describe the information that must be considered as the context in a generic way. These models provide a classification for an initial set of elements that compose the context in a certain domain (VIEIRA; TEDESCO; SALGADO, 2009). Different applications can reuse the modeled information by extending the model in order to deal with particularities of a particular application. Generic contextual models have been proposed in several areas such as pervasive systems (CHAARI et al., 2007), collaborative systems (VIEIRA; TEDESCO; SALGADO, 2005), data integration (SOUZA et al., 2008), and intelligent systems (GU; PUNG; ZHANG, 2005).

In this direction, researchers have investigated the adoption of several techniques for representation of information and knowledge about context (STRANG; LINNHOFF-POPIEN, 2004)(BETTINI et al., 2010). Vieira, Tedesco e Salgado (2009) summarize some of these techniques, as adapted and described in Table 1.

Each representation technique described in Table 1 has advantages and disadvantages. Thus, there is not a technique that is universally considered as suitable for a certain context-aware system, since different systems have different restrictions and capabilities (VIEIRA; TEDESCO; SALGADO, 2009). A hybrid approach, which combines two or more techniques, may also be adopted as a contextual model. For example, Henricksen e Indulska (2006) proposed a hybrid model that combines ontologies and a graph model based on *Object-Role Modeling* (ORM). (VIEIRA et al., 2008) proposed a hybrid model that combines ontologies and contextual graphs (BRÉZILLON, 2007) to represent the structure of the contextual information and context-aware behavior.

With respect to CARSs, in general they deal with modelling and predicting user preferences by incorporating contextual information into the recommendation process. These preferences are usually modeled as user ratings for items under specific contexts. In this way, the user ratings can be accompanied by contextual information that may be modelled of different types, each type defining a certain aspect of context such as time, location, companion, mood, and so on (ADOMAVICIUS; TUZHILIN, 2015). For instance, by considering movie recommender system, its users and movies can be described according to the following attributes (ADOMAVICIUS; TUZHILIN, 2015):

- Movie: id, title, length, release year, director, genre, among others.
- User: id, name, address, age, gender, profession, and so on.

Table 1 – Summary of techniques for representation of context (VIEIRA; TEDESCO; SALGADO, 2009).

Technique	Advantages	Disadvantages	Brief Description
"Key-Value" pair	Simple structure, and easy to implement and use.	It does not consider hierarchy and is not suitable for applications with complex structures.	A linear search with exact matching of terms.
Markup language	It provides hierarchy moreover, a markup scheme that implements the model itself.	It does not solve incompleteness and ambiguity. Also, it is not suitable for applications with complex structures.	A query language based on marking.
Topic maps	It facilitates the navigation between the contextual elements and the human reading.	It is an immature technique with a low support of tools.	Navigation for semantic networks.
Ontologies	It aggregates rules, concepts, and facts in a single model. Standards make the reuse and sharing easier. It allows semantic comprehension between humans and machines	It does not allow modelling the behavior of the context-aware system. Also, it is a recent technology with a low number of tools.	Inference engine and query languages based on OWL or frames.
Graph models	It facilitates the concepts specification and the definition of the context-aware system behavior.	It does not allow to process the concepts: mapping for data structures.	It can be translated for XML and makes XML processing.

In addition, the contextual information may consist of the following three dimensions, which can also be defined according to attributes:

- Location. The user's location when his/her is watching a movie. It may be composed by the attributes: id, name, street, city, state, country, among others.

- Temporal. The time when a movie is watched. It may be composed by the attributes: date, day of week[5], day type[6], month, year, etc.

[5] "monday", "tuesday", "wednesday", "thursday", "friday", "saturday", "sunday"

[6] "weekday" or "weekend"

- Companion. With who the user watches the movie. It may be composed by the attributes: companion type[7], companion name[8], and so on.

Given that, a user may rate (or watch) a movie depending on "where" he/she will be, "when" he/she will watch and/or "whom" he/she will be with. Beyond these three contextual dimensions illustrated in the example above, Neto e Freitas (2007) identified another three basic dimensions, referred as "5W+1H", which represents "who", "what", "where", "when", "why", and "how".

In addition, each contextual dimension can have a complex structure and hierarchy of attributes and their corresponding values. Although this complexity may be modelled by different forms, traditional models adopt a hierarchical structure of contextual information represented as trees (ADOMAVICIUS et al., 2005)(PALMISANO; TUZHILIN; GORGOGLIONE, 2008). For instance, suppose two contextual dimensions: location and temporal. These dimensions could have the following hierarchies associated with them:

- Location: Street → City → State → Country;
- Time: Date → Day of Week → Month → Year.

Besides the traditional trees for representing the hierarchical structure of contextual information, other ways have been adopted such as Online Analytical Processing (OLAP) (ADOMAVICIUS et al., 2005) and ontologies (KAMINSKAS et al., 2014).

2.3.3 Obtaining Contextual Information

An important aspect of CARS is how to obtain contextual information. Adomavicius e Tuzhilin (2015) mention three of the most common methods:

- *Explicitly.* The contextual information is obtained directly from users (LEE; KWON, 2014)(COLOMBO-MENDOZA et al., 2015). A CARS could have this information by asking direct questions about the users' contexts. For example, a user could select one of the possible contexts provided by the CARS together with the item rating.
- *Implicitly.* In this case, users are not aware about the contextual information gathering process by the CARS. This information can be implicitly obtained in several ways (OH et al., 2014)(PHAM; JUNG; VU, 2014). For instance, a CARS could detect the user location from his/her mobile device location. Another manner could be through temporal information that can be implicitly obtained from the ratings' timestamps. Therefore, the CARS does not need to interact with users to obtain their contexts.

[7] "alone", "friends", "girlfriend/boyfriend", "family", "co-workers", etc.

[8] "Joseph", "Paul", "Laura", etc. In this case, the values could have an associated id.

- *Inferring.* In this method, the contextual information is also obtained implicitly, but the use of statistical or data mining methods is required since the context cannot be obtained in a direct way (SHEPSTONE; TAN; JENSEN, 2014)(WANG; LI; XU, 2015). For example, a CARS could infer the companion (context) of a user from his/her review about a TV program through text mining techniques or observing the kind of the TV program watched by comparing it by using statistical data (e.g. an adult watching a TV program for kids probably is accompanied with kids). *Semantic interpretation* can also be used for inferring contextual information(BOYTSOV et al., 2015). Recently, some works have used the term "situation" for representing a particular contextual information which is inferred by means of *semantic interpretation*(BOUNEFFOUF, 2013)(BOYTSOV et al., 2015). Usually, the "situation" is inferred from sensor data and characterizes situations in which a user interacts with the CARS(BOUNEFFOUF, 2013). For instance, consider a user associated to: a location defined by the coordinates from his phone's GPS; the time from his phone's watch; and the meeting with some person from his agenda. From this knowledge, the CARS could infer that the user is "in a restaurant, with the general manager of a company, at midday, and it is a workday". In this way, that inferred contextual information can be called "situation" represented by three contextual dimensions (temporal, location and companion).

2.3.4 Contextual Information Relevance

Some contextual dimensions can be more relevant in a given application than some other types (ADOMAVICIUS; TUZHILIN, 2015). For example, the weather may be more relevant for recommending places to visit than for recommending movies to watch.

There are several approaches to determine the relevance of a given dimension (or type) of contextual information (ADOMAVICIUS; TUZHILIN, 2015). In particular, this relevance can be verified either manually (e.g. by using domain knowledge of a expert for a given application domain) (BRÉZILLON, 2007) or automatically (e.g. by using several existing feature selection methods from machine learning, data mining, statistics, and so on.) (GUYON; ELISSEEFF, 2003)(LIU; MOTODA, 2012)(CHATTERJEE; HADI, 2015). In a same contextual dimension, there may exist contextual attributes more relevant than others, since a contextual dimension can be modelled as a hierarchical tree (e.g. country x city attributes).

In addition, some parts of a contextual dimension may not be known or available. In this case, some authors classify the source of the contextual information according to the relevance of its acquisition and selection. The classification proposed by Adomavicius e Tuzhilin (2015) is divided into three categories:

- *Fully observable.* The relevant contextual information to the application is known explicitly as well as its structure and its values at the moment when recommendations are made (DOURISH, 2004). For example, a product recommender system may consider that only the *Temporal*, *Purchasing Purpose*, and *Companion* dimensions matter for it. In addition, the recommender system may know the entire structure (attributes and values) of all these three contextual dimensions. For instance, the "day type" attribute from the *Temporal* dimension can have three possible values: "weekday", "weekend", and "holiday".

- *Partially observable.* In this category, only some of the information about the contextual dimensions is known explicitly (PALMISANO; TUZHILIN; GORGOGLIONE, 2008). For example, the recommender system may consider all the contextual dimensions, such as *Temporal*, *Purchasing Purpose*, and *Companion*, but not know all their structure (attributes and values). Note that there may exist different levels of "partial observability". For example, a CARS could only have access to the *Temporal* dimension for a certain user, whereas for another user it knows all the other contextual dimensions (*Purchasing Purpose*, and *Companion*).

- *Unobservable.* In this category, no information about contextual dimensions is explicitly available to the CARS, and it makes recommendations by considering only the inferred context in an implicit way. For example, a CARS could build a latent predictive model to estimate unknown ratings, where unobservable context is modeled using latent variables (KOREN, 2008).

Another aspect of the contextual information relevance is whether and how its importance changes over the time. Adomavicius e Tuzhilin (2015) also classified the contextual dimension relevance into two categories:

- *Static.* The relevant contextual dimensions and their structure remains the same (static) over the time (PALMISANO; TUZHILIN; GORGOGLIONE, 2008). For example, a product recommender system could have three contextual dimensions (*Temporal*, *Purchasing Purpose*, and *Companion*) and they could not change along the entire RS lifetime. In this case, for example, the structure (attributes and values) of the *Purchasing Purpose* dimension does also not change over the time.

- *Dynamic.* In this category, contextual dimensions, attributes or values change in some way over the time (ANAND; MOBASHER, 2006). For example, a CARS (or a CARS designer) could identify that the *Companion* dimension is no longer relevant for the CARS and could remove it from the system. Besides, a CARS could change the structure of some of the contextual dimensions (e.g. by adding new attributes to the *Purchasing Purpose* dimension).

2.3.5 Context-Aware Approaches

According to (ADOMAVICIUS; TUZHILIN, 2015), there are three systematic paradigms (or approaches) found in the CARS literature:

- *Contextual pre-filtering.* In this recommendation paradigm (illustrated in Figure 9), contextual information guides the data selection for that specific context. In other words, information about the current context is used for selecting the relevant set of data (i.e., user ratings) (ADOMAVICIUS et al., 2005)(VERAS et al., 2015). Then, ratings can be predicted using any traditional collaborative-filtering recommender system on the pre-filtered data.

- *Contextual post-filtering.* In this recommendation paradigm (illustrated in Figure 9), contextual information is initially ignored and the ratings are predicted using any traditional collaborative-filtering recommender system on the entire data. Then, the resulting recommendations (or predictions) are adjusted (or filtered) depending on the contextual information of the users (PANNIELLO et al., 2009)(VERAS et al., 2015).

- *Contextual modeling.* Unlike the pre-filtering and post-filtering paradigms, in this paradigm (illustrated in Figure 9) the contextual information is used directly in the recommendation or predictive process (neither before or after it). Although the pre-filtering and post-filtering paradigms can use traditional CF-based algorithms, the *Modelling* paradigm actually needs to make "multidimensional" recommendations by considering contextual information as another dimension, beyond the users and items. Several approaches can be used in this algorithm such as predictive models (e.g. decision tree, regression, probabilistic model, among others) (ANSARI; ESSEGAIER; KOHLI, 2000) (OKU et al., 2006), matrix (or tensor) factorization (KARATZOGLOU et al., 2010)(HIDASI; TIKK, 2012)(BALTRUNAS; LUDWIG; RICCI, 2011)(KIM; YOON, 2014), heuristic calculations(ADOMAVICIUS et al., 2005), among others.

On the other hand, other ad-hoc approaches, which not necessarily need user-ratings, have also been found in CARS literature and could be used according to the paradigms described above. Véras et al. (2015) describe some of these ad-hoc approaches:

- *Contextual rules*: in this category, there are all kinds of rules that allow recommender systems to sense and to react based on their context. In general, these rules follow the same approach of "event-condition-action" (ECA) rules (MOON et al., 2006), "Key-value" rules (SONG; MOUSTAFA; AFIFI, 2012), among others.

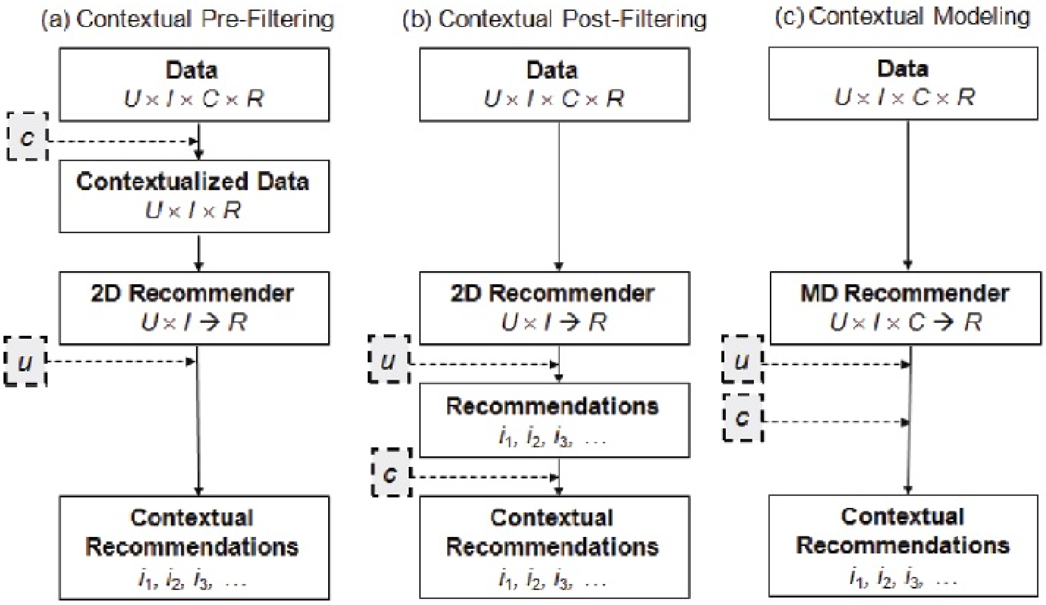


Figure 9 – Paradigms for incorporating context in recommender systems (ADOMAVICIUS; TUZHILIN, 2015).

- *Contextual Ontology*: contextual ontologies are not algorithms, but they are crucial to other knowledge-based context-awareness techniques. Most studies that used contextual ontologies also adopted some semantic-based inference. Thus, the combination of semantic-based and context-awareness techniques is present in many studies (KAMINSKAS et al., 2014)(MOE; AUNG, 2014b).

- *Similarity-based*: instead of using similarity metrics to compare user or items, in this approach, algorithms compare contexts in order to recommend items (ALHAMID et al., 2015)(VILDJIOUNAITE et al., 2009)(WANG; LI; XU, 2015). The context can be represented in several ways such as tags, key-value pairs, among others.

- *Supervised learning*: in this approach, a set of labeled examples is produced, where each example is composed by features extracted from contextual attributes (e.g. time of the day, mood, etc.). The task of supervised learning is, given a training set, to learn a function that predicts the user preferences based on the contextual features. Examples of algorithms adopted in this approach include Support Vector Machines (VILDJIOUNAITE et al., 2009), case-based reasoning (VILDJIOUNAITE et al., 2009), reinforcement learning (MOON et al., 2009), among others.

However, these ad-hoc approaches are difficult to reproduce in distinct domains, since they are usually designed for specific ones. Thus, the algorithms proposed in this thesis (described in Section 3.3.1) follow the systematic paradigms described in (ADOMAVICIUS; TUZHILIN, 2015).

2.3.6 CARS Evaluation

As described in Section 2.1.3, there are several metrics for evaluating recommender systems in general. All these metrics can be used for evaluating a CARS depending on its purpose (ADOMAVICIUS; TUZHILIN, 2015).

However, the evaluation is one of the main research issues and directions for CARS (ADOMAVICIUS; TUZHILIN, 2015). Only a few works have deeply studied the performance evaluation of several CARS approaches and techniques, besides their benefits and limitations. One of these works is presented in (PANNIELLO; TUZHILIN; GORGOGLIONE, 2014), which performed a categorical evaluation and comparison of several contextual techniques under a variety of situations. For instance, the authors compared different recommendation tasks (e.g. to recommend all relevant items, to recommend only the *top-n* relevant items, etc.), different evaluation metrics (e.g. accuracy, diversity, etc.), and the granularity of the processed contextual information, as well as other evaluation perspectives.

Another example is the work presented in (CAMPOS; DÍEZ; CANTADOR, 2014), which focused on exploring "time" as one of the most relevant and widely used contextual dimensions in many CARS. For example, the authors reviewed common evaluation practices and methodological issues related to the comparative evaluation of *time-aware* recommender systems. They also demonstrated that the choice of the assessment conditions impacts the classification (or ranking) performance of different recommendation strategies. For that, they proposed a methodological framework for a robust and fair evaluation process.

The works mentioned above represent an important step in direction to a more reproducible and standardized evaluation methodology of CARS. Analogously to (PANNIELLO; TUZHILIN; GORGOGLIONE, 2014), we performed different evaluation tasks (prediction and classification), besides verifying the recommender system performance by using contextual information from distinct dimensions, as we describe in Section 5.1.

2.4 Related Works

In this section, we present related works to this thesis, divided into two subsections. In Section 2.4.1, we present some cross-domain recommender systems based on collaborative filtering without considering context-aware techniques, whereas in Section 2.4.2, we describe related cross-domain recommender systems that use contextual information, highlighting their limitation in comparison to our proposed CD-CARS.

2.4.1 Cross-Domain Recommendation based on Collaborative Filtering

As mentioned before, the cross-domain recommendation has been addressed from various perspectives. This fact led the development of a wide range of recommendation approaches, as categorized by Cantador et al. (2015) (see Section 2.2.5).

In general, *Merging user preferences* from different domains is the most direct way to address the cross-domain recommendation problem and it is among the most widely used strategies for the cross-domain recommendation (CANTADOR et al., 2015). The massive use of this approach can be explained by the fact that it has been shown that enriching sparse user preference data in a certain domain by adding user preference data from other domains, can significantly improve the generated recommendations under cold-start and sparsity conditions (SHAPIRA; ROKACH; FREILIKHMAN, 2013)(SAHEBI; BRUSILOVSKY, 2013). Figure 10 illustrates the *Merging user preferences* approach, in which user rating matrices from source (D_S) and target (D_T) domains are merged, and traditional single-domain CF-based recommender systems can be used on the merged data to recommend items from target domain (I_T).

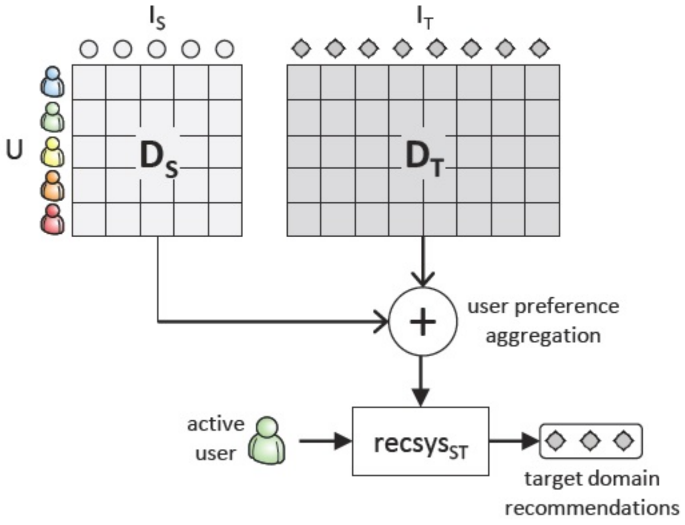


Figure 10 – Merging user preferences approach (CANTADOR et al., 2015).

Given that and the fact that our proposed CD-CARS is based on the *Merging user preferences* approach, we describe some works related to it according to the cross-domain RS perspectives described in Section 2.2. Table 2 describes the classification of related papers for these perspectives.

Berkovsky, Kuflik e Ricci (2007) focused on cross-domain mediation of user models in CF-based recommendations. In the cross-domain mediation, the user modeling data is imported from remote systems (source domains) exploiting the same CF recommendation technique as the target system (domain). Hence, both source and target domains represent the user models as a list of ratings provided by a user on both domains (user overlap). The CD-CFRS imports the complete set of nearest neighbors calculated by the remote system and uses these similarities in the target domain (cross-domain task). The prediction accuracy of that CD-CFRS was measured through the MAE metric (probabilistic) by using the EachMovie dataset (MCJONES, 1997). In this dataset, movies from different

Table 2 – Cross-Domain CF-based RS using the *Merging user preferences* approach.

Paper	**Domain Level**	**Task**	**Goal**	**Scenario**	**Evaluation**
(BERKOVSKY; KUFLIK; RICCI, 2007)	Item Attribute	Cross-Domain	Accuracy	User and Item Overlap	Probabilistic
(WINOTO; TANG, 2008)	Item	Linked-Domain	Diversity and Accuracy	User Overlap	Probabilistic
(NAKATSUJI et al., 2010)	Item	Cross-Domain	Accuracy	User Overlap	Probabilistic
(CREMONESI; TRIPODI; TURRIN, 2011)	System	Linked-Domain and Cross-Domain	Cold-start and Accuracy	User Overlap	Ranking
(SANTOS et al., 2012)	Item	Multi-Domain and Linked-Domain	Cold-start and New user	User Overlap	Ranking
(TIROSHI et al., 2013)	Item	Cross-Domain	Accuracy	User Overlap	Ranking
(SAHEBI; BRUSILOVSKY, 2013)	Item	Linked-Domain	New User and Accuracy	User Overlap	Probabilistic
(SHAPIRA; ROKACH; FREILIKHMAN, 2013)	Item	Linked-Domain	Cold-start and Accuracy	User Overlap	Probabilistic and Ranking
(LONI et al., 2014)	Item	Cross-Domain	Accuracy	User Overlap	Probabilistic
Proposed CD-CARS	Item	Linked-Domain	Cold-start, New user and Accuracy	User Overlap	Probabilistic and Ranking

genres are considered as from distinct domains (*item attribute* domain level), and one item can belong to two or more genres (domains), i.e., there is an item overlap.

Winoto e Tang (2008) investigated alternative benefits that cross-domain recommendations may have such as serendipity and diversity. For that, they applied a traditional single-domain CF-based algorithm for making cross-domain recommendations by considering aggregated ratings from source and target domains. The unique change made for them in the algorithm was in the weight of the Pearson correlation, which was modified in order to find neighbors with higher number of co-rated items in the target domain. The CD-CFRS performance was measured through the MAE metric (probabilistic) by using a real collected dataset. This dataset was composed by several domains such as movies, books, games, among others (*item* domain level). On it, a user could rate items from all domains (user overlap).

Instead of aggregating user preferences directly, several researches have focused on directed weighted graphs that link user preferences from multiple domains (NAKATSUJI et al., 2010)(CREMONESI; TRIPODI; TURRIN, 2011)(TIROSHI et al., 2013).

Nakatsuji et al. (2010) created a domain-specific-user graph (DSUG) for each domain (source and target). In the DSUG, the nodes are users and sets weighted edges between user nodes according to the similarity of users computed in each domain. Also, the DSUGs from distinct domains are connected to create a cross-domain-user graph (CDUG). Thus, the cross-domain RS performs a Random Walk with Restarts (RWR)(LOVÁSZ et al., 1996) on the CDUG from the active user node, and extracts user nodes that are present in DSUGs from the target domain that do not include the node of the active user in the source domain. The authors evaluated the cross-domain RS by using a dataset from two different domains (movies and music) with user overlap, and verified that the accuracy (measured with the MAE metric) of their method is higher than one method that predicts user preference by merging the user ratings from all domains.

Cremonesi, Tripodi e Turrin (2011) built a graph whose nodes are associated with items and whose edges reflect rating-based item similarities. In this case, the inter-domain connections are the edges between pairs of items in different domains. The authors also proposed to enhance inter-domain edges by discovering new edges and strengthening existing ones, through strategies based on the transitive closure. Through datasets from different systems (*system* domain level) and with the same item type (movies), they evaluated several CF-based (nearest neighborhood and latent factor techniques) algorithms using the built multi-domain graph. The authors estimated the accuracy of the algorithms in terms of F-metric (ranking) (CREMONESI; KOREN; TURRIN, 2010) by varying the user overlap level (sensitivity analysis) in the datasets.

Santos et al. (2012) proposed an architecture for a recommender system in an inter-application environment and compared traditional (single-domain) and inter-applications recommendations through the Breese metric (Ranking). The proposed recommender system handles different profiles from various applications (with different user-rating scales and forms) by normalizing them. Besides, its recommendation module is based on traditional collaborative filtering techniques, which are adjusted for making cross-domain recommendations (Multi-domain and Linked-domain). The authors developed a web application in order to obtain real user preferences for generating three datasets with different item domains (movies, books, bands and singers) and user-rating scales/forms. In the experiments, 60 users evaluated at least 10 items in the analyzed item domains (user overlap) so that each user have, at least, 30 preferences in the system. In these experiments, the authors evaluated several scenarios (combining distinct domains as target) in order to verify the quality of the proposed RS face to cold-start and new user issues.

Tiroshi et al. (2013) merged data from source and target domains into a single bipartite user-item graph. From it, several statistical and graph-based features of users and items were extracted. These features were exploited by a machine learning algorithm that addressed the recommendation problem as a binary classification problem. Then, they applied a Random Forest classifier (LIAW; WIENER, 2002) in order to recommend items from the target domain based on the user preferences in the source domain, present in the unified bipartite user-item graph. The authors collected a dataset containing user preferences in multiple domains (book, movie and music) extracted from social network profiles (Facebook[9], Last.fm[10], LinkedIn[11], etc.) with user overlap. They adopted Precision (ranking) as evaluation metric in order to verify the accuracy of their cross-domain RS.

Sahebi e Brusilovsky (2013) examined the impact of the size of user profiles in the source and target domains on the quality of cross-domain recommendations, and showed that aggregating ratings from a dense source domain increases the accuracy of recommendations in the target domain under cold-start conditions. Basically, the authors applied the k-Nearest Neighbor (k-NN) algorithm (LAROSE, 2005) in the aggregated ratings in order to perform recommendation in the target domain (linked-domain recommendation). For evaluating the CD-CFRS, they adopted a dataset (SAHEBI; COHEN, 2011) with two item domains (book and movie) and user overlap. The accuracy of the system was measured with the RMSE metric (Probabilistic).

Similar to (SAHEBI; BRUSILOVSKY, 2013), the work proposed in (SHAPIRA; ROKACH; FREILIKHMAN, 2013) showed significant accuracy improvement by using aggregation-based methods when the user preferences from the target domain are sparse. In this case, the authors used a dataset composed of unary Facebook “likes” as user preferences from several domains (movies, TV shows, and music). Two algorithms were adopted by them: the k-NN algorithm with Jaccard similarity (AMATRIAIN et al., 2011), since the user ratings are in unary form, and the “Facebook popularity” one, which simply lists the top-mentioned user preference items in the Facebook profiles. The CD-CFRS accuracy was mainly measured by Recall (ranking) and MAE metrics.

Finally, Loni et al. (2014) proposed a CD-CFRS with factorization machines (RENDLE, 2012) capable of transferring knowledge from different auxiliary domains to a target domains to improve rating predictions in the target domain. The CD-CFRS encodes rating matrices from multiple domains as real-valued feature vectors. With these vectors, the factorization machine finds patterns between features from the source

[9] http://www.facebook.com
[10] http://www.last.fm
[11] http://www.linkedin.com

and target domains, and estimates preferences associated with the input vectors. The CD-CFRS accuracy is evaluated through MAE and RMSE metrics. Besides, the Amazon dataset (LESKOVEC; ADAMIC; HUBERMAN, 2007) is used for evaluation purposes. This dataset is composed by user ratings from three different domains (book, music, and television) with user overlap.

It is important to notice that only four related papers are based on the "Linked-Domain" task (see Table 2) like our proposed CD-CARS. While (WINOTO; TANG, 2008), (SANTOS et al., 2012), (SHAPIRA; ROKACH; FREILIKHMAN, 2013) and (SAHEBI; BRUSILOVSKY, 2013) explored only traditional single-domain CF-based algorithms used for making cross-domain recommendations, (CREMONESI; TRIPODI; TURRIN, 2011) proposed a cross-domain CF-based algorithm and compared it with those traditional ones.

The cross-domain CF-based algorithms presented in Table 2 differ from our proposed CD-CARS by the fact that they do not take into account any contextual information for making recommendations. Therefore, in the next section, we present some related works about cross-domain algorithms that use contextual information in order to improve the quality of their recommendations.

2.4.2 Cross-Domain Recommendation based on Context-Awareness

This thesis focuses on investigating the use of contextual information to enhance cross-domain collaborative filtering recommendations. According to Fernández-Tobías et al. (2012), no previous work had addressed the cross-domain recommendation task by deploying of contextual features until then. Seminal works have been published more recently (BRAUNHOFER; KAMINSKAS; RICCI, 2013)(ZHANG; YUAN; YU, 2014)(MOE; AUNG et al., 2013)(MOE; AUNG, 2014b)(MOE; AUNG, 2014a)(KAMINSKAS et al., 2014) (TANG; WAN; ZHANG, 2014)(TEKIN; SCHAAR, 2015)(JI; SHEN, 2015), adopting various approaches to this issue, from semantic techniques to supervised learning, for instance.

Taking into account the cross-domain RS and CARS aspects described in Section 2.2 and Section 2.3, respectively, we describe and categorize some related works that make use of context-awareness techniques for providing cross-domain recommendations. Table 3 presents a classification of these works regarding cross-domain RS aspects whereas Table 4 shows the categorization of them with respect to the CARS perspectives.

TripFromTV+ (BLANCO-FERNÁNDEZ et al., 2011) selects personalized tourism resources (target domain) for Digital TV viewers, by inferring their particular preferences from the kind of TV programs (source domain) that they enjoyed and from their activity on social networking sites (user overlap). The user profile, contextual information, and item (resources) are modelled through ontologies, so, the recommendation is made with semantic reasoning methods. Specifically, relevant context information is associated with

Table 3 – Classification of context-aware-based related works regarding cross-domain RS aspects.

Paper	Domain Level	Task	Goal	Scenario	Evaluation
(BLANCO-FERNÁNDEZ et al., 2010)(BLANCO-FERNÁNDEZ et al., 2011)(BLANCO-FERNÁNDEZ et al., 2011)	Item	Cross-Domain	New Item and Diversity	User Overlap	User Satisfaction
(BRAUNHOFER; KAMINSKAS; RICCI, 2013)	Item	Cross-Domain	Diversity	No Overlap	User Satisfaction
(YUAN et al., 2012)(ZHANG; YUAN; YU, 2014)	Item	Multi-Domain	Diversity	User Overlap	Qualitative and Ranking
(MOE; AUNG et al., 2013)(MOE; AUNG, 2014b)(MOE; AUNG, 2014a)	Item	Cross-Domain	Accuracy	No Overlap	Ranking
(KAMINSKAS et al., 2014)	Item	Cross-Domain	Diversity	No Overlap	Ranking
(TANG; WAN; ZHANG, 2014)	Item Attribute	Cross-Domain	Diversity	No Overlap	Ranking
(TEKIN; SCHAAR, 2015)	Item	Multi-Domain	Diversity	No Overlap	Qualitative
(JI; SHEN, 2015)	Item	Linked-Domain	Accuracy	User Overlap	Probabilistic
Proposed CD-CARS	Item	Linked-Domain	Cold-start, New user and Accuracy	User Overlap	Probabilistic and Ranking

each tourism resource (e.g. *opening times*, *dates*, *duration*, *location* and *ticket price*) and matched by the recommendation strategy against the user's partially observed and static context (e.g *location*, *temporal*, etc.). The authors performed a simple experiment with 95 users in order to measure their satisfactions in the use of the *TripFromTV+*. That work is difficult to compare with other cross-domain RSs since its evaluation is empirical and the *TripFromTV+* adopts a knowledge-based method, which in general is domain-specific and requires a great knowledge about the domains and their interconnections.

Braunhofer, Kaminskas e Ricci (2013) addressed the cross-domain recommendation task by developing a mobile application that selects music content (target domain) that fits a place of interest (source domain) visited by the user. For that, the application used

Table 4 – Classification of context-aware-based related works with respect to CARS aspects.

Paper	Representation	Obtention	Relevance	Approach
(BLANCO-FERNÁNDEZ et al., 2010)(BLANCO-FERNÁNDEZ et al., 2011)(BLANCO-FERNÁNDEZ et al., 2011)	Ontologies	Implicitly	Partially Observable (Static)	Contextual Ontology
(BRAUNHOFER; KAMINSKAS; RICCI, 2013)	Key-Value	Explicitly and Implicitly	Partially Observable (Dynamic)	Similarity-based
(YUAN et al., 2012)(ZHANG; YUAN; YU, 2014)	Key-Value	Explicitly	Partially Observable (Dynamic)	Modelling
(MOE; AUNG et al., 2013)(MOE; AUNG, 2014b)(MOE; AUNG, 2014a)	Ontologies	Explicitly	Fully Observable (Static)	Contextual Ontology
(KAMINSKAS et al., 2014)	Ontologies	Explicitly	Fully Observable (Static)	Contextual Ontology
(TANG; WAN; ZHANG, 2014)	Key-Value	Inferred	Fully Observable (Dynamic)	Supervised Learning
(TEKIN; SCHAAR, 2015)	Key-Value	Implicitly	Partially Observable (Static)	Modelling
(JI; SHEN, 2015)	Key-Value	Implicitly	Partially Observable (Static)	Modelling
Proposed CD-CARS	Key-Value	Implicitly and Inferred	Partially Observable (Static)	Pre-Filtering, Post-Filtering and Modelling

the users' location and emotional tags (contextual information) assigned to both music tracks and point-of-interests (POIs), and adopted similarity metrics (e.g cosine, Jaccard, etc.) for establishing a match between music tracks and POIs based on their emotional tags. These tags were given explicitly by users without any user overlap between the domains. Through a live user study with 10 users, the authors evaluated if the mobile application is capable of providing recommendation with a certain grade of diversity. That work is domain-specific and does not taking into account the users' preferences in the cross-domain recommendation. In contrast, it recommends items from the target domain (music) directly related to the source domain (POI) according to their contextual information. The user's context is only used for identifying in which POI he/she is located. Therefore, the same recommendations may be made for different users located in that POI.

Yuan et al. (2012) proposed a context-aware feature selection framework for cross media recommendation in a digital library. The recommended items in that digital library (DL) can be from different domains (e.g. book, movie and music) and have different tags defined by the users representing the contextual features such as emotions, location, and so on. Thus, the set of items, users and contexts is represented by a user-item-context tensor. The authors initially applied a tensor factorization method (TUCKER, 1966) in that tensor and then used a k-NN clustering algorithm to recommend the top-n items regardless the items' domain (multi-domain recommendation task). At least, the authors performed experiments by using the Douban[12] cross media dataset with user overlap in order to evaluate the quality of cross media recommendations by means of recall and diversity metrics. As we described, the goal of that work is to recommend items in several domains aiming to improve the diversity of the system. In this case, only items from a specific domain could be recommend without taking into account the current user's context. Besides, the contextual features considered in that work are based on user tags, which can be too varied among different users and also are applied to items. In this way, the users' contexts are not considered by that work.

In (MOE; AUNG, 2014b), a cross-domain RS was developed to recommend cosmetics (target domain) related to skin care problems (source domain). The developed system represented the contextual information through ontologies. This contextual information was related to cosmetics such as *Place Zone*, *Age Level*, *Cosmetics Brand*, *Season*, and *Price Range*. The system was developed by using Taxonomic conversational case-based reasoning (Taxonomic CCBR) on ontological properties to manage personalization systematically (GUPTA, 2001), Ford-Fulkerson algorithm (PARAMESWARAN; VENETIS; GARCIA-MOLINA, 2011) applied to build the bridge of the semantic concepts between source and target domains and a technique for gathering recommendations according to the users' contexts (called TOPSIS) (JADIDI; FIROUZI; BAGLIERY, 2010). The accuracy of the developed system was measured by means of raking metrics such as Precision, Recall and F-measure in a simple dataset without user overlap and with information about cosmetics and skin care problems. Likewise (BLANCO-FERNÁNDEZ et al., 2011), the work presented in (MOE; AUNG, 2014b) relies on the extensive use of knowledge about two domains, in which their interconnections must be established a priori by the RS designer. Thus, their domain-specific approach may be difficult to be adjusted for other domains (e.g. book, movie, music, etc.).

[12] http://www.douban.com

By extending the work proposed in (BRAUNHOFER; KAMINSKAS; RICCI, 2013), Kaminskas et al. (2014) proposed a knowledge-based framework for semantic networks that link concepts from different domains. The framework propagates the node weights, in order to identify target concepts that are most related to the source concepts. Based on data from DBpedia[13] without user overlap, the authors evaluated the framework for recommending music (target domain) related to places of interest (source domain) according to location and time as contextual information explicitly defined by the users. Similar to evaluated in (BRAUNHOFER; KAMINSKAS; RICCI, 2013), the authors evaluated the knowledge-based framework by means of a empirical experimentation with some users. Therefore, the same criticism that we mentioned for (BRAUNHOFER; KAMINSKAS; RICCI, 2013) can be applied to that work.

Tang, Wan e Zhang (2014) defined a task of cross-language context-aware citation recommendation[14], aiming to recommend English citations (target domain) for a given part of the text (context) where a citation is made (e.g. introduction, motivation, related work, etc.) in a Chinese paper (source domain). This task is very challenging because the contexts and citations are written in different languages and there is a language gap when matching them. To handle this problem, they adopted a method that uses machine translation (MT) to translate contexts and/or citations, and then the problem is reduced to the monolingual context-aware citation recommendation. With this reduced problem, they proposed a bilingual context-citation embedding algorithm (called BLSRec-I), which can learn a low-dimensional joint embedding space for both contexts and citations. They evaluated the proposed methods based on a real dataset that contains Chinese contexts and English citations. In this case, there is not the concept of user or item overlap, given that the paper is considered as a "user". In this way, they adopted three ranking measures (Recall, MAP and Mean Reciprocal Rank) in order to evaluate the positions of the right citations in the ranking list for each given context. Therefore, that work is designed especially for the citation domain and intended for matching citations in the correct context, without taking into account users.

Tekin e Schaar (2015) proposed a multimedia content aggregation framework, which gathers content generated by multiple sources in order to provide content on demand for its users. They proposed a content aggregation algorithm, called DIStributed COntent Matching (DISCOM), capable of learning which content to gather and performing a matching between it and users' preferences, by exploiting similarities between user types. In that system, each user is represented together with its context, which is considered as the user's type. Based on this user type (context), the content aggregation framework requests content from one of the multimedia sources (multi-domain recommendation). Thus, the context can be represented as user information such as age, gender, among

[13] http://wiki.dbpedia.org

[14] Despite being a cross-language task, we can classify it as a cross-domain one due to its approach.

others. In addition, it may also be represented by the device type that the user is using (e.g., computer, mobile phone, etc.). The authors adopted two datasets without user or item overlap for evaluation purposes: the Yahoo! Today Module (YTM) (LI et al., 2010) and a collected one with music items. Based on these datasets, they evaluated the diversity of the recommendations generated by the content aggregation framework. A limitation of that work is the fact that the contextualized recommendations are provided for user/device types, thus, they are not personalized to a single user.

Ji e Shen (2015) proposed an improved group-aware CF-based algorithm[15] which predicts a user rating using a weighted sum of similar ratings from multiple user subgroups. The algorithm is based on matrix factorization and CodeBook Transfer (CBT) (LI; YANG; XUE, 2009a). The user subgroups are defined according to contextual information available from their ratings. This contextual information can be divided into three categories: users' contexts (age, gender, etc.), items' contexts (genre, release date, etc.), and environments' contexts from the user ratings (time, place, etc.). Experiments were done based on three datasets with distinct domains (book, movie and music) with user overlap. The accuracy of the proposed algorithm was evaluated through probabilistic measures (MAE and RMSE). As we can note, the same limitation that we mentioned for (TEKIN; SCHAAR, 2015) can be applied to that work.

In summary, the majority of the cross-domain RS described and categorized above relies on ad-hoc approaches of CARS (Contextual Ontology, Similarity-based and Supervised Learning), which may be difficult to customize to new situations, once that they are usually designed for a specific domain and do not take into account context obtained from user ratings (ADOMAVICIUS; TUZHILIN, 2015). As it can be seen from Table 4, our proposed CD-CARS, in turn, relies on the use of systematic context-aware techniques (Pre-Filtering, Post-Filtering and Modelling). These techniques have been successfully adopted for single domain RS and, in general, require little domain knowledge, since they are based on context obtained from user ratings (ADOMAVICIUS; TUZHILIN, 2015).

It is important to mention that some related works adopted the *Modelling* systematic context-aware technique (YUAN et al., 2012)(TEKIN; SCHAAR, 2015)(JI; SHEN, 2015), as it can be seen from Table 4. However, two of them (YUAN et al., 2012)(TEKIN; SCHAAR, 2015) perform different cross-domain tasks and have distinct cross-domain goals in comparison to our CD-CARS. In addition, the work proposed in (TEKIN; SCHAAR, 2015) is designed for making cross-domain recommendations with no overlap among users, in opposite to the proposed in our CD-CARS. The related work proposed in (JI; SHEN, 2015) is the most similar to the our proposed CD-CARS according to the classification in

[15] The authors consider their work as a context-aware RS by determining that a group can be viewed as a user type (context).

Table 3 and Table 4, but (JI; SHEN, 2015) differs from it once that they only propose a single systematic context-aware technique (*Modelling*) and it is also not based on the *Merging user preferences* approach from cross-domain CF-based algorithms, which allows that traditional CF-based algorithms can also be used. Finally, the work proposed in (JI; SHEN, 2015) is intended for making recommendations for group of users instead of recommending items to a singular user. Table 5 summarizes the main limitations of the related works mentioned above in comparison to our proposed CD-CARS.

Table 5 – Main limitations of context-aware-based related works in comparison to our proposed CD-CARS.

Paper	**Systematic approach**	**Accuracy Goal**	**Linked-Domain Task**	**User Overlap**	**Merging user preferences**
(BLANCO-FERNÁNDEZ et al., 2010)(BLANCO-FERNÁNDEZ et al., 2011)(BLANCO-FERNÁNDEZ et al., 2011)	No	No	No	Yes	No
(BRAUNHOFER; KAMINSKAS; RICCI, 2013)	No	No	No	No	No
(YUAN et al., 2012)(ZHANG; YUAN; YU, 2014)	Yes	No	No	Yes	Yes
(MOE; AUNG et al., 2013)(MOE; AUNG, 2014b)(MOE; AUNG, 2014a)	No	Yes	No	No	No
(KAMINSKAS et al., 2014)	No	No	No	No	No
(TANG; WAN; ZHANG, 2014)	No	No	No	No	No
(TEKIN; SCHAAR, 2015)	Yes	No	No	No	Yes
(JI; SHEN, 2015)	Yes	Yes	Yes	Yes	No
Proposed CD-CARS	Yes	Yes	Yes	Yes	Yes

2.5 Final Remarks

In this chapter, we presented the main concepts related to this thesis as well as its related works. The research about these concepts provides a background for the understanding of the proposed CD-CARS, described in the next chapter.

3 CD-CARS Proposal

In this chapter, we describe the CD-CARS proposal. For that, we formalize the cross-domain context-aware recommendation problem (Section 3.1) and model the contextual information (Section 3.2). In Section 3.3, we describe the proposed CD-CARS algorithms whereas in Section 3.3.2 we present the cross-domain algorithms that can be adopted as base, in combination with the proposed CD-CARS. At last, in Section 3.4, we mention the final remarks of this chapter.

3.1 CD-CARS Problem Formalization

As mentioned in Chapter 1, the majority of the proposed approaches to cross-domain recommendation deals with collaborative filtering (CF) (CREMONESI; TRIPODI; TURRIN, 2011)(FERNÁNDEZ-TOBÍAS et al., 2012). CF is more convenient for cross-domain recommendation due to the lack of homogeneous description of item content in different domains. It can rely only on user ratings of items, usually represented by user-rating matrices (User x Item). Also, most of the available cross-domain RS suggest items regardless of the contextual conditions, which can be important to predict the users' preferences in a particular context. Despite the large number of existing cross-domain CF-based recommender systems (CD-CFRS), the synergy between them and context-aware techniques is still little explored (FERNÁNDEZ-TOBÍAS et al., 2012)(CANTADOR; CREMONESI, 2014).

In this way, we address the cross-domain recommendation problem under the CF and context-awareness perspectives. For that, as defined by Adomavicius e Tuzhilin (2015), we consider the user ratings as a function of three dimensions:

$$CR : User \times Item \times Context \longrightarrow \text{Contextual Ratings}$$

Thus, user ratings can be stored in a multidimensional user-rating-context tensor for each item domain (e.g. books, movies, music, among others). Notice that the notion of domain adopted in this thesis is based on the "Item level" definition (described in Section 2.2.1) by considering movies and books belonging to different domains, for example.

In order to formalize our cross-domain context-aware recommendation problem, we introduce the following definitions, considering a set of 'n' source domains $(S_1, S_2, ..., S_n)$ and just one target domain (T).

Definition 1.

$U_{S_1}, U_{S_2}, ..., U_{S_n}, U_T$: *sets of users for each domain;*

$I_{S_1}, I_{S_2}, ..., I_{S_n}, I_T$: *sets of items for each domain;*

$C_{S_1}, C_{S_2}, C_{S_n}, C_T$: *sets of contextual features for each domain;*

$CR_{S_i} : U_{S_i} \times I_{S_i} \times C_{S_i}$ *(where i=1,2,...,n) and* $CR_T : U_T \times I_T \times C_T$: *contextual user-rating tensors (i.e., multidimensional matrices or cubes) for each domain;*

$U_{S,T} = (U_{S_1} \cup U_{S_2} \cup ... \cup U_{S_n}) \cap U_T \neq \oslash$: *at least one user must have preferences for items in the target domain and, at least, a source domain (user overlap);*

$I_{S,T} = I_{S_1} \cap I_{S_2} \cap ... \cap I_{S_n} \cap I_T = \oslash$: *there is no item overlap between domains;*

$C_{S,T} = C_{S_1} \cap C_{S_2} \cap ... \cap C_{S_n} \cap C_T = C_{S_1} \cup C_{S_2} \cup ... \cup C_{S_n} \cup C_T \neq \oslash$: *the same set of possible contexts is observed for user ratings in all domains (contextual overlap).*

Hence, the problem to be solved in this thesis is to estimate unknown ratings for items in a target domain (I_T) by exploiting the user-rating tensors from the source and target domains (CR_{S_i} where $i = 1, 2, ..., n$ and CR_T), assuming $U_{S,T}$, $I_{S,T}$ and $C_{S,T}$.

It is important to mention that the ratings from the contextual user-rating tensors can have different scales or forms in distinct domains. For example, ratings of music could be represented as a binary form such as “Like” or “Dislike” while the ratings of movies and books could be represented, respectively, by five-star or ten-star scales. Therefore, the recommendation algorithms have to deal with this issue. For instance, an algorithm could normalize the different scales from ratings among distinct domains (SANTOS et al., 2012).

3.2 Modelling Contextual Information

In this section, we describe how the contextual features are formalized (Section 3.2.1) as well as the contextual information is obtained and selected considering its relevance (Section 3.2.2).

3.2.1 Contextual Features Formalization

As mentioned in Section 2.3.2, contextual information can be of different “types”, each one defining a certain contextual dimension, such as time (e.g. “day of week”, “period of the day”, etc.), location (e.g. “at home”, “at work”, etc.), companion (e.g. “alone”, “with friends”, etc.), among others. Furthermore, each contextual dimension can have a hierarchical structure that can be represented as different attributes (e.g. Time: Date →

DayOfWeek → TimeOfWeek, or Date → Month → Quarter → Year) (ADOMAVICIUS; TUZHILIN, 2015).

According to the CD-CARS problem formalization described before, we modelled a set of contextual features (illustrated in Figure 11), for each domain ($C_{S_1}, C_{S_2}, ..., C_{S_n}, C_T$), as a Cartesian product of k contextual dimensions: $C_d = D_1 \times D_2 \times ... \times D_k$ (where d=$S_1, S_2, ..., S_n, T$ domains) (ADOMAVICIUS et al., 2005). Each dimension D_j (j = 1,2,...,k) can be represented by l contextual attributes ($A_1, A_2, ..., A_l$). Each attribute A_z (z = 1,2,...,l) has a set of m values (v_1,v_2,...,v_m) representing a part of the contextual information. Moreover, "Unknown", which represents a missing (or not observable) part of the contextual information, is a *default* value (v_1) for any contextual attribute.

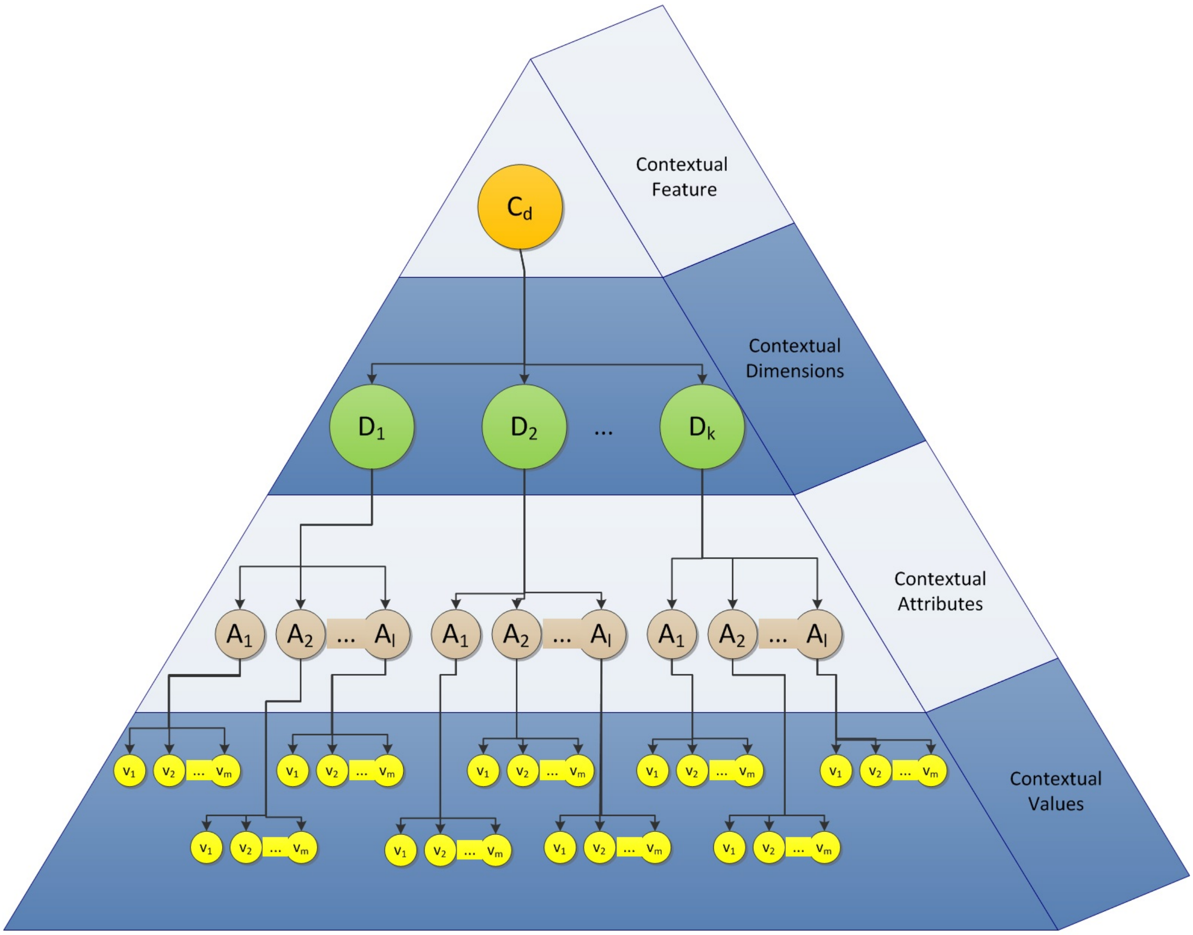


Figure 11 – A contextual feature represented by dimensions, attributes and values.

Thus, the contextual information can be represented as a tuple of w values from different contextual attributes and/or dimensions, i.e., a possible context (c') of a set of contextual features can be denoted as $c' = (v_1, v_2, ..., v_w)$, where each value v_s (s=1,2,...,w) belongs to a different contextual attribute A_z (z = 1,2,...,l) and/or dimension D_j (j = 1,2,...,k). Note that the order of these values in the tuple does not change the meaning of the represented context (c').

For instance, consider three contextual dimensions ($k = 3$): $D_1 = Temporal, D_2 = Location, D_3 = Companion$. Each one can have different hierarchical representation through contextual attributes. Suppose that D_1 has two ($l = 2$) attributes ($A_1 = Day, A_2 = DayType$), D_2 has three ($l = 3$) attributes ($A_1 = City, A_2 = State, A_3 = Country$) and D_3 has one ($l = 1$) attribute ($A_1 = CompanionType$). For each contextual attribute of those dimensions, there is a set of possible values such as:

- Temporal dimension (D_1): $A_1 = \{v_1 = Unknown, v_2 = Sunday, v_3 = Monday, v_4 = Tuesday, v_5 = Wednesday, v_6 = Thursday, v_7 = Friday, v_8 = Saturday\}$ with eight possible values ($m = 8$), $A_2 = \{v_1 = Unknown, v_2 = Weekday, v_3 = Weekend\}$ with three possible values ($m = 3$);

- Location dimension (D_2): $A_1 = \{v_1 = Unknown, v_2 = Aberdeen, ..., v_{2839} = Zurich\}$ with 2839 possible values ($m = 2839$), $A_2 = \{v_1 = Unknown, v_2 = Alabama, ..., v_{381} = Wisconsin\}$ with 381 possible values ($m = 381$), $A_3 = \{v_1 = Unknown, v_2 = Australia, ...v_{113} = Zambia\}$ with 113 possible values ($m = 113$);

- Companion dimension (D_3): $A_1 = \{v_1 = Unknown, v_2 = Alone, v_3 = Accompanied, v_4 = Family, v_5 = Friends, v_6 = Partner, v_7 = Fellows\}$ with seven possible values ($m = 7$),

Given this example, a set of contextual features could be the combination of all possible values from different attributes (six) and dimensions (three). So, by multiplying the 'm' values of each attribute, in this case, it will be approximately twenty billion different contexts resulted from the Cartesian product. Notice that this contextual feature modelling does not guarantee the consistency of the information among different attributes. For example, a context c' =$\{$*Sunday, Weekday,Recife,Alagoas,EUA, Unknown*$\}$ would be *valid* according to our modelling, but *inconsistent* considering the real contextual information (e.g., a *consistent* context could be c' =$\{$*Sunday, Weekend, Recife, Pernambuco, Brazil, Unknown*$\}$).

In this way, we let the RS application that uses this modelling responsible for obtaining consistent contextual information. In fact, despite the huge amount of possible contexts in the example given above, a real dataset obtained by the RS application, in that example, would have approximately one hundred and sixty thousand possible contexts, which is the result of the multiplication of the 'm' values of the more discrete contextual attributes from different dimensions: Day (A_1 from D_1) with $m = 8$, City (A_1 from D_2) with $m = 2839$, and CompanionType (A_1 from D_3) with $m = 7$, instead of the twenty billion ones.

It is important to mention that the proposed contextual feature modelling is based on the "Key-Value" model (referred in Section 2.3.2). In this case, the matching between the context of the recommendation, which is called "contextual criteria", and the contextual information, represented by this model in the user-ratings, is made in a linear way. In other words, once the tuple of w contextual values is established, from different contextual attributes and/or dimensions, then a contextual criteria can be used as a query term (i.e., context of the recommendation), as described in Algorithm 1. A contextual criteria can also be represented as a tuple of w contextual values from the same contextual attributes and/or dimensions as the contextual information from ratings.

Despite the "Unknown" value be always a possible one for each contextual attribute, it has distinct meanings in the contextual criteria and the contextual information from ratings. For the contextual criteria, "Unknown" (v_1) can be viewed as a part of the context to be ignored (i.e., uninformed). In this case, this value means that any value of the contextual information from ratings is acceptable for that contextual attribute and dimension, including the "Unknown" one, which for the contextual information from ratings represents a missing (or not observable) part of the contextual information, as mentioned before. Therefore, the algorithm described above considers, for contextual matching purposes, only the values different from "Unknown" on the contextual criteria. This mechanism is sufficient for the proposed CD-CARS algorithms.

Algorithm 1. *Matching between the context of the recommendation and the contextual information from ratings.*

Input: C, I, n (where C is the contextual criteria array of contextual values, and I is the contextual information array of contextual values, considering both arrays with the same size n*).*
Output: isMatched (a boolean value determining if there is a match between the context of the recommendation and the contextual information).

```
procedure contextualMatching(C, I, n)
    for v=1 to n do
        if $C[v] \neq$ "Unknown" and $C[v] \neq I[v]$ then return false
        end if
    end for
return true
end procedure
```

end

3.2.2 Obtaining and Selecting Relevant Contextual Information

In this thesis, we are not concerned about how to obtain contextual information. We let the RS application responsible for gathering this information and persisting it according to the proposed contextual model. As mentioned in Section 2.3.3, three methods are more often used to acquire contextual information: explicit, implicit, and inferred. So, depending on the data available in the datasets used by the RS application, some of these methods can be more useful than others. Although the source of contextual information is irrelevant for the proposed CD-CARS algorithms, the quality of the obtained contextual information remains relevant for them.

On the other hand, after obtaining the contextual information, if there are many contextual attributes available for a contextual dimension in the contextual model, then selecting relevant contextual information is important for the quality and performance of the CD-CARS. Taking the example previously given, if the temporal dimension (D_1) has two attributes: *Day* ($A_1 = \{v_1 = Unknown, v_2 = Sunday, v_3 = Monday, v_4 = Tuesday, v_5 = Wednesday, v_6 = Thursday, v_7 = Friday, v_8 = Saturday\}$), and *DayType* $A_2 = \{v_1 = Unknown, v_2 = Weekday, v_3 = Weekend\}$, then CD-CARS could select just one of these attributes for recommendation. The same may occur among two distinct contextual dimensions, for example, also considering the location dimension (D_2), CD-CARS could select just one of these dimensions for the recommendation.

Some types of contextual information (e.g. temporal, location, companion, etc.) can be more relevant in a given domain (e.g. books, movies, music, etc.) than others. As mentioned in Section 2.3.4, the selection of a relevant contextual dimension or attribute can be made with a feature selection method from data mining (LIU; MOTODA, 1998). In the proposed CD-CARS, for each different target domain where the recommendation takes place, we can apply the information gain measure[1] considering the user-rating as a class and each contextual attribute as a tested attribute of the information gain measure. For instance, the user-rating class could have five possible nominal values representing ratings of a five-star scale. Note that in that example we assume that all ratings from the source and target domains are in the same scale and form. However, as mentioned before, for different forms or scales of ratings among distinct domains an algorithm must normalize them(SANTOS et al., 2012). Besides, the information gain calculated for each contextual attribute may vary depending on the target domain in which the data mining method is applied.

So, from the list of most relevant attributes generated by the information gain measure, the CD-CARS could select only the contextual attribute with higher information gain value. Then, it could execute performance experiments in the selected attribute and, progressively, select the next relevant attribute of a different contextual dimension if the performance difference is significative between the previously selected attribute and the next one. In the case of selecting contextual attributes in the same dimension, however,

[1] InfoGain(Class,Attribute) = H(Class) - H(Class | Attribute), where H means “entropy”, defined in Information Theory.

the CD-CARS could select only the top attribute with higher information gain value, since the subsequent attributes are nothing more than different representations of the top attribute selected. For instance, if the top contextual attribute was *Day*, followed by the *DayType*, then selecting only the *Day* attribute is sufficient, as it represents all values of *DayType* attribute in a more discrete way.

3.3 CD-CARS Algorithms

The algorithms proposed in our work rely on the use of a base cross-domain recommender system, in which the predicted rating ($\hat{R}(u,i)$) for a particular pair of user u and item i, belonging to the target domain item set (I_T), can be formalized as:

$$\hat{R}(u,i) = CD(u,i,R_{S_1},R_{S_2},...,R_{S_n},R_T), i \in I_T \tag{3.1}$$

In which, R_{S_i} (i=1,2,...,n domains) and R_T are 2-dimensional user-rating matrices, respectively in the source and target domains. Notice that the base cross-domain RS does not take into account the contextual information. In addition, it is possible that different scales and forms of ratings from distinct domains have to be handled by the base cross-domain RS algorithm. As mentioned before, an algorithm could normalize the ratings among distinct domains (SANTOS et al., 2012). In this way, we assume that all ratings from the source and target domains are in the same scale and form.

In our CD-CARS problem, we consider contextual user-rating tensors and we need a function ($\mathcal{F}$) to make rating predictions of items (i) for users (u) in contexts (c) given the tensors from source (CR_{S_i}, where i=1,2,...,n domains) and target domains (CR_T), as defined in Equation 3.2.

$$\hat{CR}(u,i,c) = \mathcal{F}(u,i,c,CR_{S_1},CR_{S_2},...,CR_{S_n},CR_T), i \in I_T \tag{3.2}$$

The function ($\mathcal{F}$) can be implemented using any of the three proposed CD-CARS algorithms described in the following section.

3.3.1 Proposed Algorithms

We designed the algorithms according to three different context-aware RS paradigms (ADOMAVICIUS; TUZHILIN, 2015): *Pre-filtering* (PreF), *Post-filtering* (PostF) and *Modelling*. These paradigms are usually adopted in single-domain RS, but we extended their directives for the cross-domain recommendation task by taking into account the contextual user-rating tensors from different domains.

3.3.1.1 Cross-Domain PreF Algorithm

PreF algorithm initially uses contextual information to filter the contextual user-rating tensor from the target domain (CR_T) in order to obtain a two-dimension (2D) user-rating matrix. On the other hand, the contextual user-rating tensors from the source domains ($CR_{S_1}, CR_{S_2}, ..., CR_{S_n}$) are collapsed into a two-dimension (2D) user-rating matrix, by aggregating ratings for the same user-item pair in different contexts, prioritizing the user-ratings of the context of the recommendation (c). Then, the base cross-domain algorithm is applied to these matrices to produce the predicted ratings ($C\hat{R}(u, i, c)$). Figure 12 illustrates the PreF technique, which is formalized in three steps as follows.

- Step (1) Define the 2D reduced matrix (context-filtered matrix) for the target domain:

$$R_T^c(u, i) = CR_T(u, i, c) \tag{3.3}$$

The context-filtered matrix only has ratings according to:

$$\hat{R_T}(u, i) = \begin{cases} C\hat{R}_T(u, i, c), & \text{if } c = o \\ \textit{not available}, & \text{otherwise} \end{cases} \tag{3.4}$$

where 'o' represents the rating's context.

- Step (2) Define the 2D aggregated matrices (prioritizing the user-ratings with the context of the recommendation) for the source domains:

$$R_j(u, i) = CR_j(u, i, c) \tag{3.5}$$

Where $j = S_1, S_2, ..., S_n$ source domains. For each source domain 'j', the aggregated ratings are calculated as:

$$\hat{R}_j(u, i) = \begin{cases} C\hat{R}_j(u, i, c), & \text{if } c = o \\ \frac{\sum_{c \in C_j} CR_j(u,i,c)}{|C_j|}, & \text{otherwise} \end{cases} \tag{3.6}$$

where 'o' represents the rating's context.

- Step (3): Apply the base cross-domain technique using the reduced matrices:

$$C\hat{R}(u, i, c) = CD(u, i, R_{S_1}, R_{S_2}, ..., R_{S_n}, R_T^c), i \in I_T \tag{3.7}$$

In the steps above, the matching between the user-rating context (c) and the context of the recommendation (o) is made according to Algorithm 1 (page 72). Besides, we assume that the user-ratings from the source and target domains are in the same scale and form, as mentioned before.

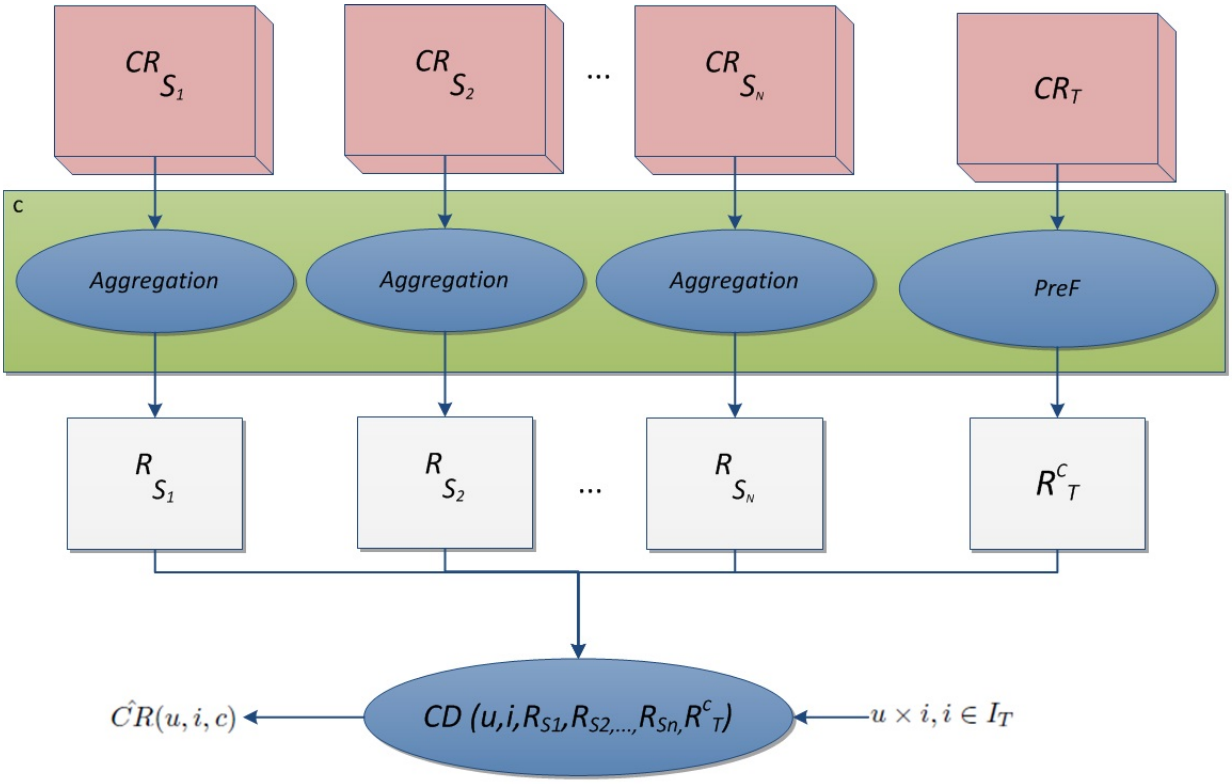


Figure 12 – The pre-filtering cross-domain recommendation is made by filtering the target contextual user-rating tensor for a given context.

Since PreF algorithm filters the contextual user-rating tensor from the target domain, it could have a few user-ratings to make recommendations in very specific contexts, thus, the recommendation process would be made using almost entirely only the user-ratings from the source domains. Due to this drawback, we also proposed other context-awareness techniques as described in the next sections.

3.3.1.2 Cross-Domain PostF Algorithm

PostF algorithm initially produces a single unified user-rating matrix by aggregating ratings for the same user-item pair in different contexts, prioritizing the user-ratings of the context of the recommendation (c). The base cross-domain is then applied using as input the aggregated rating matrices. Finally, contextual information is used to filter the ratings produced by the cross-domain algorithm. This filtering is done by considering items contained in the set of preferred item categories (e.g. comedy, action, rock, etc.) by the user in a given context (e.g. considering only comedy movies on weekdays). Figure 13 illustrates this algorithm, which is formalized in the following steps:

- Step (1) Define the 2D aggregated matrices (prioritizing the user-ratings with the context of the recommendation) for the source and target domains:

$$R_j(u,i) = CR_j(u,i,c) \tag{3.8}$$

Where $j = S_1, S_2, ..., S_n, T$ domains. For each domain 'j', the aggregated ratings are calculated as:

$$\hat{R}_j(u,i) = \begin{cases} C\hat{R}_j(u,i,c), \text{ if } c = o \\ \frac{\sum_{c \in C_j} C\hat{R}_j(u,i,c)}{|C_j|}, \text{ otherwise} \end{cases} \tag{3.9}$$

where 'o' represents the rating's context.

- Step (2): Apply the base cross-domain technique using the matrices from Step (1) and collect the predicted ratings:

$$\hat{R}(u,i) = CD(u,i,R_{S_1},R_{S_2},...,R_{S_n},R_T), i \in I_T \tag{3.10}$$

- Step (3): Given a context of recommendation (o) and user u as input, a rating produced for an item ($\hat{R}(u,i)$) is discarded if the number of "good" rated items of a given item category (g_i) is less than a threshold value (θ) in that context. Otherwise, the rating predicted by the cross-domain algorithm is maintained:

$$C\hat{R}(u,i,c) = \begin{cases} \hat{R}(u,i), \text{ if } c = o \text{ and } CP(u,c,g_i) >= \theta \\ \textit{not available}, \text{ otherwise} \end{cases} \tag{3.11}$$

where the category preferences tensor ($CP(u,c,g)$) contains the number of "good" rated items for each item category g, from different domains, observed in a context c for a user u. The definition of a "good" rated item can be made according to the scale and form of the user-ratings from distinct domains. As mentioned before, we considered that all user-ratings are normalized among the different domains. In this way, a "good" rated item could have a rating of, at least, "four" in a five-star scale, for example. We let the responsibility of this definition for the CD-CARS implementation as well as the optimal θ value, which could be calculated considering the number of "good" rated items in general. For instance, if a user has fifty "good" rated items in general (regardless their categories), then the θ value could be set to 10% of that number (i.e., $\theta = 5$), which would mean that only categories with at least five "good" rated items would be considered.

An alternative way to define Equation (3.11) is:

$$C\hat{R}(u,i,c) = \begin{cases} \hat{R}(u,i) \times (1+\omega), \text{ if } c = o \text{ and } CP(u,c,g_i) >= \theta \\ \hat{R}(u,i) \times (1-\omega), \text{ otherwise} \end{cases} \tag{3.12}$$

where ω is a factor to increase or decrease the predicted rating value ($\hat{R}(u,i)$). The θ value still has the same meaning of Equation (3.11) by defining a threshold for determining which categories of items are relevant or not according to the minimal number of “good” rated items. Thus, relevant categories will have the predicted rating value increased whereas irrelevant categories will have the predicted rating value decreased. In this case, the ω could be an empirically defined value (e.g. “0.1” value would increase or decrease by 10% the predicted rating value). Also, it could be calculated, proportionally, according to the relevance of the item category preferred by the user in a given context (e.g. the higher is the number of “good” rated items the higher is the ω value). Therefore, the PostF algorithm could adjust the predicted rating instead of filtering it. Likewise the θ value, the ω value definition is responsibility of the CD-CARS implementation.

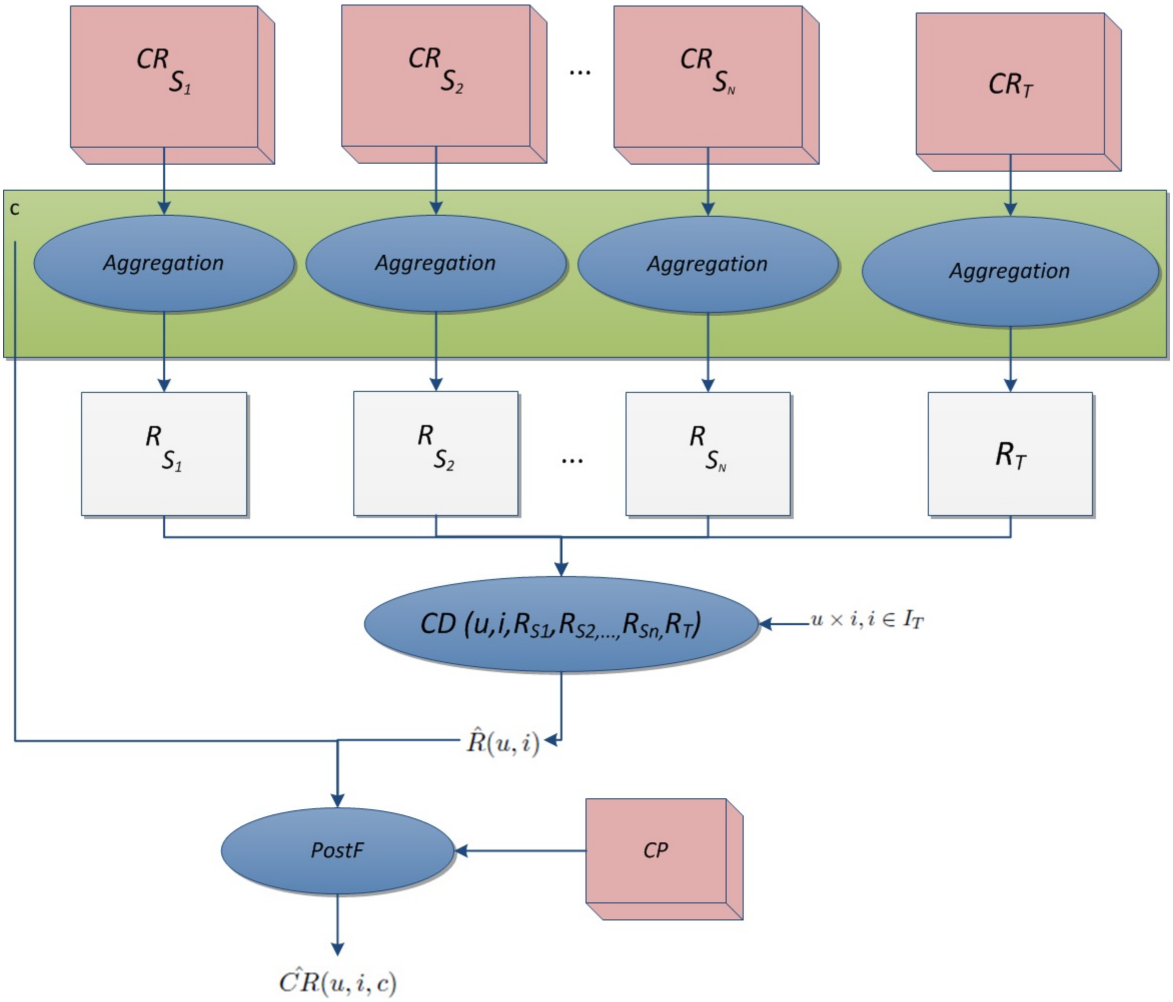


Figure 13 – The cross-domain post-filtering recommendation is made over the aggregated user-rating matrices and then post-filtered according to contextual user preferences.

It is important to mention that g could also be expressed as a set of attributes which characterize an item (e.g. user tags), instead of being expressed as item categories like item genres (e.g. comedy, action, rock, etc.), without losing generality.

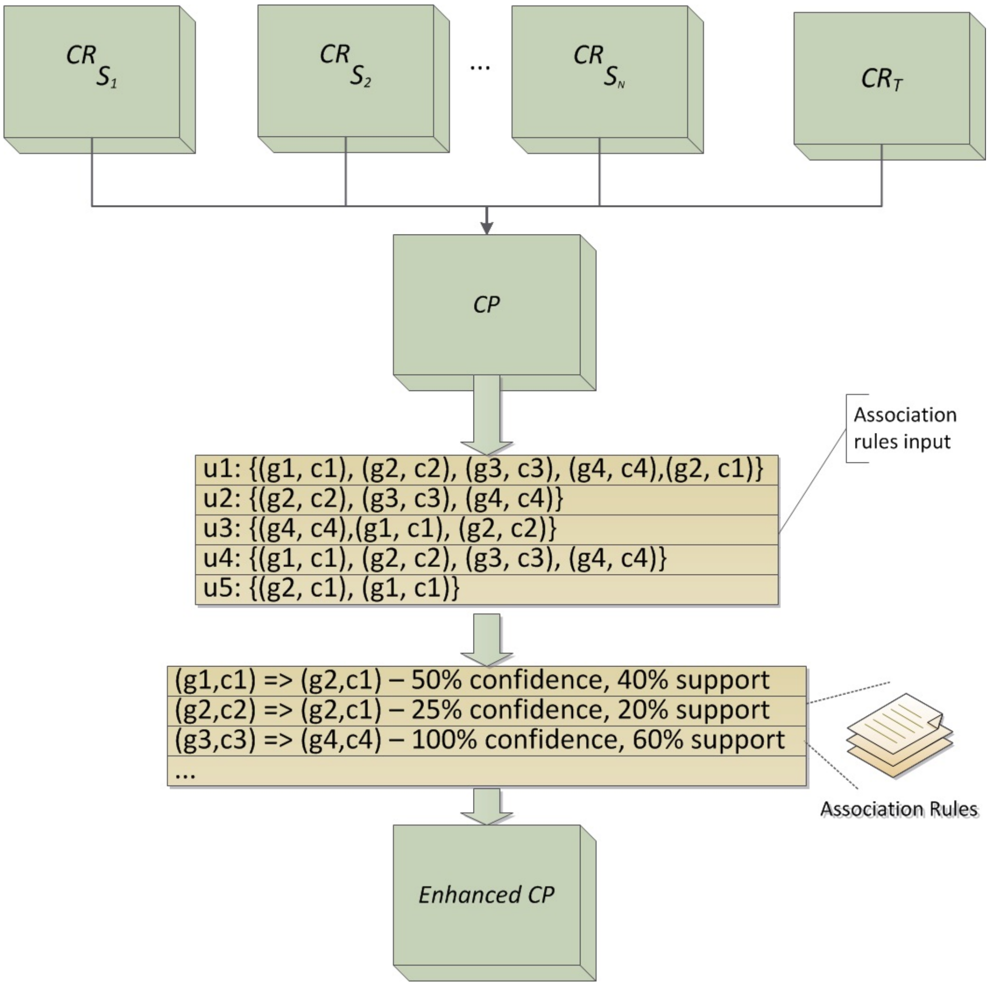
Similar to PreF algorithm, in the PostF algorithm, we apply a base cross-domain algorithm. However, none of the input tensors (from source and target domains) are filtered by context. Instead, all contextual user-rating tensors are reduced to matrices (by aggregating user-ratings from different contexts) that serve as input for the base cross-domain algorithm. After applying the base algorithm, only the post-filtered ratings are taken into account. In this process, an important task is to build the category preferences tensor ($CP(u, c, g)$).

We build the category preferences tensor from the contextual user-rating tensors of the source and target domains. Depending on the θ value, it is possible that a user only has category preferences in source domains. The same situation could happen in a scenario where a dataset contains just a few users with overlap. In these cases, the PostF algorithm would not be able to recommend items in the target domain.

In order to alleviate this problem, some techniques can be used. For example, using association rule mining (HIPP; GÜNTZER; NAKHAEIZADEH, 2000) to discover usage patterns between different domains and contexts (e.g. we could infer that users who like to read romance books on weekdays also like to watch romance movies on weekdays). Thus, we propose the enhancement of the category preferences tensor by using association rules to infer other item categories preferred by the users according to the possible contexts. Figure 14 illustrates this idea, which is only required when a user receives a recommendation in the target domain and the category preferences tensor does not have information about his/her rated item categories in that domain.

As it can be seen from Figure 14, the association rules input is generated from the category preferences tensor ($CP(u, c, g)$). Each entry of this input is extracted from a user (u) and represents a set of pairs, composed of an item category (g) and context (c) in which that user has the number of "good" rated items greater or equal than the *theta* value. With that input, we can use an algorithm for generating association rules such as, for example, the Apriori (AGRAWAL; IMIELIŃSKI; SWAMI, 1993).

After applying that algorithm and obtaining the resulting association rules, we can select only the most relevant ones according to their confidence and support levels. Optimal values for these parameters can vary depending on the dataset used in the CD-CARS application. In addition, we are interested only in "cross-domain" rules, i.e., rules that relate item categories between a source domain and the target domain. We discard rules that relate item categories between two source domains (or for the same domain), since they do not make the PostF recommendation possible when a user only has item category preferences in source domains. Finally, these rules are used for enhancing the category preferences tensor, which will have inferred item categories and contexts beyond the original preferences.



u1: {(g1, c1), (g2, c2), (g3, c3), (g4, c4),(g2, c1)}
u2: {(g2, c2), (g3, c3), (g4, c4)}
u3: {(g4, c4),(g1, c1), (g2, c2)}
u4: {(g1, c1), (g2, c2), (g3, c3), (g4, c4)}
u5: {(g2, c1), (g1, c1)}

(g1,c1) => (g2,c1) – 50% confidence, 40% support
(g2,c2) => (g2,c1) – 25% confidence, 20% support
(g3,c3) => (g4,c4) – 100% confidence, 60% support
...

Figure 14 – Category preferences tensor enhancement from association rules.

Note that source and target domains have different sets of categories (e.g. music x books). However, by using association rules in the category preferences tensor enhancement, we can make cross-domain PostF recommendations even to less related domains such as music and movies, making the CD-CARS domain-independent. We remember that we consider this relation among distinct domains according to the set of item genres of them. As more the domains have item genres in common the more related they are considered (e.g. *Book* and *Television* have several item genres in common such as “romance”, “educational”, “religion”, etc.).

3.3.1.3 Cross-Domain Modelling Algorithm

Unlike the PreF and PostF algorithms, the *Modelling* algorithm does not need to use a base cross-domain algorithm. In fact, it makes “multidimensional” recommendations by considering contextual information beyond users and items without reducing the contextual user-rating tensors.

In this thesis, we propose the extension of two single-domain context-aware *Modelling* approaches: heuristic calculations (ADOMAVICIUS et al., 2005) and matrix factorization (BALTRUNAS; LUDWIG; RICCI, 2011) (as mentioned in Section 2.3.5). This extension goes beyond the inclusion of contextual information in a single-domain CF-based algorithm since we intend to perform cross-domain context-aware recommendations. Thus, we consider four dimensions (user, item, context, and domain) instead of three (user, item, and context) for the cross-domain context-aware recommendation.

The heuristic calculation approach described in (ADOMAVICIUS et al., 2005) includes contextual information by using an n-dimensional distance metric instead of the user-user or item-item similarity metrics traditionally used in such techniques (RICCI; ROKACH; SHAPIRA, 2011) whereas the matrix factorization approach described in (BALTRUNAS; LUDWIG; RICCI, 2011) could be generalized to consider additional dimensions (e.g. item domain) for the representation of the data as a tensor of four dimensions (user, item, context, and domain).

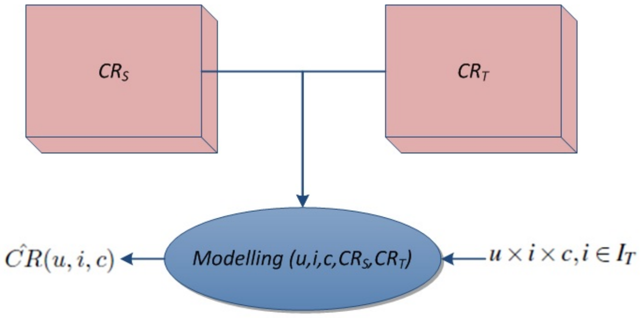

CR_S
CR_T
$\hat{CR}(u,i,c)$
Modelling (u,i,c,CR_S,CR_T)
$u \times i \times c, i \in I_T$

Figure 15 – The cross-domain modelling recommendation uses contextual information directly in the recommendation function as an explicit predictor of a user rating for an item.

The Modelling algorithms proposed in this thesis (illustrated in Figure 15) are formalized in a single step as follows.

- Step (1): Apply the extended version of the base cross-domain algorithm using the user-rating-context tensors:

$$\hat{CR}(u,i,c) = CD(u,i,c,CR_{S_1},CR_{S_2},...,CR_{S_n},CR_T), i \in I_T \tag{3.13}$$

In this way, for the Modelling algorithm based on heuristic calculations, instead of using a similarity metric only for calculating the user-user and item-item distances, it could also include other dimensions as context and item domain. For example, if the similarity metric is the Euclidian distance, it could be defined as:

$$dist[(u,i,c,d),(u',i',c',d')] = \sqrt{w_1 d_1^2(u,u') + w_2 d_2^2(i,i') + w_3 d_3^2(c,c') + w_4 d_4^2(d,d')} \tag{3.14}$$

where d_1, d_2, d_3, and d_4 are distance functions defined for dimensions User, Item, Context, and Domain, respectively, and w_1, w_2, w_3, and w_4 are the weights assigned for each of these dimensions. For instance, these weights can be set according to the relevance of the four dimensions or empirical values. As mentioned in Section 2.3.4, it is important to select properly the contextual information used in the recommendation dataset.

In addition, depending on the way how contextual information is obtained, it could be more or less relevant. For example, the contextual relevance might be low in a system that a user rates an item without explicitly consider the context in that rating, i.e., the context is dissociated from the user-rating (ADOMAVICIUS; TUZHILIN, 2015). On the other hand, a system that collects the user contextual information together with his/her rating may be more reliable, obtaining a more relevant contextual information (ADOMAVICIUS; TUZHILIN, 2015).

Therefore, the use of the Modelling algorithm based on heuristic calculations will depend on the relevance of the association between context and user-rating. To avoid this issue, we also propose to adapt some algorithms based on matrix factorization (or tensor factorization, to be more specific) that considers the contextual information, such as the proposed ones in (KARATZOGLOU et al., 2010), (HIDASI; TIKK, 2012), and (KIM; YOON, 2014). For these algorithms, the item domain (e.g. book, TV, music, etc.) could be considered as a contextual dimension. This adaptation is simple and maintains the original logic of those algorithms.

Finally, it is important to notice that the Modelling algorithm does not consider the context of the recommendation, as opposed to the PreF and PostF algorithms. On the other hand, it only takes into account the context of the user-ratings. Thus, the Modelling algorithm can recommend items without knowing the context of the user at the moment of the recommendation.

3.3.1.4 Cross-Domain Hybrid Contextual Algorithms

In the previous sections, we described three proposed CD-CARS algorithms. In this section, we discuss how these different algorithms can be combined.

One possibility is to combine the PreF and PostF algorithms, as illustrated in Figure 16. Naturally, the PreF algorithm can be used before the PostF one, since the PreF only filters out the recommendation data before the base cross-domain algorithm is applied, whereas the PostF filters out only the outcome of the base cross-domain algorithm.

Another possibility is to combine the Modelling and PostF algorithms, as illustrated in Figure 17. Again, the PostF algorithm can be used after the first proposed algorithm, which in this case is the Modelling one. Unlike the utilization of the Modelling algorithm alone, in which the context of the recommendation is not considered, in this combination is necessary to take into account the context in the moment of the recommendation once that the PostF algorithm demands this information in order to filter irrelevant items recommended by the Modelling algorithm.

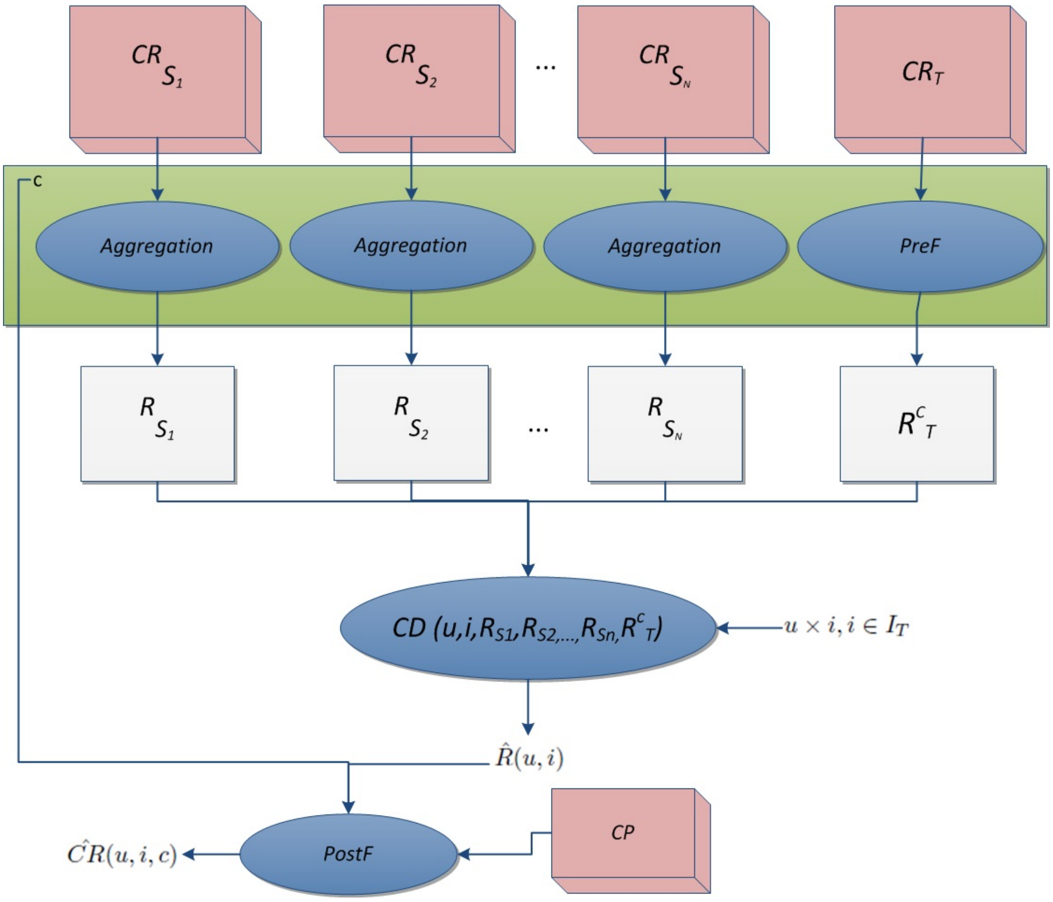


Figure 16 – The cross-domain PreF algorithm can be used before the PostF algorithm in a possible combination.

In the two hybrid algorithms mentioned above (PreF + PostF and Modelling + PostF), the combinations are made in a direct way, without requiring to adapt any proposed algorithm. Other combinations might demand adaptation of the proposed algorithms, so, this could be a future direction of our research.

3.3.2 Base Cross-Domain Algorithms

In this thesis, we propose the adoption of single-domain and cross-domain algorithms. According to the taxonomy presented in Section 2.2.5, the adopted algorithms fit in the *Aggregating knowledge* category and, more specifically, in the *Merging user preferences* approach. Section 3.3.2.1 describes the single-domain CF-based algorithms whereas Section 3.3.2.2 describes the cross-domain CF-based algorithms.

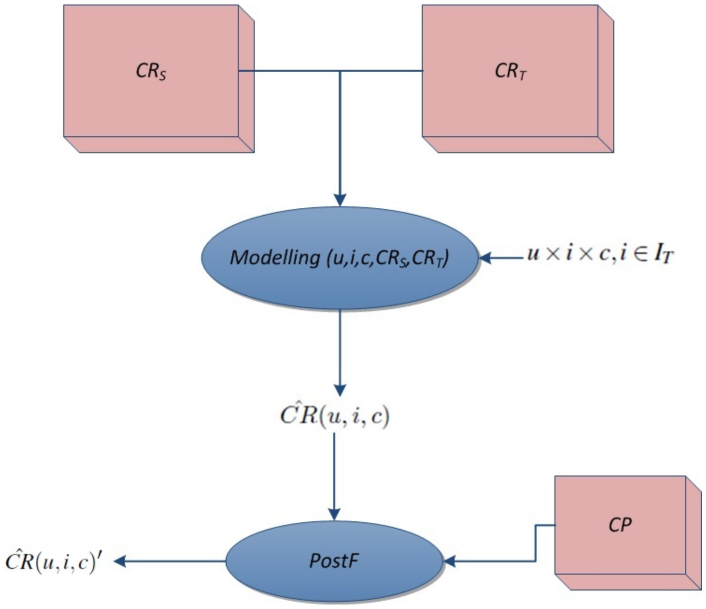

CR_S
CR_T
Modelling (u,i,c,CR_S,CR_T)
$u \times i \times c, i \in I_T$
$\hat{CR}(u,i,c)$
CP
PostF
$\hat{CR}(u,i,c)'$

Figure 17 – The cross-domain modelling algorithm can be used before the PostF algorithm in a possible combination.

3.3.2.1 Single-Domain as Cross-domain Algorithms

As stated by Cremonesi, Tripodi e Turrin (2011), if there is overlap among users and/or items, then standard single-domain CF algorithms can be used for generating cross-domain recommendations by merging user-rating matrices from different domains, considering that these matrices are normalized (see Section 2.2.5). Thus, these algorithms also can be used to make cross-domain recommendations considering our formalization of the CD-CARS problem, since we have an overlap of users among domains (see Section 3.1). Therefore, those CF-based algorithms can be used as a base cross-domain algorithm together with our proposed CD-CARS algorithms.

In this way, we can apply single-domain collaborative filtering algorithms as cross-domain (CD) technique in equations 3.7 and 3.10, for PreF and PostF algorithms, respectively. Note that the Modelling algorithm does not require a base cross-domain algorithm, as mentioned before. The following sections describe some algorithms for two traditional classes of CF-based algorithms that can be applied as a base cross-domain algorithm.

3.3.2.1.1 Neighborhood-based Algorithms

Neighborhood-based algorithms (RICCI; ROKACH; SHAPIRA, 2011) calculate the similarity between two users or items, producing a rating prediction, which is computed by averaging the ratings expressed by similar users or items, weighted with the respective similarity values. We describe some of these algorithms below (RICCI; ROKACH; SHAPIRA, 2011):

- *NNUserNgbr* computes a neighborhood consisting of the nearest 'n' users to a given user. "Nearest" users are defined by a similarity metric. In other words, the recommendations are derived from a neighborhood of the 'n' most similar users. The optimal value of 'n' can be defined through experiments.

- *ThresholdUserNgbr* computes a neighborhood through a similarity threshold and takes any users that are at least that similar to a given user. The threshold should be between -1 and 1. The higher is the threshold value, the more selective is the neighborhood. Again, the optimal threshold value can be estimated from experimentation.

- *GenericItemBasedCF* is simpler than the user-based CF algorithms described above because there is no parameter to be adjusted (like as 'n' or threshold). This item-based CF algorithm compares series of preferences expressed by many users, for one item, rather than by one user for many items (user-based). Some similarity metrics used by user-based CF algorithms also can be used in order to compute the similarity between items.

A crucial aspect of these algorithms is the similarity computation between items or users. Similarity in user-based and item-based CF algorithms can be computed by means of traditional similarity metrics, such as (RICCI; ROKACH; SHAPIRA, 2011):

- Weighted Euclidean distance similarity computes the Euclidean distance (*dist*) between two such user or items points. The equation below denotes a generic calculation of this metric:

  $$dist[(u,i),(u',i')] = \sqrt{w_1 d_1^2(u,u') + w_2 d_2^2(i,i')} \tag{3.15}$$

  where d_1 and d_2 are distance functions defined for two dimensions: User and Item, respectively, and w_1 and w_2 are the weights assigned for each of these dimensions. This similarity metric never returns a negative value, and the more similar two users are (i.e., the larger the similarity value between them is), the smaller is the distance between them. In addition, if we only need to calculate the distance between users, then we can consider $i = i'$. On the other hand, if we only need to calculate the distance between items, then we can consider $u = u'$.

- Cosine similarity (*cos*) is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. In the context of item recommendation, this measure can be employed to compute user similarities by considering a user u as a vector $\mathbf{x}_u \in \mathbb{R}^{|I|}$, where $\mathbf{x}_{ui} = r_{ui}$ if user u has rated item i, and 0 otherwise. The similarity between two users u and v is then computed as

$$cos(u,v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2 \sum_{j \in I_v} r_{vj}^2}} \tag{3.16}$$

where I_{uv} denotes the items rated by both u and v users. The same idea can be used to obtain similarities between two items i and j, according to the equation below:

$$cos(i,j) = \frac{\sum_{u \in U_{ij}} r_{ui} r_{uj}}{\sqrt{\sum_{u \in U_i} r_{ui}^2 \sum_{u \in U_j} r_{uj}^2}} \tag{3.17}$$

Cosine similarity is particularly used in positive space, where the outcome is bounded in a [0,1] interval.

- Pearson correlation (PC) similarity is a ratio of the covariance of two data sets to their standard deviations. Unlike the Cosine similarity, this metric considers the effects of mean and variance of the ratings made by users u and v. The equation below denotes the calculation of this metric:

$$PC(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r_u})(r_{vi} - \bar{r_v})}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r_u})^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r_v})^2}} \tag{3.18}$$

The same idea can be used to obtain similarities between two items i and j, according to the equation below:

$$PC(i,j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \bar{r_i})(r_{uj} - \bar{r_j})}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \bar{r_i})^2 \sum_{u \in U_{ij}} (r_{uj} - \bar{r_j})^2}} \tag{3.19}$$

While the sign of the outcome of this metric indicates whether the correlation is direct or inverse, its magnitude (ranging from 0 to 1) represents the strength of the correlation.

More sophisticated similarity metrics can also be used such as the proposed ones in (DIDAY; BOCK, 2000), (BEZERRA; CARVALHO, 2004), (BEZERRA; CARVALHO, 2011), among others.

The recommendation of these single-domain neighborhood-based algorithms, used for cross-domain purposes, is made as usual (CREMONESI; TRIPODI; TURRIN, 2011), except by the fact that only items from the target domain are recommended.

3.3.2.1.2 Matrix factorization algorithms

Matrix factorization algorithms (RICCI; ROKACH; SHAPIRA, 2011) map users and items to a latent feature space, commonly with reduced dimensionality (f). User-item interactions are modeled as inner products in that space. The latent space tries to explain ratings by characterizing both items and users on factors automatically inferred from user feedback. The major challenge of these algorithms is computing the mapping of each item and user to factor vectors. After the recommender system completes this mapping, it can easily estimate the rating a user will give to any item.

Unlike neighborhood-based algorithms, matrix factorization algorithms do not use similarity metrics, but techniques for identifying latent semantic factors like singular value decomposition (SVD) (RICCI; ROKACH; SHAPIRA, 2011). For the SVD, consider that each item i is associated with a vector $q_i \in \mathbb{R}^f$ and each user u is associated with a vector $p_u \in \mathbb{R}^f$. For a given item i, the elements of q_i measure the extent to which the item possesses those factors, positive or negative. For a given user u, the elements of p_u measure the extent of interest the user has in items that are high on the corresponding factors (again, these may be positive or negative).

The resulting dot product[2], $q_i^T p_u$, captures the interaction between user u and item i, i.e., the overall interest of the user in characteristics of the item. The final rating is created by the rule above:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u \tag{3.20}$$

where μ is the overall average rating and the parameters b_u and b_i indicate the observed deviations of user u and item i, respectively, from the average μ.

In order to learn the model parameters (b_u, b_i, p_u and q_i) the regularized squared error can be minimized:

$$\min_{b_*,q_*,p_*} \sum_{(u,i)\in\kappa} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda_4(b_i^2 + b_u^2 + \| q_i \|^2 + \| p_u \|^2) \tag{3.21}$$

The constant λ_4, which controls the extent of regularization, is usually determined by cross-validation. Minimization is typically performed by either stochastic gradient descent or alternating least squares. Alternating least squares techniques rotate between fixing the p_u's to solve for the g_i's and fixing the q_i's to solve for the p_u's. When one of these is taken as a constant, the optimization problem is quadratic and can be optimally solved (BELL; KOREN; VOLINSKY, 2007).

An easy stochastic gradient descent optimization was popularized by Koren (2008). The algorithm loops through all ratings in the training data. For each given rating r_{ui}, a prediction ($\hat{r}_{ui}$) is made, and the associated prediction error $e_{ui} = r_{ui} - \hat{r}_{ui}$ is computed. For a given training case r_{ui}, the parameters are modified to move in the opposite direction of the gradient, producing:

[2] The dot product between two vectors $x, y \in \mathbb{R}^f$ is defined as $x^T y = \sum_{k=1}^{f} x_k \cdot y_k$.

- $b_u \leftarrow b_u + \gamma \cdot (e_{ui} - \lambda_4 \cdot b_u)$
- $b_i \leftarrow b_i + \gamma \cdot (e_{ui} - \lambda_4 \cdot b_i)$
- $q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda_4 \cdot q_i)$
- $p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda_4 \cdot p_u)$

As the single-domain neighborhood-based algorithms, single-domain matrix factorization algorithms can be used as the base of cross-domain recommendations, considering that a unique merged user-rating matrix from different domain contains information about users and items. The recommendation of these single-domain neighborhood-based algorithms, used for cross-domain purposes, is made as usual (CREMONESI; TRIPODI; TURRIN, 2011), except by the fact that only items from the target domain are recommended.

It is important to mention that the single-domain matrix factorization algorithms are different from the tensor factorization described in the Modelling approach. Instead of considering the mapping between users and items, matrix factorization (or tensor factorization, to be more specific) algorithms can be generalized to consider the context for the representation of the data as a tensor (KIM; YOON, 2014).

3.3.2.2 Cross-Domain Algorithm

In the previous section, we described the single-domain CF-based algorithms adopted as a base cross-domain algorithm. One of the CD-CARS's advantages is to allow that the majority of traditional single-domain CF-based algorithms can be used in combination with the proposed CD-CARS algorithms.

In this section, we describe an actual cross-domain algorithm adopted in this thesis. Once that this algorithm is originally intended to perform cross-domain recommendations, we can directly apply it as a base cross-domain algorithm in combination with our proposed CD-CARS algorithms. For that, we adopted a neighborhood-based (CF-based) cross-domain algorithm, due to its simplicity.

The cross-domain neighborhood-based algorithm adopted is proposed by (CREMONESI; TRIPODI; TURRIN, 2011), *NNUserNgbr-transClosure*. It enhances the *NNUserNgbr* algorithm (described in Section 3.3.2.1.1), which is intended for single-domain recommendations, by improving its user-to-user or item-to-item similarities calculations with a "transitive closure" method. This improvement is achieved by discovering indirect relations among elements (i.e., transitive closure discovers all n-steps similarity paths between any pair of users, extending their neighborhood), as illustrated in Figure 18. For instance, if there exist two direct links: user A = user B = 1 (e.g. full similarity by the

Pearson metric) and user B = user C = 1, then the transitive closure allows to set user A = user C = 1.

According to (CREMONESI; TRIPODI; TURRIN, 2011), this transitive closure procedure is described as follows. Given a binary relation $\mathbf{S}$, where s_{ij} is equal to either 1 or 0, the algebraic *transitive closure* of $\mathbf{S}$ is the union of successive powers of the original matrix, i.e.:

$$\mathbf{S}_{trans} = \bigcup_{n \in \mathbb{N}} \mathbf{S}^n \tag{3.22}$$

where $\bigcup$ is the union operator. Matrix $\mathbf{S}$ is represented by the weighted connections among a set of items. However, since this matrix does not represent a binary relation, Equation (3.22) has been adapted as follows.

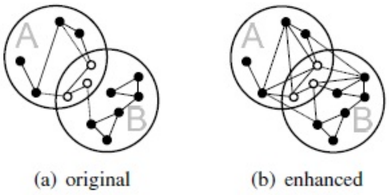


Figure 18 – Original (a) and enhanced (b) item-to-item connections. Solid circles represent items belonging to a single domain, whereas blank circles represent cross items that act as a bridge among different domains (CREMONESI; TRIPODI; TURRIN, 2011).

The "union" operator, which is defined for binary relations, has been replaced by the "maximum" operator, $\mathbf{Z} = max(\mathbf{X}, \mathbf{Y})$ where the maximum matrix $\mathbf{Z}$ between two similarity matrices $\mathbf{X}$ and $\mathbf{Y}$ has been defined so that $z_{ij} = max(x_{ij}, y_{ij})$. The maximum operator adds the similarities discovered for new links while maintaining the original values for existing connections (since original similarities are generally stronger than derived ones).

(CREMONESI; TRIPODI; TURRIN, 2011) has limited the transitive closure to only two steps. Experiments showed that a transitive closure with more than two steps did not provide any sensible improvement in the recommendation accuracy while increasing computational requirements. Thus, the enhanced item-to-item similarity matrix was computed as:

$$\mathbf{S}^* = max(\mathbf{S}, \mathbf{S}^2) \tag{3.23}$$

Except by this user-to-user (or item-to-item) similarity calculation, the remaining logic of the *NNUserNgbr-transClosure* algorithm is the same as the *NNUserNgbr* one.

3.4 Final Remarks

In this chapter, we described the CD-CARS proposal. For that, we formalized the cross-domain context-aware recommendation problem and modeled the contextual information. In addition, we described the proposed CD-CARS algorithms as well as the base cross-domain algorithms adopted in this thesis.

Regarding the CD-CARS problem formalization, we remember that we assumed that all ratings from the source and target domains are in the same scale and form. As mentioned before, an algorithm could normalize the ratings among distinct domains (SANTOS et al., 2012). In the same way, the proposal of the CD-CARS algorithms considers that all ratings from different domains are normalized.

As outlined in Section 3.2, the proposed contextual feature modelling is based on the "Key-Value" model (mentioned in Section 2.3.2), since it is simple and relatively easy to implement and use (VIEIRA; TEDESCO; SALGADO, 2009)(BETTINI et al., 2010). Besides, this kind of contextual model allows a quickly matching between the context of the recommendation and the context represented in the user-ratings, so, that model is sufficient for the proposed CD-CARS algorithms.

One of the advantages of the proposed CD-CARS algorithms is the possibility of using traditional single-domain and cross-domain CF-based algorithms as a base algorithm, which is used in combination with the proposed ones. The adoption of other algorithms of different approaches (e.g. content-based filtering, semantic-based, and so on.) for serving as a base cross-domain algorithm may be studied in future research.

A particular common aspect between the *PreF* and *PostF* algorithms relies on the aggregation of user-ratings from different contexts in order to reduce the contextual user-rating tensors into matrices. Despite the proposal of these algorithms is concerned about a theoretical scenario, in which a user can have multiple ratings for the same item in distinct contexts, this usually does not happen in real datasets, as it is the case of the adopted ones in this thesis. Thus, this important aspect of those algorithms could not be experimented.

Taking into account the *PreF* and *Modelling* algorithms, we can say that they are capable of making domain-independent cross-domain recommendations, i.e., they do not use any information about the item attributes (*unlike the PostF approach*), so, any "kind of item" (domain) can be used in such approaches independently. Indeed, the pre-filtering approach only reduces the dataset to be used by any traditional CD-CFRS, which is capable of making domain-independent cross-domain recommendations (CREMONESI; TRIPODI; TURRIN, 2011)(FERNÁNDEZ-TOBÍAS et al., 2012). Also, a heuristic-based modelling approach only changes the user/item similarity metric used by any traditional CD-CFRS, so, the domain-independent cross-domain recommendation can be made as well.

On the other hand, for making domain-independent cross-domain recommendations with the post-filtering approach, the used dataset must have the description of item categories/genres (or any other attribute that categorizes them). Hence, the post-filtering approach filters out (or adjusts), from the cross-domain recommended item list, the items of a category according to the category preferences of the users. These preferences are obtained according to the users' contexts and can be generated by several approaches, from simple arithmetic calculations to more sophisticated ones as the solution described in this chapter.

In the next chapter, we will present an implementation of the CD-CARS proposal and an experimental evaluation performed on real datasets.

4 CD-CARS Implementation

This chapter describes particular details of an implementation of the CD-CARS proposal. For that, we investigate the problem outlined in Section 3.1 taking into account three contextual dimensions and three distinct domains from real datasets. Section 4.1 describes the properties of CD-CARS datasets with different contextual information and domains and how they were obtained. Also in this section, the process of selecting relevant contextual attributes and values is shown. Section 4.2 presents how the contextual model is implemented through the extension of a traditional framework available in the area of single-domain recommender systems. Section 4.3 describes two of the proposed algorithms (see Section 3.3.1) considering the implemented contextual model. Section 4.4 describes implementation details of two base cross-domain algorithms. Finally, Section 4.5 presents the final remarks of this chapter.

4.1 Dataset Acquisition

One of the main difficulties for evaluating cross-domain recommender systems is the lack of publicly available data, representing the ratings of the same users on items classified in multiple domains (TANG et al., 2012). Although there are several datasets from different domains (television, music, books, etc.) separately, just a few of them have user-overlapped ratings, i.e., all users only have ratings in a single domain. However, our problem demands a cross-domain dataset with contextual information. In other words, this dataset must have at least a small number of users that have some ratings in the source and target domains (i.e. a cross-domain dataset with some level of user overlap) and some of these ratings must contain contextual information.

In order to achieve that, we extracted two datasets based on the dataset from (LESKOVEC; ADAMIC; HUBERMAN, 2007), since it was not designed for evaluating cross-domain context-aware recommendations. This dataset contains product metadata and review information about different Amazon products (Books, music CDs, DVDs, and so on)[1] and we implemented a method to extract only its relevant information for our problem. For instance, we removed duplicated ratings and irrelevant information (e.g. number of votes, Amazon sales rank, product reviews, etc.), besides the creation of methods for gathering contextual information of three contextual dimensions (see Section 4.1.1).

One of the extracted datasets was used for evaluating the CD-CARS algorithms in two more related domains (*Book* and *Television*, named as “book-television dataset”) and another for evaluating it in two less related domains (*Book* and *Music*, named as

[1] https://snap.stanford.edu/data/amazon-meta.html

"book-music dataset"). Both datasets contain a set of ratings, which are composed of:

- User ID: a positive integer value;
- Item ID: a positive integer value;
- Rating value: a positive integer value, defined explicitly by the user on a five-star scale; and
- Contextual information: an array of integer values that represent contextual values. Each index from the array represents a distinct contextual attribute (e.g. country) of a certain contextual dimension (e.g. *Location*), as described in Section 3.2.1.

Unfortunately, the contextual information was not available directly in the original dataset, so, we had to obtain the contextual information implicitly from the ratings' dates, users' Web accounts (from their Amazon IDs), and by inferring the ratings' reviews, as detailed in the Section 4.1.1.

Finally, we discarded users that had less than twenty ratings and did not have ratings in both domains (book/television for one dataset, and book/music for the other one) from datasets, which means that only overlapped users were included (full overlap). From these datasets, we created reduced versions of them in order to evaluate the sensitivity of the CD-CARS algorithms for datasets with different levels of user overlap. Thus, beyond the full-overlapped "book-television" and "book-music" datasets, we have four variations of each one of them: two datasets with 10% of overlap (one for each domain as a target), and two with 50% of overlap (one for each domain as a target). We generated these reduced versions from the full-overlapped datasets by removing all ratings for items from the target domain of the users chosen randomly according to the overlap percentage. For example, the "book-television" dataset with 10% of user overlap and *Television* domain as the target has 10% of the users with ratings in both domains (source and target) and the remaining 90% of them have ratings only for items in the *Book* domain (source). So, this dataset can be used for evaluating the cross-domain recommendation in the *Television* domain as a target when just a few number of users has ratings in the target domain.

Besides the set of ratings, the extracted datasets contain information about items such as item ID, title, domain, and categories. We extracted those categories from the original dataset for all items in all domains (book, television, and music). The categories of the *Book* and *Television* domains were mapped into a single set of 27 categories[2] while

[2] Based on Amazon's Movie&TV categories: unknown, action&adventure, international, animation, anime, boxed sets, classics, comedy, documentary, drama, educational, health, religion, fantasy, LGBT, holiday&seasonal, horror, artistical, kids&family, war, musicals, mystery, romance, sci-fi, special, sports, westerns

the *Music* domain had 19 categories[3].

Also, the extracted datasets contain information about users such as user ID, Amazon user ID, and address (obtained from the Amazon user ID, as detailed in Section 4.1.1.2). At last, they contain the user-rating reviews, which are used for obtaining companion contextual information (as detailed in Section 4.1.1.3).

4.1.1 Obtaining Contextual Information

As mentioned in Section 2.3.3, three methods are more often used to gather contextual information: explicit, implicit, and inferred. According to the contextual information available in the extracted datasets, we considered three contextual dimensions in the CD-CARS implementation. For two of them (*Temporal* and *Location*), we implicitly obtained the contextual information from the user ratings, while for one of them (*Companion*), we inferred the contextual information from the user-rating reviews.

4.1.1.1 Temporal Dimension

The contextual information in the *Temporal* dimension can be directly extracted from user-rating timestamps, which are present in the majority of the datasets containing user-ratings. In this way, the timestamps could be transformed into several contextual attributes with different values and hierarchical levels. For example, timestamps could represent the "period of the day" attribute (Dawn, Morning, Afternoon and Night) as well as "day type" (weekend or weekday), as illustrated in Figure 19. A discussion about the hierarchical levels and possible values of contextual attributes can be found in Section 3.2.2.

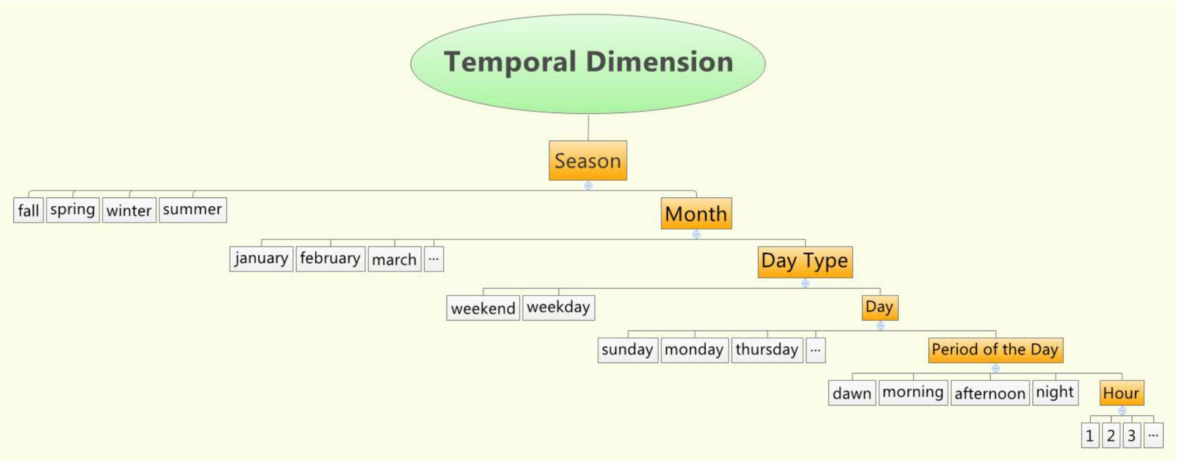


Figure 19 – Example of a temporal dimension with its possible contextual attributes and values in a hierarchical view.

[3] Based on Amazon's CD music categories: unknown, jazz, rock, classic rock, international, classical, pop, blues, gospel, dance, new age, country, folk, vocal, alternative rock, hard rock, kids&family, rap, special

In this implementation, the real datasets used in the experiments only had date information of the user-ratings. For that reason, we could not extract contextual attributes related to the rating time (e.g. rating hour) or “period of the day”. So, only contextual attributes related to day or month could be extracted, such as “day type” or “period of the year”.

It is important to mention that the contextual information extracted from user-rating date is not entirely reliable because a user can rate an item and consume it in distinct moments, which clearly generates an impact in the temporal context. For instance, a user could watch a movie on Saturday and rate it only on Sunday.

This temporal gap could be more frequent for the “period of the day” attribute, since a user could watch a movie in the afternoon and, due to its duration, rate it only at night, when that movie ends. However, despite this risk, we believe that there are rating patterns according to the users' contexts in the same way that there are consumption patterns from these users' contexts. In other words, we can say that there are users that usually rate (instead of watching) comedy movies on Sundays, for example. Thus, we expect that the risk of gathering the temporal context in an implicit way is minimal, which can be observed in the evaluation of the proposed recommender system.

4.1.1.2 Location Dimension

Usually, the contextual information in the *Location* dimension can be implicitly collected when the user is using some device with Internet access or with a Global Positioning System (GPS), for example. However, this information is not available in the real datasets adopted in this implementation. On the other hand, all user-ratings in the datasets have information about the user IDs (and the real Amazon user IDs), as mentioned before.

From the actual Amazon user IDs, we created a web crawler responsible for extracting the address information in the profile web pages from the users' accounts, which can be accessed at a Uniform Resource Locator (URL)[4] containing the Amazon user IDs. The web crawler simply makes a GET request from the Hypertext Transfer Protocol (HTTP) and receives an HyperText Markup Language (HTML) page. This page is parsed by a regular expression based on a particular HTML tag in order to extract a string containing the user's address information. However, this string is defined by the user and is not standardized.

In this way, after obtaining the not-standardized address information from the user profile web page, we used a Representational State Transfer (REST) web service, called

[4] http://www.amazon.com/gp/pdp/profile/AMAZONUSERID/ is an example of URL, where “AMAZONUSERID” represents the real Amazon user ID, omitted here for privacy issues.

Google Maps Geocoding[5], that provides an Application Programming Interface (API) to retrieve the standardized address information of the users. This service requires a string address as parameter and returns a JavaScript Object Notation (JSON) document with multi-level address information about that address, such as “country”, “locality” (representing a “city”), “administrative_area_level_1” (representing a “state”, for example), “administrative_area_level_2” (representing a “county” or “district”, for example), and so on (“administrative_area_level_N”, with ‘N’ greater than two, representing more specific administrative areas). Listing 4.1 shows a response example for the requested string “recife”.

Listing 4.1 – Example of a response (JSON document) from google maps geocoding api for the input “recife”.

```
{
   "results" : [
      {
         "address_components" : [
            {
               "long_name" : "Recife",
               "short_name" : "Recife",
               "types" : [ "locality", "political" ]
            },
            {
               "long_name" : "Recife",
               "short_name" : "Recife",
               "types" : [ "administrative_area_level_2", "political" ]
            },
            {
               "long_name" : "Pernambuco",
               "short_name" : "PE",
               "types" : [ "administrative_area_level_1", "political" ]
            },
            {
               "long_name" : "Brazil",
               "short_name" : "BR",
               "types" : [ "country", "political" ]
            }
         ],
         ...
      }
}
```

The standardized address (composed by country, state, and city) is extracted from the JSON document (through a JSON parser) and persisted into a database, serving as an

[5] A developer key is required to use the web service, as described at https://developers.google.com/maps/documentation/geocoding/intro

address catalog. This mechanism avoids future unnecessary requests to the service. Thus, for each address searched in the web service, we persisted the search string (raw address) and its corresponding standardized address. This information is also persisted together with the users' information. Figure 20 illustrates the process for gathering the location contextual information, described above.

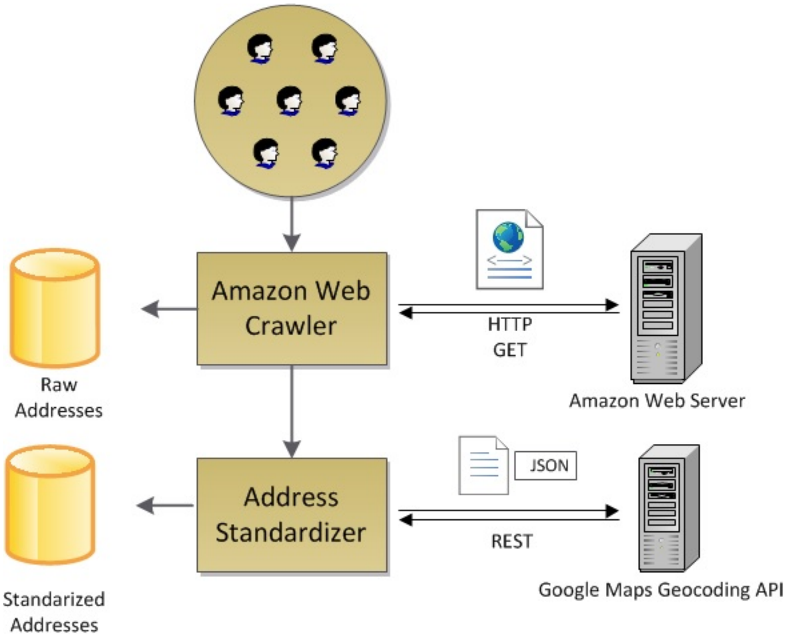


Figure 20 – Process for gathering the location contextual information from the user information.

Since we only obtained static address information (country, state, and city were obtained from users' web profiles), we could not extract contextual attributes related to the abstract locations, as, "place" attribute (at home, at work, in a movie theater, etc.). Therefore, only contextual attributes related to geographical location could be extracted, as illustrated in Figure 21. It is important to mention that each user has the same location context for all his/her ratings once that his/her location context was extracted from his/her address defined in his/her static web profile.

Despite the users' geographical locations are extracted, there is no guarantee that they are actually in those locations where they are rating (or consuming) an item, given that their locations are extracted from their web profiles, which are defined in their initial registration in the system. However, we believe that the majority of the items are rated by the users in their registered location. Finally, it is important to say that not all users had the address information available at their web profile, thus, many users did not have any contextual information about their location.

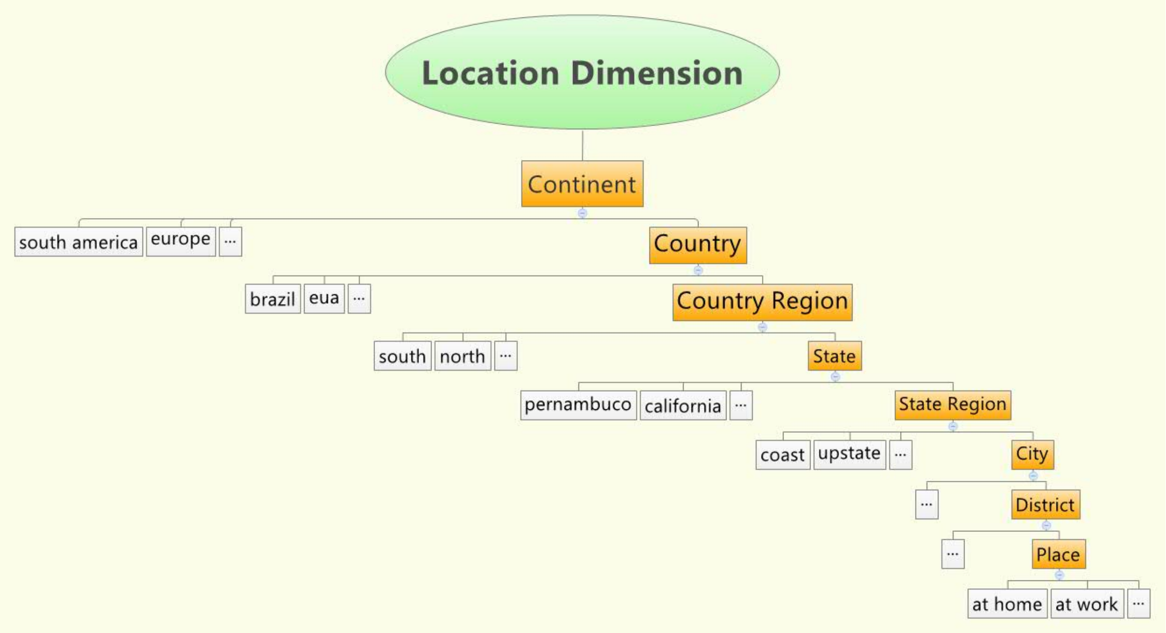


Figure 21 – Example of a location dimension with its possible contextual attributes and values in a hierarchical view.

4.1.1.3 Companion Dimension

In opposite to the previous contextual dimensions, in which we implicitly obtained the contextual information, the contextual information of the *Companion* dimension was inferred from the user-rating reviews available in the real datasets. For that, we implemented a method based on (BAUMAN; TUZHILIN, 2014).

(BAUMAN; TUZHILIN, 2014) proposed an unsupervised text mining algorithm for discovering relevant contextual information from the user-generated reviews. Initially, they observed that contextual information appears more likely in specific reviews (those that describe specific details of an item, such as a book or movie) than in generic reviews (that describes overall comments about an item). After they cluster the user reviews into two groups (specific and generic), they try to find key-words or topics describing the contextual information in those reviews. They state that these topics appear more frequently in the specific reviews than in the generic ones. In this way, they compare the frequencies of key-words (or topics) appearing in the specific and the generic reviews and then select these key-words that have high-frequency ratios, assuming that the selected key-words should contain most of the contextual information among the user reviews. Finally, they inspect the list of the selected key-words by manually identifying the relevant context-related topics.

In this way, we applied that method to our dataset with some adaptations by considering different domains. This method is described in the following. In the first step of the method, we separated reviews into specific and generic reviews by using the measures proposed in the original method (BAUMAN; TUZHILIN, 2014):

- LogSentences: logarithm of the number of sentences in the review plus one[6].
- LogWords: logarithm of the number of words used in the review plus one.
- VBDsum: logarithm of the number of verbs in the past tenses in the review plus one.
- Vsum: logarithm of the number of verbs in the review plus one.
- VRatio - the ratio of VBDsum and Vsum ($\frac{VBDsum}{Vsum}$).

With those measures, we used the classical K-means clustering method (JAIN, 2010) to separate all the reviews into the "specific" and "generic" clusters, as described in (BAUMAN; TUZHILIN, 2014). However, we applied the clustering separately for each domain (*Book*, *Television*, and *Music*). This was the first adaptation in the original method, which is intended for single domain reviews. As a result, the vast majority of the reviews (99.8%) were classified as "specific" for all domains in our dataset. This result might have occurred due to the nature of user-rating reviews available in the original dataset, in which they were analyzed by the dataset provider in order to maintain only relevant user-rating reviews. It is important to mention that the majority of user-ratings (76%) did not have reviews (only did the rating values), considering both datasets.

Given that the great majority of the reviews were classified as "specific", we simplified the word-based and LDA-based (BLEI; NG; JORDAN, 2003) methods proposed in (BAUMAN; TUZHILIN, 2014), since these methods rely on the separation of specific and generic reviews.

The adapted word-based method is explained below:

1. For each review R_i, identify the set of nouns N_i appearing in it.
2. For each noun n_k, determine its weighted frequencies $w^s(n_k)$ corresponding to the specific (s) reviews, as follows

$$w^s(n_k) = \frac{|R_i \,:\, R_i \in specific\, and\, n_k \in N_i|}{|R_i \,:\, R_i \in specific|} \tag{4.1}$$

3. Filter out the words n_k that have overall low frequency, i.e.,

$$w(n_k) = \frac{|R_i \,:\, n_k \in N_i|}{|R_i \,:\, R_i \in specific|} < \alpha, \tag{4.2}$$

where α is a threshold value for the application (e.g., $\alpha = 0.005$)

[6] The authors of the proposed method added one to avoid the problem of having empty reviews when logarithm becomes $-\infty$

4. For each remaining noun n_k left after filtering in the previous step, find the set of senses $synset(n_k)$ using WordNet[7] (MILLER, 1995).

5. Combine senses into groups g_t having close meanings using WordNet taxonomy distance. Words with several distinct meanings can be represented in several distinct groups.

6. For each group g_t determine its weighted frequencies $w^s(g_t)$ through frequencies of its members as:

$$w^s(g_t) = \frac{|R_i \,:\, R_i \in specific\, and\, g_t \cap N_i \neq \varnothing|}{|R_i \,:\, R_i \in specific|} \tag{4.3}$$

7. Sort groups by its weighted frequencies $w^s(g_t)$ in its descending order.

The adapted LDA-based method is described below:

1. Build an LDA model on the set of the specific reviews.

2. Apply this LDA model to all the user-generated reviews in order to obtain the set of topics T_i for each review R_i with a probability higher than a certain threshold level.

3. For each topic t_k from the generated LDA model, determine its weighted frequencies $w^s(t_k)$ corresponding to the specific (s) reviews, as follows

$$w^s(t_k) = \frac{|R_i \,:\, R_i \in specific\, and\, t_k \in T_i|}{|R_i \,:\, R_i \in specific|} \tag{4.4}$$

4. Filter out the topics t_k that have overall low frequency, i.e.,

$$w(t_k) = \frac{|R_i \,:\, t_k \in T_i|}{|R_i \,:\, R_i \in specific|} < \alpha, \tag{4.5}$$

   where α is a threshold value for the application (e.g., $\alpha = 0.005$)

5. Sort topics by its weighted frequencies $w^s(t_k)$ in its descending order.

Note, that we did not use the generic reviews in both adapted methods described above, unlike they are originally (BAUMAN; TUZHILIN, 2014). In addition, after generating the sorted lists of key-words (or topics), we manually selected in the list of topics for each item domain only the topics related to the *Companion* contextual dimension. In contrast, in the original method, there is no restriction about the contextual dimensions extracted from the key-words (or topics).

[7] WordNet is a large lexical database of English. Words are grouped into sets of cognitive synonyms, each expressing a distinct concept. Function synset(word) returns a list of lemmas of this word that represent distinct concepts.

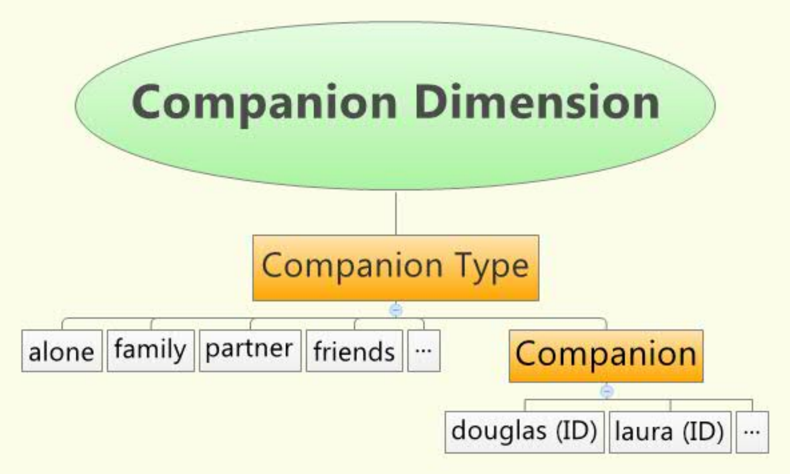


Figure 22 – Example of a companion dimension with its possible contextual attributes and values in a hierarchical view.

In this way, we identified six contextual values (alone, accompanied[8], family, friends, partner, and colleagues[9]) for only one contextual attribute of "companion", from the word groups and topics selected. This contextual attribute is high-level and likewise other contextual dimensions, it could be expanded into more granular contextual attributes (illustrated in Figure 22), as, specific family members (father, mother, siblings, and so on.), or yet, the person itself (e.g. by pointing the companion's name) that the user is with. However, we let the contextual attribute of the *Companion* dimension in high-level due to the source of contextual information (inferred from reviews), in which has a few number of reviews with particular description of companion (specific family members or companion's name).

Furthermore, contextual information of other contextual dimensions could be discovered (e.g. the task or purchase purpose). However, we did not consider other contextual dimensions due to the small percentage of user-rating reviews with their respective inferred contexts, in contrast to the *Companion* dimension, which had 85% of the total of user-rating reviews with the inferred contexts. This observation is reasonable because a few users usually mention their temporal contexts in the reviews and inferring the companion context of user-ratings seems to be easier than inferring the temporal context, for example.

In order to evaluate the classification performance of the implemented method for the companion extraction, we adopted the same methodology described by the authors of that method in (BAUMAN; TUZHILIN, 2014). In this way, for each item domain we randomly selected 300 reviews from the entire set of user-reviews (i.e., book, television,

[8] User-ratings are classified in this high-level contextual value only when a more particular value could not be inferred, such as "family".

[9] In opposition to the "friends" value, in this contextual value are considered only co-workers, classmates, etc.

and music - 900 reviews in total), however, we let 50 reviews for each contextual value (i.e., alone, accompanied, family, friends, partner, and colleagues). Hence, we manually labeled these reviews according to their contextual values and measured the accuracy of the contextual classification by comparing the labeled reviews to the classified reviews. The accuracy was calculated considering the number of correct classifications in comparison to the total of tested reviews. Table 6 reports the results of this empirical evaluation considering the different domains and contextual values.

Table 6 – Classification accuracy of the companion extraction.

Target	Overall	Contextual Values					
Domain	Accuracy	Alone	Accompanied	Family	Friends	Partner	Colleague
Book	19.67%	94%	3%	8%	2%	8%	3%
TV	17%	76%	9%	5%	4%	6%	2%
Music	10.83%	52%	2%	4%	1%	5%	1%

As it can be seen from table, the implemented method did not have a good performance in the companion extraction task. *Book* was the domain with better results in general, while the *Music* had the worst ones. This result may be associated with the length of the user reviews, which is greater in the *Book* domain in comparison to the other ones. In addition, the average implemented method achieved better results for the *Alone* contextual value than for other values. This result may have occurred due to the great presence of personal pronoun “I” in the user reviews, which can be considered as a “topic” by the implemented method. For that reason, that method infers that the user was “alone”.

4.1.2 Selecting Relevant Contextual Attributes and Values

As mentioned in Section 2.3.4, there are several approaches to determine the relevance of a given type of contextual information (contextual dimension). If we consider that only relevant dimensions (*Location*, *Temporal* and *companion*) are present in the datasets, we have to determine the relevance of the contextual attributes of these dimensions. For that, each user-rating in the datasets has contextual information about all contextual dimensions and their attribute variations, as described below:

- Temporal - in this dimension, we persisted two contextual attributes: day (with values: Sunday to Saturday) and day type (values: weekend and weekday);
- Location - in this dimension, we persisted three contextual attributes: country, state, and city[10];
- Companion - in this dimension, we persisted one contextual attribute: companion type (with values: alone, accompanied, family, friends, partner, and colleagues).

[10] 112 countries, 380 states, and 2838 cities were found in the datasets used in this thesis

Given these contextual dimensions and their attributes, we applied a data mining method (as mentioned in Section 3.2.2) to select only the most relevant contextual attributes of each contextual dimension. For that, we adopted the *InfoGainAttributeEval* method from Weka (HALL et al., 2009), which evaluates the worth of an attribute by measuring the information gain for a "class". The output of this method is a ranking of the attribute list which indicates the importance of the attributes in the task of classification.

In our case, the task of classification serves to analyze the influence of the distinct contextual attributes in the user-rating value (class). In this way, we applied the *InfoGainAttributeEval*[11] with the user-rating value as a class (five possible values: 1 to 5) and six attributes (day, day type, country, state, city, companion type) for the two datasets used in this thesis, considering all target domains separately. Table 7 and Table 8 report the information gain values[12] of the contextual attributes in different target domains for these datasets.

Table 7 – Information gain of contextual attributes in different target domains for the book-television dataset.

Target	Temporal Dimension		Location Dimension			Companion Dimension
Domain	Day	Day Type	Country	State	City	Companion Type
Book	**2.6e-4**	1.1e-5	2.3e-3	7.9e-3	**2.8e-2**	**9.9e-5**
TV	**2.3e-4**	8e-5	5e-3	1e-2	**3.6e-2**	**4.6e-5**

Table 8 – Information gain of contextual attributes in different target domains for the book-music dataset.

Target	Temporal Dimension		Location Dimension			Companion Dimension
Domain	Day	Day Type	Country	State	City	Companion Type
Book	**3.9e-4**	3.1e-5	2.2e-3	9e-3	**3.2e-2**	**1.2e-4**
Music	**2.4e-4**	2.6e-5	4.6e-3	8.3e-3	**3.6e-2**	**2.5e-5**

As these tables demonstrate, the information gain was similar in both datasets and their respective domains. For them, the "day" attribute was the most relevant in the *Temporal* dimension as well as the "city" attribute and the "companion type" attribute were the most worth in their respective contextual dimensions. In addition, the "city" attribute is the most relevant to them, followed by the "day" and "companion type" attributes, in that order. Therefore, we have chosen only these three attributes for the evaluation of the proposed CD-CARS. However, there is no guarantee that the quality of recommendation is better in the "city" attribute than in the others. The quality may depend on how good the recommendation algorithms explore the contextual information available. Thus, this analysis does not discard the necessity of experimental evaluations

[11] The weka.attributeSelection.Ranker was the ranker used with the configuration: -T (generateRanking) -1.7976931348623157E308 (threshold) -N (startSet) -1 (numToSelect).

[12] These values range between 0 and 1, where a higher value represents a more discriminating feature

for measuring the quality of recommendations in different (or with less information gain) contexts.

It is important to mention that this analysis considers each attribute independently, and so do not take into account any correlation between distinct contextual attributes. For that reason, we also made experimental evaluations combining the selected contextual attributes (see Chapter 5). Moreover, we selected just one contextual attribute for each contextual dimension, but, other relevant contextual attributes could be used in the *Location* dimension once not all user-ratings have information about their particular location (e.g. instead of "city", a user can only have information about "country" or "state"). Therefore, a CD-CARS implementation could consider any contextual information available although, for evaluation purposes, we have selected only the contextual attributes with higher information gain per dimension in order to minimize the evaluation cost by considering several contextual attributes and their combinations.

Another aspect of the contextual information selection refers to selecting the most relevant contextual values of the contextual attributes. In order to verified this aspect, we had generalized the *Companion* values into only two categories: "alone" or "not-alone", which contained all the other values such as "accompanied", "family", "friends", "partner", and "colleagues". However, we verified that the information gain was higher when the *Companion* values were more granular, so, we kept all the original companion values separately. However, we remember that the quality of the inferred contextual information is low, which may impact in the calculation of the information gain.

4.1.3 Cross-Domain Datasets Description

In this section, we describe the properties of two extracted datasets. One of them for evaluating the CD-CARS in two more related domains (*Book* and *Television*, named as "book-television dataset", described in Section 4.1.3.1) and another considering two less related domains (*Book* and *Music*, named as "book-music dataset", described in Section 4.1.3.2).

4.1.3.1 Book-Television dataset

Table 9 summarizes the properties of the "book-television dataset", which can be split into two single-domain sub-datasets and three or more samples considering ratings from specific contextual dimensions. As it can be seen from the table, the *Books* domain has more ratings (≃64% from total) than *Television* domain (≃36% from total). Also, the table summarizes the cross-domain dataset properties according to three contextual dimensions (*Temporal*, *Location* and *Companion*). In the table, we can see that 100% of the ratings have information about *Temporal* dimension, while almost 45% and 20% of them, respectively, have information about *Location* (city) and *Companion* dimensions. In Section 4.1.2, we described why these contextual dimensions were chosen.

Table 9 – Cross-domain and single-domain “book-television dataset” properties with 100% of user overlap.

Dataset	Users	Items	Ratings	Ratings per	
				User	Item
Cross-domain (both domains)	15341	194615	1249949	81.47	6.42
Books (single-domain)	15341	165896	805102	52.48	4.85
Television (single-domain)	15341	28719	444847	28.99	15.48
Temporal context (both domains)	15341	194615	1249949	81.47	6.42
Location (city) context (both domains)	7405	118020	557018	75.22	4.72
Companion context (both domains)	13598	76295	251707	18.51	3.30
Location (city) AND Companion contexts (both domains)	6846	52032	131257	19.17	2.52

Note that, in Table 9, all ratings from the full cross-domain dataset have a *Temporal* context associated with[13], since they have this information extracted from dates, as described in Section 4.1. For this reason, we omitted the properties of the combination between the *Temporal* and *Location* (or *Companion*) dimensions, since the number of ratings for these combinations is the same as the number of ratings considering only the *Location* (or *Companion*) dimension alone. For example, if we consider the dataset by selecting ratings only with *Location* (city) and *Temporal* contextual dimensions, then we will have a sample of the dataset with 557018 ratings, which is the same number of ratings from the *Location* (city) dimension alone.

It is important to mention that Table 9 shows the properties of the “book-television dataset” with full overlap between users. However, samples of that dataset are used with other user overlap levels in order to perform a sensitivity evaluation (see Section 2.2.6.3). As mentioned in Section 4.1, we generated reduced versions from the full-overlapped datasets by removing all ratings for items from the target domain of the users chosen randomly according to the overlap percentage. In this way, Table 10 and Table 11 present, respectively, the “book-television dataset” properties by considering user overlap levels of 50% and 10% when *Television* is the target domain while Table 12 and Table 13 show the “book-television dataset” properties by considering the same user overlap levels when *Book* is the target domain.

4.1.3.2 Book-Music dataset

Table 14 summarizes the properties of the “book-music dataset”. We can see on the table that the *Books* domain has more ratings (≃72% from total) than *Music* domain (≃28% from total). In addition, 100% of the ratings have information about *Temporal*

[13] See the number of ratings in the 1st and 4th rows.

Table 10 – “book-television dataset” properties with 50% of user overlap when “TV” is the target domain.

Dataset	Users	Items	Ratings	Ratings per	
				User	Item
Cross-domain (both domains)	15341	188402	1011324	65.92	5.37
Books (single-domain)	15341	165896	805102	52.48	4.85
Television (single-domain)	7671	22506	206222	26.88	9.16
Temporal context (both domains)	15341	188402	1011324	65.92	5.37
Location (city) context (both domains)	7405	113049	446297	60.27	3.95
Companion context (both domains)	13012	73353	206512	15.87	2.82
Location (city) AND Companion contexts (both domains)	6571	49216	106664	16.24	2.17

Table 11 – “book-television dataset” properties with 10% of user overlap when “TV” is the target domain.

Dataset	Users	Items	Ratings	Ratings per	
				User	Item
Cross-domain (both domains)	15341	178646	851680	55.57	4.38
Books (single-domain)	15341	165896	805102	52.48	4.85
Television (single-domain)	1534	12750	46578	30.36	3.65
Temporal context (both domains)	15341	178646	851680	55.57	4.38
Location (city) context (both domains)	7405	103862	529307	71.48	5.10
Companion context (both domains)	12546	68084	332271	26.49	4.88
Location (city) AND Companion contexts (both domains)	6363	44830	248592	39.06	5.55

Table 12 – “book-television dataset” properties with 50% of user overlap when “Book” is the target domain.

Dataset	Users	Items	Ratings	Ratings per	
				User	Item
Cross-domain (both domains)	15341	131456	819335	53.41	6.23
Books (single-domain)	7671	102737	374488	48.82	3.65
Television (single-domain)	15341	28719	444847	28.99	15.48
Temporal context (both domains)	15341	131456	819335	53.41	6.23
Location (city) context (both domains)	7405	90838	581614	78.54	6.40
Companion context (both domains)	12425	53684	361950	29.13	6.74
Location (city) AND Companion contexts (both domains)	6298	36102	281688	44.73	7.80

dimension, while almost 46% and 11% of them, respectively, have information about *Location* (city) and *Companion* dimensions.

Note that, due to the same reason mentioned in Section 4.1.3.1, we omitted the properties of the combination between other contextual dimensions, remaining only the combination between the *Location* and *Companion* dimensions.

Table 13 – “book-television dataset” properties with 10% of user overlap when “Book” is the target domain.

Dataset	Users	Items	Ratings	Ratings per User	Ratings per Item
Cross-domain (both domains)	15341	62722	514965	33.57	8.21
Books (single-domain)	1534	34003	70118	45.71	2.06
Television (single-domain)	15341	28719	444847	28.99	15.48
Temporal context (both domains)	15341	62722	514965	33.57	8.21
Location (city) context (both domains)	7087	43859	244913	34.56	5.58
Companion context (both domains)	11420	24291	104353	9.14	4.30
Location (city) AND Companion contexts (both domains)	5788	17147	55803	9.64	3.25

Table 14 – Cross-domain and single-domain “book-music dataset” properties with 100% of user overlap.

Dataset	Users	Items	Ratings	Ratings per User	Ratings per Item
Cross-domain (both domains)	13189	219034	1031386	78.20	4.71
Books (single-domain)	13189	162449	742844	56.32	4.57
Music (single-domain)	13189	56585	288542	21.88	5.10
Temporal context (both domains)	13189	219034	1031386	78.20	4.71
Location (city) context (both domains)	6951	132830	478510	68.84	3.60
Companion context (both domains)	11519	75754	207010	17.97	2.73
Location (city) AND Companion contexts (both domains)	6412	53999	116100	18.11	2.15

Table 15 – “book-music dataset” properties with 50% of user overlap when “Music” is the target domain.

Dataset	Users	Items	Ratings	Ratings per User	Ratings per Item
Cross-domain (both domains)	13189	208427	897227	68.03	4.31
Books (single-domain)	13189	162449	742844	56.32	4.57
Music (single-domain)	6595	45978	154383	23.41	3.36
Temporal context (both domains)	13189	208427	897227	68.03	4.31
Location (city) context (both domains)	6921	120509	407003	58.81	3.38
Companion context (both domains)	11315	75233	197571	17.46	2.63
Location (city) AND Companion contexts (both domains)	6303	53474	110730	17.57	2.07

Regarding the sensitivity analysis, Table 15 and Table 16 present, respectively, the “book-music dataset” properties by considering user overlap levels of 50% and 10% when “Music” is the target domain while Table 17 and Table 18 show the “book-music dataset” properties by considering the same user overlap levels when *Book* is the target domain.

Table 16 – “book-music dataset” properties with 10% of user overlap when “Music” is the target domain.

Dataset	Users	Items	Ratings	Ratings per	
				User	Item
Cross-domain (both domains)	13189	177368	770030	58.38	4.34
Books (single-domain)	13189	162449	742844	56.32	4.57
Music (single-domain)	1319	14919	27186	20.61	1.82
Temporal context (both domains)	13189	177368	770030	58.38	4.34
Location (city) context (both domains)	6897	102369	350664	50.84	3.43
Companion context (both domains)	11141	74144	190478	17.10	2.57
Location (city) AND Companion contexts (both domains)	6203	52438	106797	17.22	2.04

Table 17 – “book-music dataset” properties with 50% of user overlap when “Book” is the target domain.

Dataset	Users	Items	Ratings	Ratings per	
				User	Item
Cross-domain (both domains)	13189	154329	635947	48.22	4.12
Books (single-domain)	6595	97744	347405	52.68	3.56
Music (single-domain)	13189	56585	288542	21.88	5.10
Temporal context (both domains)	13189	154329	635947	48.22	4.12
Location (city) context (both domains)	6482	104249	317421	48.97	3.04
Companion context (both domains)	8209	51833	116360	14.17	2.25
Location (city) AND Companion contexts (both domains)	4578	36396	66499	14.53	1.83

Table 18 – “book-music dataset” properties with 10% of user overlap when “Book” is the target domain.

Dataset	Users	Items	Ratings	Ratings per	
				User	Item
Cross-domain (both domains)	13189	87993	347805	26.37	3.95
Books (single-domain)	1319	31408	59263	44.93	1.89
Music (single-domain)	13189	56585	288542	21.88	5.10
Temporal context (both domains)	13189	87993	347805	26.37	3.95
Location (city) context (both domains)	5945	57465	172780	29.06	3.00
Companion context (both domains)	5454	15949	35785	6.56	2.24
Location (city) AND Companion contexts (both domains)	3071	10310	19971	6.50	1.94

4.2 Contextual Model Implementation

In this section, we describe how the contextual model, outlined in Section 3.2, was implemented. To implement this contextual model, we extended the implementation of the Mahout[14] framework (OWEN et al., 2011). Figure 23 and Figure 24 show two class diagrams that represent the contextual data model considering the extension/realization of three Mahout entities (*RecommenderBuilder*, *AbstractDataModel* and *IDRescorer*).

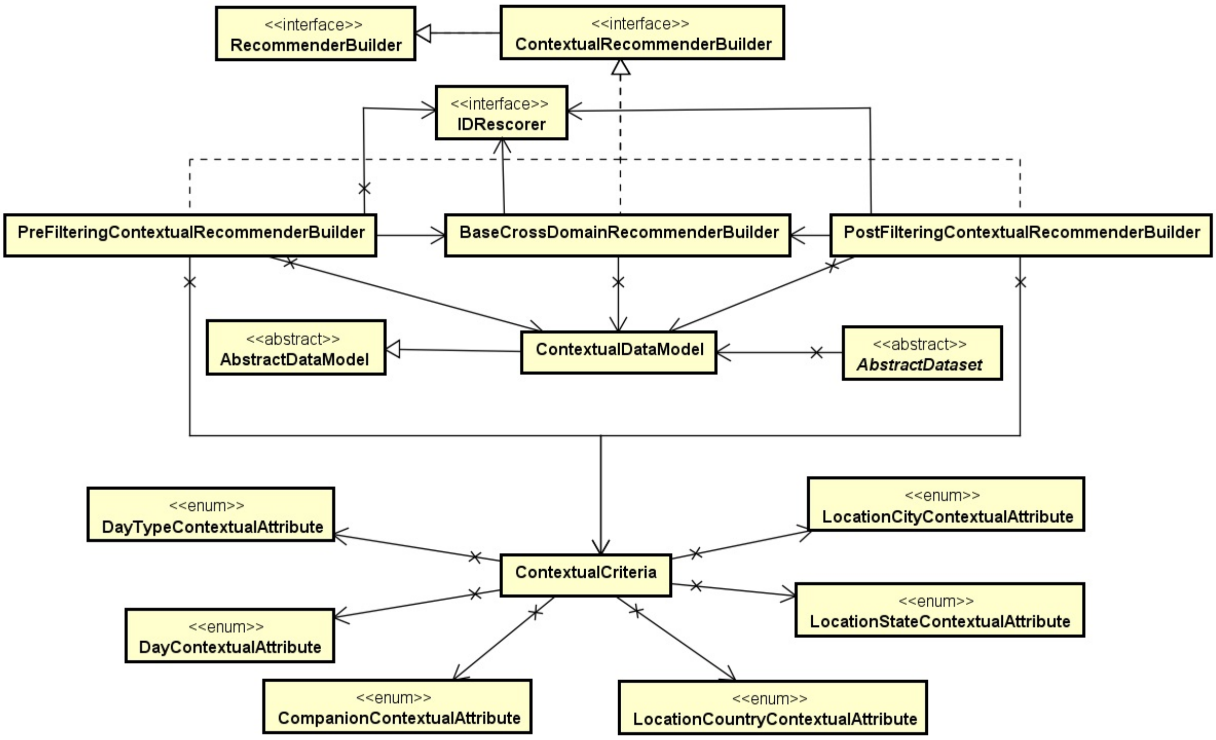

<<interface>>
RecommenderBuilder
<<interface>>
ContextualRecommenderBuilder
<<interface>>
IDRescorer
PreFilteringContextualRecommenderBuilder
BaseCrossDomainRecommenderBuilder
PostFilteringContextualRecommenderBuilder
<<abstract>>
AbstractDataModel
ContextualDataModel
<<abstract>>
AbstractDataset
<<enum>>
DayTypeContextualAttribute
<<enum>>
LocationCityContextualAttribute
ContextualCriteria
<<enum>>
DayContextualAttribute
<<enum>>
LocationStateContextualAttribute
<<enum>>
CompanionContextualAttribute
<<enum>>
LocationCountryContextualAttribute

Figure 23 – Data model class diagram focusing contextual aspects of the CD-CARS implementation.

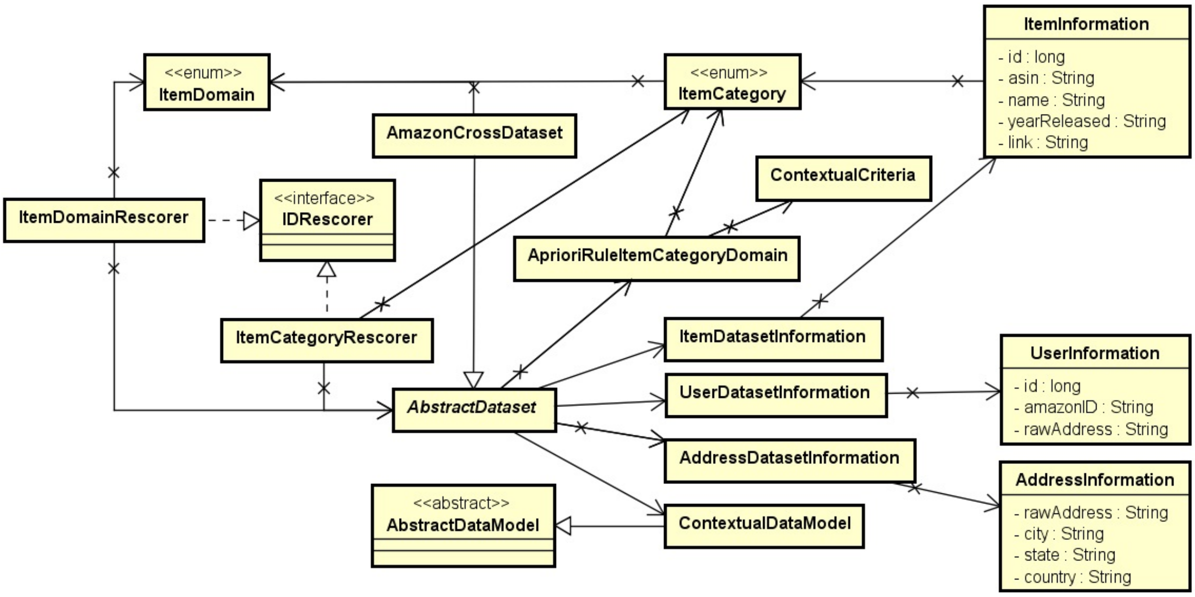


Figure 24 – Data model class diagram focusing dataset aspects of the CD-CARS implementation.

[14] Apache Mahout open source project is a machine learning library under Apache Software Foundation - http://mahout.apache.org/

In the following, we describe the main entities represented in those class diagrams:

- *RecommenderBuilder* - this Mahout interface guides the implementation of classes responsible for building recommender algorithms in order to be evaluated based on a given realization of the *AbstractDataModel* Mahout abstract class. In turn, these recommender algorithms must implement the *Recommender* Mahout interface to recommend items for a user.

- *AbstractDataModel* - this Mahout abstract class implements some basic methods defined by the *DataModel* Mahout interface, which represents a repository of information about users and their associated preferences for items (i.e., user-ratings).

- *ContextualRecommenderBuilder* - this novel interface extends the *RecommenderBuilder* Mahout interface in order to guide the building of recommender algorithms capable of taking advantage of context-awareness features.

- *PreFilteringContextualRecommenderBuilder* - this class implements the *ContextualRecommenderBuilder* interface. It is responsible for building and preparing the pre-filtering recommender algorithm (see Section 3.3.1.1) for evaluation according to three parameters: target domain (restricting source domains declared in the *ItemDomainRescorer* class, showed in Figure 24), contextual data model (represented by the *ContextualDataModel* class), and the context of the recommendation (represented by the *ContextualCriteria* class).

- *PostFilteringContextualRecommenderBuilder* - this class implements the *ContextualRecommenderBuilder* interface and follows the same logic as the *PreFilteringContextualRecommenderBuilder*, but, considering the post-filtering recommender algorithm (see Section 3.3.1.2) instead of the pre-filtering one.

- *BaseCrossDomainRecommenderBuilder* - this class implements the *ContextualRecommenderBuilder* interface. It is responsible for building and preparing the base cross-domain recommender algorithm (see Section 3.3.2) for evaluation according to two parameters: target domain (restricting source domains declared in the *ItemDomainRescorer* class, described later) and contextual data model (represented by the *ContextualDataModel* class). In contrast to the *PreFilteringContextualRecommenderBuilder* and the *PostFilteringContextualRecommenderBuilder*, this class does not take into account the context of the recommendation (represented by the *ContextualCriteria* class), since it does not use any contextual information to recommend items in the target domain. However, it is used by the *PreFilteringContextualRecommenderBuilder* and *PostFilteringContextualRecommenderBuilder* classes that bring the contextual data model, which in turn can be used by the

BaseCrossDomainRecommenderBuilder only for evaluation purposes, for example, in specific contexts as a baseline (without interfering in the recommendation process).

- *ContextualDataModel* - while the *AbstractDataModel* Mahout abstract class contains only user-ratings, this novel class contains, besides the user-ratings, contextual information about these user-ratings, thus, the *ContextualDataModel* extends the *AbstractDataModel*. It is important to mention that the implementation of the *AbstractDataModel* is worried about performance issues (OWEN et al., 2011) and, for that reason, represents all preferences of a user through a *PreferenceArray* object. This object contains a single user ID, an array of item IDs, and an array of preference values. In our implementation, we extended the *PreferenceArray* with a multidimensional array of contextual feature codes (*ContextualPreferenceArray*).

- *ContextualCriteria* - this novel class encapsulates the context of the recommendation represented by a list of contextual values (according to their contextual dimensions and attributes), which in turn are implemented as enumeration objects that realize the *AbstractContextualAttribute* interface. The list of contextual values is composed of six enumeration objects, one for each implemented *AbstractContextualAttribute*.

- *IDRescorer* - this Mahout interface allows the realization of classes that can filter out items from the recommended item list according to several attributes such as an item genre (e.g. action, comedy, etc.) or an item domain (e.g. book, music,etc.). The developer that creates a class to implement the *IDRescorer* is free to choose the appropriate attribute for his purposes. Therefore, this Mahout interface makes the cross-domain recommendation possible, as well as the post-filtering recommendation (see Section 3.3.1.2), since both item genre and item domain can be used as a filter.

- *ItemDomainRescorer* - this class implements the *IDRescorer* Mahout interface in order to filter out from the recommend item list those items that belong to the set of items from the source domains. These domains are specified through *ItemDomain* enumeration objects. For that, the *ItemDomainRescorer* depends on the information from a dataset (e.g. we created an *AmazonCrossDataset* class, which is an implementation of the *AbstractDataset* class and contains only three domains: *Book*, *Television* and *Music*).

- *ItemCategoryRescorer* - this class implements the *IDRescorer* Mahout interface in order to filter out from the recommend item list those items that belong to the set of categories specified by *ItemCategory* enumeration objects. Thus, this class is useful for the post filtering recommendation algorithm and also depends on the information from a dataset, which contains a set of categories retrieved from the set of items.

- *AbstractDataset* - this abstract class encapsulates, besides the contextual data model, a set of meta-information about its users (*UserDatasetInformation*), items

(*ItemDatasetInformation*), addresses (*AddressDatasetInformation*) and association rules (*AprioriRuleItemCategoryDomain*).

- *AprioriRuleItemCategoryDomain* - this class generates and persists a set of association rules used by the post filtering recommendation (see Section 3.3.1.2). These general rules are generated from all user preferences in the contextual data model, and relate item categories (*ItemCategory* enumeration objects) between different domains, e.g. who likes action movies also likes rock music - {ACTION, TV} => {ROCK, MUSIC}. In addition, for each generated rule, the class maintain its confidence and support levels, besides all combinations of contexts (represented by a *ContextualCriteria*) from the instances that generated that rule. These combination of contexts are used a posteriori by the *PostFilteringStrategyRecommendation* class (described in Section 4.3.2) that generated more specific rules (e.g. {ACTION, TV, WEEKDAY} => {ROCK, MUSIC, WEEKEND}) by considering particular contexts. This decision avoids the generation of several rules with context that are not used by the Post-Filtering algorithm, once that these rules only are used when a user does not have any category preferences in a specific context.

- *ItemDatasetInformation* - this class represents the set of *ItemInformation* objects. An *ItemInformation* object contains information about an item, such as ID, name/title, year released, category, domain, link, and so on.

- *UserDatasetInformation* - this class represents the set of *UserInformation* objects. A *UserInformation* object contains information about a user, such as ID, Amazon user ID, raw address (a not-standardized string), and so on.

- *AddressDatasetInformation* - this class represents the set of *AddressInformation* objects. An *AddressInformation* object contains information about a standardized address, such as raw address string, city, state, country, and so on.

4.3 Proposed Algorithms Implementation

In this thesis, we implemented two of the proposed algorithms (Section 3.3): Pre-Filtering (PreF) and Post-Filtering (PostF). In the next subsections, we will describe some implementation particularities of these algorithms considering the contextual model entities mentioned in Section 4.2 and other Mahout's entities.

4.3.1 Pre-filtering Implementation

We implemented the PreF algorithm according to its proposal, described in Section 3.3.1.1. The PreF implementation filters out from the *ContextualDataModel* the target domain user-ratings whose contexts (*ContextualPreferenceArray*) are "contained" in the context of the recommendation (defined by the *ContextualCriteria*). In other words, the user-rating context must be the same context as the context of the recommendation without considering unknown contextual attributes of the user-rating context. Besides, it is important to remember that the user-ratings are only filtered in the target domain (*ItemDomain*), which is verified through the *ItemDomainRescorer* and the *ItemDatasetInformation* classes (described before).

For illustrating the pre-filtering process, consider a context of the recommendation and the contexts of a set of user-ratings, both respectively represented by *ContextualCriteria* and *ContextualPreferenceArray* entities in Figure 25. While the *ContextualPreferenceArray* contains a multidimensional array representing different user-ratings (first index of the array) and its contextual values in a sequence representing different contextual attributes (second index of the array), the *ContextualCriteria* contains a list of enumeration objects representing different contextual attributes, which have a contextual code for each contextual value. The same order of contextual attributes declared in the *ContextualPreferenceArray* is automatically used in the *ContextualCriteria*, since the *ContextualFileAttributeSequence* instance determines a unified order of the contextual attributes.

As it can be seen in Figure 25, the user (*userId* = '1') has ratings for five items (*itemIds* from '1' to '5'), where each rating for an item has an array of contextual codes in a specific order (e.g. *itemIds*[0] = 1, *ratings*[0]=4.0, *contextualPreferences*[0] = {1,1,0,-1,-1,-1}). Each contextual code represents a contextual value of a different contextual attribute in the sequence defined by the *ContextualFileAttributeSequence* instance[15]. In this way, considering that the *ContextualCriteria* from the figure can be expressed by the following contextual code sequence: {1 ("SUNDAY"), 1 ("WEEKEND"), -1 ("UNKNOWN"),-1 ("UNKNOWN"),-1 ("UNKNOWN"),2 ("FAMILY")}, so, only one user-rating (*contextualPreferences*[4]) is discarded, whereas the others four ones (*contextualPreferences*[0] to *contextualPreferences*[3]) are maintained by the PreF implementation. This process is illustrated in Figure 26, in which a green square means "a matching" between the contextual attribute values from the user-rating contexts and the recommendation context, whereas a red square means "a non-matching". Finally, a yellow square means that the contextual attribute was not considered for matching between the user-rating contexts and the recommendation context, since one (or both) of these contextual attributes is "unknown".

[15] The six codes are represented respectively by the *DayContextualAttribute*, *DayTypeContextualAttribute*, *LocationCountryContextualAttribute*, *LocationStateContextualAttribute*, *LocationCityContextualAttribute*, and *CompanionContextualAttribute* enumerations

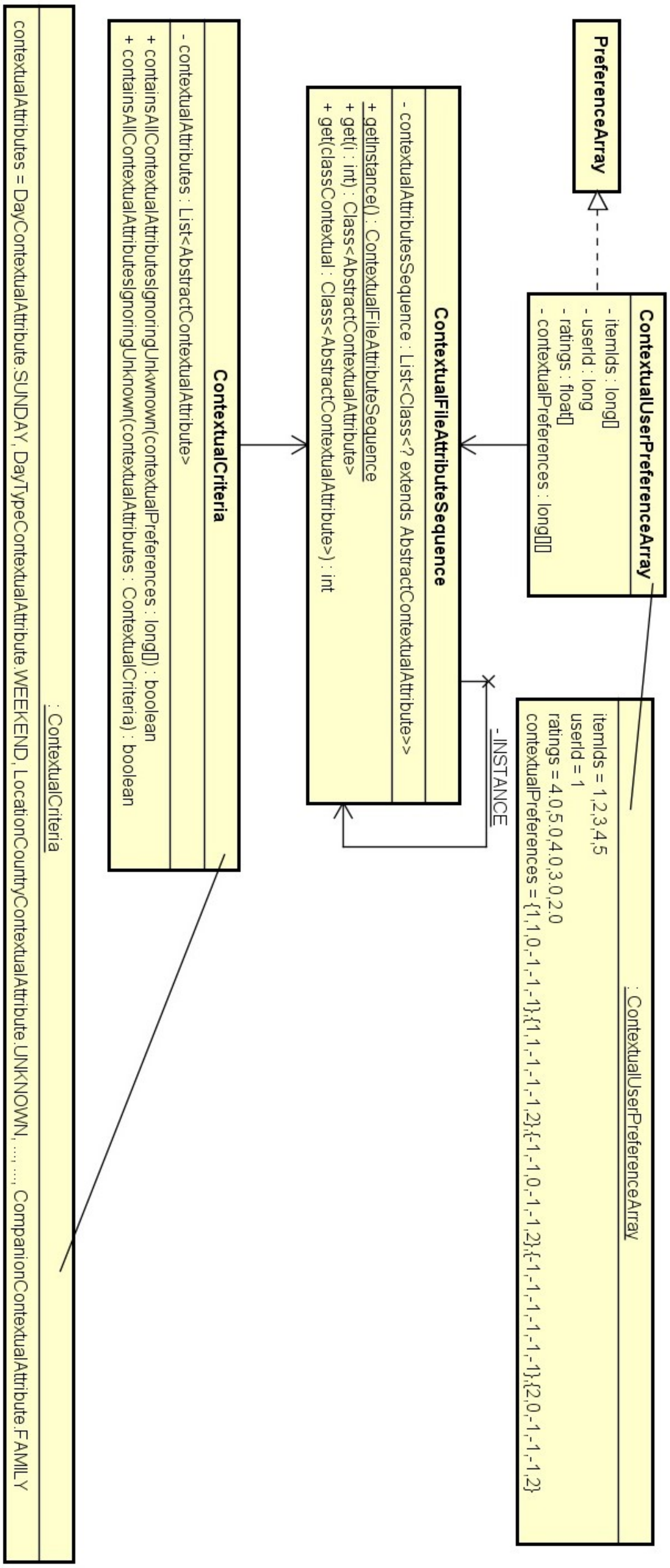

PreferenceArray
ContextualUserPreferenceArray
- itemIds : long[]
- userId : long
- ratings : float[]
- contextualPreferences : long[][]
ContextualFileAttributeSequence
- contextualAttributesSequence : List<Class<? extends AbstractContextualAttribute>>
+ getInstance() : ContextualFileAttributeSequence
+ get(i : int) : Class<AbstractContextualAttribute>
+ get(classContextual : Class<AbstractContextualAttribute>) : int
-INSTANCE
: ContextualUserPreferenceArray
itemIds = 1,2,3,4,5
userId = 1
ratings = 4.0,5.0,4.0,3.0,2.0
contextualPreferences = {1,1,0,-1,-1,-1},{1,1,-1,-1,-1,2},{-1,-1,0,-1,-1,2},{-1,-1,-1,-1,-1,-1},{2,0,-1,-1,-1,2}
ContextualCriteria
- contextualAttributes : List<AbstractContextualAttribute>
+ containsAllContextualAttributesIgnoringUnkwnown(contextualPreferences : long[]) : boolean
+ containsAllContextualAttributesIgnoringUnknown(contextualAttributes : ContextualCriteria) : boolean
: ContextualCriteria
contextualAttributes = DayContextualAttribute.SUNDAY, DayTypeContextualAttribute.WEEKEND, LocationCountryContextualAttribute.UNKNOWN, ..., ..., CompanionContextualAttribute.FAMILY

Figure 25 – Class diagram illustrating entities used by the pre-filtering class.

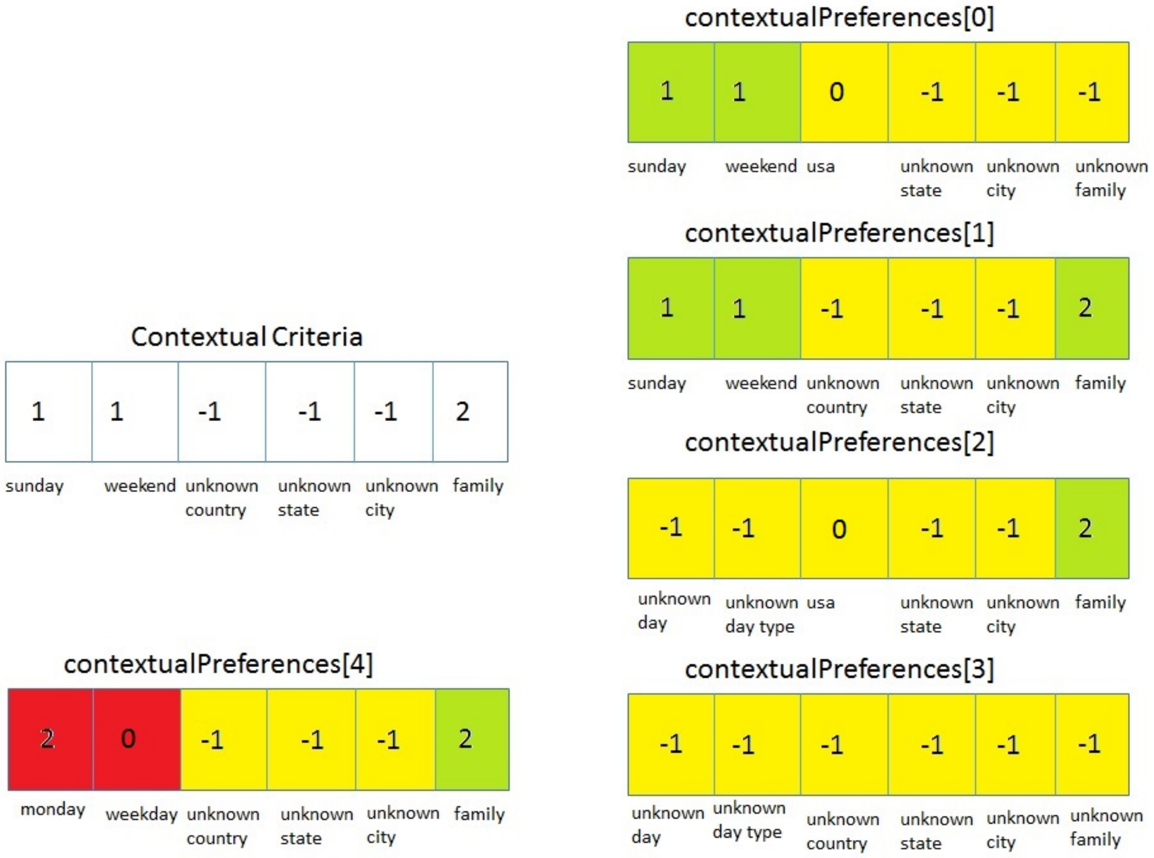

contextualPreferences[0]
1 1 0 -1 -1 -1
sunday weekend usa unknown state unknown city unknown family
contextualPreferences[1]
1 1 -1 -1 -1 2
sunday weekend unknown country unknown state unknown city family
Contextual Criteria
1 1 -1 -1 -1 2
sunday weekend unknown country unknown state unknown city family
contextualPreferences[2]
-1 -1 0 -1 -1 2
unknown day unknown day type usa unknown state unknown city family
contextualPreferences[4]
2 0 -1 -1 -1 2
monday weekday unknown country unknown state unknown city family
contextualPreferences[3]
-1 -1 -1 -1 -1 -1
unknown day unknown day type unknown country unknown state unknown city unknown family

Figure 26 – Example of the pre-filtering process considering the context of user-ratings and the recommendation context.

4.3.2 Post-filtering Implementation

In contrast to the PreF proposal, the PostF one (described in Section 3.3.1.2) allows that several strategies be implemented. Thus, we investigated some strategies to perform the PostF recommendation by varying its threshold value (θ). For instance, we set θ to 2/3 of the frequency of the most preferred category. So, suppose that a user has given good ratings (at least 4.0 in a scale from 1.0 to 5.0), in a given context, for thirty religion books, twenty-five educational books, twenty comedy books, nineteen romance books, and ten action books, as illustrated in Figure 27. By applying the threshold strategy, the minimal value of occurrences is twenty (2/3 of 30, which is the frequency of the most preferred category - religion) for an item category to be maintained on the recommendation list. So, only religion (30 occurrences), comedy (20 occurrences) and educational books (25 occurrences) are included in the resulting recommendation, i.e., books of other categories are ignored such as romance (19 occurrences) and action (10 occurrences) ones.

The optimal value of θ can be set through experiments, as well as the minimal value to consider a good rating, which will vary depending on the rating value interval (e.g. in an interval of 1-10 we could consider 8.0 or more like a good rating). The higher the θ value, the less the number of categories included in the users' preferred categories.

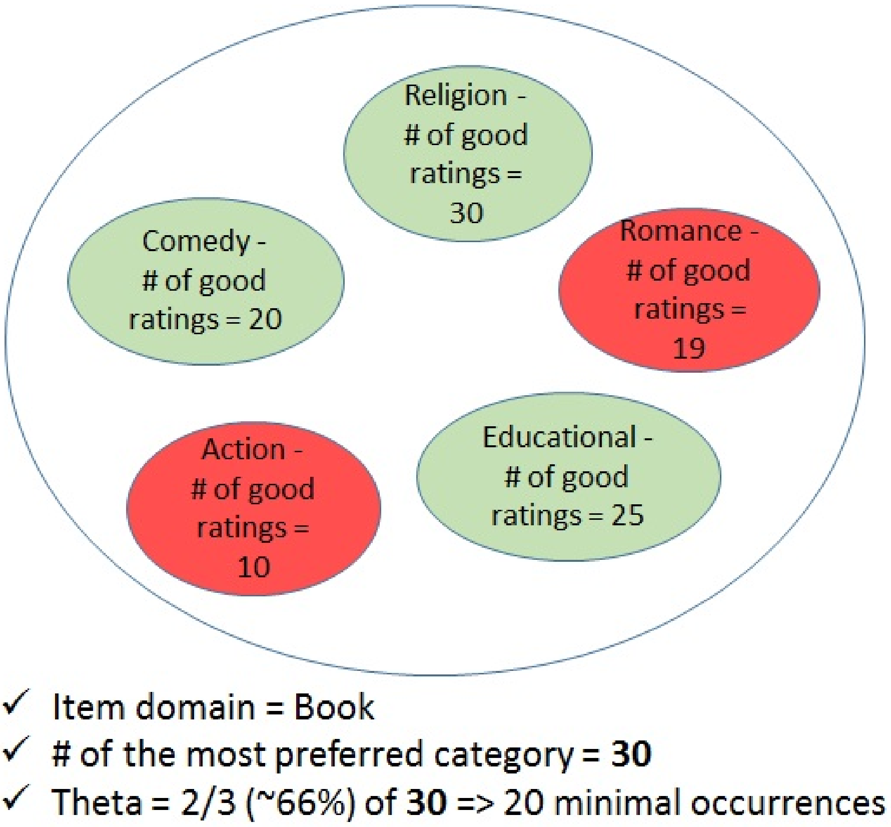

Religion - # of good ratings = 30
Comedy - # of good ratings = 20
Romance - # of good ratings = 19
Action - # of good ratings = 10
Educational - # of good ratings = 25
✓ Item domain = Book
✓ # of the most preferred category = 30
✓ Theta = 2/3 (~66%) of 30 => 20 minimal occurrences

Figure 27 – Example of selected categories in the post-filtering recommendation.

Figure 28 shows the main entities of the post-filtering implementation. From this figure, we can see that *PostFilteringStrategyRecommendation* class uses two distinct databases of preferred categories:

- *UserCategoriesPrefsInContextsByDomain* - this database contains all categories of rated items for each user in the dataset. However, only the categories of well-rated (or "good") items were considered (at least 4.0 in a scale from 1.0 to 5.0). Besides, the number of occurrences of each category is associated with their observed contexts and domains (e.g. {RELIGION,BOOK,WEEKDAY} = 5, which means that a user rated five books as "good" in weekdays). Thus, this class contains one contextual preference tensor $CP(u, c, g)$ (described in Section 3.3.1.2) for each item domain.

- *CategoryContextDomainRulesMap* - when a user does not have any information about preferred categories in a given domain, general association rules are necessary so that the post-filtering algorithm can recommend items even in a domain that the user has not ratings (see Section 3.3.1.2). In this way, this database increments the set of rules initially generated by the *AprioriRuleItemCategoryDomain* class with contextual information. For instance, a {RELIGION,BOOK} => {RELIGION,TV} rule from the *AprioriRuleItemCategoryDomain* could be transformed into a {RELIGION,BOOK,WEEKDAY} => {RELIGION,TV,WEEKEND} rule. Each part of a rule is represented by a *RuleTuple* class, which contains information about the context (*ContextualCriteria*), item category (*ItemCategory*) and item domain (*ItemDomain*). In this case, rules are composed by one precedent *RuleTuple* and one consequent *RuleTuple*. Later in this section, we will detail the implementation of the process of association rules generation.

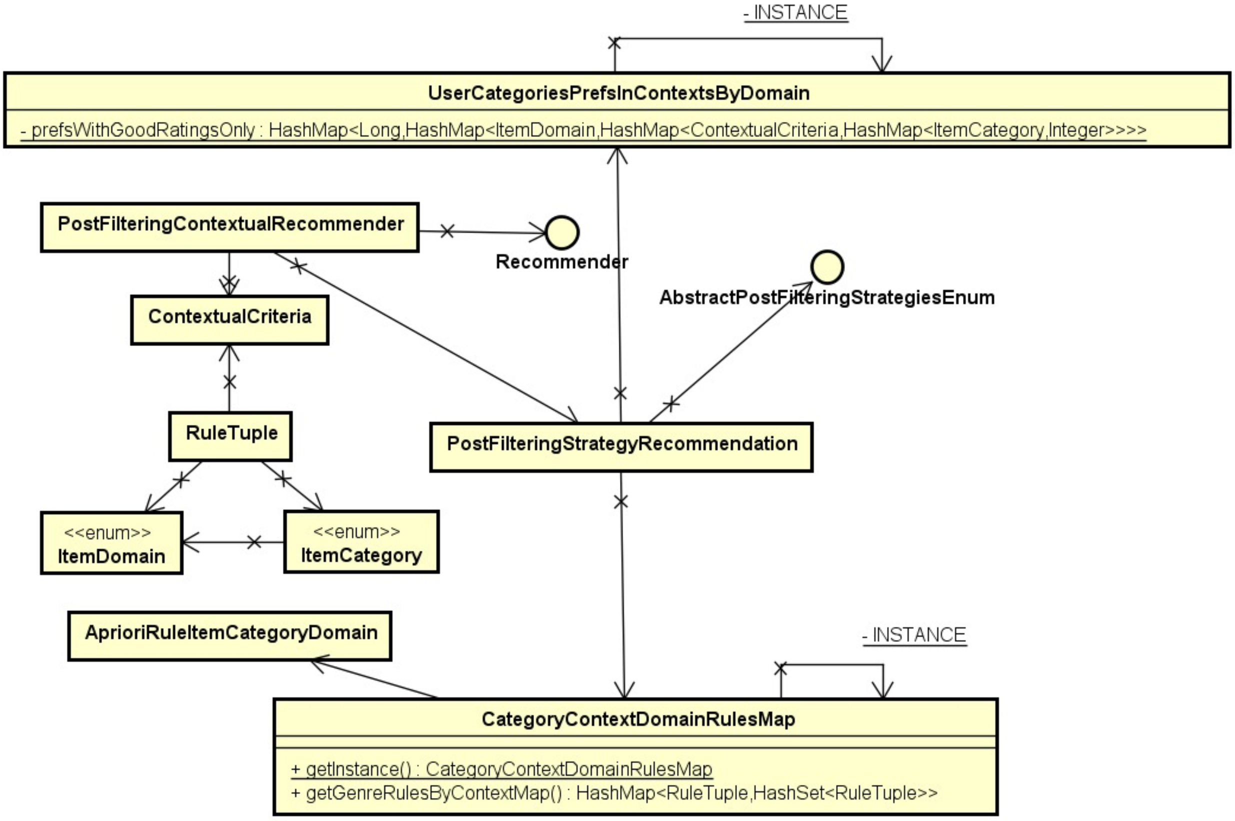

- INSTANCE
UserCategoriesPrefsInContextsByDomain
- prefsWithGoodRatingsOnly : HashMap<Long,HashMap<ItemDomain,HashMap<ContextualCriteria,HashMap<ItemCategory,Integer>>>>
PostFilteringContextualRecommender
Recommender
AbstractPostFilteringStrategiesEnum
ContextualCriteria
RuleTuple
PostFilteringStrategyRecommendation
<<enum>>
ItemDomain
<<enum>>
ItemCategory
AprioriRuleItemCategoryDomain
- INSTANCE
CategoryContextDomainRulesMap
+ getInstance() : CategoryContextDomainRulesMap
+ getGenreRulesByContextMap() : HashMap<RuleTuple,HashSet<RuleTuple>>

Figure 28 – A class diagram illustrating the main post-filtering entities.

It is important to mention that these databases are updated periodically on demand, i.e., in execution time just when it is necessary. We made this design decision once that we are concerned about performance issues given that the generation of association rules is costly if we consider the multi-dimensional *RuleTuple* entities (item category, item domain and context). In order to generate these rules, we used the *AprioriRuleItemCategoryDomain* and *CategoryContextDomainRulesMap* implementations in a two-step process described in Algorithm 2 (1-step) and in Algorithm 3 (2-step).

Algorithm 2. AprioriRuleItemCategoryDomain *algorithm for association rules generation (1-step).*

Input: d *(dataset),* cl *(minimal confidence level),* sl *(minimal support level),* gr *(minimal "good" rating threshold value)*
Output: rs *(rules with category and domain information)*

```
procedure generatePreferredCategoriesMapByUser(d,gr)
    Create a uc data structure containing a list of users where each user (u) has a set of categories (cs)
    for each u ∈ d do
        Get the array (ur) of user ratings (r) from u
        for each r ∈ ur do
            v = value from r
            if v ≥ gr then
                i = item from r
                ic = categories from i
                acs = cs from uc for u
                for each c ∈ ic do
                    if c ⊈ acs then
                        Add c in the acs
                        Add acs in uc for u
                    end if
                end for
            end if
        end for
    end for
return uc
end procedure
```

```
procedure generateAprioriCategoryDomainRules(uc, d, cl, sl)
    for $i = 0$; i < size of the categories set from d; $i = i + 1$ do
        for $j = 0$; j < size of the categories set from d; $j = j + 1$ do
            di = domain from category i
            dj = domain from category j
            if $di \neq dj$ then
                Create a precedent tuple (pt) composed by the category i and di
                Create a consequent tuple (ct) composed by the category j and dj
                $ptc = 0$
                $ctc = 0$
                for each $u \in uc$ do
                    acs = cs from uc for u
                    if category i ⊂ acs then
                        $ptc = ptc + 1$
                        if category j ⊂ acs then
                            $ctc = ctc + 1$
                        end if
                    end if
                end for
                Create a rule (r) composed by pt, ct, ptc and ctc.
                Add r in rs
            end if
        end for
    end for
    for each r ∈ rs do
        Get ptc and ctc from r
        nu = number of users from uc
        $rcl = ctc/ptc$
        $rsl = ctc/nu$
        if rcl < cl or rsl < sl then
            Remove r from rs
        end if
    end for
return rs
end procedure
```

end

Algorithm 3. CategoryContextDomainRulesMap *algorithm for association rules generation (2-step).*

Input: rs*, which is a set of rules from the 1-step algorithm;* d *(dataset);* gr *(minimal "good" rating threshold value);* uc *data structure containing a list of users where each user (*u*) has a set of categories (*cs*); a rule tuple condition (*rtc*) composed by an item category* ict*, an item domain* idt *and contextual criteria (*cc*);* cl *(minimal confidence level); and* sl *(minimal support level)*

Output: crs *(set of rules with category, domain and contextual information)*

```
procedure addContextualInformationInAprioriCategoryDomainRules(rs, d, gr, uc)
    for each rule ∈ rs do
        for each u ∈ d do
            Get cs from uc for u
            Get the precedent tuple (pt) from rule
            Get the consequent tuple (ct) from rule
            Get ic1 from pt
            Get ic2 from ct
            if ic1 $\subset$ cs and ic2 $\subset$ cs then
                Get the array (ur) of user ratings (r) from u
                Create an empty set of contexts for the rule precedent category (cpt)
                Create an empty set of contexts for the rule consequent category (cct)
                for each r $\in$ ur do
                    v = value from r
                    if v $\geq$ gr then
                        i = item from r
                        ic = categories from i
                        if ic1 $\subset$ ic then
                            urc = context from r
                            Add urc in cpt
                        else
                            if ic2 $\subset$ ic then
                                Add urc in cct
                            end if
                        end if
                    end if
                end for
                if cpt $\neq \varnothing$ and cct $\neq \varnothing$ then
                    for $i = 0$; i < cpt size; $i = i + 1$ do
                        for $j = 0$; j < cct size; $j = j + 1$ do
                            Add context i from cpt in pt for the rule
                            Add context j from cct in ct for the rule
                        end for
                    end for
                end if
            end if
        end for
    end for

end procedure
procedure getAprioriCategoryDomainContextRules(rs, rtc, cl, sl)
    for each rule $\in$ rs do
        Get the precedent tuple (pt) from rule
        Get the consequent tuple (ct) from rule
        Get the item category (ic) from pt
        Get the item domain (id) from pt
        Get ict from rtc
        Get idt from rtc
        Get cc from rtc
        if ic = ict and id = idt then
            Get the total number contexts (nc) from pt
            $ptc = 0$
            $ctc = 0$
            for $i = 0$; $i <$ nc; $i = i + 1$ do
                if contextualMatching(cc,context i from pt,size of the array cc) then //According to Algorithm 1
                    $ptc = ptc + 1$
                    if contextualMatching(cc,context i from ct,size of the array cc) then
                        $ctc = ctc + 1$
                    end if
                end if
            end for
            $rcl = ctc/ptc$
            $rsl = ctc/nc$
            if rcl < cl and rsl < sl then
                Get the item category (icc) from ct
                Get the item domain (idc) from ct
                Create a contextual rule (cr) containing the rtc as precedent, and a consequent rule tuple composed by icc, idc and cc
                Add cr in crs
            end if
        end if
    end for
return crs
end procedure
```

end

Both algorithms described above are based on Apriori algorithm (AGRAWAL; IMIELIŃSKI; SWAMI, 1993). They can be seen as simplified versions of it since we are interested only in a subset of rules. More precisely, once that the PostF algorithm only uses the rules base when a user does not have contextual preferences in the recommendation domain, so, only rules that relate categories among different domains (source and target) are necessary. For example, we could obtain rules like "{RELIGION,BOOK} => {RELIGION,TV}" or "{ACTION,TV} => {ROCK,MUSIC}" from Algorithm 2.

As it can be seen in this algorithm, first, we perform the *GeneratePreferredCategoriesMapByUser* procedure considering only good ratings (we set the *minGoodRatingValue* to 4.0), and then we apply the *GenerateAprioriCategoryDomainRules* procedure, resulting in a 1-step rule base which contains rules among different domains considering only their item categories. For that, this procedure makes all possible combinations of precedent and consequent item categories of different domains (see lines 2 and 3 in the *GenerateAprioriCategoryDomainRules* procedure).

Furthermore, there are two Apriori parameters (*minConfidenceLevel* and *minSupportLevel*) that can be used to define a minimal threshold for considering rules or not. If the minimum confidence and support levels for mining the rules are high, then the algorithm may not obtain enough rules for the PostF recommendation. Again, these rules are only required when a user does not have any category preference in the context of the recommendation for the target domain.

Therefore, for the datasets used in this implementation, we set the *minConfidenceLevel* to 0.7 and the *minSupportLevel* to 0.01. Thus, the 1-step algorithm obtained 25 rules[16] for the "book-television dataset" and 23 rules[17] for the "book-music dataset", both with full user overlap. If we take into account the number of categories in the three domains (27 for *Book* and *Television*, and 19 for *Music*), then we can say that almost a half of these categories have a rule with an associated category, inferred by the 1-step algorithm. This also means that the PostF algorithm will not be able to recommend items of the categories that do not have any association rule.

We used the same values for *minConfidenceLevel* and *minSupportLevel* parameters in Algorithm 3. This 2-step algorithm has the set of rules generated by the 1-step algorithm as an input and generates a rule base containing rules among different domains considering their item categories and also their contexts. In this case, examples of inferred rules could be like "{ACTION,TV,WEEKEND_FAMILY} => {ROCK,MUSIC,WEEKEND_FAMILY}" or "{RELIGION,BOOK,WEEKDAY_CANADA} => {GOSPEL,MUSIC,WEEKDAY_CANADA}". Note that both precedent and consequent contexts of these rules are the same (according to the line 27 in the *getAprioriCategoryDomainContextRules* procedure). This rule "kind" (with the same precedent and consequent contexts) is sufficient for our purposes since the PostF algorithm only needs to infer a set of preferred categories in the target domain for a user with preferred categories in the source domain according to the

[16] Examples of generated rules: {DRAMA,TV => FANTASY,BOOK} (confidence=0.75 and support=0.5), {ARTISTIC,BOOK => DRAMA,TV} (confidence=0.70 and support=0.12), {WESTERNS,TV => DOCUMENTARY,BOOK} (confidence=0.70 and support=0.09), {HEALTH,TV => EDUCATIONAL,BOOK} (confidence=0.74 and support=0.03), among others.

[17] Examples of generated rules: {POP,MUSIC => FANTASY,BOOK} (confidence=0.74 and support=0.33), {CLASSICAL,MUSIC => DOCUMENTARY,BOOK} (confidence=0.73 and support=0.14), {NEW AGE,MUSIC => EDUCATIONAL,BOOK} (confidence=0.72 and support=0.05), {KIDS,BOOK => KIDS,MUSIC} (confidence=0.78 and support=0.03), among others.

context of the recommendation. This context is used to filter the users' preferred item categories from the source domain in order to obtain the inferred item categories in the target domain.

Finally, it is important to remember that the PostF algorithm is applied after the base cross-domain recommendation by filtering recommended items out in the target domain according to the users' item category preferences (inferred or not).

4.4 Base Cross-domain Algorithm Implementation

As mentioned in Section 3.3.2, we apply collaborative filtering algorithms as base cross-domain algorithm. In this implementation, we adopted a neighborhood-based (user-based similarity) algorithm, due to its simplicity and based on a preliminary battery of experiments using other CF-based algorithms (e.g. item-based neighborhood and matrix factorization). This algorithm also has been used as a baseline for cross-domain recommendation purposes in (CREMONESI; TRIPODI; TURRIN, 2011), which proposed an enhanced version of that user-based algorithm, aiming to make cross-domain CF recommendations under user overlap conditions. The implementation of these algorithms is described in the following.

Algorithm 4. *Item rating estimation with the implementation of the NNUserNgbr algorithm.*

Input: ux *(the user),* un *(the user neighborhood calculated by any traditional similarity metric),* i *(the item, from the target domain)*
Output: er *(estimated rating)*

```
procedure ESTIMATEPREFERENCEINNNUSERNGBR(ux, un, i)
    if un $\neq \varnothing$ then
        p = 0.0
        ts = 0.0
        c = 0
        for each u $\in$ un do
            if u $\neq$ ux then
                if u has preference for i then
                    Get the preference value (pv) of u for i
                    Get the similarity value (sv) between u and ux
                    $p = p + (sv \times pv)$
                    $ts = ts + sv$
                    $c = c + 1$
                end if
            end if
        end for
        if $c > 0$ and $ts \neq 0.0$ then
            $er = p/ts$ return er
        end if
    end if
end procedure
```

end

Algorithm 4 presents the item rating estimation with the implementation of the *NNUserNgbr* algorithm. It is important to notice that the item must be from the target domain, so, the cross-domain recommendation can be seen as a reduced version of a traditional single-domain CF-based recommendation. Besides, the user neighborhood can be calculated by any similarity metric described in Section 3.3.2.1.1.

In addition, we implemented the enhanced version of the user-based algorithm proposed in (CREMONESI; TRIPODI; TURRIN, 2011) (*NNUserNgbr-transClosure*), as mentioned in Section 3.3.2.1.1. This algorithm differs from the *NNUserNgbr* only by the fact that the user's neighborhood calculation is extended. Thus, the Algorithm 4 also represents the item estimation process of the *NNUserNgbr-transClosure* algorithm considering the calculation of the extended user neighborhood, which is presented in Algorithm 5.

Algorithm 5. *User neighborhood calculation for the NNUserNgbr-transClosure algorithm.*

Input: ux *(user),* un *(the user neighborhood calculated by any traditional similarity metric),* mn *(neighborhood size limit)*
Output: unt

```
procedure userNeighborhoodInNNUserNgbr-transClosure(ux, un,mn)
    if un $\neq \varnothing$ then
        Create a data structure (tm) containing a list of users where each user u has a set of similar users with their respective similarity values (sv)
        for each uA $\in$ un do
            if uA $\neq$ ux then
                Get the uA neighborhood (uAn) according to any traditional similarity metric
                for each uB $\in$ uAn do
                    if uB $\not\subseteq$ un and uB $\neq$ ux and uB $\neq$ uA then
                        Get the similarity value (svAB) between uB and uA
                        if uB $\not\subseteq$ tm then
                        Create a data structure of users (su) containing the uA associated with the svAB value
                        Add su in tm for uB
                    else
                        Get su in tm for uB
                        Add uA associated with the svAB value in su
                    end if
                end if
            end for
        end if
    end for
    unt = un
    Get the minimum similarity value (msv) from unt
    Get the less similar user (lsu) from unt
    for each uA ∈ tm do
        s = 0.0
        c = 0
        Get su from uA
        for each uB ∈ su do
            Get svAB from su for uB
            Get the similarity value between uB and ux (svBU) from unt
            $s = s + (svAB \times svBU)$
            $c = c + 1$
        end for
        if c > 0 then
            $ns = s/c$
            Get number of similar users ∈ unt (nsu)
            if mn > nsu then
                Add in unt the uA associated with the respective ns value
                if ns < msv then
                    msv = ns
                    lsu = uA
                end if
            else
                if ns > msv then
                    Add in unt the uA associated with the respective ns value
                    Remove lsu from unt
                    Get the minimum similarity value (msv) from unt
                    Get the less similar user (lsu) from unt
```

```
49:                 end if
50:             end if
51:         end if
52:     end for
53: end if
54: return unt
55: end procedure
```

end

Note that the implemented user neighborhood calculation in Algorithm 5 can be considered as a two-step similarity path between two users, as described in Section 3.3.2.1.1. Besides, a similarity metric is used in this calculation (see line 6 in the *userNeighborhoodIn-NNUserNgbr-transClosure* procedure) once that the "transclosure" process only extends the similarities discovery among users. Finally, the maximum number of 'nearest neighbors' is maintained by the "transclosure" process, which eliminates the smaller user similarities from the user neighborhood (see lines 46 to 48 from the *userNeighborhoodInNNUserNgbr-transClosure* procedure).

4.5 Final Remarks

In this chapter, we presented particular details of an implementation of two proposed CD-CARS algorithms (PreF and PostF) as well as the implementation of two base cross-domain algorithms (*NNUserNgbr* and *NNUserNgbr-transClosure*). In addition, we showed the properties of two CD-CARS datasets ("book-television" and "book-music") with different contextual information (*Temporal*, *Location* and *Companion*) and domains (*Book*, *Television* and *Music*) and how they were obtained.

Also in this chapter, we described the process of selecting relevant contextual attributes and values through a data mining method (*InfoGainAttributeEval* from Weka tool (HALL et al., 2009)). Finally, we presented the implementation of the contextual model through the extension of the Mahout framework (OWEN et al., 2011).

In the next chapter, we describe and discuss experimental evaluations of the implemented algorithms through the CD-CARS datasets.

5 CD-CARS Evaluation

This chapter presents an experimental evaluation of two proposed CD-CARS algorithms in comparison to cross-domain CF-based ones. For that, Section 5.1 describes the evaluation methodology adopted and the algorithms' settings. Section 5.2 describes the evaluation results for each dataset used in the experiments as well as a discussion about their findings. Finally, Section 5.3 presents the final remarks of this chapter.

5.1 Evaluation Methodology

In this section, we describe the adopted methodology to evaluate the proposed algorithms as well as their configurations. Besides, we describe how the statistical significance of the results is verified.

5.1.1 Settings of the Algorithms

Before evaluating the proposed CD-CARS algorithms, we performed a preliminary battery of experiments in the two datasets mentioned in Section 4.1.3 in order to adjust the settings of the base single-domain CF-based algorithm (*NNUserNgbr*) adopted in the CD-CARS evaluation. As mentioned in Section 3.3.2.1.1, that algorithm can also be used to perform cross-domain recommendations, thus, we intend to verify its performance for single-domain and cross-domain scenarios.

In this way, we adjusted the *NNUserNgbr* settings according to several experiments performed in the *Book* domain for each dataset, i.e., performing a single-domain recommendation, once that *Book* is a common domain to the both datasets. We set the 'n' parameter of the *NNUserNgbr* algorithm to "475" and selected the *Euclidian distance* as the similarity metric for it. The same configuration and similarity metric were adopted for another base cross-domain CF-based algorithm (*NNUserNgbr-transClosure*) and for evaluation in other domains (*Television* and *Music*).

As the proposed CD-CARS algorithms, PreF and PostF, can be performed in combination with the base *NNUserNgbr* and *NN-UserNgbr-transClosure* ones, so, the base algorithms were used with the same settings described before. In addition to these settings, we set the PostF threshold (θ) value to "2/3" of the frequency of the most preferred category, and only the categories of items that had good ratings (four or more in a five-star scale) were considered in computation of the frequency of the users' preferred categories (see Section 3.3.1.2). In the PostF algorithm, we also set "0.7" and "0.01", respectively, for the association rule confidence and support levels. This decision was also made by considering preliminary experiments in the *Book* domain as target by observing the PostF performance in the two datasets.

We evaluated the proposed algorithms in comparison to the baseline ones by using *Probabilistic* and *Classification* measures, as described in the following sections.

5.1.2 Predictive Performance

We measured the predictive performance of the algorithms by using the Mean Average Error (MAE) and Root Mean Squared Error (RMSE) metrics (SHANI; GUNAWARDANA, 2011). MAE is a measure of the deviation of recommendations from their actual user-rating values. For each ratings-prediction pair (p_i, q_i) this metric treats the absolute error between them. The MAE is computed by first summing these absolute errors of the 'N' corresponding ratings-prediction pairs and then computing the average. Formally,

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{5.1}$$

Analogously, RMSE computes the square root of the average of the square of all of the error (punishing large errors), by means of this formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (p_i - q_i)^2}{N}} \tag{5.2}$$

These metrics evaluate the performance of a RS by comparing the numerical recommendation scores against the actual user ratings for the user-item pairs in the test dataset. In this way, for a single dataset adopted in the CD-CARS evaluation, we split it into the training and test sets for each target domain (e.g. *Music*) and context under test (e.g. "on sunday with friends").

The training set is composed by 100% of ratings from source domain, 100% of ratings from the target domain in which their contexts are not under test and 90% of ratings from target domain in which their contexts are under test. The test set is composed by 10% of ratings from target domain in which their contexts are under test.

Figure 29 illustrates the process of splitting training and test sets considering the target domain and context under test. It avoids the waste of ratings in the test set for those ones that are not used in the target domain and context under test. That process can be seen as *Hold-out*, according to the partitioning ways of evaluation data described in Section 2.2.6.1.

Finally, for each target domain and context under test, we performed each evaluated algorithm five times in order to verify its standard deviation and apply statistical tests.

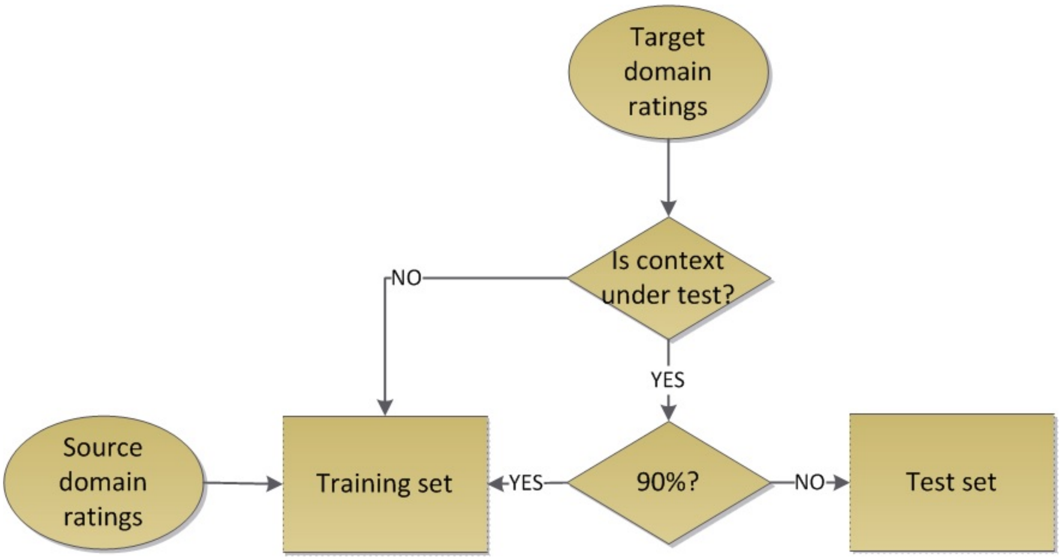


Figure 29 – Splitting training and test sets considering the target domain and context under test.

5.1.3 Classification Performance

Regarding the classification performance of the algorithms, we adopted the F-metric proposed by Cremonesi, Tripodi e Turrin (2011), which is calculated according to the Precision and Recall values used for evaluating top-N recommendations and obtained through a testing methodology described in (CREMONESI; KOREN; TURRIN, 2010).

Analogously to (CREMONESI; KOREN; TURRIN, 2010), we randomly extracted, approximately, 1.4% of the ratings from the original dataset in order to build a probe set, therefore, the training set was composed by 98.6% of the rating from the full dataset. On the other hand, the test set was composed exclusively by the 5-star ratings (the maximum rating value for that evaluation dataset) from the probe set, thus, the not 5-star ratings from the probe set were discarded.

However, we adapted this methodology by considering the target domain and context under test in order to fulfill the training and probe sets in a similar way than is made in the probabilistic evaluation, as illustrated in Figure 29. This avoids the waste of ratings in the probe set for those ones that are not used in the target domain and context under test. Likewise the predictive performance evaluation, the dataset used in the classification performance is partitioned as *Hold-out*, according to the partitioning ways of evaluation data described in Section 2.2.6.1.

After those steps, we trained the algorithms with the training set and for each rating in the test set, given by a user 'u' for an item 'i' from the target domain:

- We predict the ratings for the item 'i' and for 100 additional items[1] from the target

[1] The original method empirically adopts 1000 additional items, but the authors let this number free to

domain randomly chosen from the ones unrated by the user ‘u’; and

- In decreasing order, we sort the list of 101 items[2] according to the predicted ratings. If the item ‘i’ appears in the top-N recommendation list, we have a “hit”.

In this way, *Precision*, *Recall* and *F-metric* values, according to (CREMONESI; KOREN; TURRIN, 2010), are defined as:

$$Recall(N) = \frac{\#hits}{|test\ set|} \tag{5.3}$$

$$Precision(N) = \frac{\#hits}{N * |test\ set|} \tag{5.4}$$

$$F\text{-}metric(N) = \frac{2 * Recall(N) * Precision(N)}{Recall(N) + Precision(N)} \tag{5.5}$$

Likewise the probabilistic evaluation, for each target domain and context under test, we performed each evaluated algorithm five times. The execution of several trials is not specified by the methodology proposed in (CREMONESI; KOREN; TURRIN, 2010), but we believe that as more executions are made the more reliable should be the results.

Finally, in the evaluation results of the algorithms for a particular target domain and user overlap level, we show their classification performances through the F-metric curves by varying the number of top ‘N’ items (from one to twenty[3]). Besides, given that most of the online recommender systems (e.g. Amazon[4], IMDB[5], etc.) usually recommend up to five items in their basic layout (CREMONESI; TRIPODI; TURRIN, 2011), we fixed the top ‘N’ value to “five” to verify the variation of the F-metric value across different user overlap levels (sensitivity evaluation).

5.1.4 Sensitivity Evaluation

As mentioned in Section 2.2.6.3, the performance of a cross-domain RS can be affected by the density of the target domain data and the user overlap between source and target domains. In this way, we evaluated the quality of the cross-domain algorithms by varying the percentage of the user overlap (10%, 50%, and 100%). Section 4.1.3 describes the properties of the datasets adopted in the CD-CARS evaluation regarding the different overlap levels.

be chosen depending on the dataset used.

[2] In the original method, the authors adopted 1001 items.

[3] We have chosen this maximum top ‘N’ value by observing the convergence in the F-metric curves of the algorithms.

[4] http://www.amazon.com

[5] http://www.imdb.com

In addition, we studied the impact of the density of the target domain data in comparison to the density of the source domain data. Thus, we evaluated the quality of the cross-domain algorithms by varying the target domain in both datasets. One of them for more related domains (*Book* and *Television*), where the *Book* domain has more data than the *Television* domain, and another for less related domains (*Book* and *Music*), also where the *Book* domain has more data than the *Music* domain. Therefore, we expect that enriching sparse user preference data in a certain domain by adding user preference data from other domain can significantly improve the quality of cross-domain recommendations (SAHEBI; BRUSILOVSKY, 2013) (FERNÁNDEZ-TOBÍAS et al., 2012).

5.1.5 Statistical Significance Analysis

In order to verify the statistical significance of the evaluation results, we adopted the nonparametric *Mann–Whitney U test* (WINTER; DODOU, 2010), also called *Mann–Whitney–Wilcoxon* (MWW) or *Wilcoxon rank-sum test*. This test verifies the null hypothesis, which states that two samples are statistically the same, against an alternative hypothesis, which especially can determine if a particular population tends to have larger values than the other. In addition, unlike the t-test it does not require the assumption of normal distributions (WINTER; DODOU, 2010).

In this way, we applied the statistical significance tests with a confidence level of 95% for all user overlap levels, contextual dimensions and target domains. These tests were applied with support from "R" software tool (R Core Team, 2015). For the tests of predictive performance, we verified if the errors of the baseline algorithms were greater than the errors of the proposed ones[6]. For the tests of classification performance, we verified if the F-metric values of the proposed algorithms were greater than the F-metric values of the baselines ones[7], considering the F-metric values for N=5. In both cases, the applied Wilcoxon tests were not paired given that the samples were independent among the algorithms.

5.2 Evaluation Results

According to the datasets and evaluation methodology described before, we present and discuss the results of the proposed CD-CARS algorithms in comparison to the baseline cross-domain CF-based algorithms. For that, we divided the experiments into two datasets. Section 5.2.1 shows the evaluation results about two *related* domains (*Book* and *Television*), whereas Section 5.2.2 presents the evaluation results about two less *related* domains (*Book* and *Music*). Finally, we discuss the evaluation results in Section 5.2.3.

[6] wilcox.test(baseline,proposed_algorithm, paired=FALSE,alternative = "greater") by using the "R" software tool.

[7] wilcox.test(proposed_algorithm,baseline, paired=FALSE,alternative = "greater") by using the "R" software tool.

5.2.1 Book-Television Results

As mentioned before, we evaluated the quality of the cross-domain algorithms by varying the target domain for each dataset in order to study the impact of the density of the target domain data in comparison to the density of the source domain data. Thus, the following sections present the results considering a different domain as a target.

5.2.1.1 Television as Target Domain

According to the contextual dimensions present in the datasets, we describe the evaluation results for each contextual dimension in the following sections. In addition, we show the results for a combination of contextual dimensions in Section 5.2.1.1.4.

5.2.1.1.1 Temporal Dimension

Table 19 reports the overall predictive performance of the recommender algorithms, considering all contextual values from the *Temporal* dimension and different user overlap levels for the *Television* domain as target. The rows 1 and 2 from the table show the *NNUserNgbr* predictive performance when it is applied in the single-domain and cross-domain recommendations. As it can be seen, the simple addition of user ratings from other domain (*Book*), by using the same algorithm for cross-domain recommendation, improved the recommendation performance in, approximately, 8–21% (MAE) and 2–10% (RMSE) depending on the user overlap level.

Table 19 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Temporal dimension (source domain: Book, and target domain: Television).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.721 ± 0.024	1.020 ± 0.048	0.454 ± 0.008	0.759 ± 0.020	0.412 ± 0.006	0.734 ± 0.012
NNUserNgbr (cross-domain)	0.598 ± 0.022	0.922 ± 0.044	0.417 ± 0.007	0.742 ± 0.018	0.324 ± 0.005	0.661 ± 0.010
NNUserNgbr-transClosure	0.251 ± 0.014	0.548 ± 0.028	0.256 ± 0.005	0.556 ± 0.012	0.217 ± 0.002	0.531 ± 0.004
PreF with NNUserNgbr-transClosure	**0.129 ± 0.020**	**0.382 ± 0.057**	**0.132 ± 0.006**	**0.374 ± 0.016**	**0.151 ± 0.003**	**0.413 ± 0.007**
PostF with NNUserNgbr-transClosure	0.216 ± 0.012	0.486 ± 0.024	0.210 ± 0.003	0.469 ± 0.005	0.173 ± 0.004	0.445 ± 0.006

In addition, Table 19 presents the overall performance of the PreF and PostF algorithms, besides the base *NNUserNgbr-transClosure* algorithm, which outperformed the *NNUserNgbr* algorithm (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 33–58% (MAE) and 19–40% (RMSE) depending on the user overlap levels.

As it can be seen from table, the PreF predictive performance was better than the *NNUserNgbr-transClosure* and the PostF algorithms in all user overlap levels. The improvement achieved by the PreF algorithm in comparison to the *NNUserNgbr-transClosure* one varied in, approximately, 30–48% (MAE) and 22–32% (RMSE) depending on the user overlap level. The PostF predictive performance was also better than the *NNUserNgbr-transClosure* algorithm in all user overlap levels, however, its improvement was smaller than the achieved by the PreF algorithm. Figure 30 illustrates the predictive performance (MAE) of the proposed algorithms over different user overlap levels.

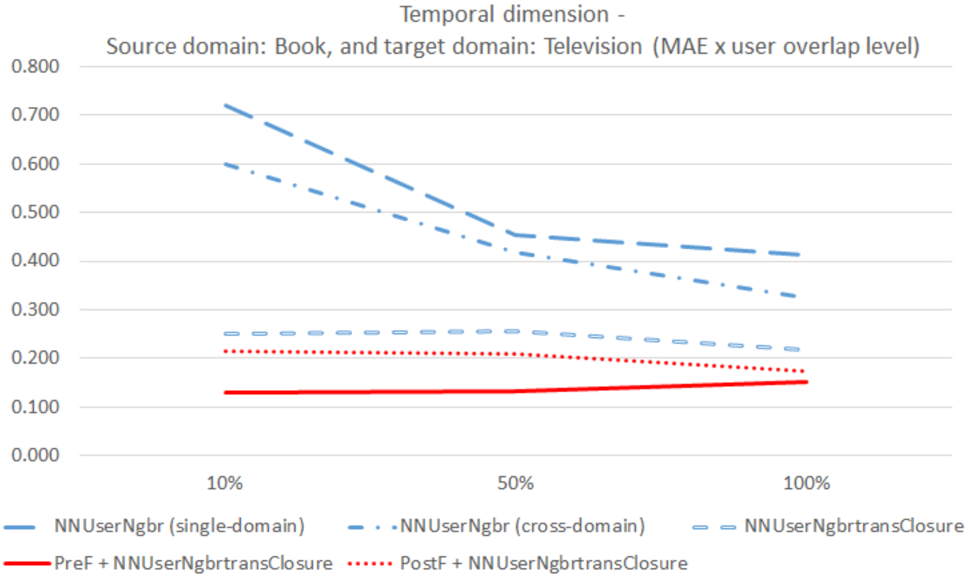
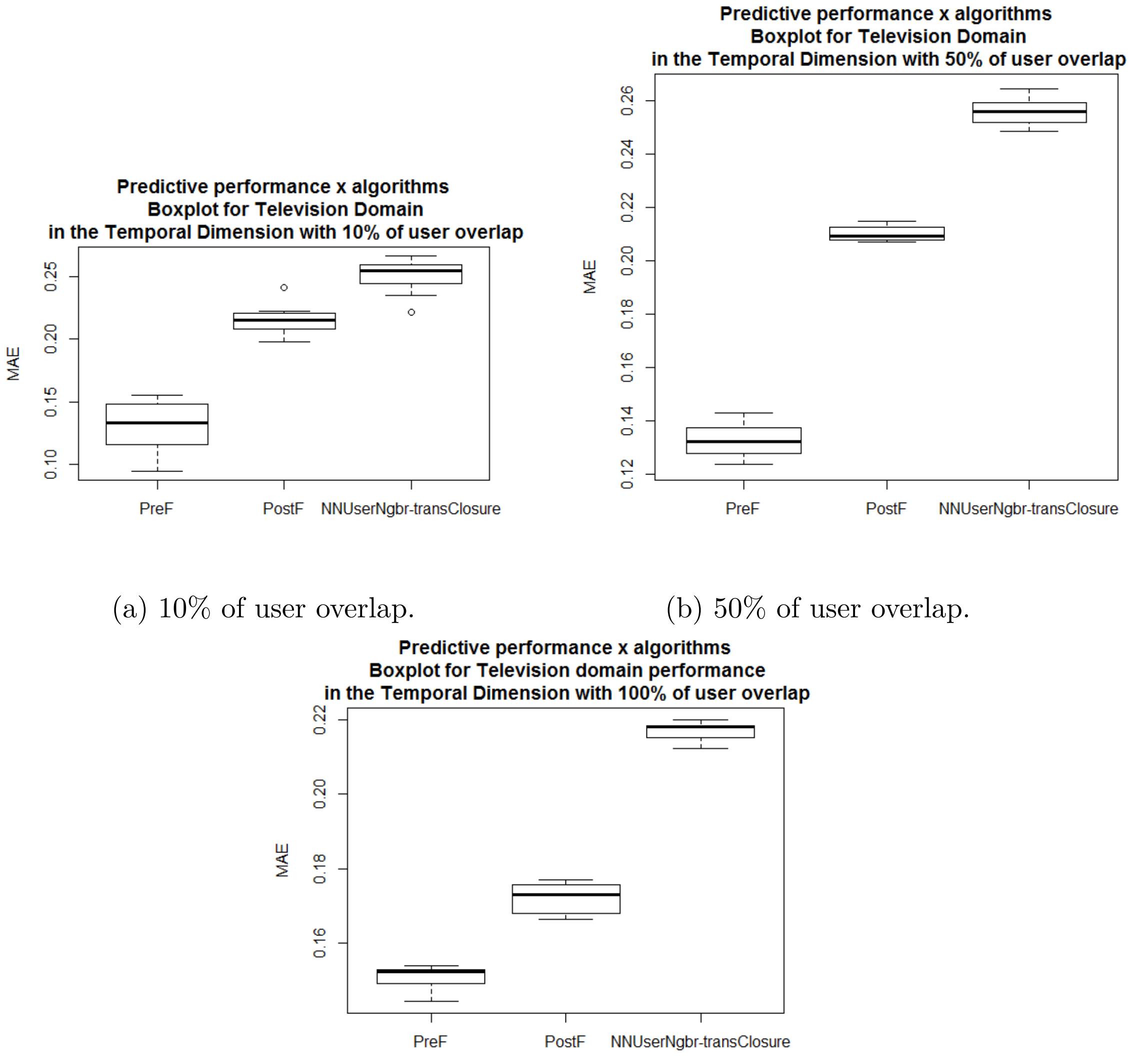


Figure 30 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the temporal dimension (source domain: book, and target domain: television).

The statistical significance tests verified that both the PreF and PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* baseline algorithm for all user overlap levels[8]. Figure 31a, Figure 31b and Figure 31c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

Regarding the classification performance, Figures 32a, 32b, and 32c present the results of the F-metric at different N values (between one and twenty), respectively, in 10%, 50%, and 100% of user overlap level for the *Television* domain as target, considering the *Temporal* dimension. As it can be seen, in all user overlap levels and top 'N' values,

[8] p-value=0.005413 and W=100 for all tests, except in the comparison between the baseline and the PostF algorithms when the user overlap level was 10%, where the W=97 and the p-value=0.037



(a) 10% of user overlap.

(b) 50% of user overlap.

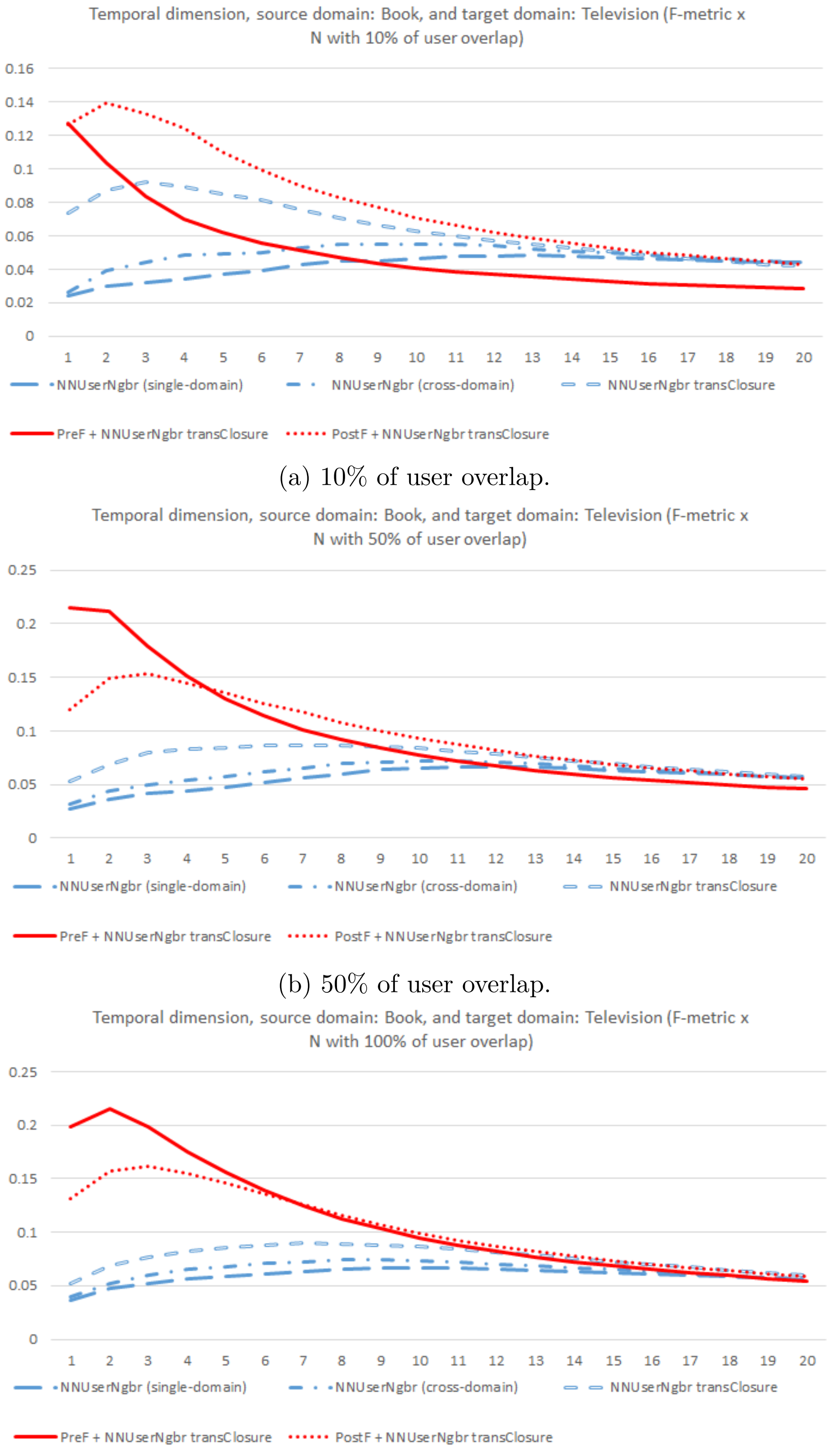
(c) 100% of user overlap.

Figure 31 – Overall prediction performance (MAE) boxplots for television domain in the temporal dimension with different user overlap levels (source domain: book).

the PostF classification performance was better or similar than the baseline algorithms. The PreF performance was better than the baseline ones for low top ‘N’ values when there were 10% and 50% of user overlap levels, and for any top ‘N’ value when there was 100% of user overlap. In addition, the PostF classification performance was better or similar than the PreF one.

Figure 33 shows the variation of the F-metric value in different user overlap levels by fixing the top ‘N’ value to five. The statistical significance tests verified that the PostF F-metric values were greater than the *NNUserNgbr-transClosure* baseline algorithm for all user overlap levels[9]. Besides, the PreF F-metric values were greater than the

[9] p-value=0.005413 and W=99 for all tests



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 32 – F-metric performance x top 'N' items for the television domain in the temporal dimension with different user overlap levels (source domain: book).

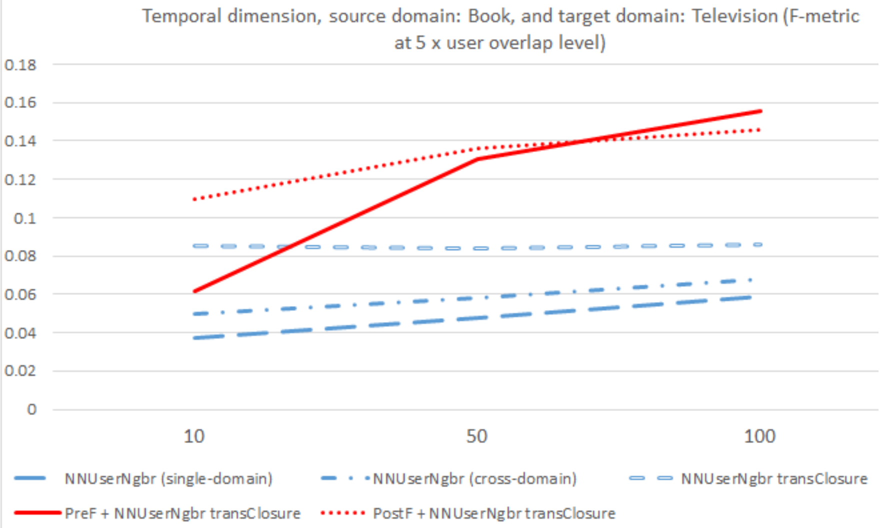


Figure 33 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the temporal dimension (target domain: television, and source: book).

NNUserNgbr-transClosure algorithm for 50% and 100% user overlap levels[10]. For 10% of user overlap, the *NNUserNgbr-transClosure* F-metric value was greater than the PreF algorithm[11].

5.2.1.1.2 Location Dimension

Table 20 reports the overall predictive performance of the recommender algorithms, considering all contextual values from the *Location* dimension and different user overlap levels for the *Television* domain as target. As it can be seen from the table, the addition of user ratings from other domain (*Book*), by using the same algorithm for cross-domain recommendation (corresponding to the two first rows of the table), improved the predictive performance in, approximately, 20–36% (MAE) and 8–20% (RMSE) depending on the user overlap level.

Also, Table 20 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 21–58% (MAE) and 11–36% (RMSE) depending on the user overlap levels.

As it can be seen from table, the PostF predictive performance was better than the *NNUserNgbr-transClosure* algorithm in all user overlap levels, with an improvement that varied in, approximately, 5–16% (MAE) and 4–14% (RMSE) depending on the user overlap level. In addition, if we consider the high standard deviation (std) of the PreF

[10] p-value=0.005413 and W=99 for all tests
[11] p-value=0.005413 and W=100

Table 20 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Location dimension (source domain: Book, and target domain: Television).

Algorithm	10% overlap		50% overlap		Full overlap	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.721 ± 0.024	1.020 ± 0.048	0.454 ± 0.008	0.759 ± 0.020	0.412 ± 0.006	0.734 ± 0.012
NNUserNgbr (cross-domain)	0.573 ± 0.022	0.865 ± 0.044	0.363 ± 0.007	0.691 ± 0.018	0.261 ± 0.005	0.582 ± 0.010
NNUserNgbr-transClosure	0.240 ± 0.017	0.550 ± 0.039	0.247 ± 0.006	0.545 ± 0.013	0.206 ± 0.003	0.513 ± 0.006
PreF with NNUserNgbr-transClosure	0.305 ± 0.385	0.433 ± 0.541	0.212 ± 0.045	0.588 ± 0.123	0.242 ± 0.027	0.602 ± 0.064
PostF with NNUserNgbr-transClosure	**0.200 ± 0.010**	**0.468 ± 0.028**	**0.233 ± 0.004**	**0.519 ± 0.011**	**0.194 ± 0.005**	**0.484 ± 0.011**

algorithm showed in Table 20, then we can say that the PostF outperformed the PreF algorithm for all user overlap levels.

Figure 34 (MAE) and Figure 35 (RMSE) illustrate the predictive performance of the proposed algorithms over different user overlap levels. Note that for this case we showed the figures for both predictive metrics, since we observed a difference in the PreF performance depending on the predictive metric used in the evaluation.

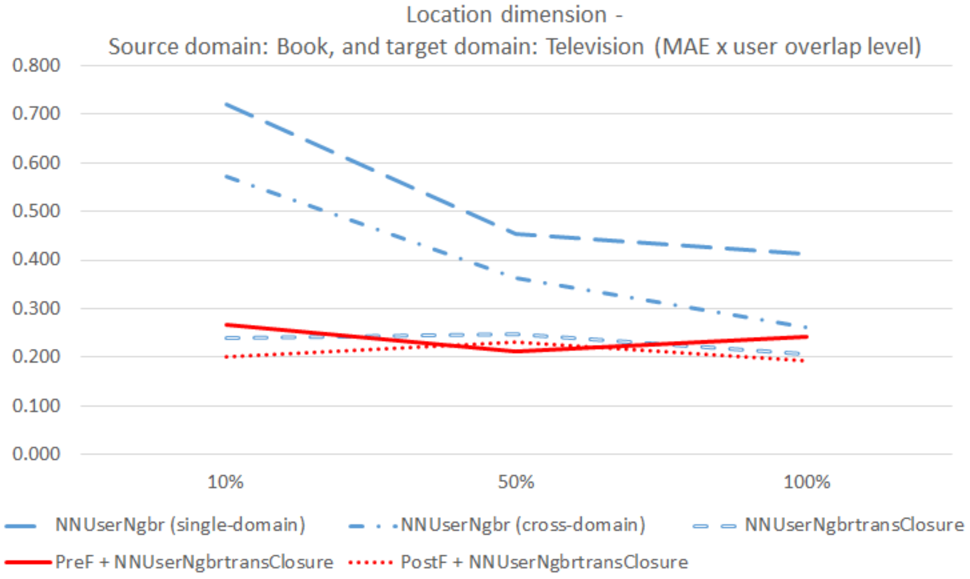


Figure 34 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the location dimension (source domain: book, and target domain: television).

The statistical significance tests verified that the PostF predictive errors (MAE)

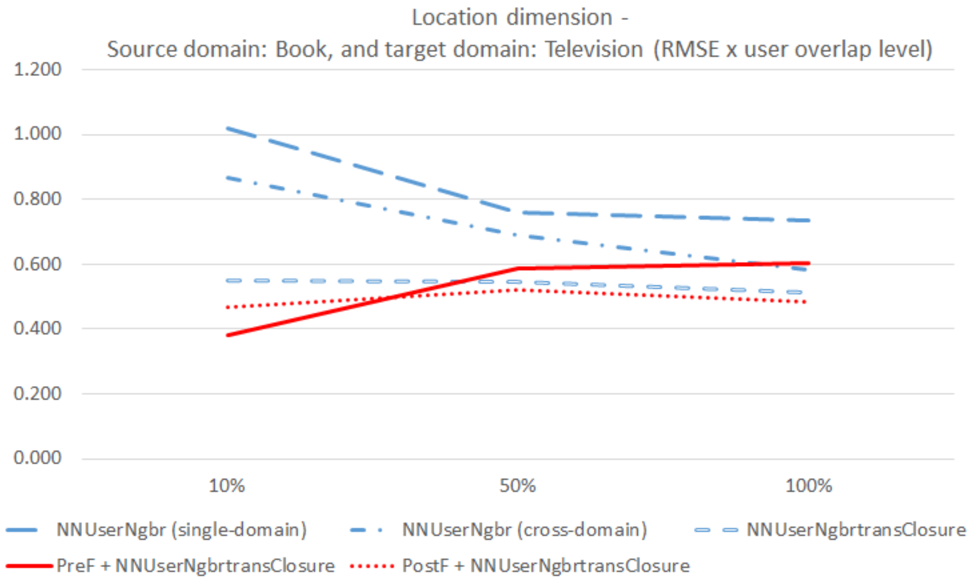


Figure 35 – Overall prediction error (RMSE) for cross-domain algorithms by varying user overlap level in the location dimension (source domain: book, and target domain: television).

were less than the *NNUserNgbr-transClosure* baseline algorithm for all user overlap levels[12]. On the other hand, the PreF predictive errors (MAE) were statistically similar to the *NNUserNgbr-transClosure* algorithm for 10% and 50% of user overlap levels[13]. For 100% of user overlap, the *NNUserNgbr-transClosure* predictive error was statistically less than the PreF one[14]. Figure 36a, Figure 36b and Figure 36c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

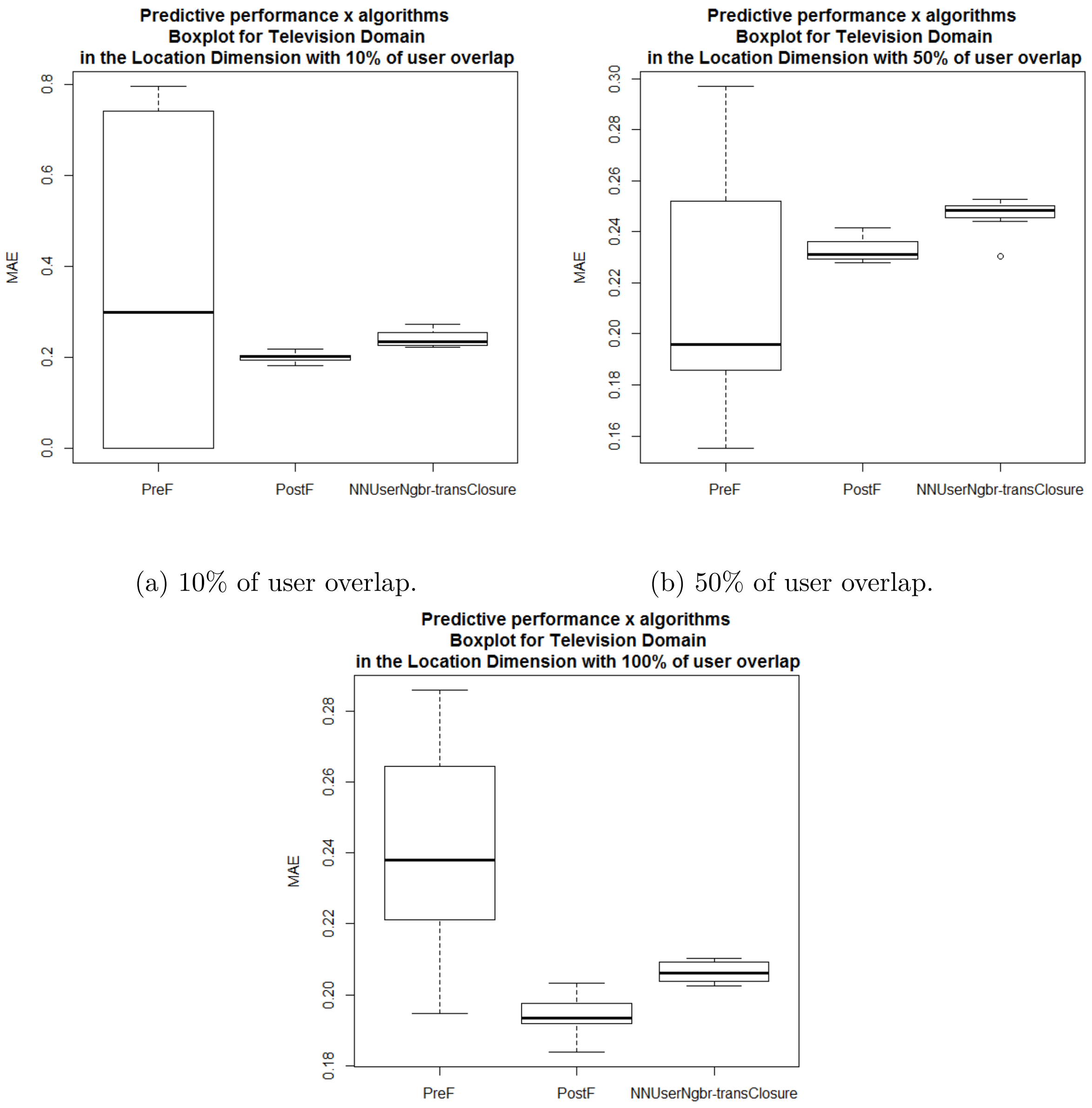
With respect to the classification performance, Figures 37a, 37b, and 37c present the results of the F-metric at different top ‘N’ values (between one and twenty), respectively, in 10%, 50%, and 100% of user overlap levels for the *Television* domain as target, considering the *Location* dimension. As it can be seen, in all user overlap levels considering ‘N’ up to five, the PostF classification performance was better than the baseline algorithms, whereas for 50% and 100% of user overlap, the PostF outperformed them for ‘N’ up to ten. On the other hand, the PreF classification performance was worse than all other algorithms in all user overlap levels and ‘N’ values.

Figure 38 shows the variation of the F-metric value in different user overlap levels by fixing the top ‘N’ value to five. The statistical significance tests verified that the PostF F-metric values were greater than the *NNUserNgbr-transClosure* baseline algorithm for

[12] For 10% of user overlap, W=100 and p-value=0.005413. For 50% of user overlap, W=95 and p-value=0.0001028. Finally, for 100% of user overlap, W=98 and p-value=0.02165

[13] For 10% of user overlap, W=30 and p-value=0.09391. For 50% of user overlap, W=71 and p-value=0.0615

[14] W=90 and p-value=0.0007523



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

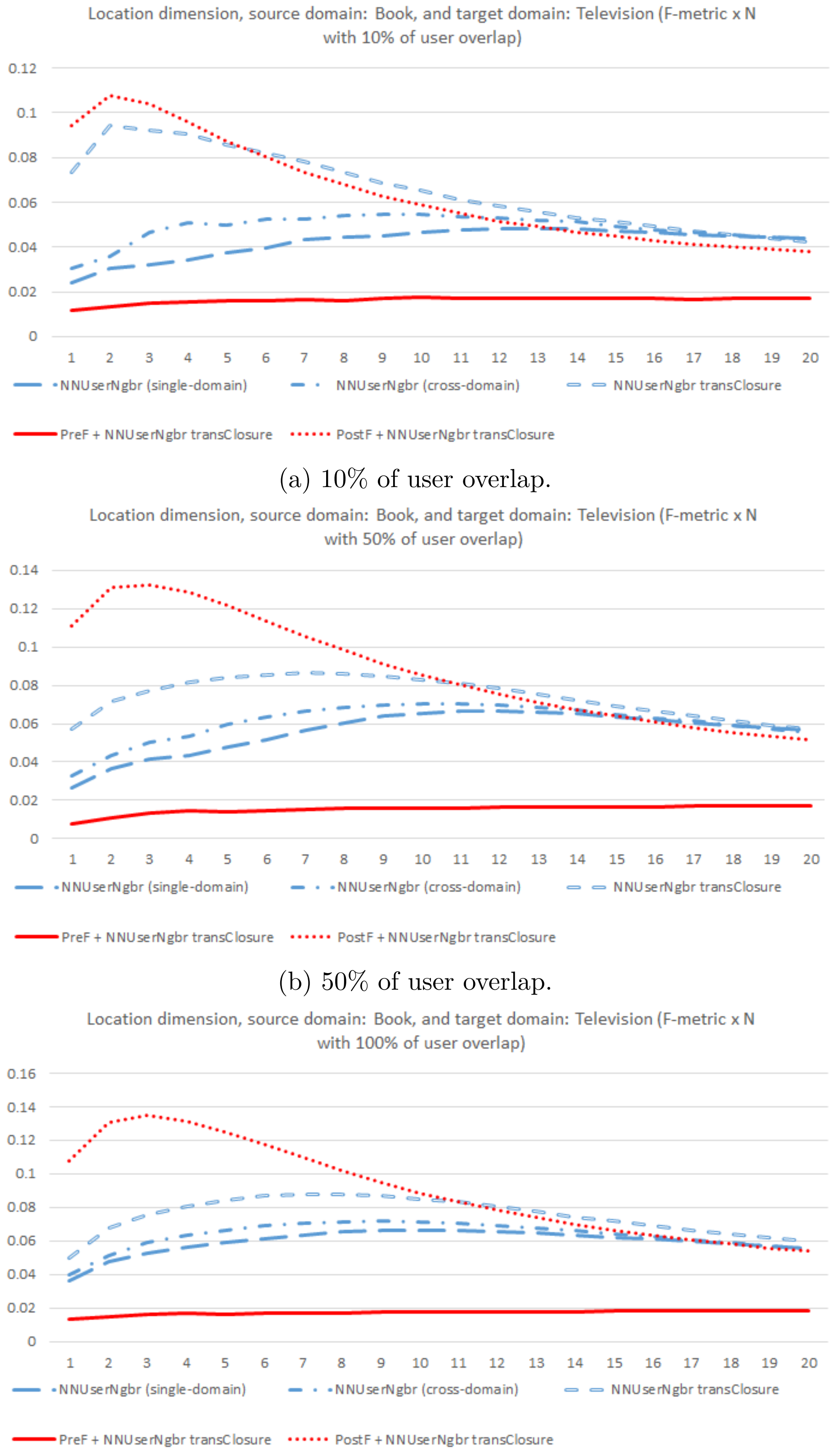
Figure 36 – Overall prediction performance (MAE) boxplots for television domain in the location dimension with different user overlap levels (source domain: book).

50% and 100% of user overlap levels[15], whereas their performances were statistically similar for 10% of user overlap[16]. Besides, the *NNUserNgbr-transClosure* F-metric values were greater than the PreF algorithm for all user overlap levels[17].

[15] p-value=0.005413 and W=99 for all tests

[16] p-value=0.2 and W=71

[17] p-value=0.005413 and W=99 for all tests



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 37 – F-metric performance x top 'N' items for the television domain in the location dimension with different user overlap levels (source domain: book).

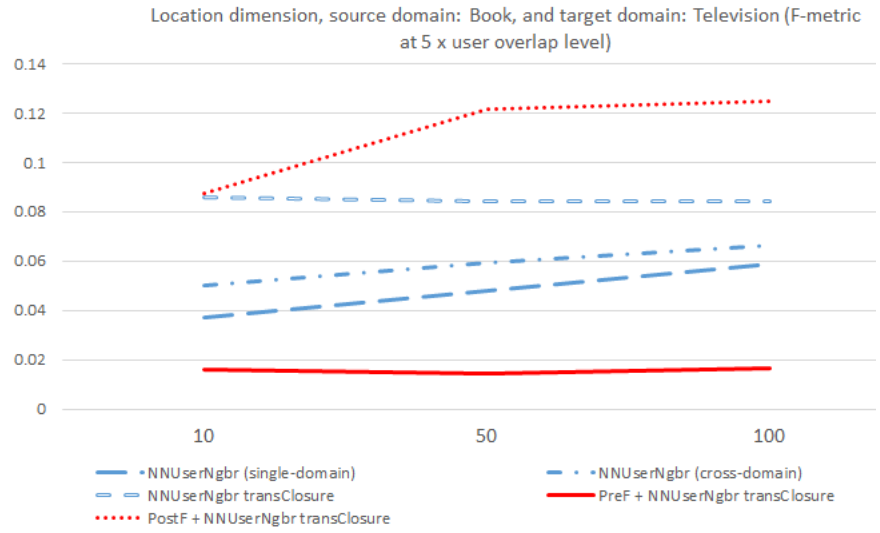


Figure 38 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the location dimension (target domain: television, and source: book).

5.2.1.1.3 Companion Dimension

Table 21 shows the overall predictive performance of the recommender algorithms, considering all contextual values from the *Companion* dimension and different user overlap levels for the *Television* domain as target. As it can be seen from the table, the addition of user ratings from other domain (*Book*), by using the same algorithm for cross-domain recommendation (corresponding to the two first rows of the table), improved the predictive performance in, approximately, 16–28% (MAE) and 10–14% (RMSE) depending on the user overlap level.

Table 21 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Companion dimension (source domain: Book, and target domain: Television).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.721 ± 0.024	1.020 ± 0.048	0.454 ± 0.008	0.759 ± 0.020	0.412 ± 0.006	0.734 ± 0.012
NNUserNgbr (cross-domain)	0.583 ± 0.022	0.881 ± 0.044	0.380 ± 0.007	0.680 ± 0.018	0.295 ± 0.005	0.625 ± 0.010
NNUserNgbr-transClosure	0.249 ± 0.029	0.539 ± 0.048	0.279 ± 0.011	0.587 ± 0.024	0.246 ± 0.003	0.574 ± 0.006
PreF with NNUserNgbr-transClosure	0.931 ± 0.111	1.308 ± 0.158	0.858 ± 0.021	1.204 ± 0.026	0.842 ± 0.010	1.169 ± 0.012
PostF with NNUserNgbr-transClosure	**0.232** ± **0.035**	**0.492** ± **0.065**	**0.256** ± **0.013**	**0.540** ± **0.025**	**0.221** ± **0.006**	**0.523** ± **0.010**

Also, Table 21 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 16–57% (MAE) and 8–38% (RMSE) depending on the user overlap levels.

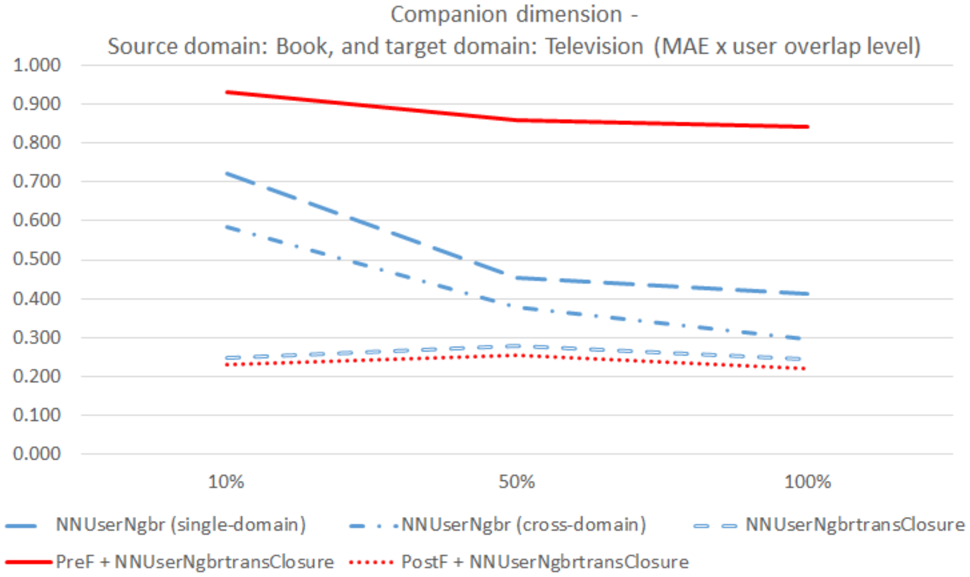


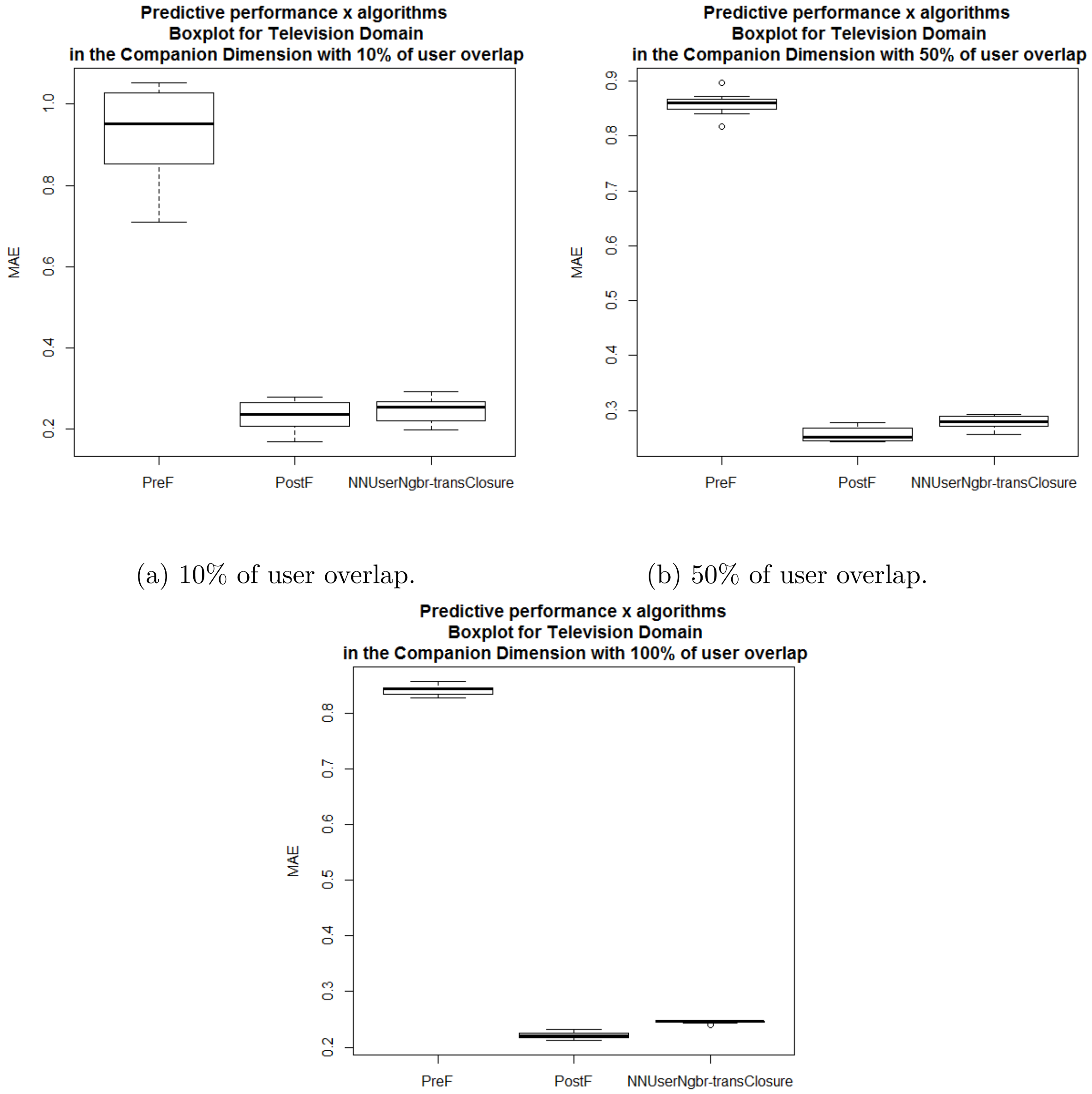
Figure 39 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the companion dimension (source domain: book, and target domain: television).

As it can be seen from table, the PostF predictive performance was better than the *NNUserNgbr-transClosure* algorithm in all user overlap levels, with an improvement that varied in, approximately, 6–10% (MAE) and 8–9% (RMSE) depending on the user overlap level. In addition, the predictive performance of the PreF algorithm was worse than all other algorithms in the *Companion* dimension considering all user overlap levels, as showed in Table 21. Figure 39 illustrates the predictive performance (MAE) of the proposed algorithms over different user overlap levels.

Except when the user overlap level was 10%, the statistical significance tests verified that the PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* algorithm for all other user overlap levels[18]. On the other hand, the *NNUserNgbr-transClosure* predictive errors (MAE) were statistically less than the PreF ones for all user overlap levels[19]. Figure 40a, Figure 40b and Figure 40c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

[18] For 10% of user overlap, W=65 and p-value=0.1399. For 50% of user overlap, W=89 and p-value=0.001045. Finally, for 100% of user overlap, W=100 and p-value=0.005413

[19] W=89 and p-value=0.001045 for all tests



(a) 10% of user overlap.

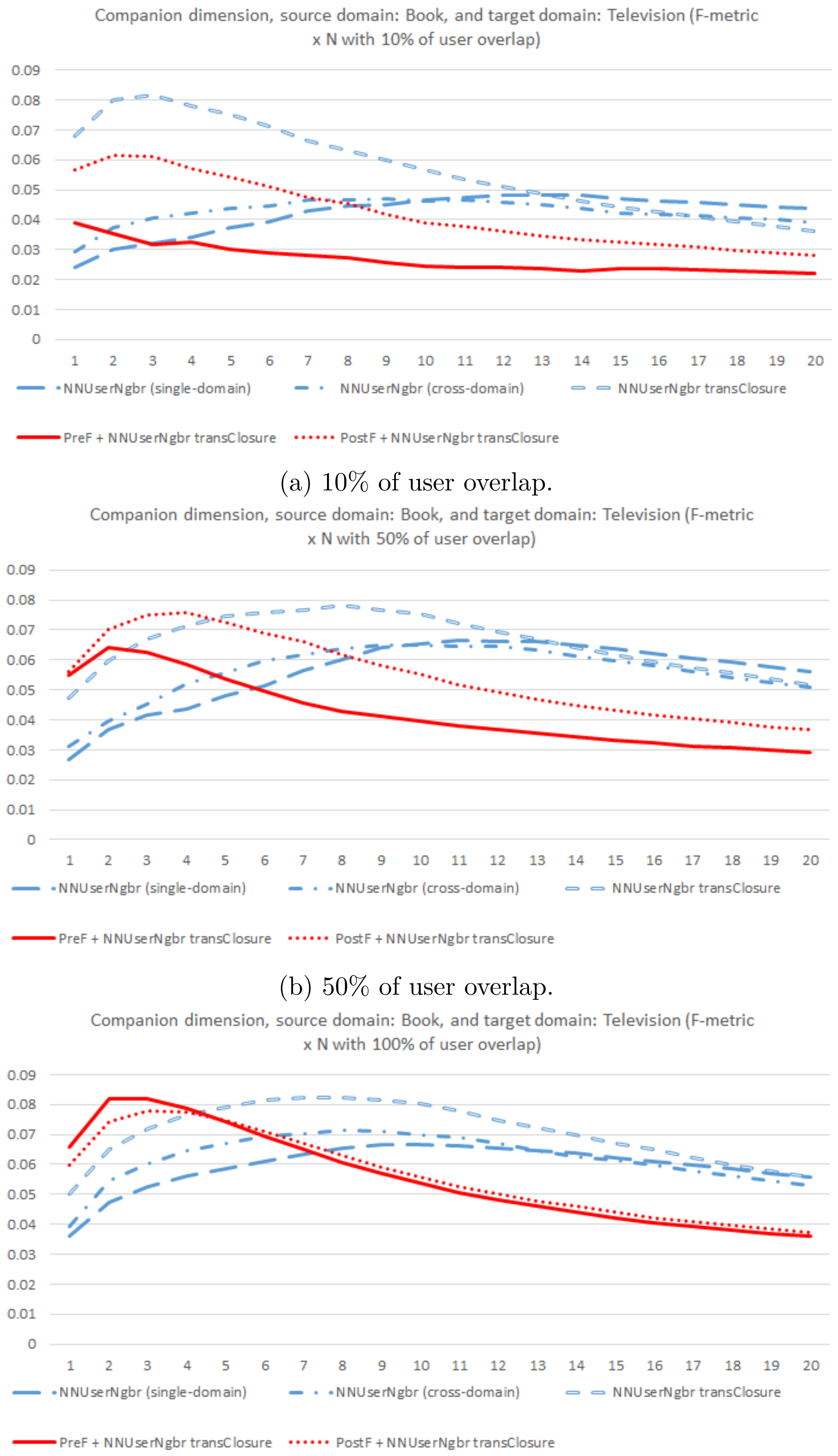
(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 40 – Overall prediction performance (MAE) boxplots for television domain in the companion dimension with different user overlap levels (source domain: book).

Figures 41a, 41b, and 41c present the results of the F-metric at different top ‘N’ values (between one and twenty), respectively, in 10%, 50%, and 100% of user overlap level for the *Television* domain as target, considering the *Companion* dimension. As it can be seen, the proposed algorithms only outperformed the baseline for 50% and 100% of user overlap with low values of top ‘N’.

Figure 42 shows the variation of the F-metric value in different user overlap levels by fixing the top ‘N’ value to five. The statistical significance tests verified that the *NNUserNgbr-transClosure* F-metric values were greater than the PostF ones for 10% and



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 41 – F-metric performance x top ‘N’ items for the television domain in the companion dimension with different user overlap levels (source domain: book).

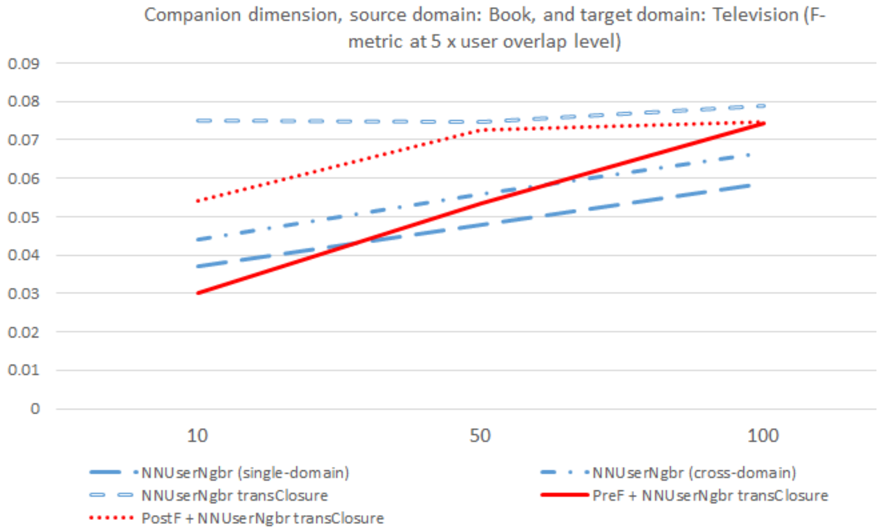


Figure 42 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the companion dimension (target domain: television, and source: book).

50% of user overlap levels[20], whereas their performances were statistically similar for 100% of user overlap[21]. Besides, the *NNUserNgbr-transClosure* F-metric values were greater than the PreF algorithm for all user overlap levels[22].

5.2.1.1.4 Combining Contextual Dimensions

In the previous sections, we presented the evaluation results regarding the contextual dimensions separately. In this section, we report the results for a combination of two contextual dimensions considering the same evaluation metrics and methodology described before. As mentioned in Section 4.1.2, an important aspect of context-aware recommender systems is to determine the relevance of contextual dimensions, attributes (or even values) in order to select only the contextual features that actually matter for evaluation (or recommendation) purposes.

We have seen in Section 4.1.2 that two of the contextual dimensions (*Temporal* and *Location*) provide a greater information gain than the *Companion* dimension, which is confirmed by the results presented in the previous sections. In this way, we evaluated the combination of those two contextual dimensions (*Temporal* and *Location*), aiming to verify its performance in comparison to their own performances evaluated separately.

Table 22 reports the overall predictive performance of the recommender algorithms, considering *all contextual value combinations* from the *Temporal* and *Location* dimensions with different user overlap levels for the *Television* domain as target. As it was observed

[20] p-value=0.005413 and W=99 for all tests
[21] p-value=0.75 and W=60
[22] p-value=0.003913 and W=100 for all tests

in the previous sections, the addition of user ratings from the *Book* domain also improved the predictive performance of the *NNUserNgbr* in, approximately, 20–37% (MAE) and 9–20% (RMSE) depending on the user overlap level (rows 1 and 2 from the table).

Table 22 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual value combinations from the temporal and location dimensions (source domain: Book, and target domain: Television).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.721 ± 0.024	1.020 ± 0.048	0.454 ± 0.008	0.759 ± 0.020	0.412 ± 0.006	0.734 ± 0.012
NNUserNgbr (cross-domain)	0.571 ± 0.022	0.860 ± 0.044	0.360 ± 0.007	0.689 ± 0.018	0.259 ± 0.005	0.580 ± 0.010
NNUserNgbr-transClosure	**0.224** ± **0.017**	**0.482** ± **0.039**	0.250 ± 0.006	0.552 ± 0.013	0.207 ± 0.003	0.515 ± 0.006
PreF with NNUserNgbr-transClosure	0.396 ± 0.390	0.761 ± 0.560	0.720 ± 0.045	1.050 ± 0.123	0.333 ± 0.027	0.739 ± 0.064
PostF with NNUserNgbr-transClosure	0.226 ± 0.010	0.503 ± 0.028	**0.190** ± **0.004**	**0.433** ± **0.011**	**0.161** ± **0.005**	**0.437** ± **0.011**

Also, Table 22 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 20–60% (MAE) and 11–43% (RMSE) depending on the user overlap levels.

As it can be seen from table, except when the user overlap level was 10%, the PostF predictive performance was better than the *NNUserNgbr-transClosure* algorithm in all other user overlap levels, with an improvement that varied in, approximately, 22–24% (MAE) and 15–21% (RMSE) depending on the user overlap level. Despite the *NNUserNgbr-transClosure* algorithm have outperformed the PostF when the user overlap level was 10%, they had a similar performance, separated only by their standard deviations. Figure 43 illustrates the predictive performance (MAE) of the proposed algorithms over different user overlap levels.

The statistical significance tests verified that the PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* algorithm for the 50% and 100% user overlap levels[23]. When the user overlap level was 10%, the applied tests could not determine any statistical difference between the *NNUserNgbr-transClosure* and PostF predictive

[23] In both cases, with W=100 and p-value=0.005413

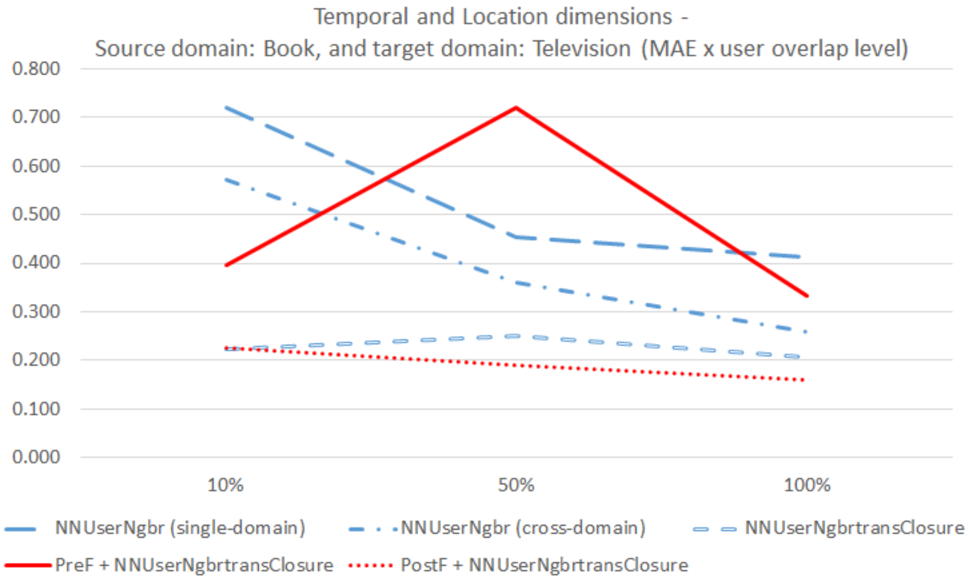


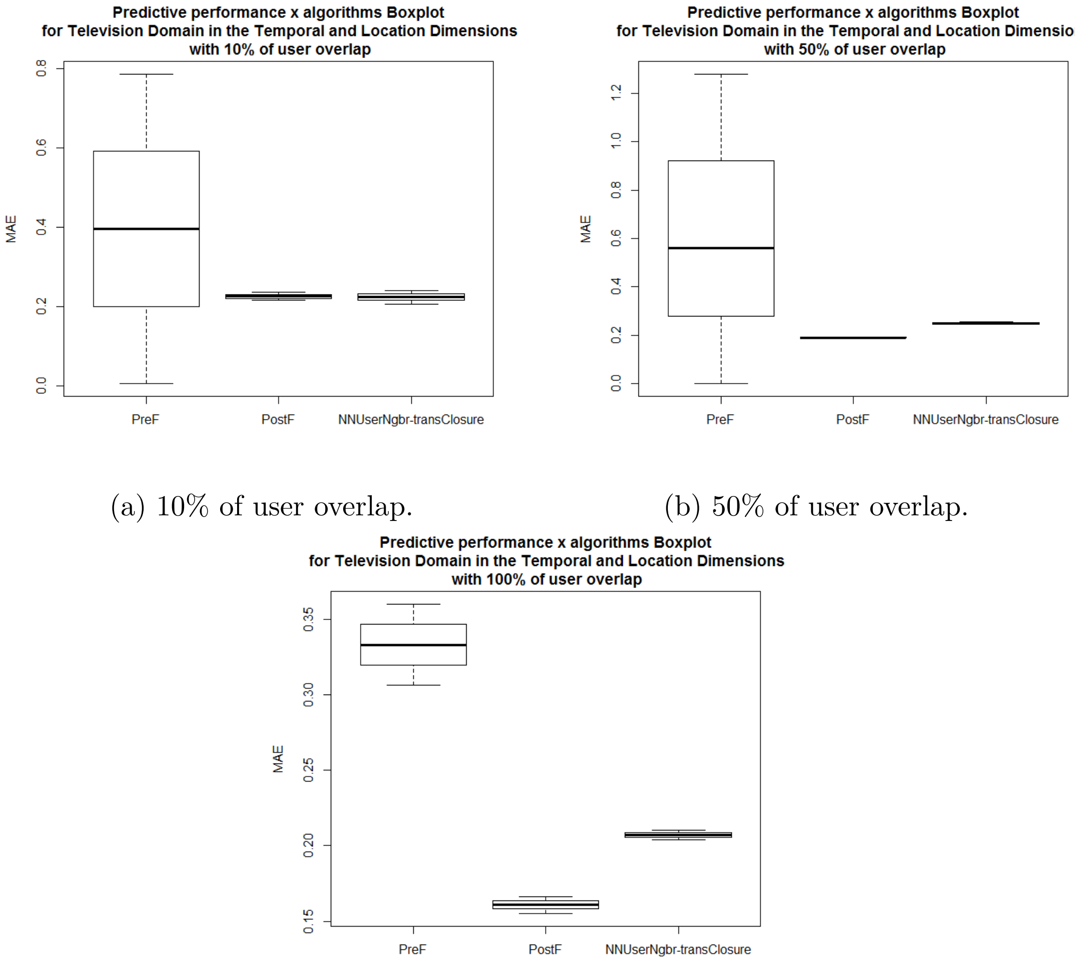
Figure 43 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the temporal and location dimensions (source domain: book, and target domain: television).

errors (MAE)[24]. On the other hand, the applied tests verified that the *NNUserNgbr-transClosure* predictive errors (MAE) were less than the PreF ones for all user overlap levels[25]. Figure 44a, Figure 44b and Figure 44c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

Taking into account the classification performance, Figures 45a, 45b, and 45c present the results of the F-metric at different top 'N' values, respectively, in 10%, 50%, and 100% of user overlap level for the *Television* domain as target, considering the combination between the *Temporal* and *Location* dimensions. For all user overlap levels and top 'N' values, the PostF classification performance was better or similar than the baseline algorithms, while the PreF classification performance was worse than all other algorithms.

Figure 46 shows the variation of the F-metric value in different user overlap levels by fixing the top 'N' value to five. The statistical significance tests verified that the PostF F-metric values were greater than the *NNUserNgbr-transClosure* baseline algorithm for all user overlap levels[26]. On the other hand, the applied tests also verified that *NNUserNgbr-transClosure* F-metric values were greater than the PreF ones for all user overlap levels[27].

[24] W=41 and p-value=0.5
[25] For all cases, with W=100 and p-value=0.003914
[26] p-value=0.005413 and W=97 for all tests
[27] p-value=0.003968 and W=100 for all tests



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 44 – Overall prediction performance (MAE) boxplots for television domain in the temporal and location dimensions with different user overlap levels (source domain: book).

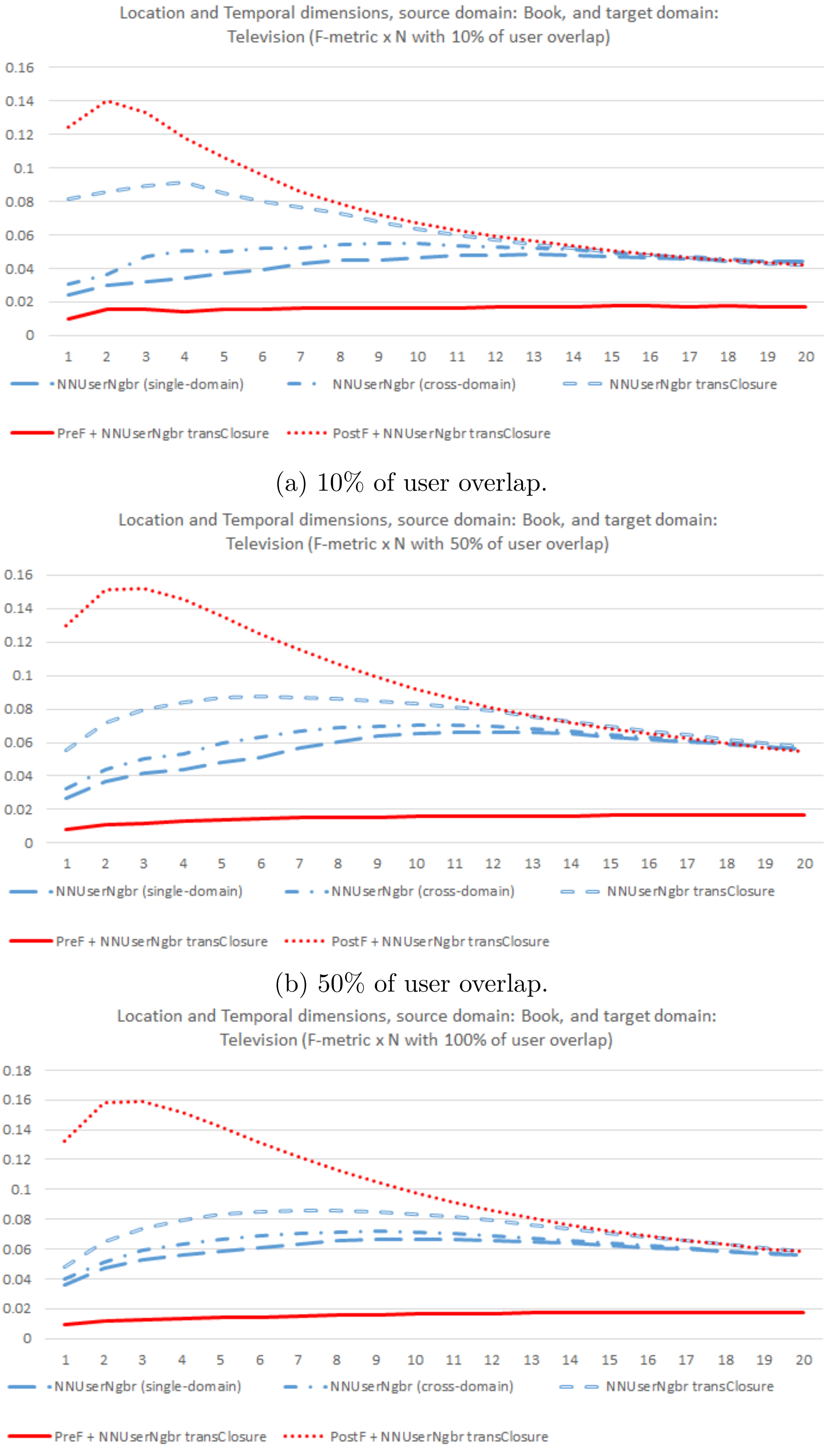
5.2.1.2 Book as Target Domain

In the Section 5.2.1.1, we presented the results for the *Television* target domain, which had fewer ratings in the cross-domain dataset in comparison to *Book* source domain (as described in Section 4.1.3.1). In this section, we present the results when *Book* is the target domain and *Television* is the source domain.

According to the contextual dimensions present in Section 4.1.3.1, we describe the evaluation results for each contextual dimension in the following sections. In addition, we show the results for a combination of contextual dimensions in Section 5.2.1.2.4.

5.2.1.2.1 Temporal Dimension

Table 23 reports the overall predictive performance of the recommender algorithms, considering all contextual values from the *Temporal* dimension and different user overlap



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 45 – F-metric performance x top 'N' items for the television domain in the temporal and location dimensions with different user overlap levels (source domain: book).

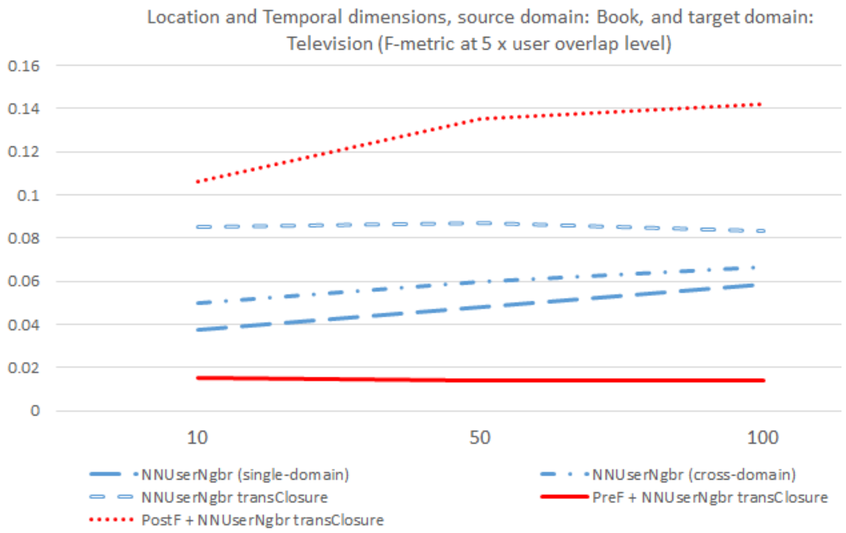


Figure 46 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the temporal and location dimensions (target domain: television, and source: book).

Table 23 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Temporal dimension (source domain: Television, and target domain: Book).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.643 ± 0.022	0.963 ± 0.040	0.500 ± 0.008	0.794 ± 0.020	0.401 ± 0.006	0.719 ± 0.010
NNUserNgbr (cross-domain)	0.497 ± 0.020	0.836 ± 0.042	0.367 ± 0.007	0.674 ± 0.018	0.297 ± 0.005	0.620 ± 0.008
NNUserNgbr-transClosure	0.133 ± 0.007	0.361 ± 0.021	0.180 ± 0.002	0.437 ± 0.006	0.181 ± 0.002	0.459 ± 0.005
PreF with NNUserNgbr-transClosure	0.122 ± 0.009	0.340 ± 0.009	**0.116 ± 0.002**	**0.340 ± 0.006**	**0.114 ± 0.001**	**0.330 ± 0.005**
PostF with NNUserNgbr-transClosure	**0.120 ± 0.008**	**0.319 ± 0.019**	0.153 ± 0.003	0.375 ± 0.009	0.155 ± 0.003	0.399 ± 0.007

levels for the *Book* domain as target. The rows 1 and 2 from the table show the *NNUserNgbr* predictive performance when it is applied in the single-domain and cross-domain recommendations. As it can be seen, the simple addition of user ratings from other domain (*Television*), by using the same algorithm for cross-domain recommendation, improved the recommendation performance in, approximately, 22–26% (MAE) and 13–15% (RMSE) depending on the user overlap level.

In addition, Table 23 presents the overall performance of the PreF and PostF algorithms, besides the base *NNUserNgbr-transClosure* algorithm, which outperformed the *NNUserNgbr* algorithm (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 38–73% (MAE) and 25–56% (RMSE) depending on the user overlap levels.

As it can be seen from the table, the PreF predictive performance was better than the *NNUserNgbr-transClosure* for all user overlap levels, and better than the PostF algorithm for 50% and 100% of user overlap levels. The improvement achieved by the PreF algorithm in comparison to the *NNUserNgbr-transClosure* one varied in, approximately, 8–37% (MAE) and 5–28% (RMSE) depending on the user overlap level. The PostF predictive performance was better than the PreF algorithm for 10% of user overlap, and better than the *NNUserNgbr-transClosure* for all user overlap levels. Figure 47 illustrates the predictive performance (MAE) of the proposed algorithms over different user overlap levels.

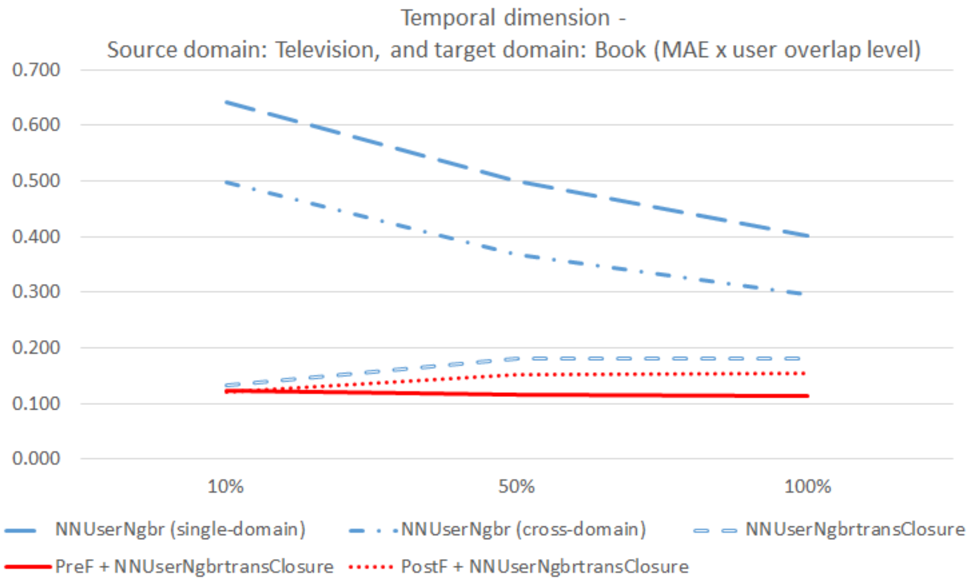
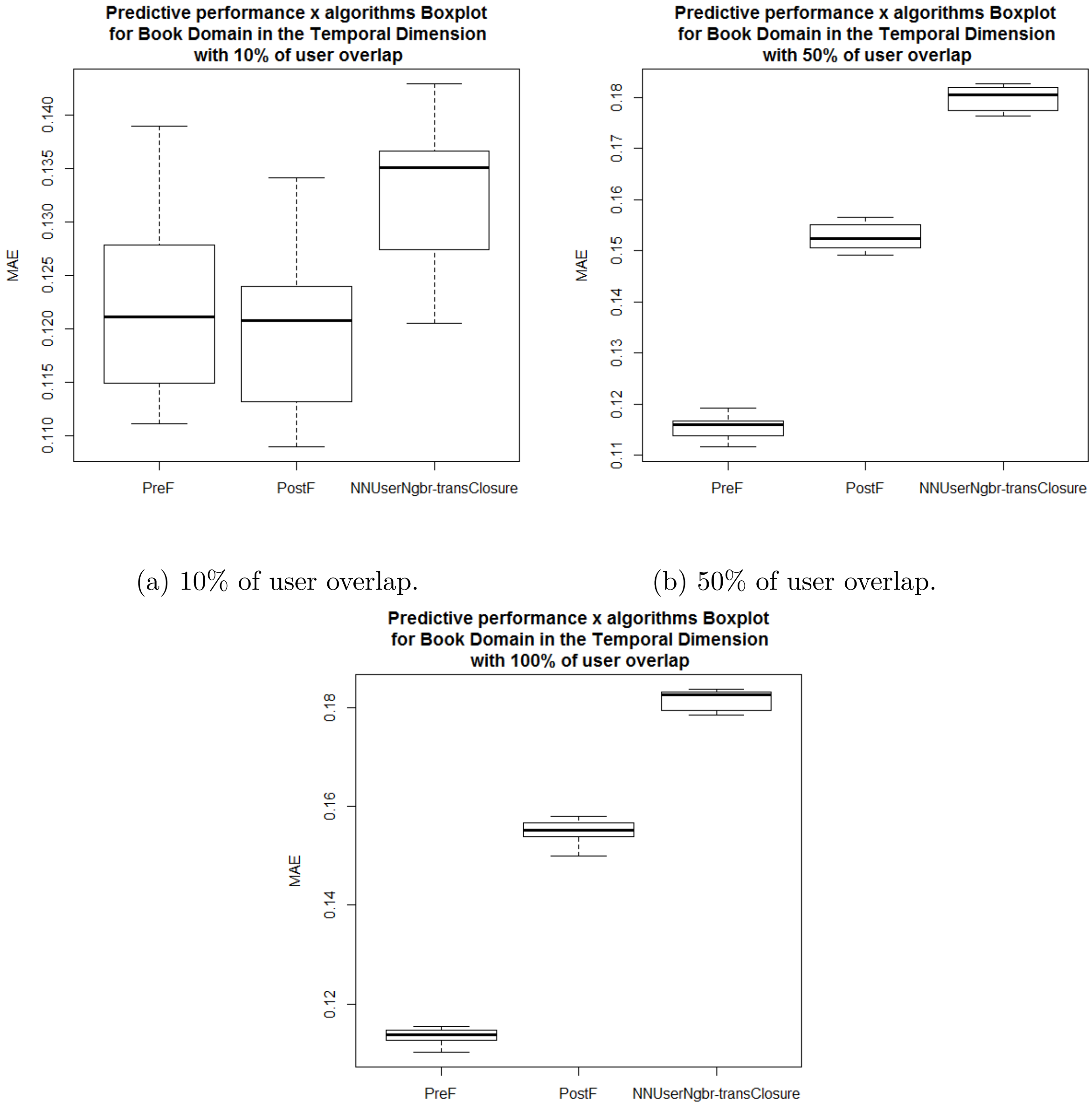


Figure 47 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the temporal dimension (source domain: television, and target domain: book).

The statistical significance tests verified that both the PreF and PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* baseline algorithm for all user overlap levels[28]. Figure 48a, Figure 48b and Figure 48c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

Regarding the classification performance, Figures 49a, 49b, and 49c present the results of the F-metric at different N values (between one and twenty), respectively, in 10%, 50%, and 100% of user overlap level for the *Book* domain as target, considering the *Temporal* dimension. As it can be seen, in all user overlap levels and top 'N' values, the

[28] p-value=0.005413 and W=100 for all tests, except when there was 10% of user overlap, where W=79 and p-value=0.0144 (PreF x *NNUserNgbr-transClosure*), and W=87 and p-value=0.001943 (PostF x *NNUserNgbr-transClosure*)



(a) 10% of user overlap.

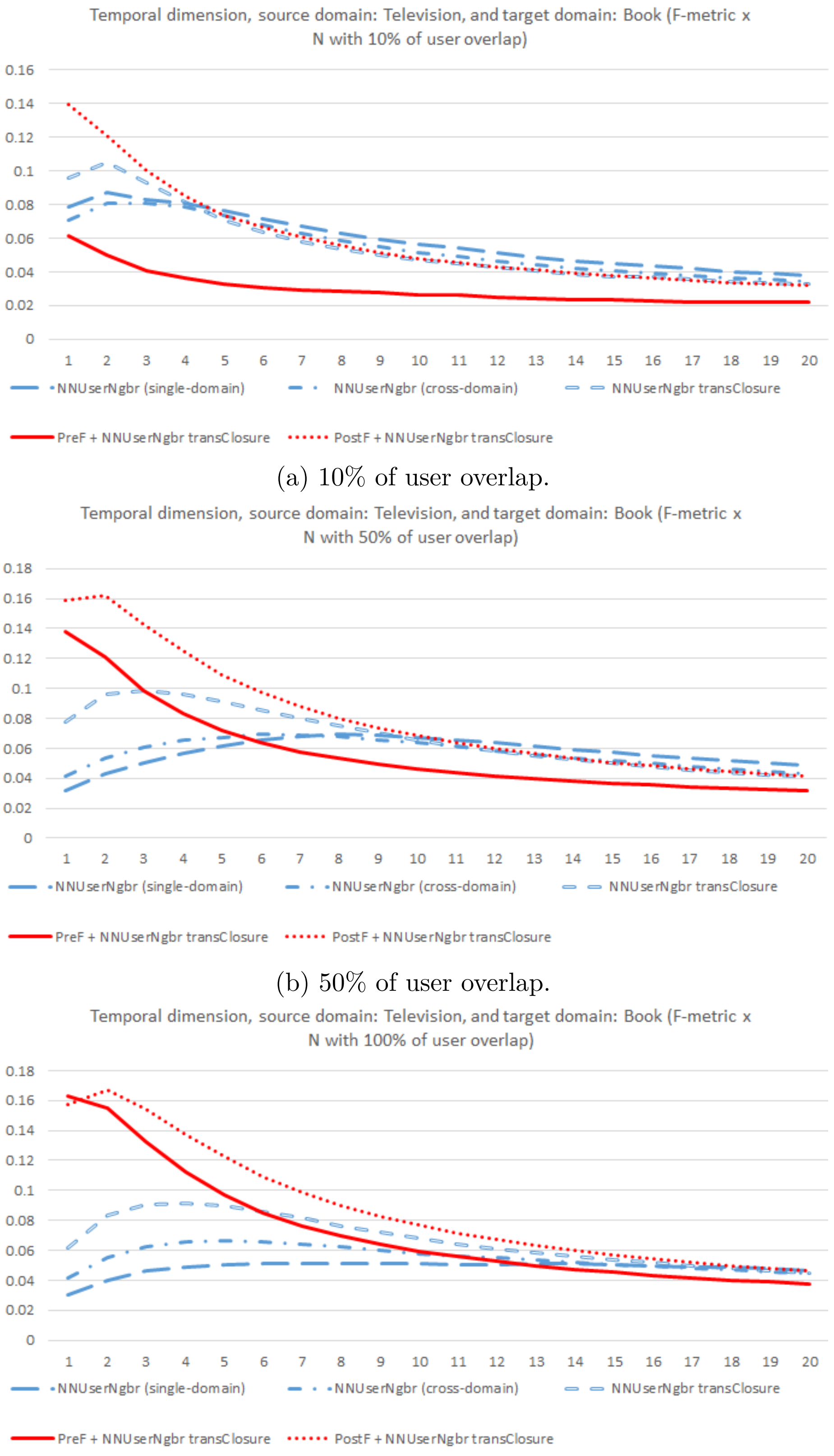
(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 48 – Overall prediction performance (MAE) boxplots for book domain in the temporal dimension with different user overlap levels (source domain: television).

PostF classification performance was better or similar than the *NNUserNgbr-transClosure* baseline algorithm, whereas the PreF was only better than that baseline for low top 'N' values with 50% and 100% of user overlap levels. In addition, the PostF classification performance was better than the PreF one for all user overlap levels and top 'N' values.

Figure 50 shows the variation of the F-metric value in different user overlap levels by fixing the top 'N' value to five. The statistical significance tests verified that the PostF F-metric values were greater than the *NNUserNgbr-transClosure* baseline algorithm for



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 49 – F-metric performance x top ‘N’ items for the book domain in the temporal dimension with different user overlap levels (source domain: television).

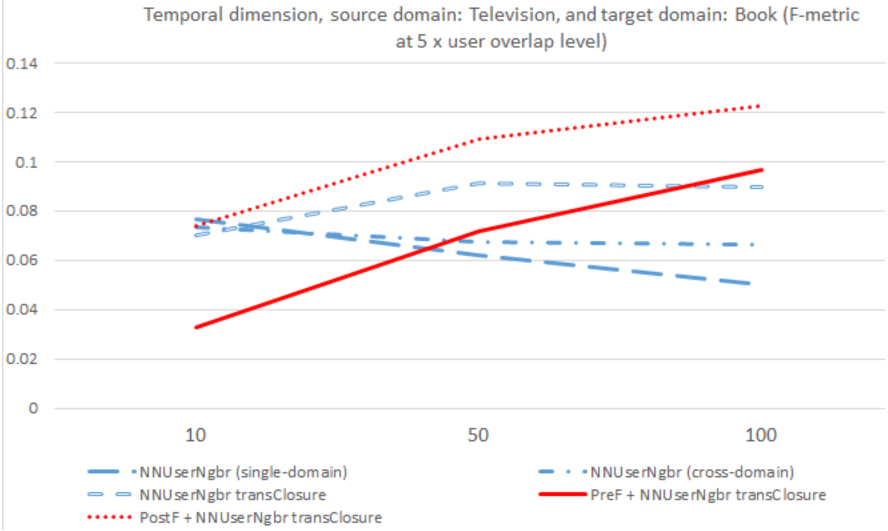


Figure 50 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the temporal dimension (target domain: book, and source: television).

50% and 100% of user overlap levels[29], whereas their performances were statistically similar for 10% of user overlap[30]. The PreF F-metric value was greater than the *NNUserNgbr-transClosure* algorithm for 100% of user overlap[31], whereas the opposite occurred when the user overlap levels were 10% and 50%[32].

5.2.1.2.2 Location Dimension

Table 24 reports the overall predictive performance of the recommender algorithms, considering all contextual values from the *Location* dimension and different user overlap levels for the *Book* domain as target. As it can be seen from the table, the addition of user ratings from other domain (*Television*), by using the same algorithm for cross-domain recommendation (corresponding to the two first rows of the table), improved the predictive performance in, approximately, 25–30% (MAE) and 13–18% (RMSE) depending on the user overlap level.

Also, Table 24 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 38–73% (MAE) and 25–60% (RMSE) depending on the user overlap levels. As it can be seen from table, the PostF predictive performance was better or similar than the *NNUserNgbr-transClosure* algorithm in all user overlap levels, with an

[29] p-value=0.003968 and W=100 for all tests
[30] p-value=0.35 and W=60
[31] W=77 and p-value=0.02163
[32] p-value=0.003968 and W=100 for both tests

Table 24 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Location dimension (source domain: Television, and target domain: Book).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.643 ± 0.022	0.963 ± 0.040	0.500 ± 0.008	0.794 ± 0.020	0.401 ± 0.006	0.719 ± 0.010
NNUserNgbr (cross-domain)	0.470 ± 0.020	0.830 ± 0.042	0.349 ± 0.007	0.645 ± 0.018	0.297 ± 0.005	0.617 ± 0.008
NNUserNgbr-transClosure	**0.125 ± 0.015**	**0.329 ± 0.034**	0.177 ± 0.006	0.427 ± 0.013	0.182 ± 0.003	0.460 ± 0.006
PreF with NNUserNgbr-transClosure	0.184 ± 0.285	0.447 ± 0.503	**0.121 ± 0.045**	**0.314 ± 0.087**	0.179 ± 0.019	0.432 ± 0.040
PostF with NNUserNgbr-transClosure	**0.127 ± 0.004**	**0.330 ± 0.012**	0.173 ± 0.004	0.410 ± 0.011	**0.175 ± 0.003**	**0.438 ± 0.009**

improvement that varied in, approximately, 2–4% (MAE) and 3–4% (RMSE) depending on the user overlap level.

In addition, we can see in Table 24 that the PostF outperformed the PreF algorithm for the majority of the user overlap levels (10% and 100%). As the results presented in Section 5.2.1.1.2 (source domain: *Book*, target domain: *Television*, and *Location* dimension), the PreF predictive performance had a high standard deviation. As mentioned in that section, this issue may be caused by the PreF feature of filtering ratings from the target domain for untested contexts, especially in the *Location* dimension, where there are several cities with a low number of ratings. Figure 51 illustrates the predictive performance (MAE) of the proposed algorithms over different user overlap levels.

The statistical significance tests verified that both the PreF and PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* baseline algorithm for 50% of user overlap level[33]. For 10% and 100% of user overlap, there was not a significant difference between the performance of the PostF algorithm and the *NNUserNgbr-transClosure* baseline algorithm[34]. The same occurred between the baseline and PreF algorithms for 100% of user overlap[35], whereas the *NNUserNgbr-transClosure* predictive errors was less than the PreF one[36] for 10% of user overlap. Figure 52a, Figure 52b and Figure 52c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

With respect to the classification performance, Figures 53a, 53b, and 53c present

[33] W=95 and p-value=0.0001028 (PreF x NNUserNgbr-transClosure), while W=77 and p-value=0.02163 (PostF and NNUserNgbr-transClosure)

[34] W=44 and p-value=0.6847 (10% of user overlap), while W=96 and p-value=0.06495 (100% of user

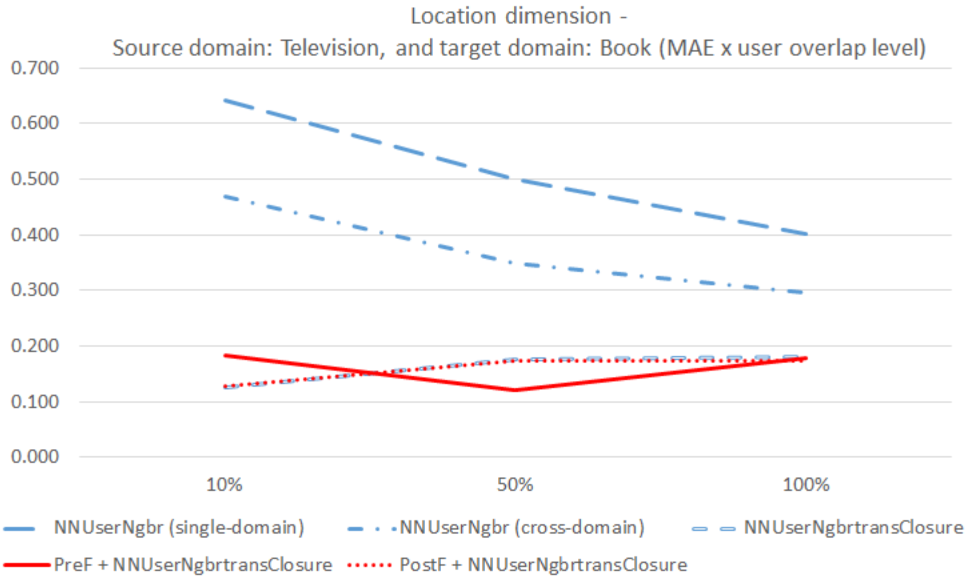


Figure 51 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the location dimension (source domain: television, and target domain: book).

the results of the F-metric at different top ‘N’ values (between one and twenty), respectively, in 10%, 50%, and 100% of user overlap level for the *Book* domain as target, considering the *Location* dimension. As it can be seen, for 10% and 50% of overlap levels and low top ‘N’ values, the PostF classification performance was better or similar than the *NNUserNgbr-transClosure* baseline algorithm, whereas for 100% of user overlap this can be seen for any top ‘N’ value. On the other hand, the PreF classification performance was worse than all other algorithms in all user overlap levels and ‘N’ values.

Figure 54 shows the variation of the F-metric value in different user overlap levels by fixing the top ‘N’ value to five. The statistical significance tests verified that the PostF F-metric values were greater than the *NNUserNgbr-transClosure* baseline algorithm for 50% and 100% of user overlap levels[37], whereas the opposite from this was observed for 10% of user overlap level[38]. The applied tests also verified that the *NNUserNgbr-transClosure* F-metric values were greater than the PreF ones for all user overlap levels[39].

5.2.1.2.3 Companion Dimension

Table 25 shows the overall predictive performance of the recommender algorithms, considering all contextual values from the *Companion* dimension and different user overlap levels for the *Book* domain as target. As it can be seen from the table, the addition of user

overlap)

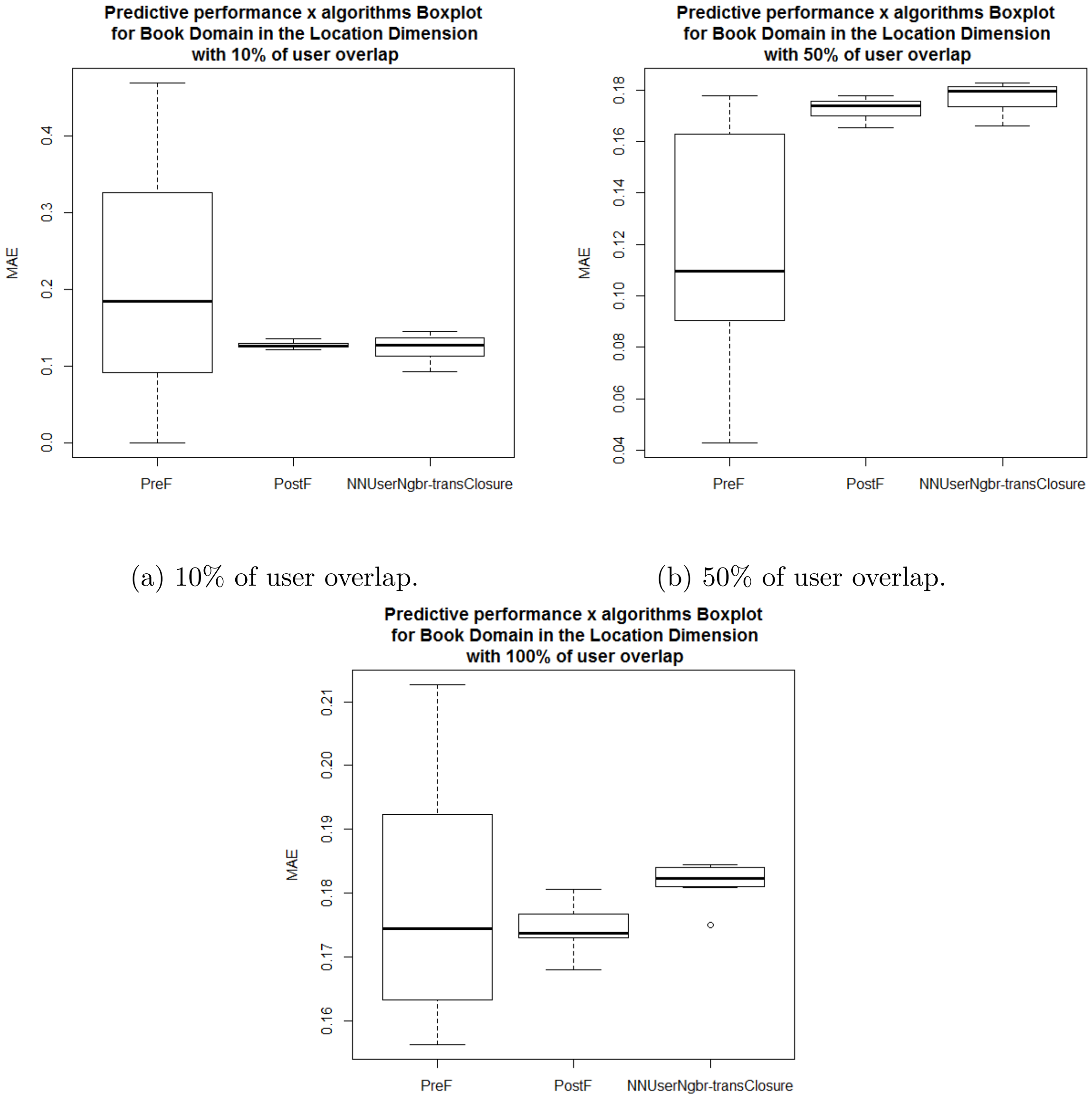
[35] W=96 and p-value=0.06495

[36] W=77 and p-value=0.02163

[37] p-value=0.005814 and W=97 for all tests

[38] p-value=0.005814 and W=97

[39] p-value=0.003913 and W=100 for all tests



(a) 10% of user overlap.

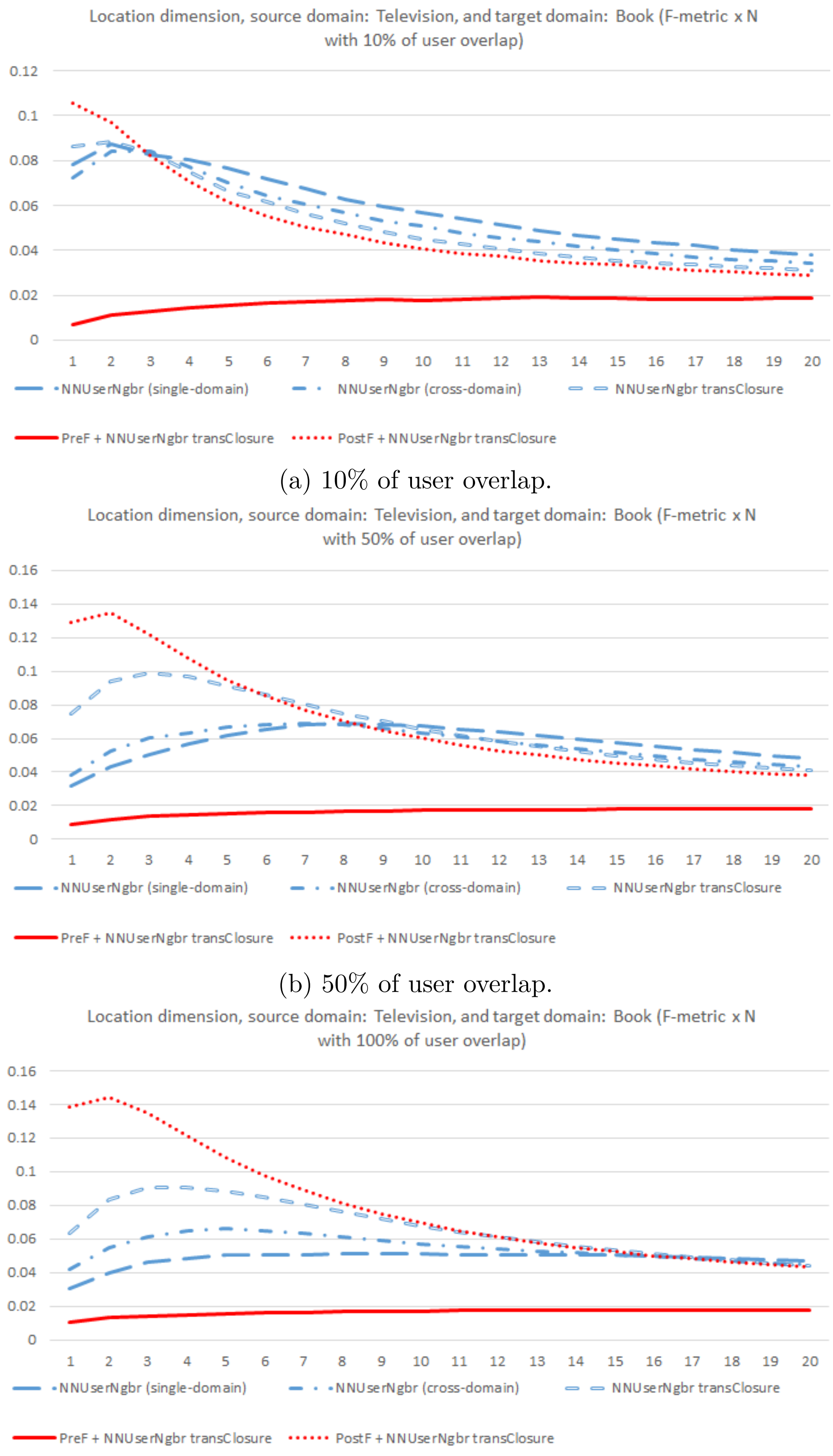
(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 52 – Overall prediction performance (MAE) boxplots for book domain in the location dimension with different user overlap levels (source domain: television).

ratings from other domain (*Television*), by using the same algorithm for cross-domain recommendation (corresponding to the two first rows of the table), improved the predictive performance in, approximately, 12% (MAE) and 11% (RMSE) for 10% of user overlap, and in, approximately, 11% (MAE) and 7% (RMSE) for 50% of user overlap.

Also, Table 25 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 28–59% (MAE) and 17–38% (RMSE) depending on the user overlap level.



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 53 – F-metric performance x top 'N' items for the book domain in the location dimension with different user overlap levels (source domain: television).

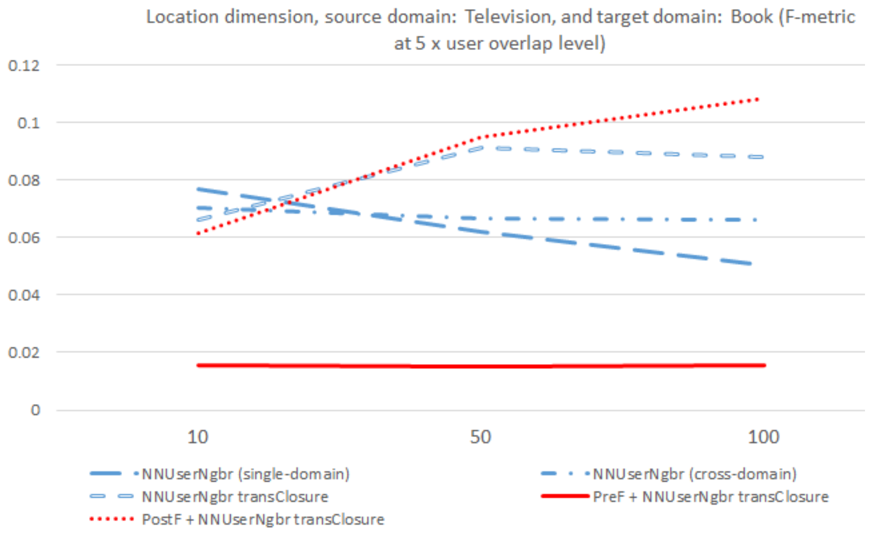


Figure 54 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the location dimension (target domain: book, and source: television).

Table 25 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Companion dimension (source domain: television, and target domain: book).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.643 ± 0.022	0.963 ± 0.040	0.500 ± 0.008	0.794 ± 0.020	0.401 ± 0.006	0.719 ± 0.010
NNUserNgbr (cross-domain)	0.565 ± 0.020	0.851 ± 0.042	0.442 ± 0.007	0.739 ± 0.018	0.409 ± 0.005	0.747 ± 0.008
NNUserNgbr-transClosure	**0.229** ± **0.039**	**0.519** ± **0.087**	0.265 ± 0.014	0.563 ± 0.031	0.293 ± 0.007	0.616 ± 0.014
PreF with NNUserNgbr-transClosure	0.611 ± 0.289	0.872 ± 0.346	0.757 ± 0.051	1.069 ± 0.059	0.789 ± 0.016	1.104 ± 0.022
PostF with NNUserNgbr-transClosure	0.241 ± 0.043	**0.520** ± **0.079**	**0.264** ± **0.014**	**0.544** ± **0.039**	**0.277** ± **0.007**	**0.575** ± **0.010**

As it can be seen from table, the PostF predictive performance was better than the *NNUserNgbr-transClosure* algorithm in, approximately, 0.4% (MAE) and 3% (RMSE) for 50% of user overlap, and in, approximately, 5% (MAE) and 6% (RMSE) for 100% of user overlap, however, the predictive performance of these algorithms were similar when the user overlap level was 10%.

In addition, the predictive performance of the PreF algorithm was worse than all other algorithms in the *Companion* dimension, as showed in Table 25. Figure 55 illustrates the predictive performance (MAE) of the proposed algorithms over different user overlap levels.

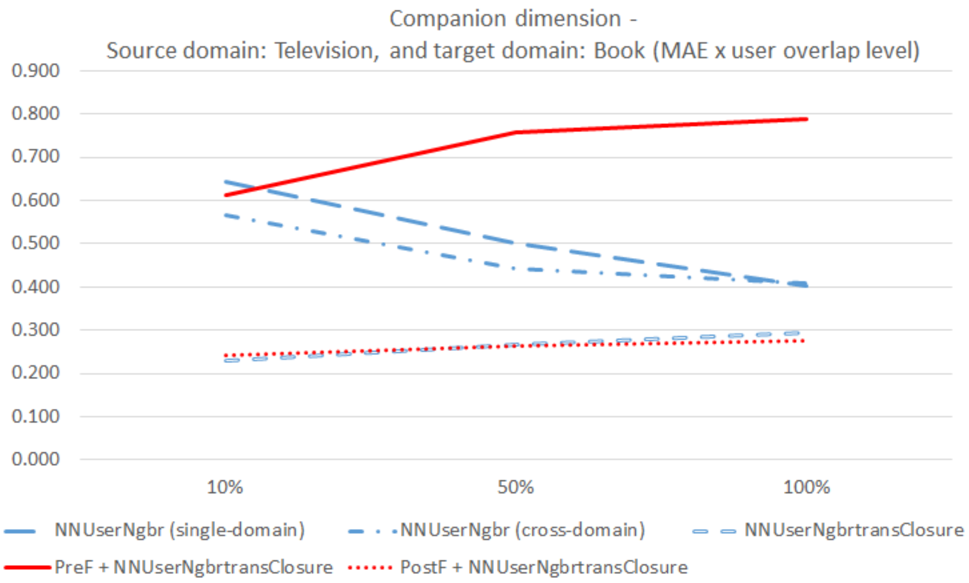


Figure 55 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the companion dimension (source domain: television, and target domain: book).

The statistical significance tests verified that the PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* algorithm for 100% of user overlap[40]. For 10% and 50% of user overlap levels, the applied tests could not verify a statistical difference between the *NNUserNgbr-transClosure* and PostF predictive errors (MAE)[41]. The applied tests also verified that the *NNUserNgbr-transClosure* predictive errors (MAE) were less than the PreF ones for all user overlap levels[42]. Figure 56a, Figure 56b and Figure 56c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

Figures 57a, 57b, and 57c present the results of the F-metric at different top ‘N’ values (between one and twenty), respectively, in 10%, 50%, and 100% of user overlap level for the *Book* domain as target, considering the *Companion* dimension. As it can be seen, the proposed algorithms only outperformed the baseline for 100% of user overlap with low values of top ‘N’.

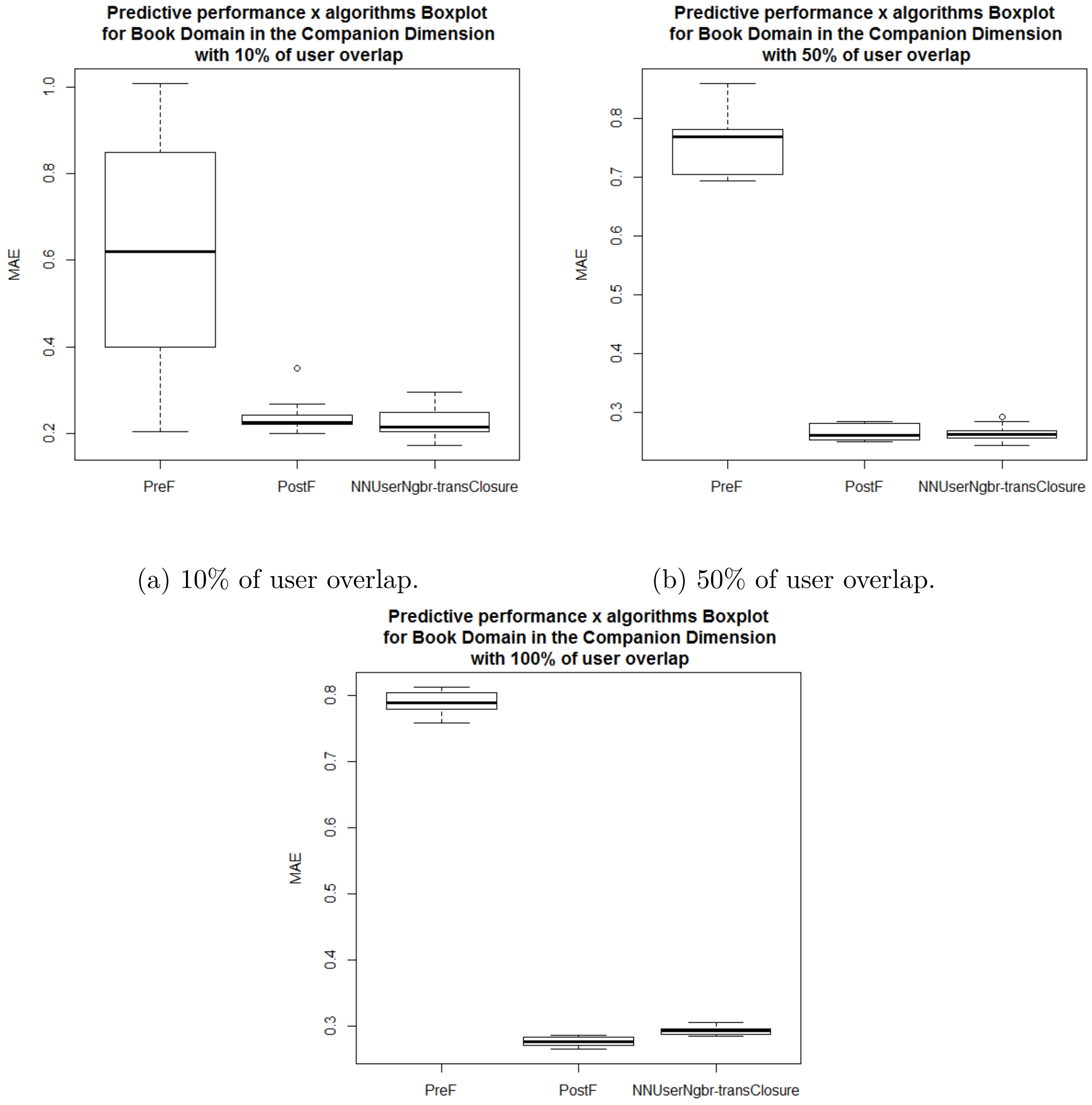
Figure 58 shows the variation of the F-metric value in different user overlap levels by fixing the top ‘N’ value to five. The statistical significance tests verified that the *NNUserNgbr-transClosure* F-metric values were greater than the both proposed algorithms for all user overlap levels[43].

[40] W=99 and p-value=0.01083

[41] For 10% of user overlap, W=60 and p-value=0.2406, while for 50%, W=55 and p-value=0.3697

[42] W=99 and p-value=0.01083 for all tests

[43] p-value=0.005814 and W=97 for all tests between the PostF and baseline algorithms, while p-value=0.003913 and W=100 for all tests between the PreF and baseline algorithms



(a) 10% of user overlap.

(b) 50% of user overlap.

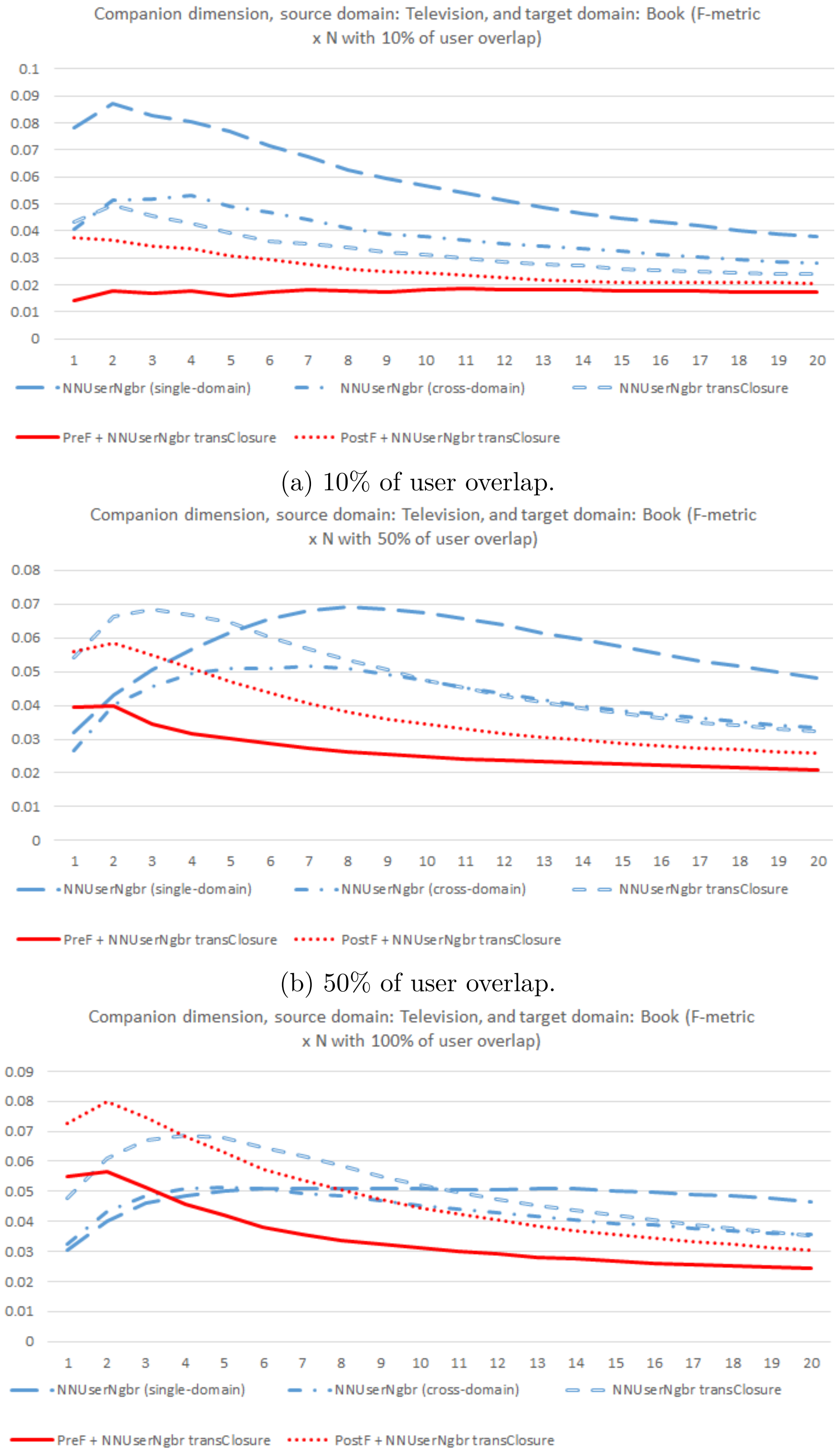
(c) 100% of user overlap.

Figure 56 – Overall prediction performance (MAE) boxplots for book domain in the companion dimension with different user overlap levels (source domain: television).

5.2.1.2.4 Combining Contextual Dimensions

In the previous sections, we presented the evaluation results regarding the contextual dimensions separately. In this section, we report the results for a combination of two contextual dimensions considering the same evaluation metrics and methodology described before.

We have seen in Section 4.1.2 that two of the contextual dimensions (*Temporal* and *Location*) provide a greater information gain than the *Companion* dimension, which is confirmed by the results presented in the previous sections. In this way, we evaluated



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 57 – F-metric performance x top 'N' items for the book domain in the companion dimension with different user overlap levels (source domain: television).

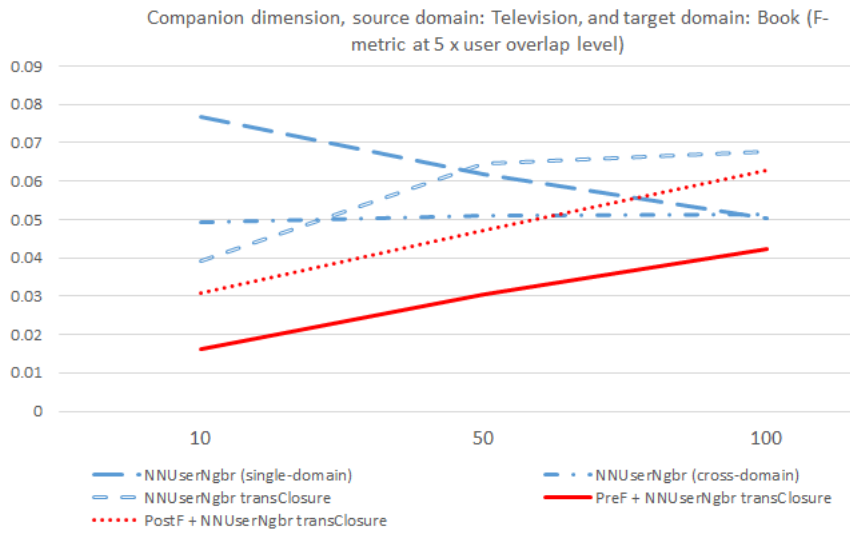


Figure 58 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the companion dimension (target domain: book, and source: television).

the combination of those two contextual dimensions (*Temporal* and *Location*), aiming to verify its performance in comparison to the performance of them evaluated separately.

Table 26 reports the overall predictive performance of the recommender algorithms, considering *all contextual value combinations* from the *Temporal* and *Location* dimensions with different user overlap levels for the *Book* domain as target. As it was observed in the previous sections, the addition of user ratings from the *Television* domain also improved the predictive performance of the *NNUserNgbr* in, approximately, 25–30% (MAE) and 13–18% (RMSE) depending on the user overlap level (rows 1 and 2 from the table).

Also, Table 26 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 39–75% (MAE) and 25–60% (RMSE) depending on the user overlap levels.

As it can be seen from table, except when the user overlap level was 10% (by considering only the MAE metric), the PostF predictive performance was better than the *NNUserNgbr-transClosure* algorithm in all other user overlap levels, with an improvement that varied in, approximately, 10–20% (MAE) and 8–17% (RMSE) depending on the user overlap level. Despite the *NNUserNgbr-transClosure* algorithm have outperformed the PostF when the user overlap level was 10%, they had a similar performance, separated only by their standard deviations. Figure 59 illustrates the predictive performance (MAE) of the proposed algorithms over different user overlap levels.

The statistical significance tests verified that the PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* algorithm for the 50% and 100% user overlap

Table 26 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual value combinations from the temporal and location dimensions (source domain: Television, and target domain: Book).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.643 ± 0.022	0.963 ± 0.040	0.500 ± 0.008	0.794 ± 0.020	0.401 ± 0.006	0.719 ± 0.010
NNUserNgbr (cross-domain)	0.470 ± 0.020	0.830 ± 0.042	0.349 ± 0.007	0.645 ± 0.018	0.297 ± 0.005	0.617 ± 0.008
NNUserNgbr-transClosure	**0.117** ± **0.017**	0.328 ± 0.039	0.176 ± 0.006	0.418 ± 0.013	0.180 ± 0.003	0.457 ± 0.006
PreF with NNUserNgbr-transClosure	0.344 ± 0.396	0.713 ± 0.545	0.200 ± 0.045	0.423 ± 0.126	0.331 ± 0.027	0.699 ± 0.064
PostF with NNUserNgbr-transClosure	**0.121** ± **0.010**	**0.309** ± **0.028**	**0.158** ± **0.004**	**0.382** ± **0.011**	**0.143** ± **0.005**	**0.378** ± **0.011**

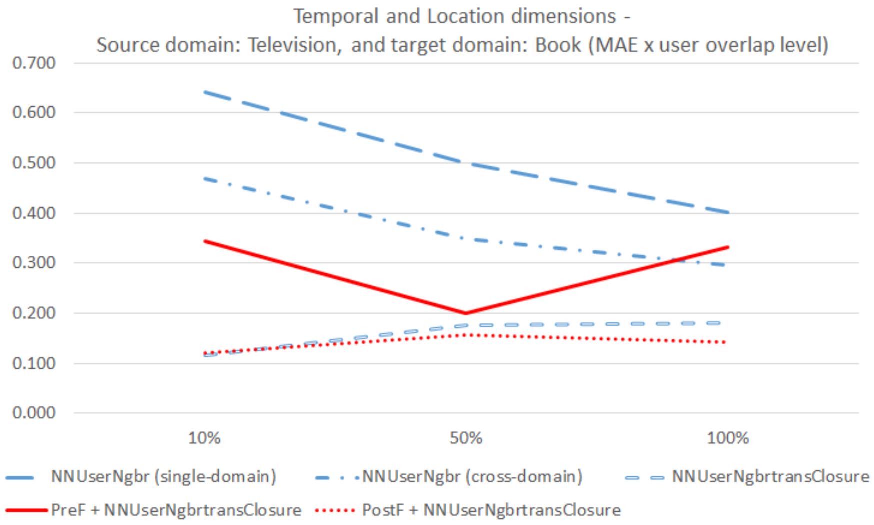


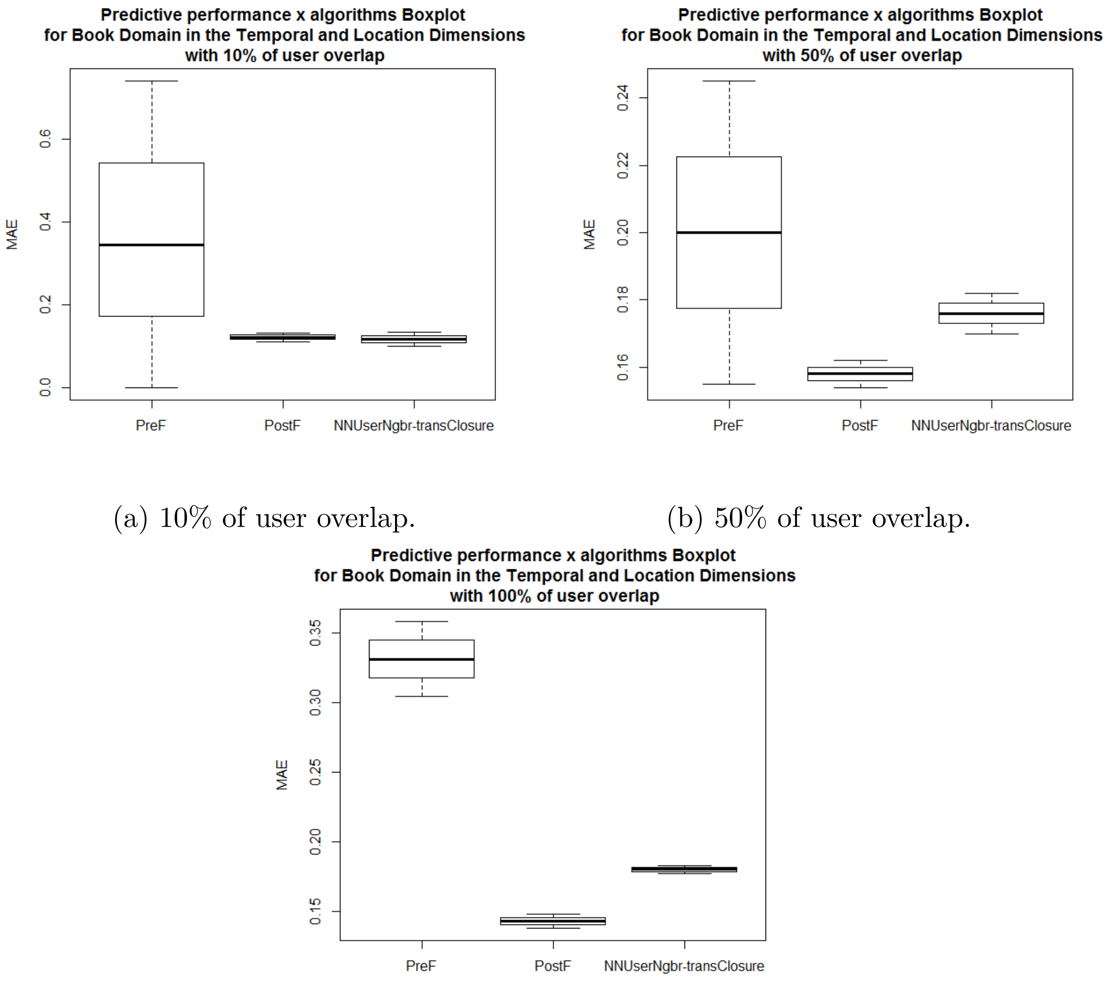
Figure 59 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the temporal and location dimensions (source domain: television, and target domain: book).

levels[44]. When the user overlap level was 10%, the *NNUserNgbr-transClosure* predictive error (MAE) was statistically similar to the PostF one[45]. On the other hand, the applied tests verified that the *NNUserNgbr-transClosure* predictive errors (MAE) were less than the PreF ones for all user overlap levels[46]. Figure 99a, Figure 99b and Figure 99c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

[44] In both cases, with W=99 and p-value=0.005413

[45] W=41 and p-value=0.5

[46] For all cases, with W=100 and p-value=0.003914



(a) 10% of user overlap.

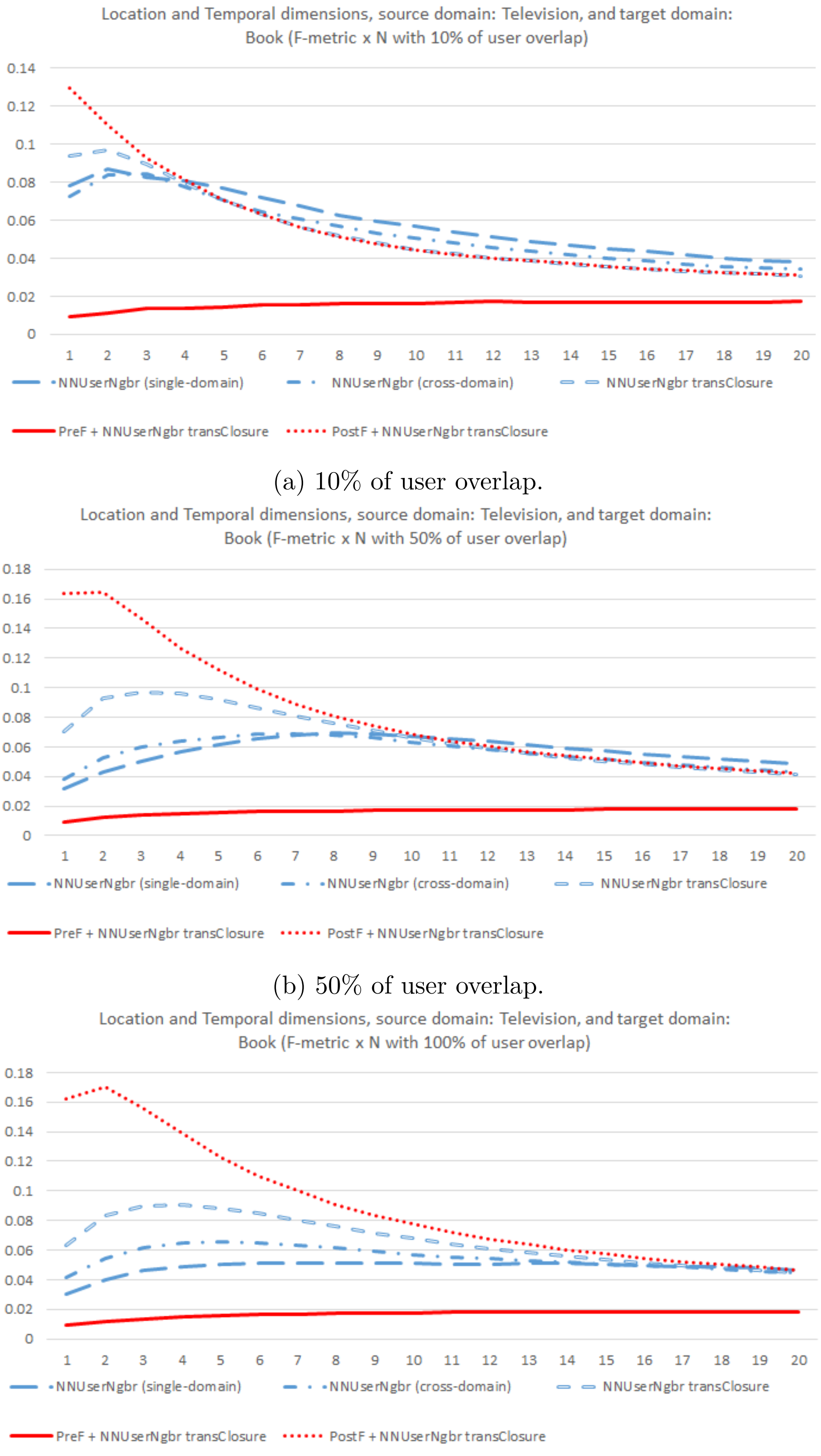
(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 60 – Overall prediction performance (MAE) boxplots for book domain in the temporal and location dimensions with different user overlap levels (source domain: television).

Taking into account the classification performance, Figures 61a, 61b, and 61c present the results of the F-metric at different top 'N' values, respectively, in 10%, 50%, and 100% of user overlap level for the *Television* domain as target, considering the *Temporal* and *Location* dimensions. For all user overlap levels and top 'N' values, the PreF classification performance was worse than the all other algorithms. This result also occurred when the evaluation was performed for the *Television* as target by combining the two contextual dimensions (see Section 5.2.1.1.4). On the other hand, the PostF classification performance was better or similar than the *NNUserNgbr-transClosure* baseline algorithm in all user overlap levels and top 'N' values.

Figure 62 shows the variation of the F-metric value in different user overlap levels by fixing the top 'N' value to five. The statistical significance tests verified that the PostF F-metric values were greater than the *NNUserNgbr-transClosure* baseline algorithm for



Location and Temporal dimensions, source domain: Television, and target domain: Book (F-metric x N with 10% of user overlap)

(a) 10% of user overlap.

Location and Temporal dimensions, source domain: Television, and target domain: Book (F-metric x N with 50% of user overlap)

(b) 50% of user overlap.

Location and Temporal dimensions, source domain: Television, and target domain: Book (F-metric x N with 100% of user overlap)

0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
NNUserNgbr (single-domain)
NNUserNgbr (cross-domain)
NNUserNgbr transClosure
PreF + NNUserNgbr transClosure
PostF + NNUserNgbr transClosure

(c) 100% of user overlap.

Figure 61 – F-metric performance x top ‘N’ items for the book domain in the temporal and location dimensions with different user overlap levels (source domain: television).

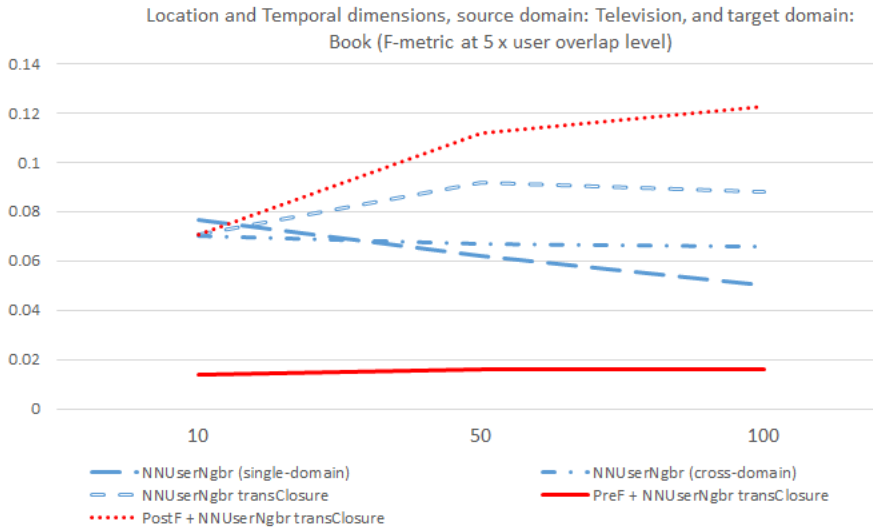


Figure 62 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the temporal and location dimensions (target domain: book, and source: television).

50% and 100% of user overlap levels[47], whereas their performances were statistically similar for 10% of user overlap[48]. The applied tests also verified that *NNUserNgbr-transClosure* F-metric values were greater than the PreF ones for all user overlap levels[49].

5.2.1.3 Summary

In this section, we provide a summary of the results from the evaluation of the "book-television dataset". Figure 63 shows a dispersion diagram illustrating the predictive performance (MAE) for the algorithms by varying target domain (*Book* and *Television*), contextual dimension and user overlap levels, whereas Figure 64 shows the same, but considering the RMSE metric. It is important to mention that these figures do not take into account the standard deviation and the statistical significance of the results.

Table 27 presents the predictive performance (MAE) achieved by the PreF and PostF algorithms in comparison to the best baseline algorithm (*NNUserNgbr-transClosure*), by taking into account their statistical significance[50] and different target domain, contextual dimension and user overlap levels.

Regarding the classification performance, Figure 65 presents a dispersion diagram illustrating the F-metric performance (with N=5) for the algorithms by varying target domain, contextual dimension and user overlap levels. Once again, it is important to mention that we are not considering the standard deviation and the statistical significance of the results in that figure.

[47] p-value=0.005814 and W=97 for all tests

[48] p-value=0.5 and W=51

[49] p-value=0.003968 and W=100 for all tests

[50] In the table, "**" means that the result could not be considered statistically significant.

Table 28 shows the classification performance improvement (F-metric with N=5) obtained by the PreF and PostF algorithms in comparison to the best baseline algorithm (*NNUserNgbr-transClosure*), by taking into account their statistical significance[51] and different target domain, contextual dimension and user overlap levels.

As it can be seen, at least one proposed algorithm (PreF or PostF) achieved the best predictive performance among the algorithms (or it was similar to the best one) in all scenarios (with distinct target domains, contextual dimensions, and user overlap levels). By considering the classification metric, the PostF algorithm achieved the best performance among the algorithms (or it was similar to the best one) in the majority of the scenarios. By summarizing the main findings from the evaluation results described in this section, we can say that:

- In all scenarios (with different target domains, contextual dimensions and user overlap levels), the addition of user ratings from an auxiliary (source) domain improved the predictive performance of the *NNUserNgbr* algorithm, which was not designed for making cross-domain recommendations. This fact can be also observed in almost all scenarios regarding the classification performance of that algorithm. Note that this occurred even when a source domain had less ratings than the target domain.

- In all scenarios (with different target domains, contextual dimensions and user overlap levels), the *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one by considering their predictive performances. Regarding their classification performances this fact also occurred in almost all scenarios.

- The proposed algorithms (PreF and PostF) had better predictive and classification performances in the *Temporal* dimension than others dimensions. This fact contrasts to the information gain calculated in Section 4.1.2. In this contextual dimension, the PostF outperformed the *NNUserNgbr-transClosure* algorithm in all scenarios (user overlap levels and target domains) by considering either their predictive or classification performances. The PreF outperformed the *NNUserNgbr-transClosure* algorithm in all scenarios (user overlap levels and target domains) by considering the predictive performance. With respect to the classification performance, the PreF outperformed the *NNUserNgbr-transClosure* algorithm for 100% of user overlap (regardless the target domain) and for 50% of user overlap when the *Television* was the target domain.

- If we make a comparison between the proposed algorithms in the *Temporal* dimension considering different evaluation metrics (predictive or classification), we see distinct relations between the algorithms' results. While the PostF algorithm outperforms

[51] In the table, "**" means that the result could not be considered statistically significant.

the PreF one by considering the classification performance in almost all scenarios (user overlap levels and target domains), the opposite happens when we take the predictive performance into account.

- The more is the user overlap level the better is the classification performance of the PostF algorithm, especially in the *Temporal* and *Location* dimensions (or their combinations). The same can be observed for the PreF algorithm, but only considering the *Temporal* dimension. Note that this fact did not seem to occur when we considered the predictive performance of the algorithms.

- The predictive and classification performances of the PreF algorithm were more affected than the PostF ones by considering the quantity of the contextual information present in the user ratings (see Section 4.1.3). The more particular were the tested contexts the worse were the PreF performances (e.g. in the *Location* dimension and in the combination of *Location* and *Temporal* dimensions). In this way, the PreF performances had a high variation, depending on the contextual information present in the user ratings, whereas the PostF performances were more uniform and similar to the *NNUserNgbr-transClosure* algorithm.

- Regarding the low quality of the contextual information obtained in the *Companion* dimension (see Section 4.1.1.3), we can see that both proposed algorithms had low predictive and classification performances in comparison to other dimensions. This could also have occurred due to the low quantity of contextual information present in the user ratings for that contextual dimension. In particular, the PostF algorithm achieved a good performance by considering only the predictive metrics.

- With respect to the combination of contextual dimensions (*Temporal* and *Location*), we can see the PostF predictive and classification performances in that combination were close to their own performances using only the *Temporal* dimension as single source of contextual information, whereas the PreF predictive and classification performances were similar to their own performances using only the *Location* dimension. The classification performances of both algorithms were reduced with the addition of contextual information from other contextual dimension. In particular, the predictive performance of the PostF algorithm was slightly improved depending on the user overlap level and target domain, whereas the predictive performance of the PreF algorithm was decreased in any case.

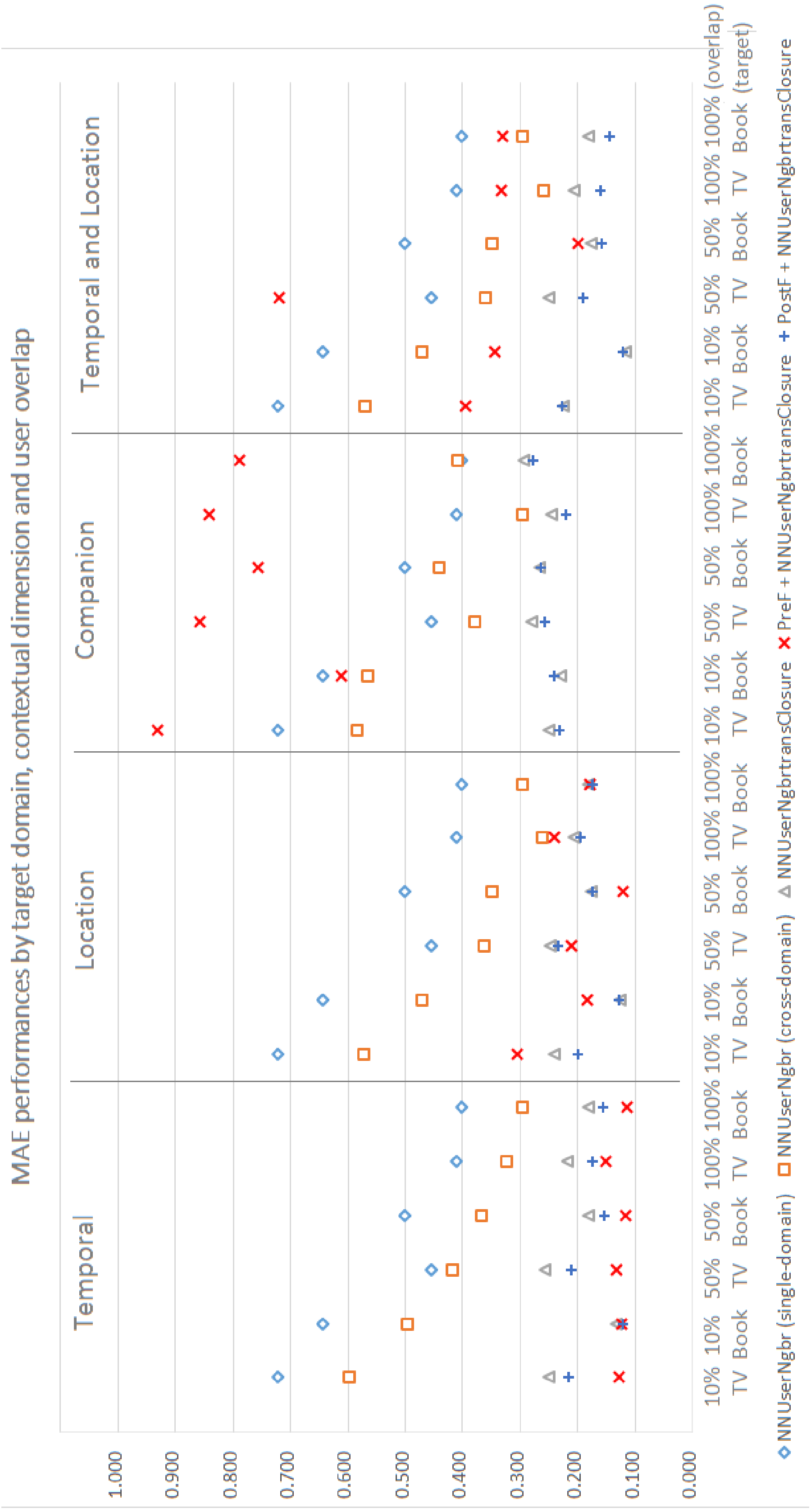


Figure 63 – Predictive performance (MAE) for the algorithms by varying target domain (book and TV), contextual dimension and user overlap levels (dispersion diagram).

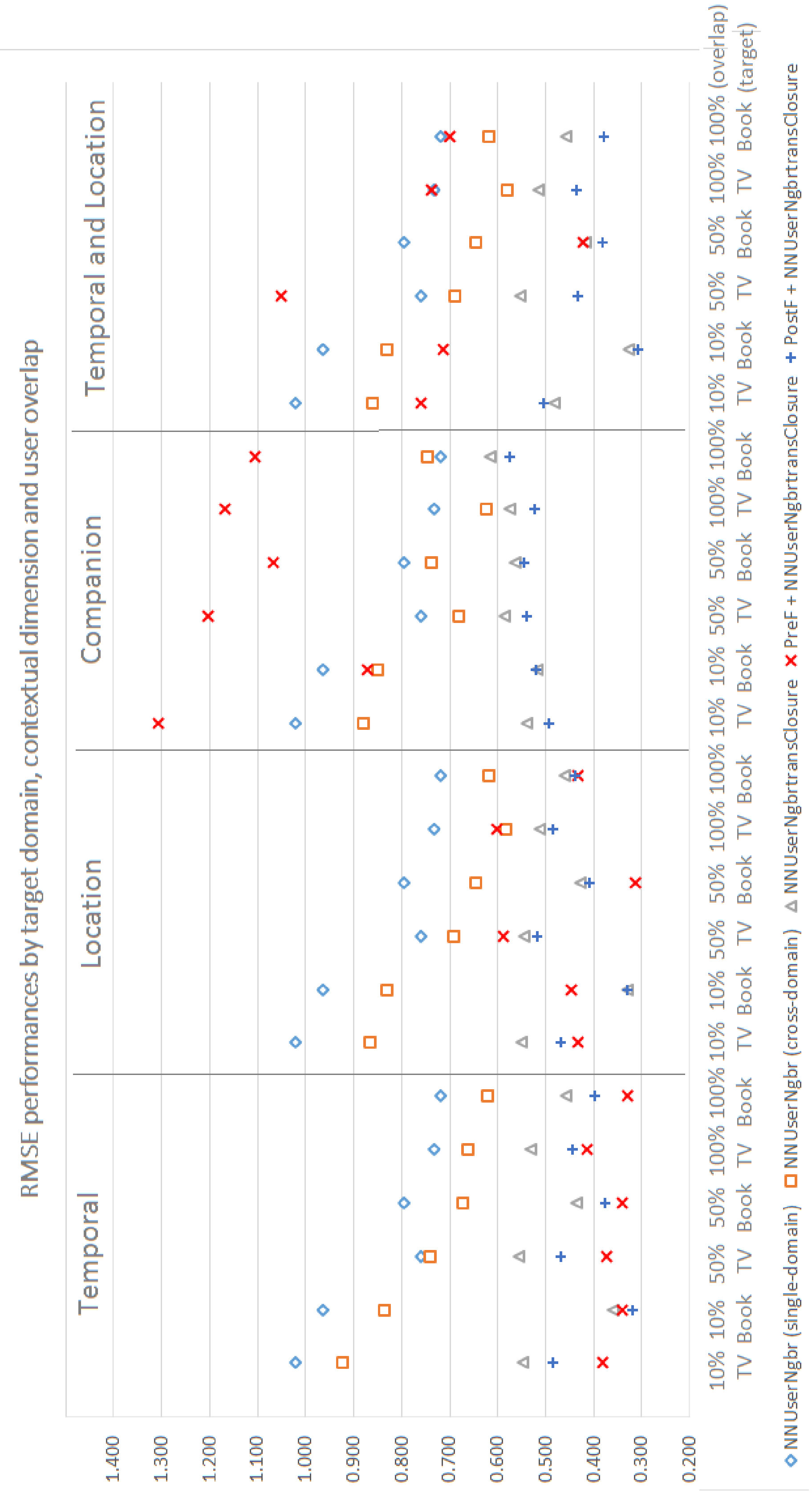


Figure 64 – Predictive performance (RMSE) for the algorithms by varying target domain (book and TV), contextual dimension and user overlap levels (dispersion diagram).

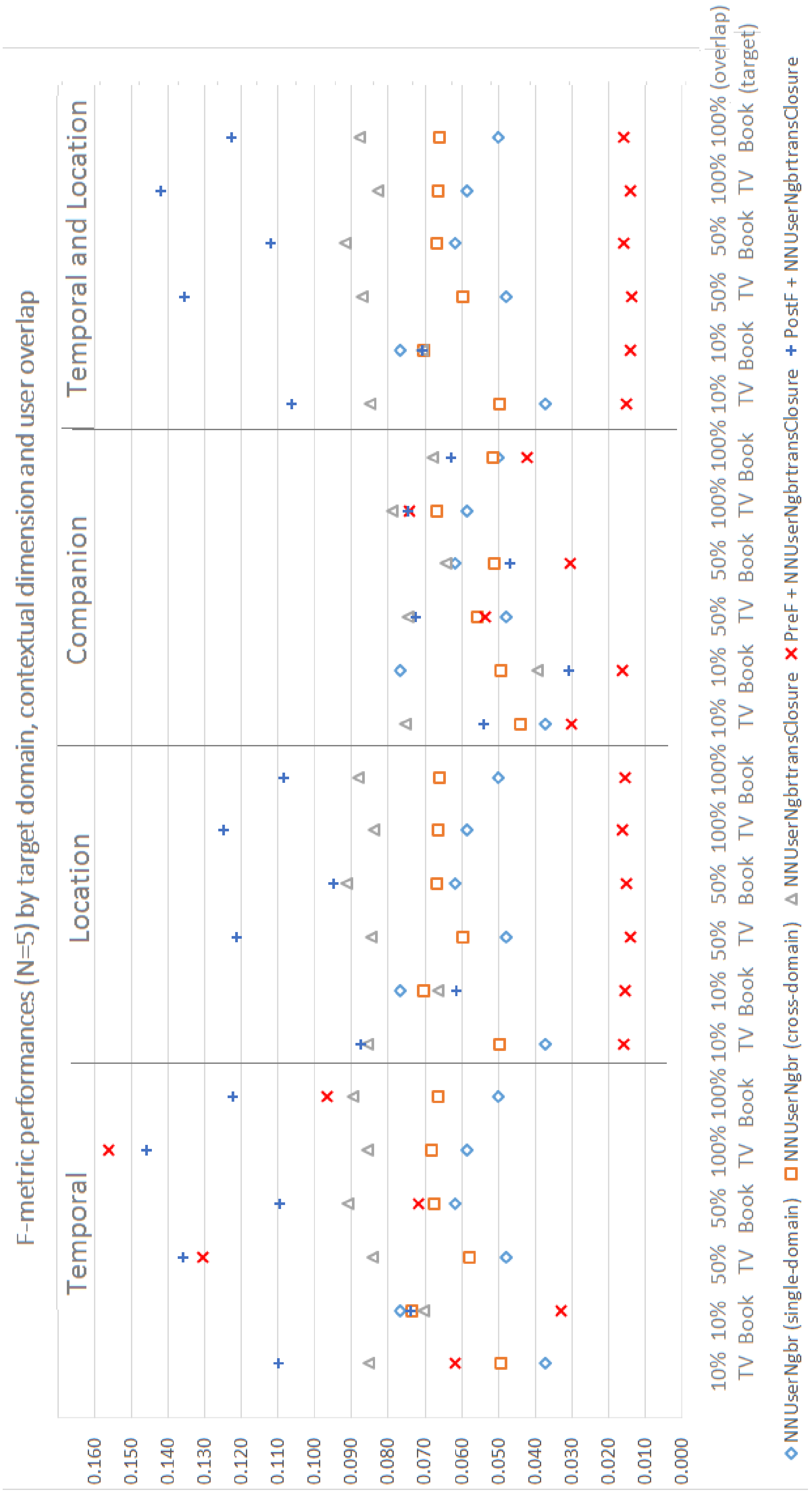


Figure 65 – Classification performance (F-metric with N=5) for the algorithms by varying target domain (book and TV), contextual dimension and user overlap levels (dispersion diagram).

Table 27 – Overall predictive performance (MAE) of the proposed algorithms in comparison to the best baseline one by varying target domain (book and TV), contextual dimension and user overlap levels.

Contextual dimension	**Target Domain**	**User Overlap Level**	**PreF Improvement**	**PostF Improvement**
Temporal	TV	10%	48.6%	13.9%
	Book	10%	8%	9.7%
	TV	50%	48.4%	18%
	Book	50%	35.7%	15%
	TV	100%	30.4%	20.3%
	Book	100%	37.4%	14.6%
Location	TV	10%	-26.8%**	16.7%**
	Book	10%	-46.7%	-1.8%**
	TV	50%	14.2%**	5.7%
	Book	50%	31.9%	2.5%
	TV	100%	-17.2%	5.9%
	Book	100%	1.7%**	4%**
Companion	TV	10%	-273.7%	6.7%**
	Book	10%	-166.8%	-5.2%**
	TV	50%	-207.9%	8.2%
	Book	50%	-185.2%	0.4%**
	TV	100%	-242.4%	10.2%
	Book	100%	-169.4%	5.6%
Temporal and Location	TV	10%	-76.8%	-0.9%**
	Book	10%	-194%	-3.4%**
	TV	50%	-188%	24%
	Book	50%	-13.6%	10.2%
	TV	100%	-60.9%	22.2%
	Book	100%	-83.9%	20.6%

5.2.2 Book-Music Results

As mentioned before, we evaluated the quality of the cross-domain algorithms by varying the target domain for each dataset in order to study the impact of the density of the target domain data in comparison to the density of the source domain data. Thus, the following sections present the results considering a different domain as a target (*Music* and *Book*).

5.2.2.1 Music as Target Domain

According to the contextual dimensions present in the datasets, we describe the evaluation results for each contextual dimension in the following sections. In addition, we show the results for a combination of contextual dimensions in Section 5.2.2.1.4.

Table 28 – Overall classification performance (F-metric with N=5) of the proposed algorithms in comparison to the best baseline one by varying target domain (book and TV), contextual dimension and user overlap levels.

Contextual dimension	**Target Domain**	**User Overlap Level**	**PreF Improvement**	**PostF Improvement**
Temporal	TV	10%	-38%	22.4%
	Book	10%	-113.4%	4.7%**
	TV	50%	35.4%	38%
	Book	50%	-27.2%	16.7%
	TV	100%	45%	41.2%
	Book	100%	7.3%	26.7%
Location	TV	10%	-435.2%	1.9%**
	Book	10%	-329.4%	-8.1%
	TV	50%	-491.7%	30.4%
	Book	50%	-496.7%	3.8%
	TV	100%	-414%	32.7%
	Book	100%	-467.1%	18.7%
Companion	TV	10%	-148.4%	-39.1%
	Book	10%	-142.4%	-26.8%
	TV	50%	-39%	-2.8%
	Book	50%	-112.2%	-36.9%
	TV	100%	-6.2%	-5.8%**
	Book	100%	-60.3%	-7.5%
Temporal and Location	TV	10%	-457.2%	20%
	Book	10%	-404.2%	0.05%**
	TV	50%	-532.5%	35.7%
	Book	50%	-482.4%	18.1%
	TV	100%	-488.4%	41.6%
	Book	100%	-456%	28.4%

5.2.2.1.1 Temporal Dimension

Table 29 reports the overall predictive performance of the recommender algorithms, considering all contextual values from the *Temporal* dimension and different user overlap levels for the *Music* domain as target. The rows 1 and 2 from the table show the *NNUserNgbr* predictive performance when it is applied in the single-domain and cross-domain recommendations. As it can be seen, the simple addition of user ratings from other domain (*Book*), by using the same algorithm for cross-domain recommendation, improved the recommendation performance in, approximately, 5–7% (MAE) and 1–2% (RMSE) depending on the user overlap level.

In addition, Table 29 presents the overall performance of the PreF and PostF algorithms, besides the base *NNUserNgbr-transClosure* algorithm, which outperformed the *NNUserNgbr* algorithm (performed for cross-domain purposes) by achieving an im-

Table 29 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Temporal dimension (source domain: Book, and target domain: Music).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.684 ± 0.026	0.972 ± 0.053	0.654 ± 0.016	0.970 ± 0.030	0.588 ± 0.010	0.903 ± 0.017
NNUserNgbr (cross-domain)	0.634 ± 0.022	0.949 ± 0.051	0.610 ± 0.013	0.943 ± 0.028	0.557 ± 0.008	0.891 ± 0.014
NNUserNgbr-transClosure	0.307 ± 0.010	0.631 ± 0.013	0.458 ± 0.013	0.793 ± 0.020	0.480 ± 0.006	0.816 ± 0.011
PreF with NNUserNgbr-transClosure	**0.185** ± **0.054**	0.616 ± 0.166	**0.171** ± **0.005**	**0.476** ± **0.007**	**0.212** ± **0.007**	**0.550** ± **0.016**
PostF with NNUserNgbr-transClosure	0.257 ± 0.031	**0.519** ± **0.038**	0.400 ± 0.004	0.707 ± 0.002	0.423 ± 0.005	0.725 ± 0.010

provement that varied in, approximately, 13–51% (MAE) and 8–33% (RMSE) depending on the user overlap levels.

As it can be seen from table, the PreF predictive performance was better than the *NNUserNgbr-transClosure* and PostF algorithms in all user overlap levels, except when the user overlap level was 10% if we only consider the RMSE metric instead of the MAE one. The improvement achieved by the PreF algorithm in comparison to the *NNUserNgbr-transClosure* one varied in, approximately, 39–55% (MAE) and 2–40% (RMSE) depending on the user overlap level. The PostF predictive performance was also better than the *NNUserNgbr-transClosure* algorithm in all user overlap levels, however, its improvement was smaller than the achieved by the PreF algorithm.

Figure 66 (MAE) and Figure 67 (RMSE) illustrate the predictive performance of the proposed algorithms over different user overlap levels. Note that for this case we showed the figures for both predictive metrics, since we observed a difference in the PreF performance depending on the predictive metric used in the evaluation.

The statistical significance tests verified that both the PreF and PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* baseline algorithm for all user overlap levels[52]. Figure 68a, Figure 68b and Figure 68c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

Regarding the classification performance, Figures 69a, 69b, and 69c present the results of the F-metric at different N values (between one and twenty), respectively, in

[52] p-value=0.005413 and W=97 for all tests.

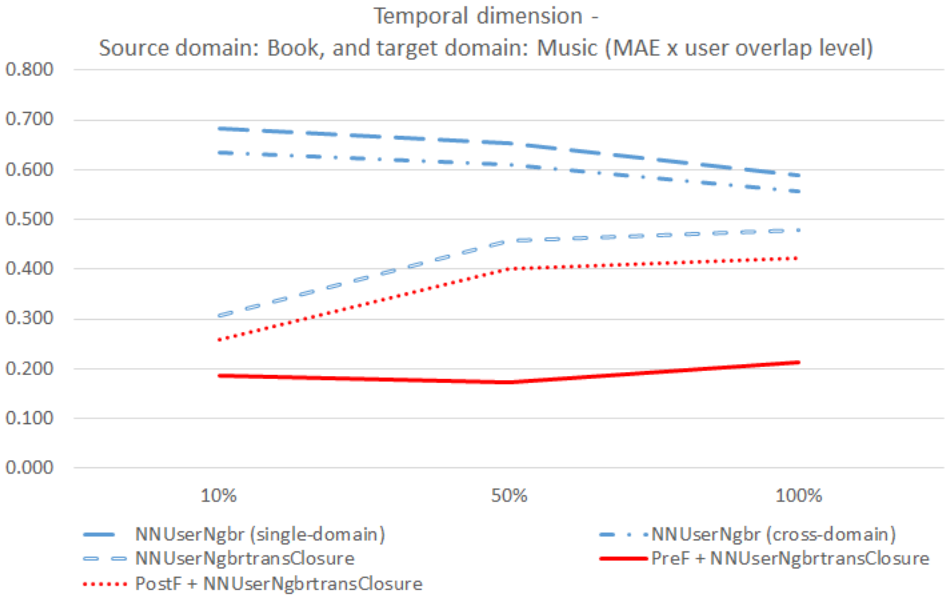


Figure 66 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the temporal dimension (source domain: book, and target domain: Music).

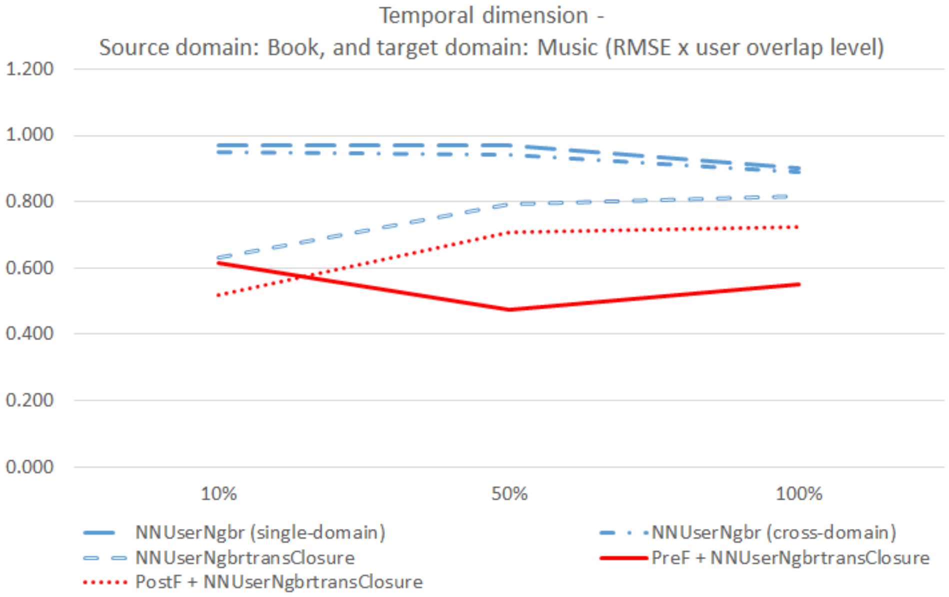
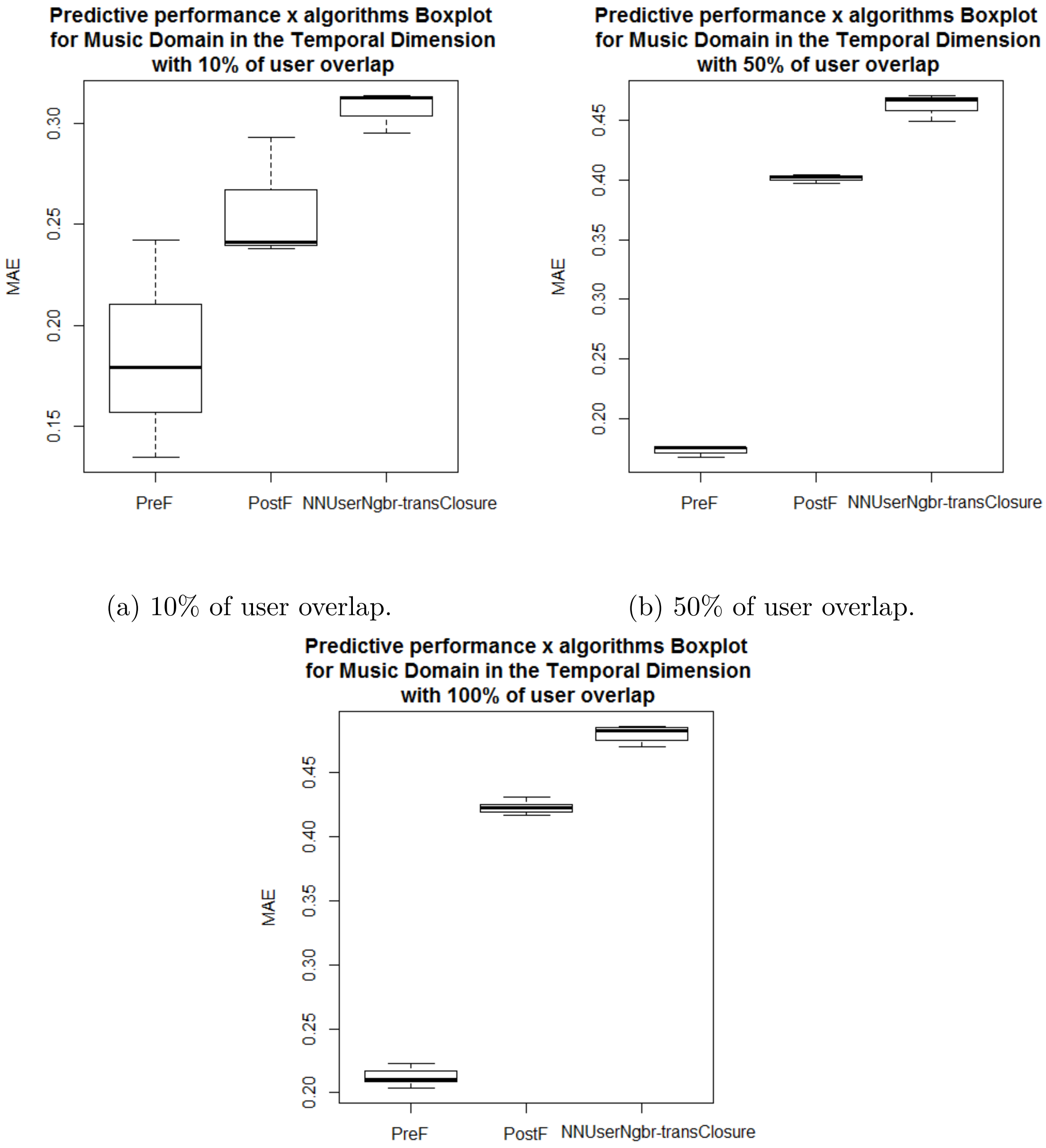


Figure 67 – Overall prediction error (RMSE) for cross-domain algorithms by varying user overlap level in the temporal dimension (source domain: book, and target domain: Music).

10%, 50%, and 100% of user overlap level for the *Music* domain as target, considering the *Temporal* dimension. As it can be seen, in all user overlap levels and top 'N' values, the PostF classification performance was better or similar than the baseline algorithms, whereas the PreF was only better than the baseline ones for 50% and 100% of user overlap levels with low top 'N' values. In addition, the PostF classification performance was better or similar than the PreF one for all user overlap levels and top 'N' values.



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

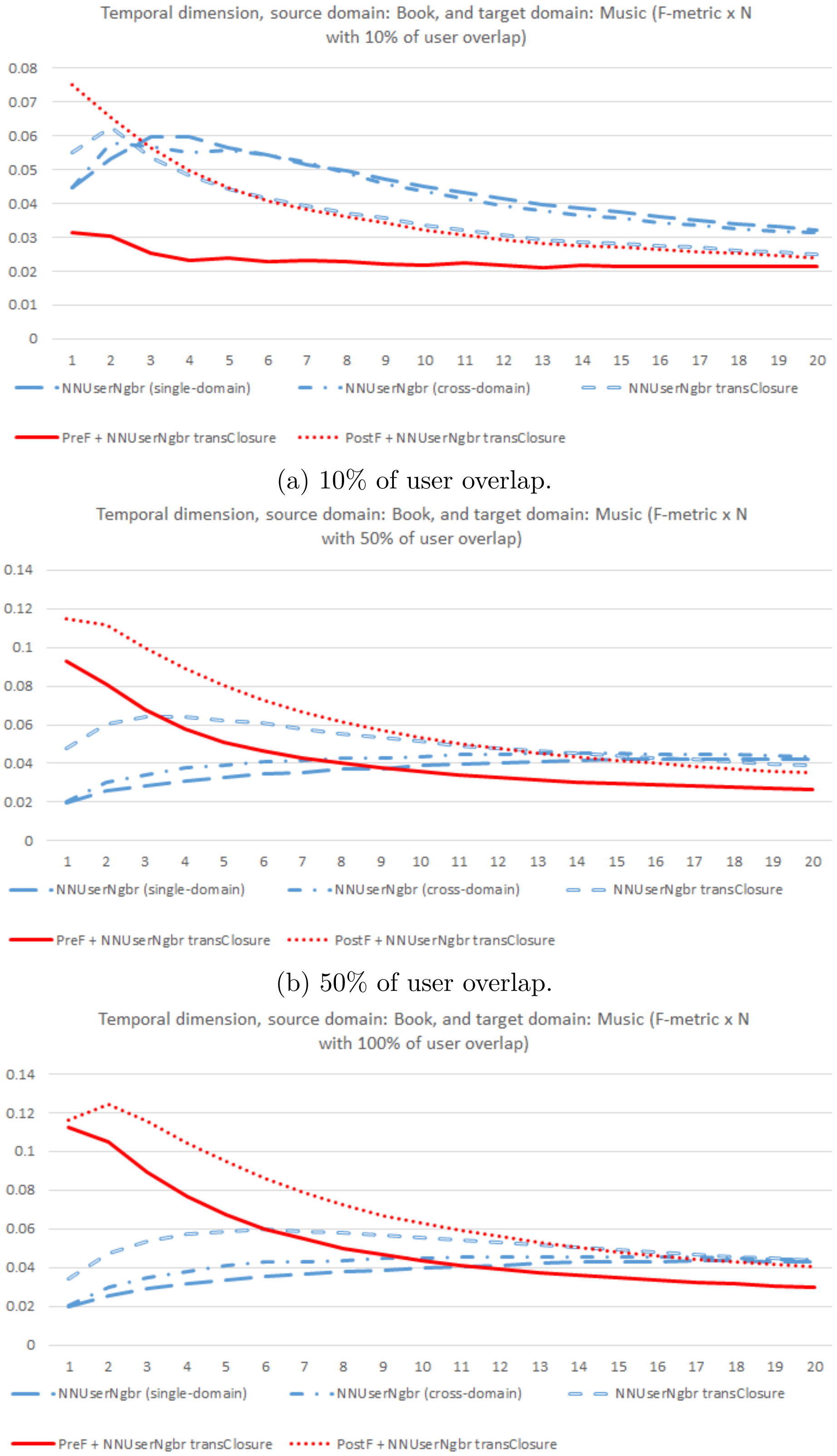
Figure 68 – Overall prediction performance (MAE) boxplots for Music domain in the temporal dimension with different user overlap levels (source domain: book).

Figure 70 shows the variation of the F-metric value in different user overlap levels by fixing the top 'N' value to five. The statistical significance tests verified that the PostF F-metric values were greater than the *NNUserNgbr-transClosure* baseline algorithm for 50% and 100% user overlap levels[53], whereas for 10% of user overlap the tests could not verify any statistical difference between their performances [54]. Besides, the PreF F-metric value was greater than the *NNUserNgbr-transClosure* one for 100% of user overlap[55],

[53] p-value=0.005413 and W=99 in both cases

[54] W=80 and p-value=0.1

[55] p-value=0.005413 and W=99



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 69 – F-metric performance x top ‘N’ items for the Music domain in the temporal dimension with different user overlap levels (source domain: book).

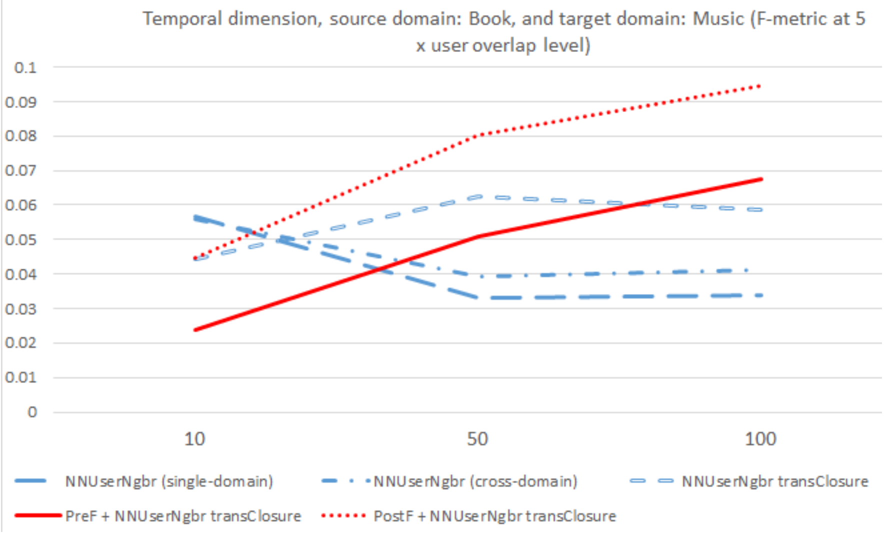


Figure 70 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the temporal dimension (target domain: Music, and source: book).

whereas the opposite from this was observed for 10% and 50% of user overlap levels[56].

5.2.2.1.2 Location Dimension

Table 30 reports the overall predictive performance of the recommender algorithms, considering all contextual values from the *Location* dimension and different user overlap levels for the *Music* domain as target. As it can be seen from the table, the addition of user ratings from other domain (*Book*), by using the same algorithm for cross-domain recommendation (corresponding to the two first rows of the table), improved the predictive performance in, approximately, 6–7% (MAE) and 3–4% (RMSE) depending on the user overlap level (50% or 100%). Note that the *NNUserNgbr* predictive performance was better than its own performance, considering the cross-domain scenario, when the user overlap level was 10%.

Also, Table 30 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 11–54% (MAE) and 6–32% (RMSE) depending on the user overlap levels. As it can be seen from table, the PostF predictive performance was better than the *NNUserNgbr-transClosure* algorithm in all user overlap levels, with an improvement that varied in, approximately, 3–16% (MAE) and 4–19% (RMSE) depending on the user overlap level.

In addition, if we consider the high standard deviation of the PreF algorithm showed in Table 30, then we can say that the PostF outperformed the PreF algorithm for

[56] p-value=0.006812 and W=97 for both tests

Table 30 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Location dimension (source domain: Book, and target domain: Music).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.684 ± 0.026	0.972 ± 0.053	0.654 ± 0.016	0.970 ± 0.030	0.588 ± 0.010	0.903 ± 0.017
NNUserNgbr (cross-domain)	0.710 ± 0.022	1.082 ± 0.051	0.602 ± 0.013	0.932 ± 0.028	0.552 ± 0.008	0.877 ± 0.014
NNUserNgbr-transClosure	0.326 ± 0.065	0.731 ± 0.084	0.468 ± 0.008	0.808 ± 0.013	0.487 ± 0.004	0.822 ± 0.008
PreF with NNUserNgbr-transClosure	**0.246 ± 0.238**	**0.528 ± 0.321**	**0.284 ± 0.022**	**0.514 ± 0.020**	**0.208 ± 0.033**	**0.541 ± 0.130**
PostF with NNUserNgbr-transClosure	**0.273 ± 0.011**	**0.591 ± 0.027**	0.437 ± 0.007	0.775 ± 0.003	0.473 ± 0.005	0.791 ± 0.014

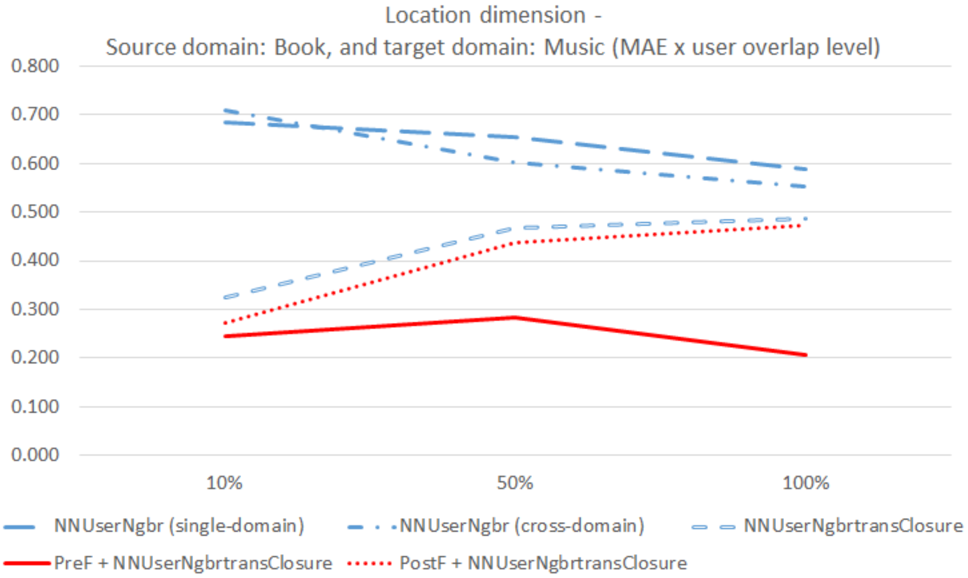
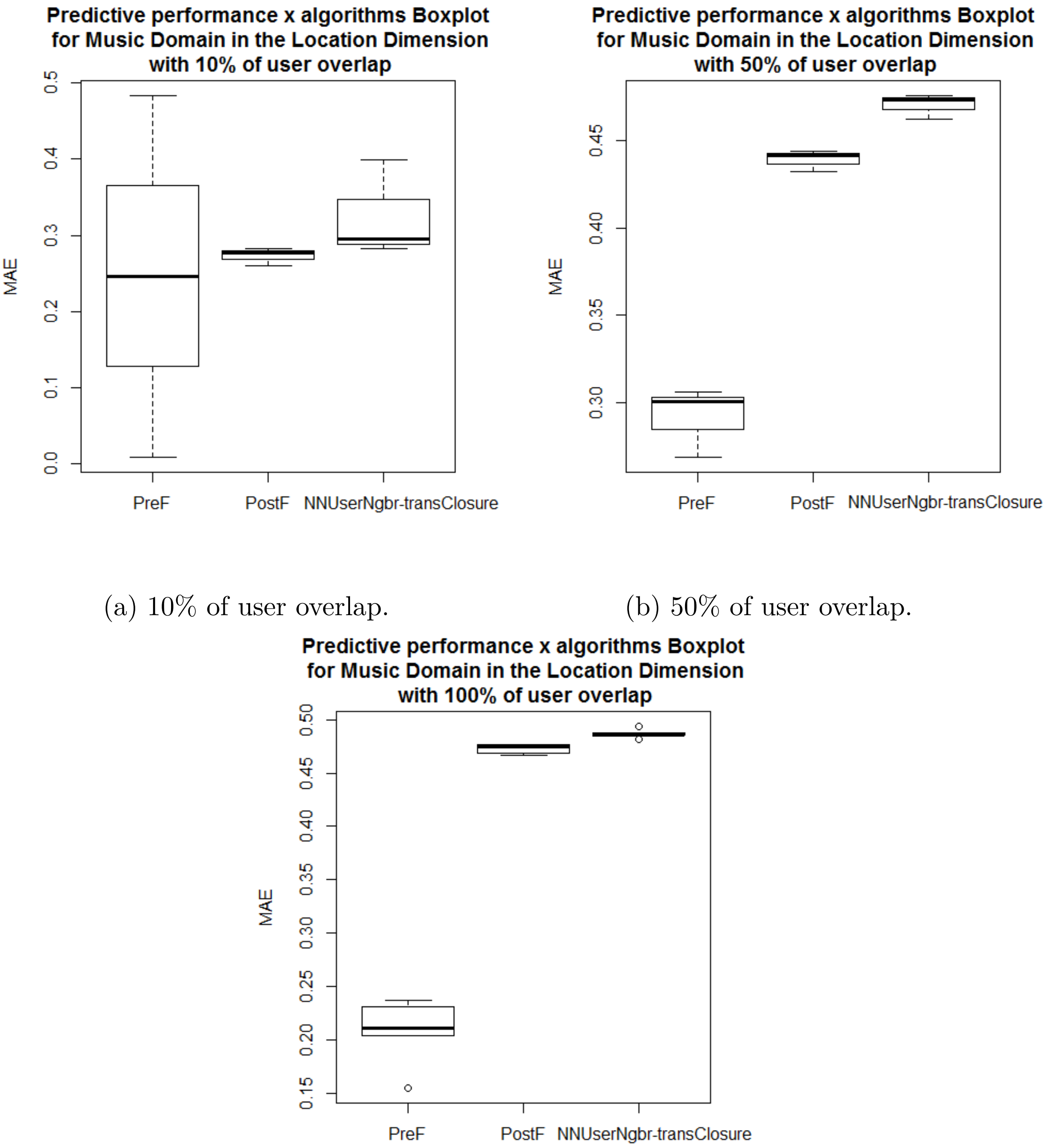


Figure 71 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the location dimension (source domain: book, and target domain: Music).

10% of user overlap level. For 50% and 100% of user overlap levels, the PreF algorithm had a low standard deviation and achieved the best predictive performance among the algorithms. Figure 71 illustrates the predictive performance (MAE) of the proposed algorithms over different user overlap levels.

The statistical significance tests verified that both the PreF and PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* baseline algorithm for 50%



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 72 – Overall prediction performance (MAE) boxplots for Music domain in the location dimension with different user overlap levels (source domain: book).

and 100% of user overlap levels[57]. For 10% of user overlap, the applied tests could not verify a statistical difference between the performance of the proposed algorithms and *NNUserNgbr-transClosure*[58]. Figure 72a, Figure 72b and Figure 72c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

[57] For all tests, W=99 and p-value=0.005413

[58] W=60 and p-value=0.35 (PreF x NNUserNgbr-transClosure), while W=80 and p-value=0.1 (PostF x NNUserNgbr-transClosure)

With respect to the classification performance, Figures 73a, 73b, and 73c present the results of the F-metric at different top ‘N’ values (between one and twenty), respectively, in 10%, 50%, and 100% of user overlap level for the *Music* domain as target, considering the *Location* dimension. As it can be seen, for 50% and 100% of user overlap the PostF outperformed the baseline algorithms for ‘N’ up to ten. On the other hand, the PreF classification performance was worse than all other algorithms in all user overlap levels and ‘N’ values.

Figure 74 shows the variation of the F-metric value in different user overlap levels by fixing the top ‘N’ value to five. The statistical significance tests verified that the PostF F-metric values were greater than the *NNUserNgbr-transClosure* baseline algorithm for 50% and 100% of user overlap levels[59], whereas the opposite from this was observed for 10% of user overlap[60]. On the other hand, the applied tests also verified that the *NNUserNgbr-transClosure* F-metric values were greater than the PreF ones for all user overlap levels[61].

5.2.2.1.3 Companion Dimension

Table 31 shows the overall predictive performance of the recommender algorithms, considering all contextual values from the *Companion* dimension and different user overlap levels for the *Music* domain as target. As it can be seen from the table, the addition of user ratings from other domain (*Book*), by using the same algorithm for cross-domain recommendation (corresponding to the two first rows of the table), improved the predictive performance in, approximately, 3–17% (MAE) and 6–16% (RMSE) depending on the user overlap level.

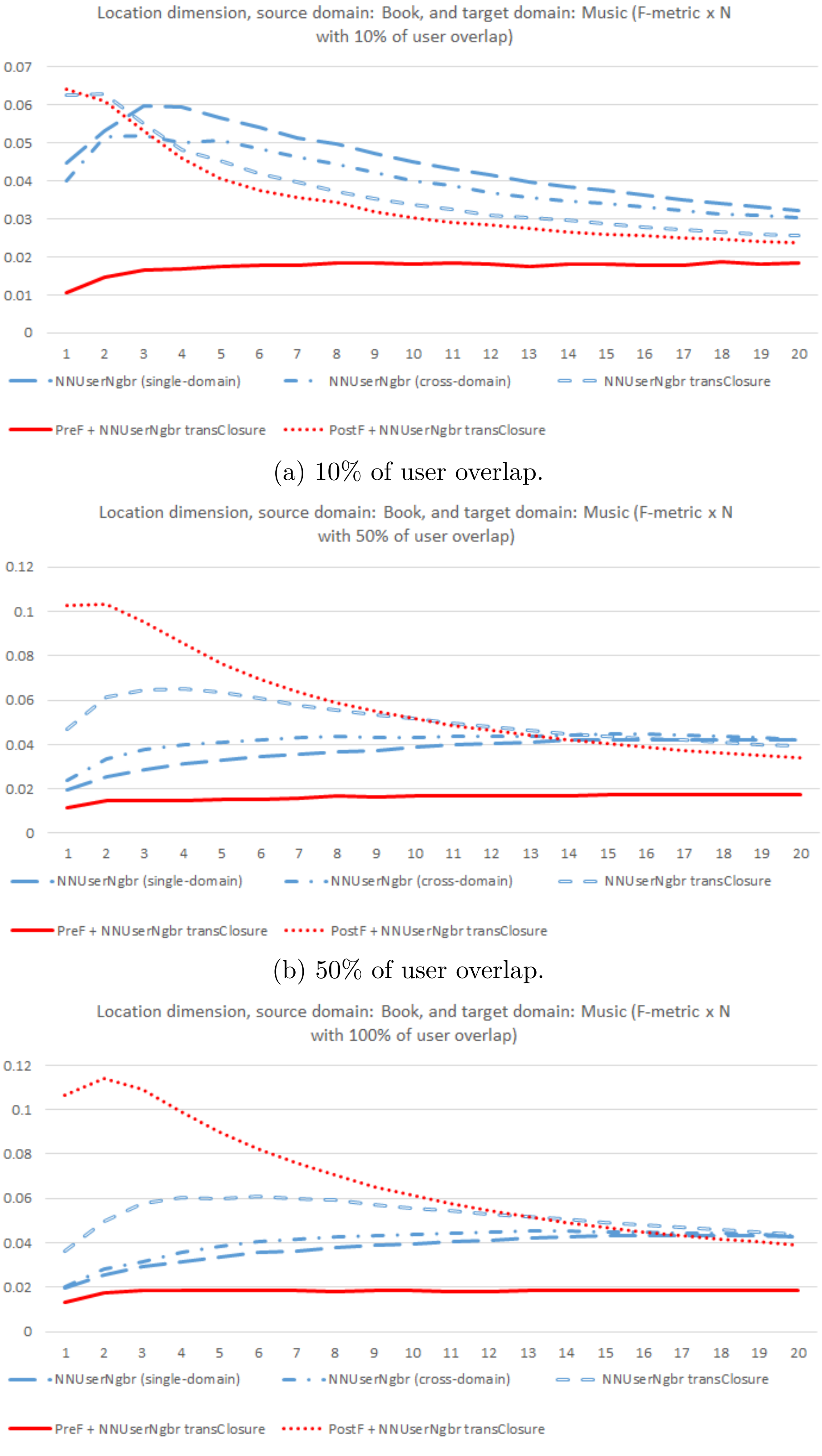
Also, Table 31 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 9–68% (MAE) and 4–47% (RMSE) depending on the user overlap levels. As it can be seen from table, the PostF predictive performance was better than the *NNUserNgbr-transClosure* algorithm for 50% and 100% of user overlap levels if we do not consider the standard deviation of the algorithms for 50% of user overlap. Its improvement in comparison to the *NNUserNgbr-transClosure* performance varied in, approximately, 3–11% (MAE) and 6–11% (RMSE) depending on the user overlap level.

In addition, the predictive performance of the PreF algorithm was worse than all other algorithms in the *Companion* dimension for 50% and 100% of user overlap levels, as showed in Table 31. Figure 75 (MAE) and Figure 76 (RMSE) illustrate the predictive

[59] p-value=0.005413 and W=99 for all tests

[60] p-value=0.006314 and W=97

[61] p-value=0.003912 and W=100 for all tests



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 73 – F-metric performance x top ‘N’ items for the Music domain in the location dimension with different user overlap levels (source domain: book).

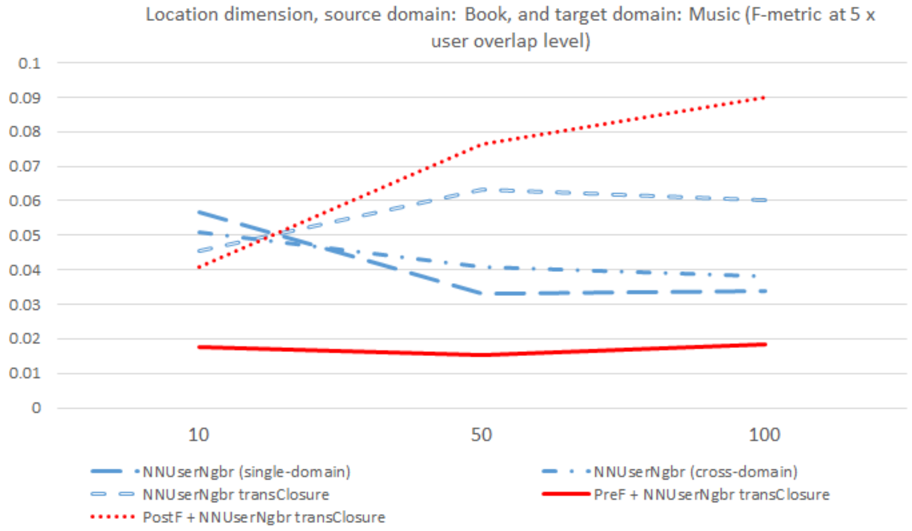


Figure 74 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the location dimension (target domain: Music, and source: book).

Table 31 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Companion dimension (source domain: Book, and target domain: Music).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.684 ± 0.026	0.972 ± 0.053	0.654 ± 0.016	0.970 ± 0.030	0.588 ± 0.010	0.903 ± 0.017
NNUserNgbr (cross-domain)	0.660 ± 0.022	0.818 ± 0.051	0.537 ± 0.013	0.823 ± 0.028	0.535 ± 0.008	0.845 ± 0.014
NNUserNgbr-transClosure	**0.205** ± **0.122**	**0.430** ± **0.188**	**0.479** ± **0.004**	**0.785** ± **0.014**	0.486 ± 0.018	0.797 ± 0.022
PreF with NNUserNgbr-transClosure	0.438 ± 0.221	0.540 ± 0.283	0.666 ± 0.058	1.076 ± 0.024	0.711 ± 0.052	1.079 ± 0.090
PostF with NNUserNgbr-transClosure	0.381 ± 0.076	0.807 ± 0.279	**0.465** ± **0.044**	**0.736** ± **0.102**	**0.431** ± **0.016**	**0.707** ± **0.016**

performance of the proposed algorithms over different user overlap levels. Note that for this case we showed the figures for both predictive metrics, since we observed a difference in the PreF and PostF performances depending on the predictive metric used in the evaluation.

The statistical significance tests verified that the PostF predictive error (MAE) was less than the *NNUserNgbr-transClosure* algorithm for 100% of user overlap level[62], whereas none statistical difference between their performances was observed when the user

[62] W=100 and p-value=0.003968

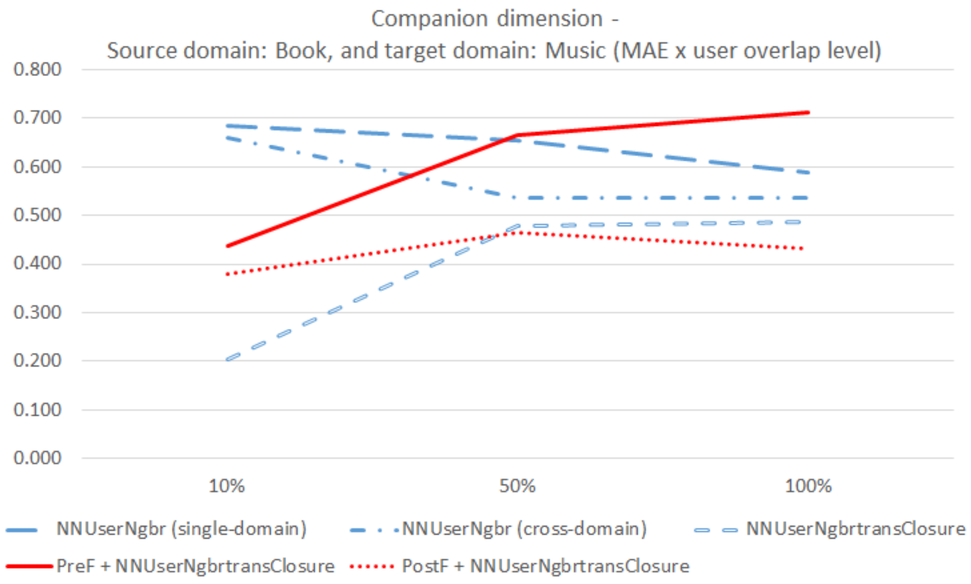


Figure 75 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the companion dimension (source domain: book, and target domain: Music).

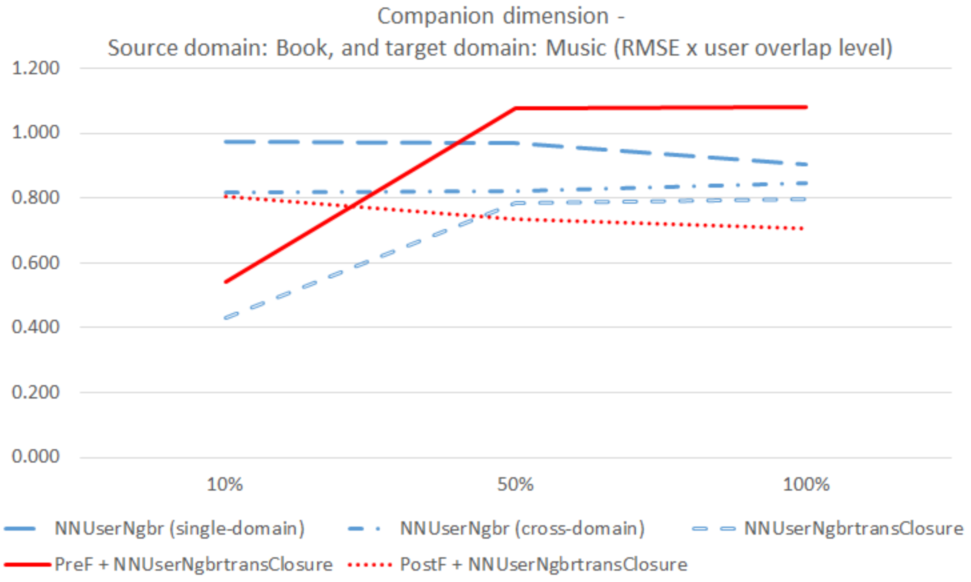


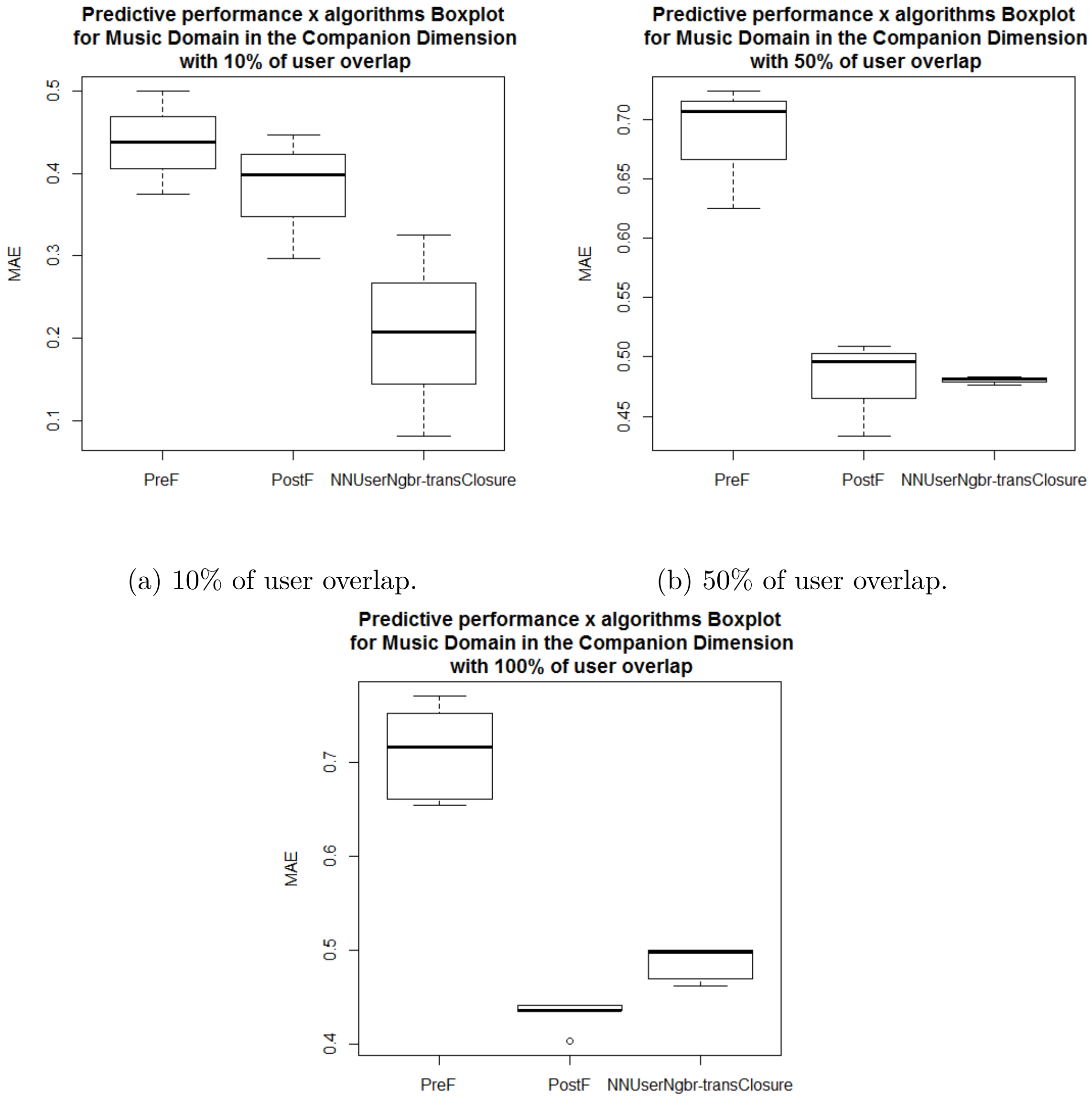
Figure 76 – Overall prediction error (RMSE) for cross-domain algorithms by varying user overlap level in the companion dimension (source domain: book, and target domain: Music).

overlap levels were 10%[63] and 50%[64]. The applied tests also verified that the *NNUserNgbrtransClosure* predictive errors (MAE) were less than the PreF ones for all user overlap levels[65]. Figure 77a, Figure 77b and Figure 77c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

[63] W=55 and p-value=0.5

[64] W=30 and p-value=0.8

[65] W=100 and p-value=0.003968 for all tests



(a) 10% of user overlap.

(b) 50% of user overlap.

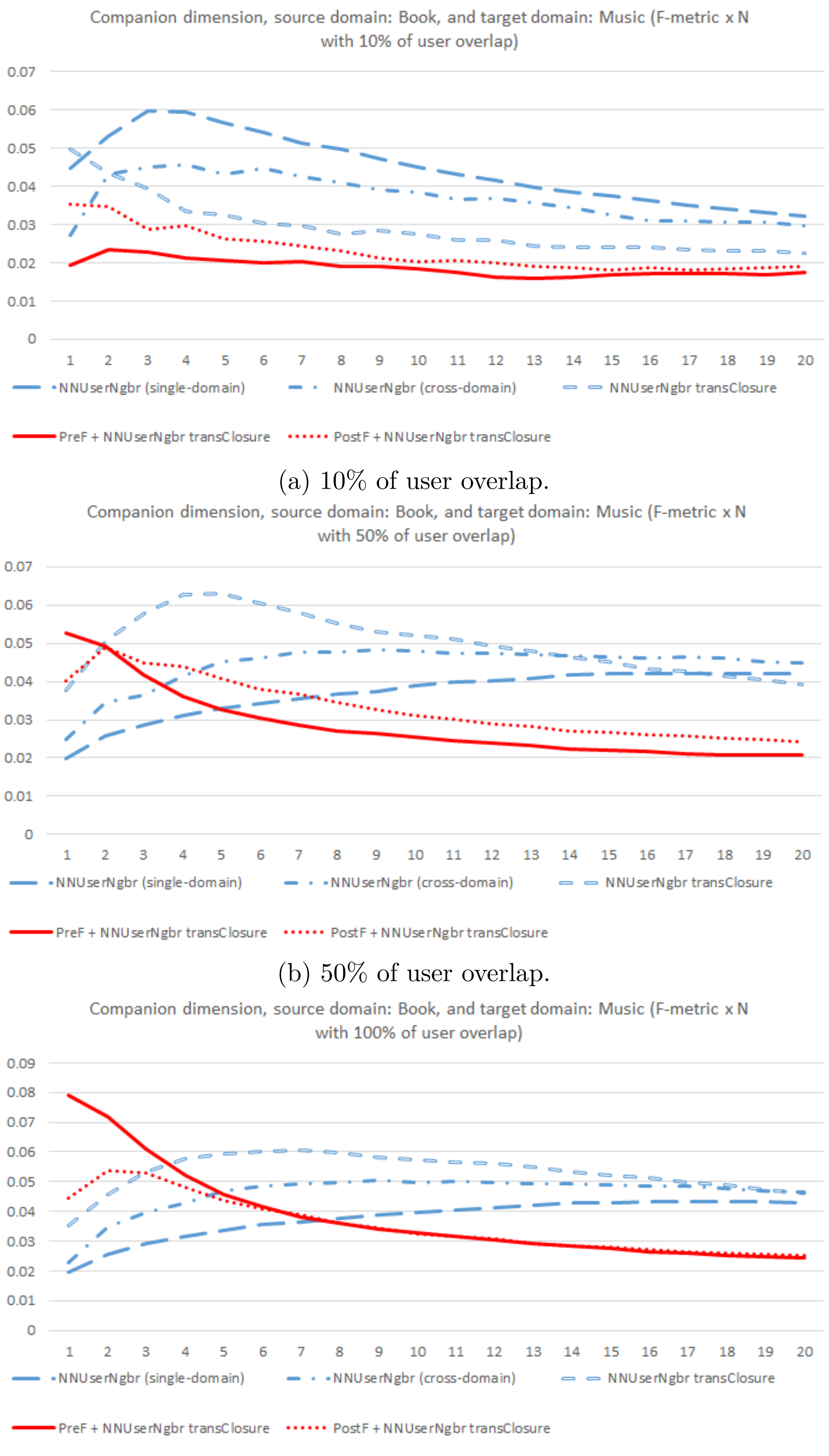
(c) 100% of user overlap.

Figure 77 – Overall prediction performance (MAE) boxplots for Music domain in the companion dimension with different user overlap levels (source domain: book).

Figures 78a, 78b, and 78c present the results of the F-metric at different top ‘N’ values (between one and twenty), respectively, in 10%, 50%, and 100% of user overlap level for the *Music* domain as target, considering the *Companion* dimension. As it can be seen, the proposed algorithms only outperformed the baseline ones for 50% and 100% of user overlap with very low values of top ‘N’. Figure 79 shows the variation of the F-metric value in different user overlap levels by fixing the top ‘N’ value to five.

The statistical significance tests verified that the *NNUserNgbr-transClosure* F-metric values were greater than the both proposed algorithms for all user overlap levels[66].

[66] p-value=0.005814 and W=97 for all tests between the PostF and baseline algorithms, while p-



Companion dimension, source domain: Book, and target domain: Music (F-metric x N with 10% of user overlap)

(a) 10% of user overlap.

Companion dimension, source domain: Book, and target domain: Music (F-metric x N with 50% of user overlap)

(b) 50% of user overlap.

Companion dimension, source domain: Book, and target domain: Music (F-metric x N with 100% of user overlap)

(c) 100% of user overlap.

Figure 78 – F-metric performance x top 'N' items for the Music domain in the companion dimension with different user overlap levels (source domain: book).

value=0.003913 and W=100 for all tests between the PreF and baseline algorithms

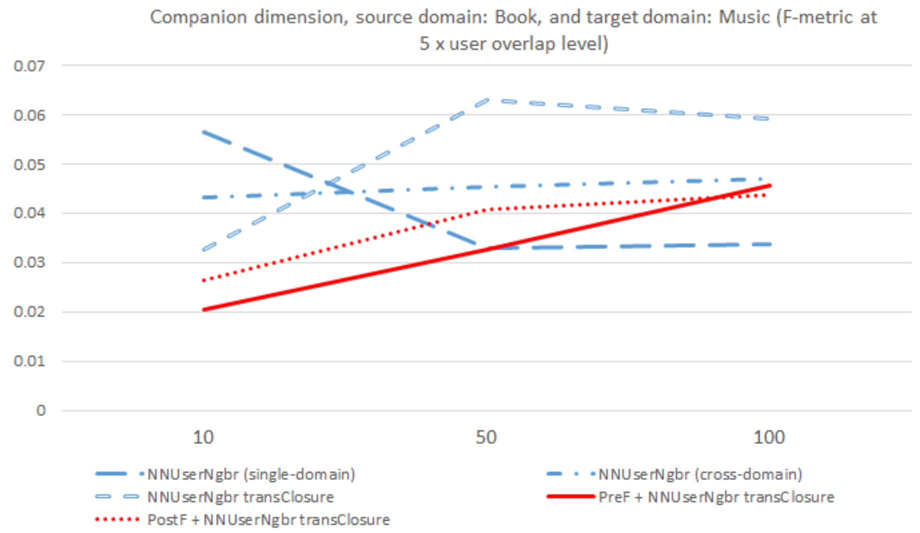


Figure 79 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the companion dimension (target domain: Music, and source: book).

5.2.2.1.4 Combining Contextual Dimensions

In the previous sections, we presented the evaluation results regarding the contextual dimensions separately. In this section, we report the results for a combination of two contextual dimensions considering the same evaluation metrics and methodology described before.

Table 32 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual value combinations from the temporal and location dimensions (source domain: Book, and target domain: Music).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.684 ± 0.026	0.972 ± 0.053	0.654 ± 0.016	0.970 ± 0.030	0.588 ± 0.010	0.903 ± 0.017
NNUserNgbr (cross-domain)	0.708 ± 0.022	1.076 ± 0.051	0.597 ± 0.013	0.929 ± 0.028	0.548 ± 0.008	0.874 ± 0.014
NNUserNgbr-transClosure	**0.302** ± **0.060**	**0.628** ± **0.078**	0.474 ± 0.009	0.818 ± 0.014	0.489 ± 0.004	0.825 ± 0.008
PreF with NNUserNgbr-transClosure	**0.302** ± **0.278**	0.755 ± 0.426	0.485 ± 0.040	0.741 ± 0.036	**0.265** ± **0.036**	**0.642** ± **0.156**
PostF with NNUserNgbr-transClosure	**0.304** ± **0.021**	**0.632** ± **0.036**	**0.338** ± **0.005**	**0.621** ± **0.002**	0.376 ± 0.004	**0.706** ± **0.010**

Table 32 reports the overall predictive performance of the recommender algorithms, considering *all contextual value combinations* from the *Temporal* and *Location* dimensions with different user overlap levels for the *Music* domain as target. As it was observed in the previous sections, the addition of user ratings from the *Book* domain also improved the predictive performance of the *NNUserNgbr* algorithm in, approximately, 6–8% (MAE) and 3–4% (RMSE) depending on the user overlap level (for 50% and 100% of user overlap).

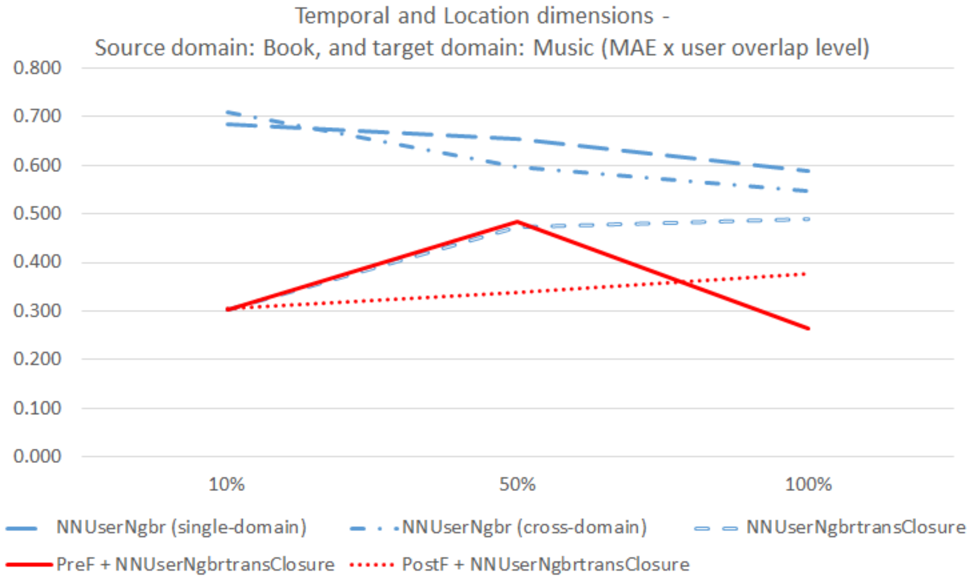


Figure 80 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the temporal and location dimensions (source domain: book, and target domain: music).

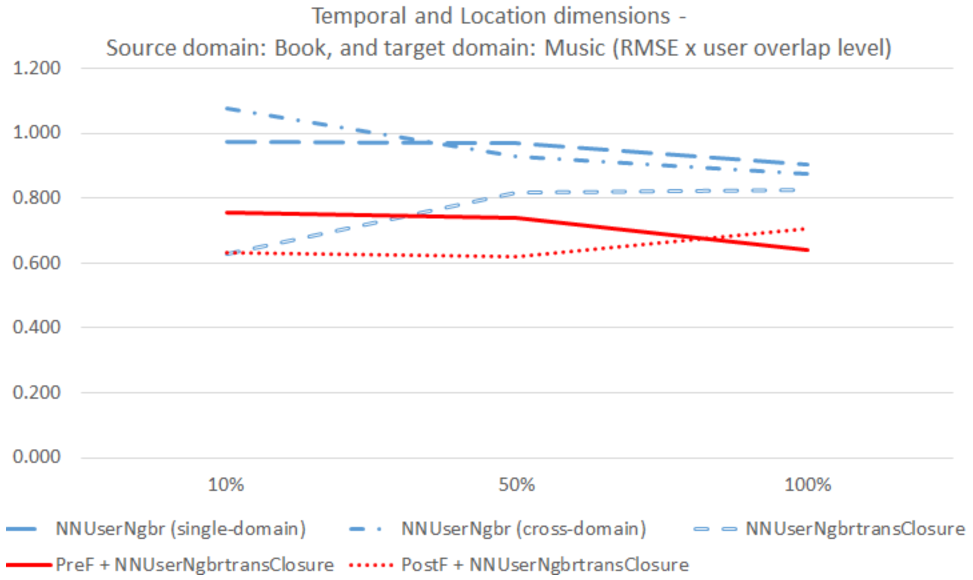


Figure 81 – Overall prediction error (RMSE) for cross-domain algorithms by varying user overlap level in the temporal and location dimensions (source domain: book, and target domain: music).

Furthermore, Table 32 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUser-Ngbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 10–57% (MAE) and 5–41% (RMSE) depending on the user overlap levels. As it can be seen from table, except when the user overlap level was 10%, the PostF predictive performance was better than the *NNUserNgbr-transClosure* algorithm in all other user overlap levels, with an improvement that varied in, approximately, 23–28% (MAE) and 14–24% (RMSE) depending on the user overlap level. Despite the *NNUserNgbr-transClosure* algorithm have outperformed the PostF when the user overlap level was 10%, they had a similar performance, separated only by their standard deviations.

Note that the PreF predictive performance had a high standard deviation, observed in Table 32, as occurred in the evaluation of the *Location* dimension alone. In addition, if we consider the high standard deviation of the PreF algorithm showed in Table 32, then we can say that the PostF outperformed the PreF algorithm for 10% and 50% of user overlap level. For 100% of user overlap, the PreF algorithm had a low standard deviation and achieved the best predictive performance among the algorithms.

Figure 80 (MAE) and Figure 81 (RMSE) illustrate the predictive performance of the proposed algorithms over different user overlap levels. Note that for this case we showed the figures for both predictive metrics, since we observed a difference in the PreF and PostF performances depending on the predictive metric used in the evaluation.

The statistical significance tests verified that the PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* algorithm for the 50% and 100% user overlap levels[67]. The applied tests also verified that the PreF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* algorithm for 100% user overlap[68], whereas the opposite from this was observed for 50% of user overlap[69]. When the user overlap level was 10%, the applied tests could not verify a statistical difference between the performance of the proposed algorithms and *NNUserNgbr-transClosure*[70]. Figure 82a, Figure 82b and Figure 82c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

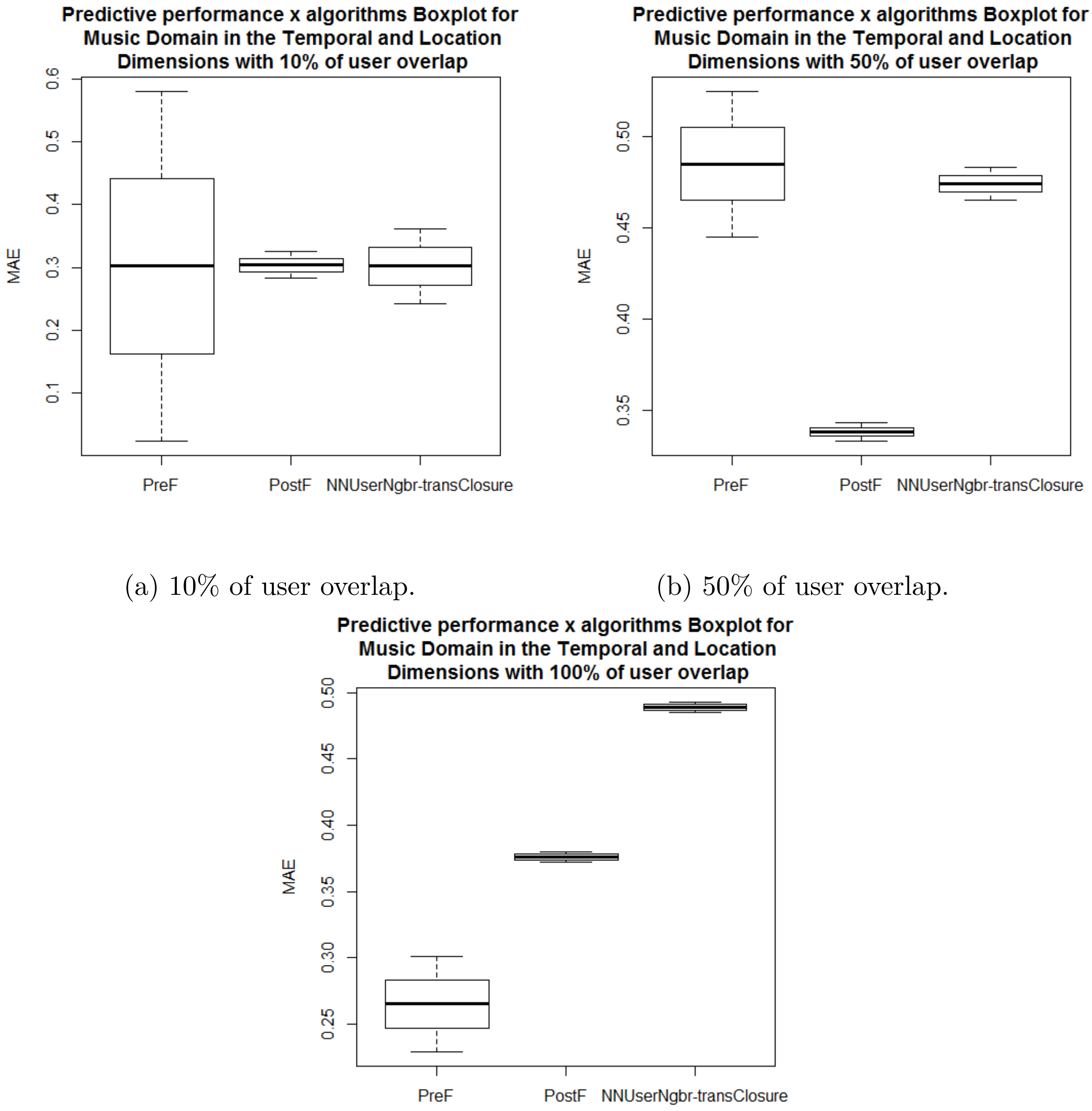
Taking into account the classification performance, Figures 83a, 83b, and 83c present the results of the F-metric at different top 'N' values, respectively, in 10%, 50%, and 100% of user overlap level for the *Music* domain as target, considering the *Temporal* and *Location* dimensions. For 50% and 100% of user overlap levels, the PostF classification performance was better or similar than the baseline algorithms for all top 'N' values, while for 10% of user overlap the PostF classification performance was better or similar than them for very low top 'N' values (1 to 3). On other hand, the PreF classification

[67] In both cases, with W=99 and p-value=0.005413

[68] W=100 and p-value=0.003968

[69] W=99 and p-value=0.005413

[70] W=61 and p-value=0.5 (PostF x *NNUserNgbr-transClosure*), whereas W=71 and p-value=0.8 (PreF x *NNUserNgbr-transClosure*)



(a) 10% of user overlap.

(b) 50% of user overlap.

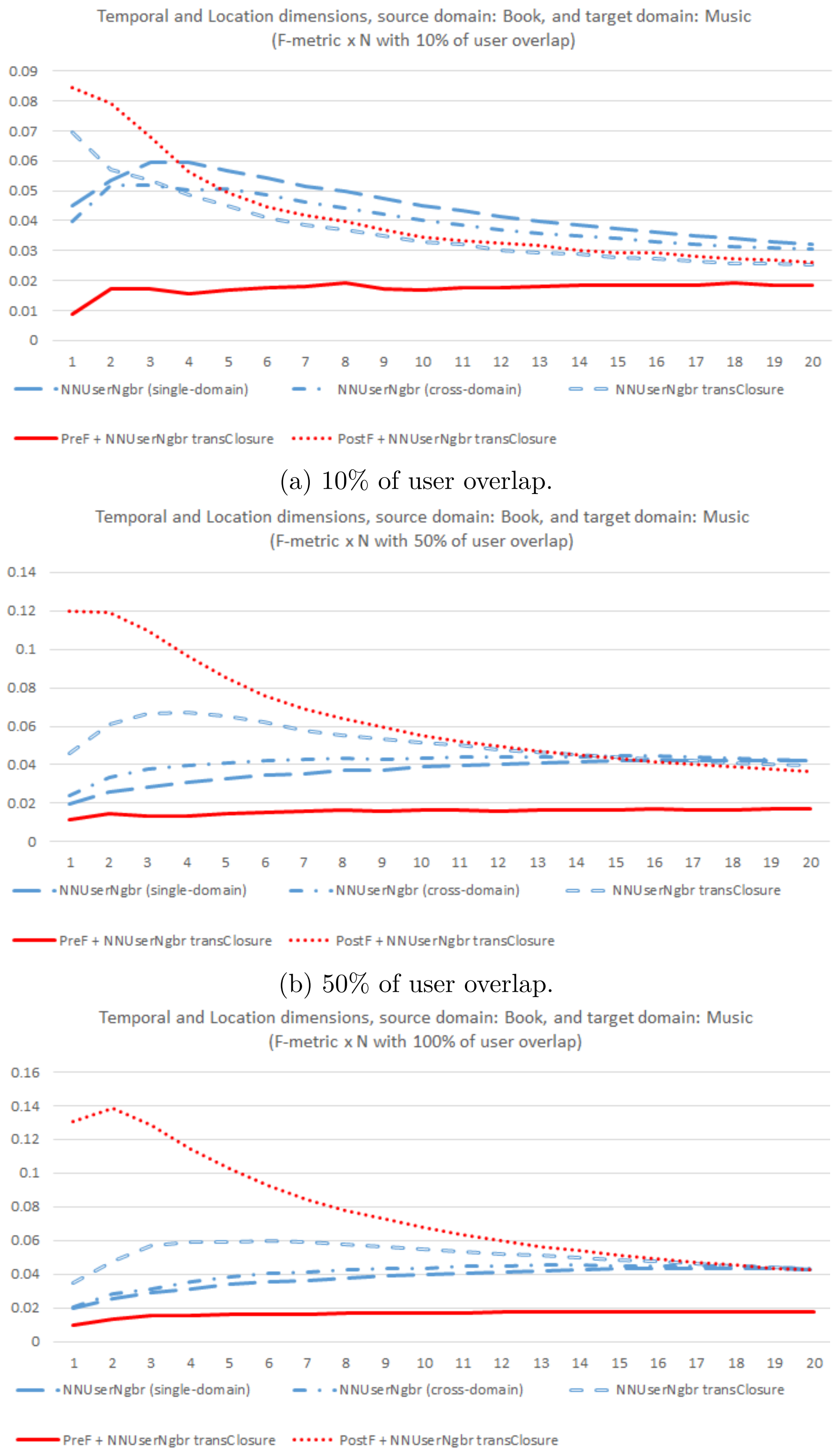
(c) 100% of user overlap.

Figure 82 – Overall prediction performance (MAE) boxplots for Music domain in the temporal and location dimensions with different user overlap levels (source domain: book).

performance was worse than all other algorithms.

Figure 84 shows the variation of the F-metric value in different user overlap levels by fixing the top ‘N’ value to five. The statistical significance tests verified that the PostF F-metric values were greater than the *NNUserNgbr-transClosure* baseline algorithm for all user overlap levels[71]. On the other hand, the applied tests also verified that the *NNUserNgbr-transClosure* F-metric values were greater than the PreF ones for all user

[71] For 10% of user overlap, with W=97 and the p-value=0.037, and p-value=0.005413 and W=97 for the other user overlap levels



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 83 – F-metric performance x top ‘N’ items for the Music domain in the temporal and location dimensions with different user overlap levels (source domain: book).

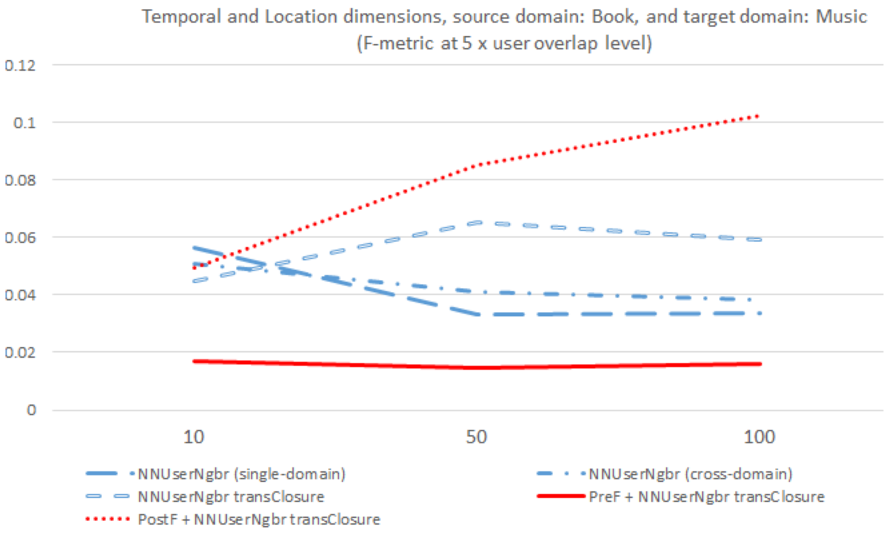


Figure 84 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the temporal and location dimensions (target domain: Music, and source: book).

overlap levels[72].

5.2.2.2 Book as Target Domain

In the Section 5.2.2.1, we presented the results for the *Music* target domain, which had fewer ratings in the cross-domain dataset in comparison to *Book* source domain (as described in Section 4.1.3.2). In this section, we present the results when *Book* is the target domain and *Music* is the source domain.

According to the contextual dimensions present in Section 4.1.3.2, we describe the evaluation results for each contextual dimension in the following sections. In addition, we show the results for a combination of contextual dimensions in Section 5.2.2.2.4.

5.2.2.2.1 Temporal Dimension

Table 33 reports the overall predictive performance of the recommender algorithms, considering all contextual values from the *Temporal* dimension and different user overlap levels for the *Book* domain as target. The rows 1 and 2 from the table show the *NNUserNgbr* predictive performance when it is applied in the single-domain and cross-domain recommendations. As it can be seen, likewise the results showed in Section 5.2.2.1.1 (source domain: *Book*, target domain: *Music*, and *Temporal* dimension), the simple addition of user ratings from other domain (*Music*), by using the same algorithm for cross-domain recommendation, improved the recommendation performance in, approximately, 9–13% (MAE) and 6–7% (RMSE) depending on the user overlap level.

[72] p-value=0.003914 and W=99 for all tests

Table 33 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Temporal dimension (source domain: Music, and target domain: Book).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.671 ± 0.024	1.002 ± 0.048	0.505 ± 0.008	0.792 ± 0.020	0.419 ± 0.006	0.735 ± 0.012
NNUserNgbr (cross-domain)	0.610 ± 0.022	0.942 ± 0.044	0.443 ± 0.007	0.740 ± 0.018	0.363 ± 0.005	0.679 ± 0.010
NNUserNgbr-transClosure	0.130 ± 0.012	0.368 ± 0.034	0.197 ± 0.001	0.451 ± 0.005	0.202 ± 0.003	0.473 ± 0.006
PreF with NNUserNgbr-transClosure	**0.109 ± 0.018**	**0.346 ± 0.063**	**0.087 ± 0.005**	**0.340 ± 0.006**	**0.090 ± 0.002**	**0.300 ± 0.005**
PostF with NNUserNgbr-transClosure	**0.114 ± 0.004**	**0.349 ± 0.011**	0.176 ± 0.002	0.399 ± 0.004	0.175 ± 0.003	0.418 ± 0.007

In addition, Table 33 presents the overall performance of the PreF and PostF algorithms, besides the base *NNUserNgbr-transClosure* algorithm, which outperformed the *NNUserNgbr* algorithm (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 44–78% (MAE) and 30–60% (RMSE) depending on the user overlap levels.

As it can be seen from the table, the PreF predictive performance was better than the *NNUserNgbr-transClosure* for all user overlap levels. Besides, it was better than the PostF algorithm for all user overlap levels if we do not consider the standard deviation. The improvement achieved by the PreF algorithm in comparison to the *NNUserNgbr-transClosure* one varied in, approximately, 16–56% (MAE) and 6–36% (RMSE) depending on the user overlap level. The PostF predictive performance was also better than the *NNUserNgbr-transClosure* for all user overlap levels. Figure 85 illustrates the predictive performance (MAE) of the proposed algorithms over different user overlap levels.

The statistical significance tests verified that the PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* baseline algorithm for all user overlap levels[73]. Also, the tests verified that, except for 10% of user overlap[74], the PreF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* algorithm for all other user overlap levels[75]. Figure 86a, Figure 86b and Figure 86c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

[73] p-value=0.003968 and W=97 for all tests

[74] W=79 and p-value=0.1

[75] p-value=0.003968 and W=97 for all tests

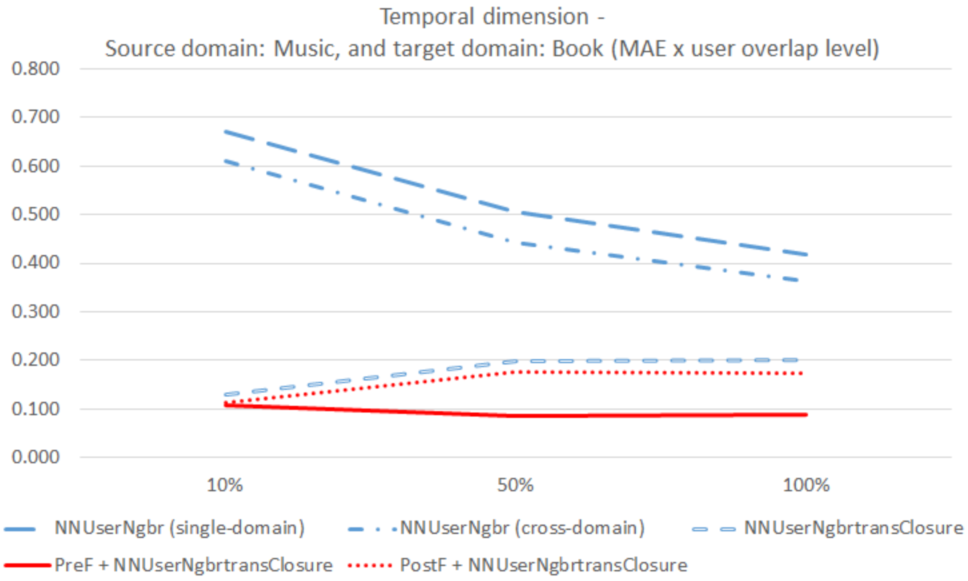


Figure 85 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the temporal dimension (source domain: Music, and target domain: book).

Regarding the classification performance, Figures 87a, 87b, and 87c present the results of the F-metric at different N values (between one and twenty), respectively, in 10%, 50%, and 100% of user overlap level for the *Book* domain as target, considering the *Temporal* dimension. As it can be seen, the PostF and PreF classification performances were better or similar than the *NNUserNgbr-transClosure* baseline algorithm for 50% and 100% of user overlap levels (with any top 'N' value for the PostF algorithm and with very low top 'N' values for the PreF one).

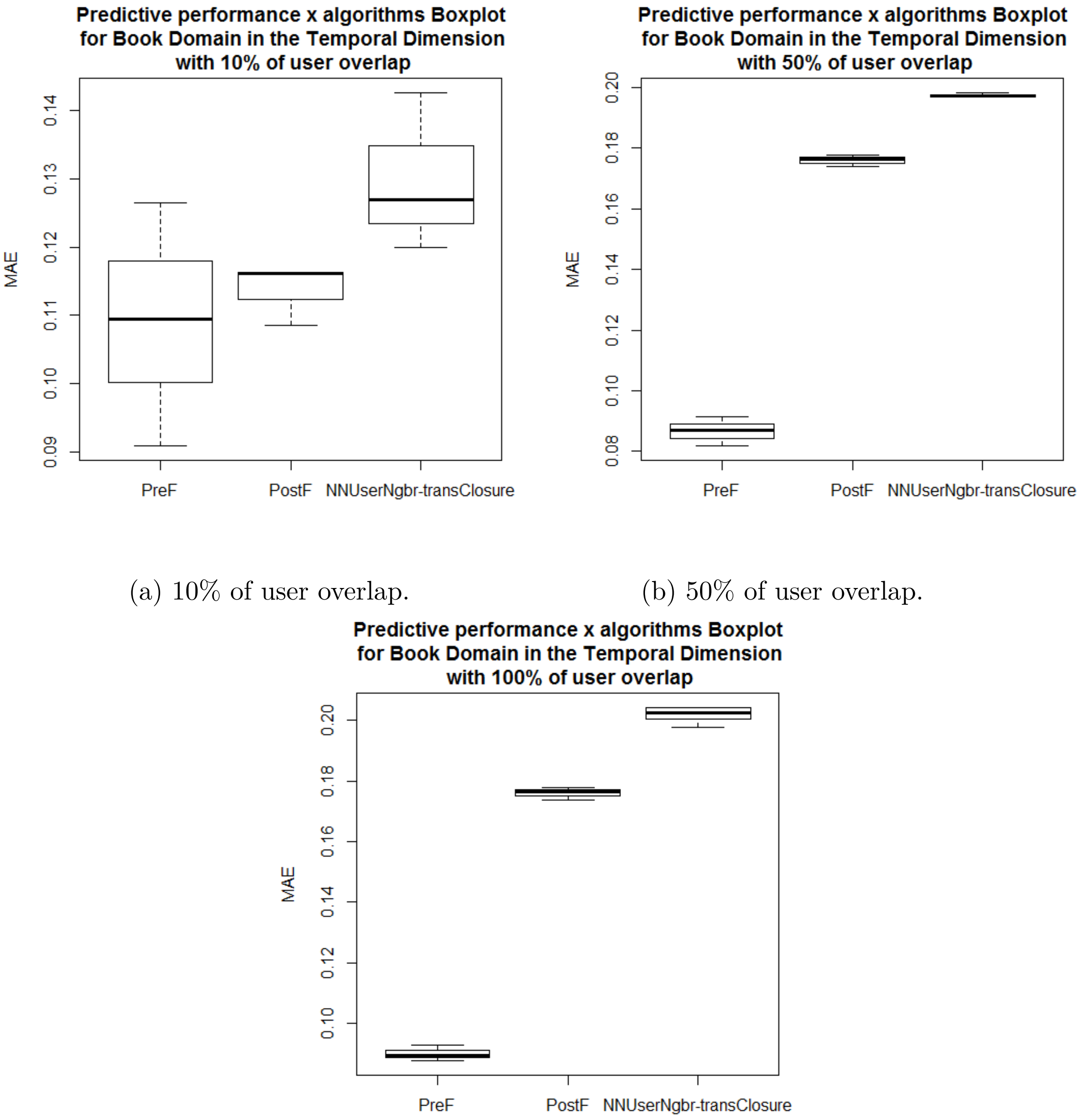
In addition, for 10% of user overlap, the PostF classification performance was better than the *NNUserNgbr-transClosure* baseline algorithm only for very low top 'N' values (1 and 2), whereas the PreF classification performance was worse than all algorithms. Finally, the PostF classification performance was better than the PreF one for all user overlap levels and top 'N' values.

Figure 88 shows the variation of the F-metric value in different user overlap levels by fixing the top 'N' value to five. The statistical significance tests verified that the PostF F-metric values were greater than the *NNUserNgbr-transClosure* baseline algorithm for 50% and 100% of user overlap levels[76], whereas their performances were statistically similar for 10% of user overlap[77]. The applied tests also verified that the *NNUserNgbr-transClosure* F-metric values were greater than the PreF ones for all user overlap levels[78].

[76] p-value=0.005413 and W=97 for all tests

[77] p-value=0.35 and W=60

[78] W=77 and p-value=0.02163 for 10% of user overlap. For 50% and 100%, p-value=0.005413 and W=97



(a) 10% of user overlap.

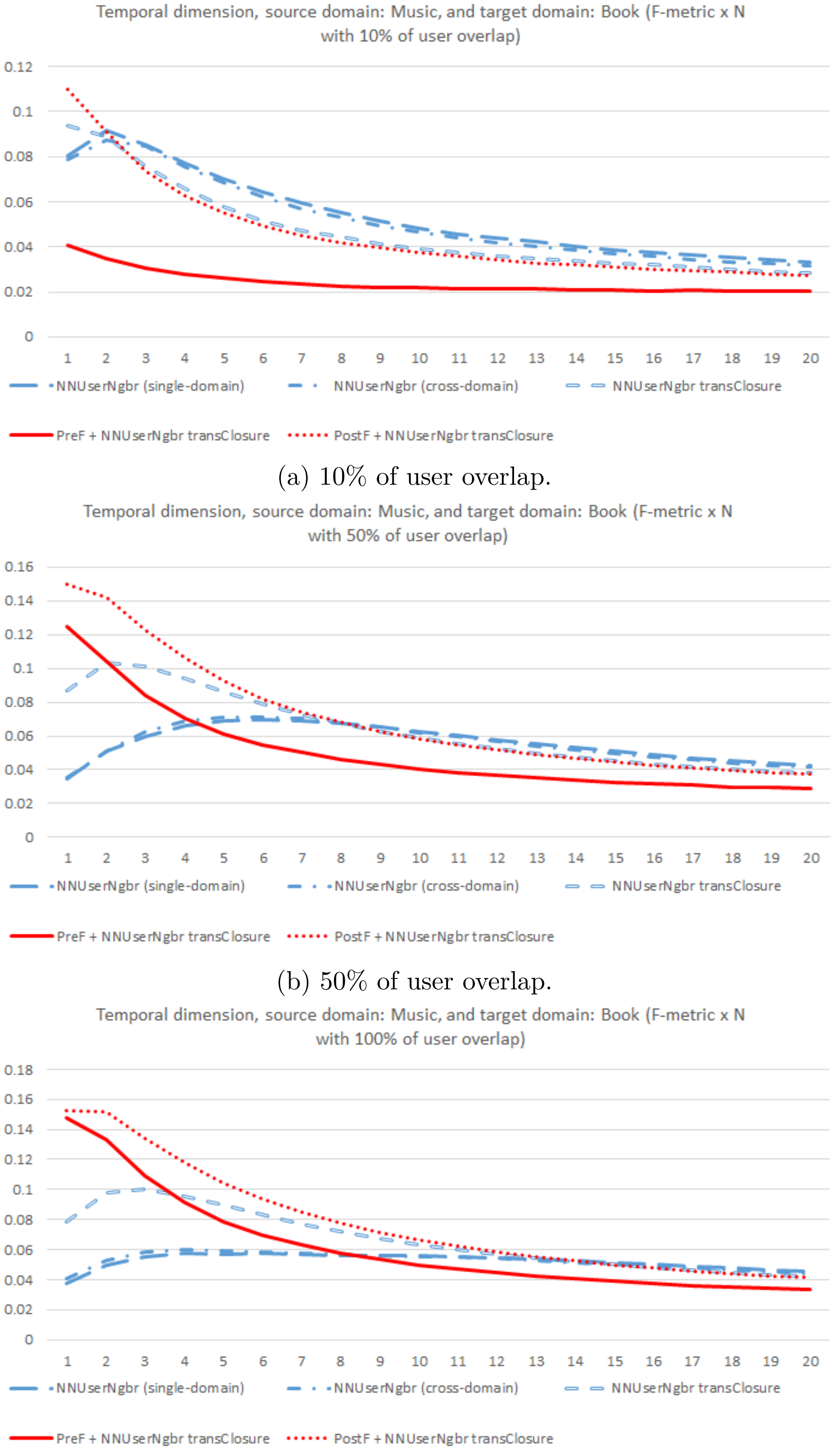
(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 86 – Overall prediction performance (MAE) boxplots for book domain in the temporal dimension with different user overlap levels (source domain: Music).

5.2.2.2.2 Location Dimension

Table 34 reports the overall predictive performance of the recommender algorithms, considering all contextual values from the *Location* dimension and different user overlap levels for the *Book* domain as target. As it can be seen from the table, the addition of user ratings from other domain (*Music*), by using the same algorithm for cross-domain recommendation (corresponding to the two first rows of the table), improved the predictive performance in, approximately, 10–12% (MAE) and 6–10% (RMSE) depending on the user overlap level.



Temporal dimension, source domain: Music, and target domain: Book (F-metric x N with 10% of user overlap)

(a) 10% of user overlap.

Temporal dimension, source domain: Music, and target domain: Book (F-metric x N with 50% of user overlap)

(b) 50% of user overlap.

Temporal dimension, source domain: Music, and target domain: Book (F-metric x N with 100% of user overlap)

(c) 100% of user overlap.

Figure 87 – F-metric performance x top 'N' items for the book domain in the temporal dimension with different user overlap levels (source domain: Music).

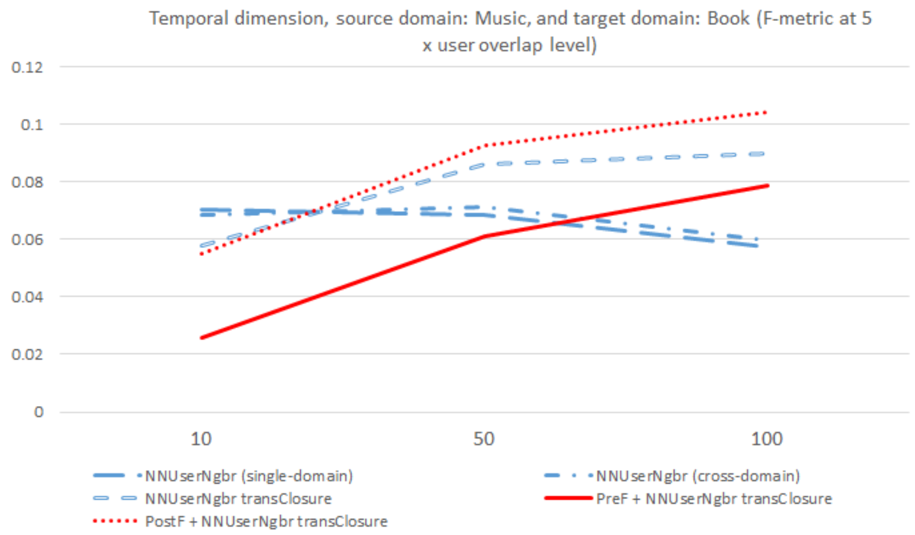


Figure 88 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the temporal dimension (target domain: book, and source: Music).

Table 34 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Location dimension (source domain: Book, and target domain: Music).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.671 ± 0.024	1.002 ± 0.048	0.505 ± 0.008	0.792 ± 0.020	0.419 ± 0.006	0.735 ± 0.012
NNUserNgbr (cross-domain)	0.587 ± 0.022	0.897 ± 0.044	0.450 ± 0.007	0.739 ± 0.018	0.369 ± 0.005	0.678 ± 0.010
NNUserNgbr-transClosure	0.120 ± 0.010	0.362 ± 0.036	0.200 ± 0.002	0.443 ± 0.006	0.199 ± 0.004	0.467 ± 0.007
PreF with NNUserNgbr-transClosure	0.122 ± 0.206	0.349 ± 0.480	**0.149 ± 0.102**	**0.406 ± 0.276**	**0.096 ± 0.050**	**0.293 ± 0.151**
PostF with NNUserNgbr-transClosure	**0.109 ± 0.016**	**0.317 ± 0.036**	**0.193 ± 0.004**	**0.428 ± 0.009**	0.193 ± 0.003	0.449 ± 0.010

Also, Table 34 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 45–79% (MAE) and 31–59% (RMSE) depending on the user overlap levels. As it can be seen from table, the PostF predictive performance was better than the *NNUserNgbr-transClosure* algorithm in all user overlap levels, with an improvement that varied in, approximately, 3–9% (MAE) and 3–12% (RMSE) depending on the user overlap level.

In addition, we can see in Table 34 that the PostF outperformed the PreF algorithm for the majority of the user overlap levels (10% and 50%) if we consider the high standard deviation of the PreF algorithm. Figure 89 illustrates the predictive performance (MAE) of the proposed algorithms over different user overlap levels.

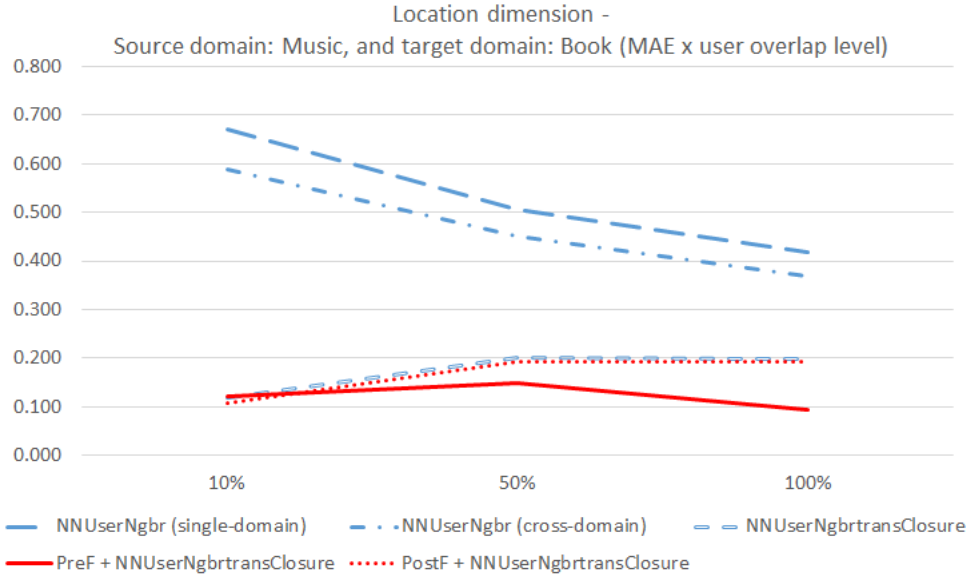


Figure 89 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the location dimension (source domain: book, and target domain: Music).

The statistical significance tests verified that, except when the user overlap level was 10%[79], the PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* baseline algorithm for all other user overlap levels[80]. The applied tests also verified that the PreF predictive error (MAE) was less than the *NNUserNgbr-transClosure* one for 100% of user overlap[81], whereas their performances were statistically similar when the user overlap level were 10%[82] and 50%[83]. Figure 90a, Figure 90b and Figure 90c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

With respect to the classification performance, Figures 91a, 91b, and 91c present the results of the F-metric at different top 'N' values (between one and twenty), respectively, in 10%, 50%, and 100% of user overlap level for the *Book* domain as target, considering the *Location* dimension.

As it can be seen, the PostF classification performance was better or similar than the *NNUserNgbr-transClosure* baseline algorithm for 100% of user overlap with any top 'N' value, whereas for 50% of user overlap it was better than the baseline algorithm only

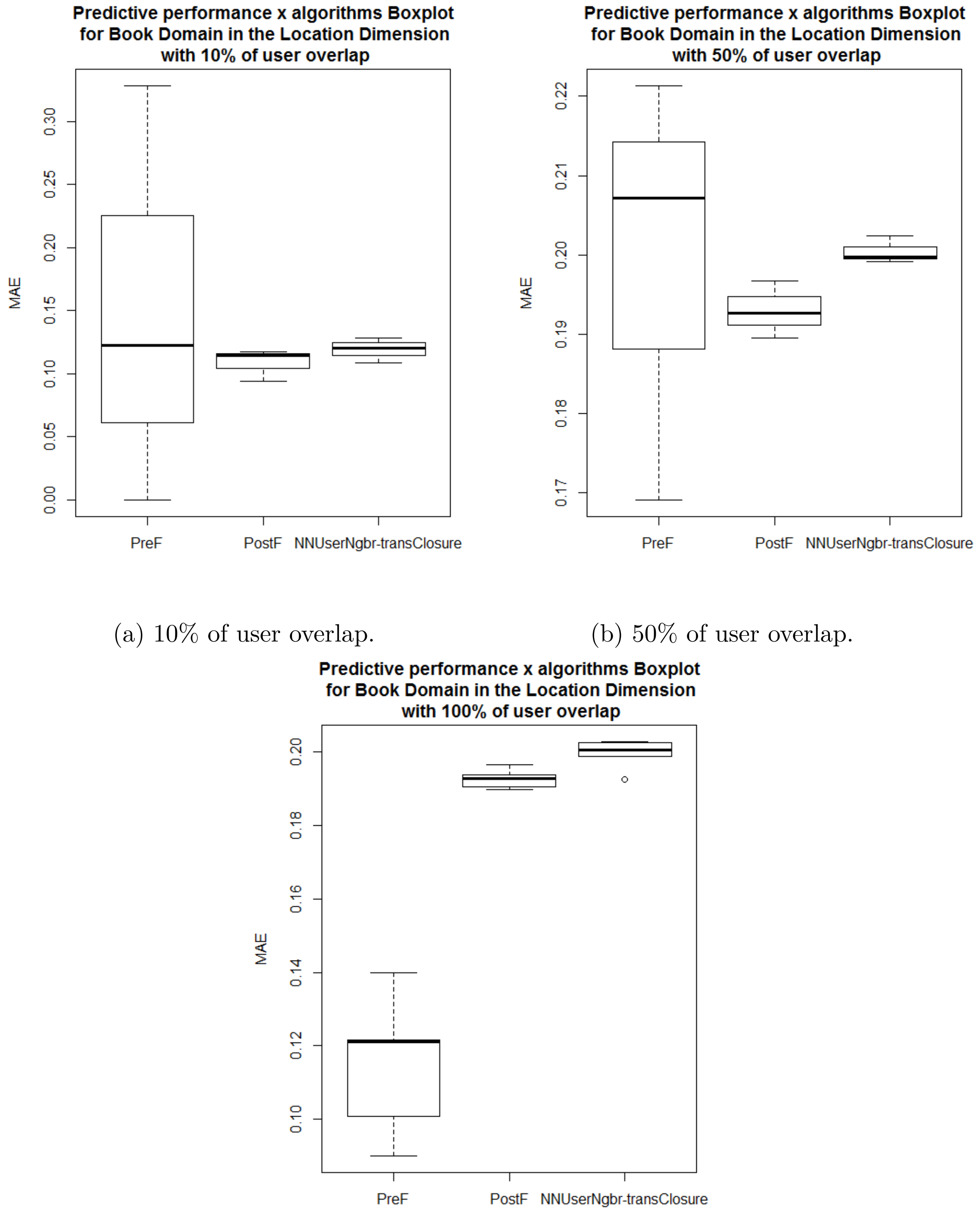
[79] W=76 and p-value=0.2

[80] For all tests, W=22 and p-value=0.02778

[81] W=97 and p-value=0.003968

[82] W=40 and p-value=0.65

[83] W=30 and p-value=0.8



(a) 10% of user overlap.

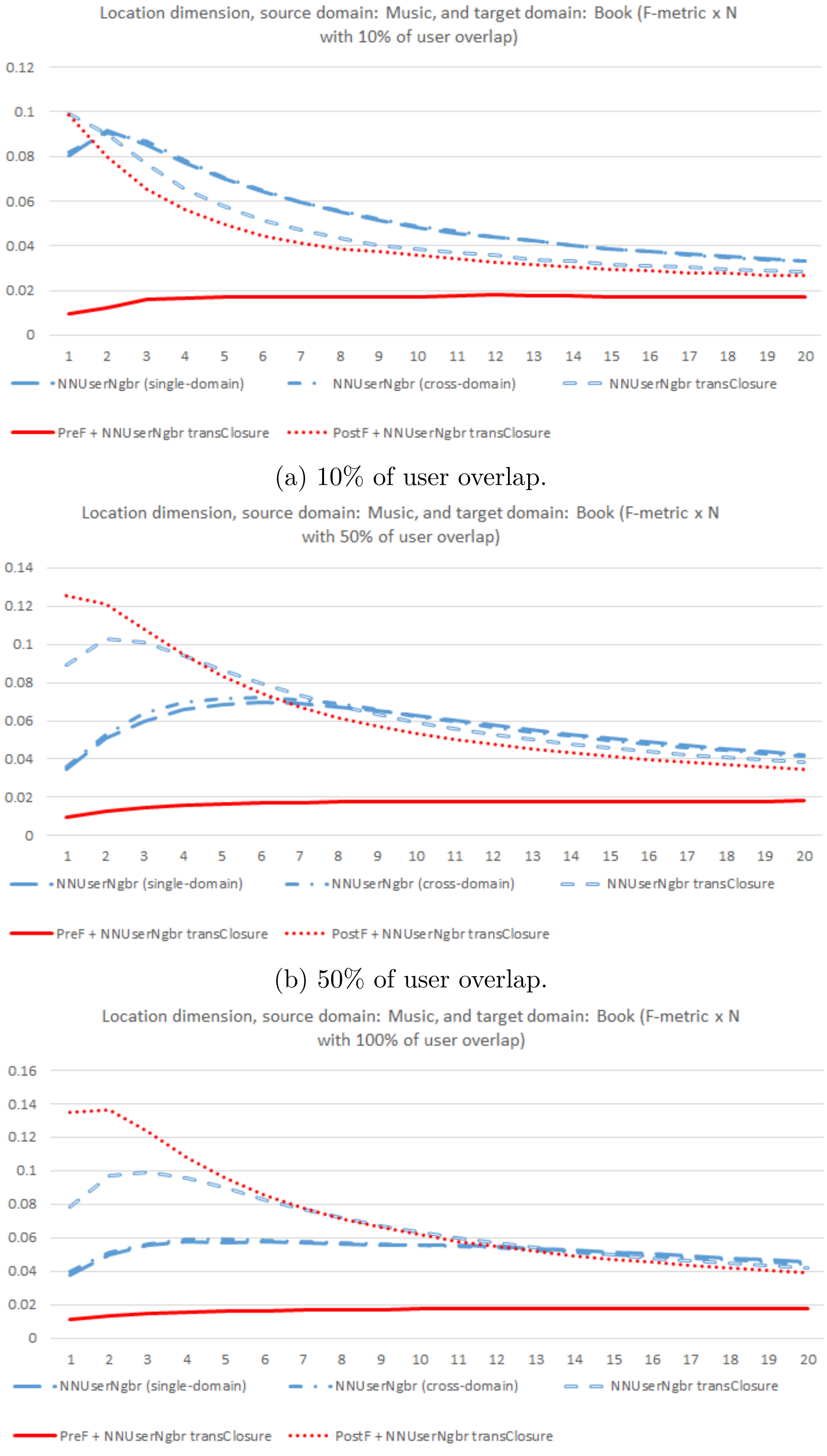
(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 90 – Overall prediction performance (MAE) boxplots for Music domain in the location dimension with different user overlap levels (source domain: book).

for low top ‘N’ values (1 to 3). For 10% of user overlap the PostF was outperformed by the baseline algorithm. On the other hand, the PreF classification performance was worse than all other algorithms in all user overlap levels and ‘N’ values.

Figure 92 shows the variation of the F-metric value in different user overlap levels



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 91 – F-metric performance x top ‘N’ items for the book domain in the location dimension with different user overlap levels (source domain: Music).

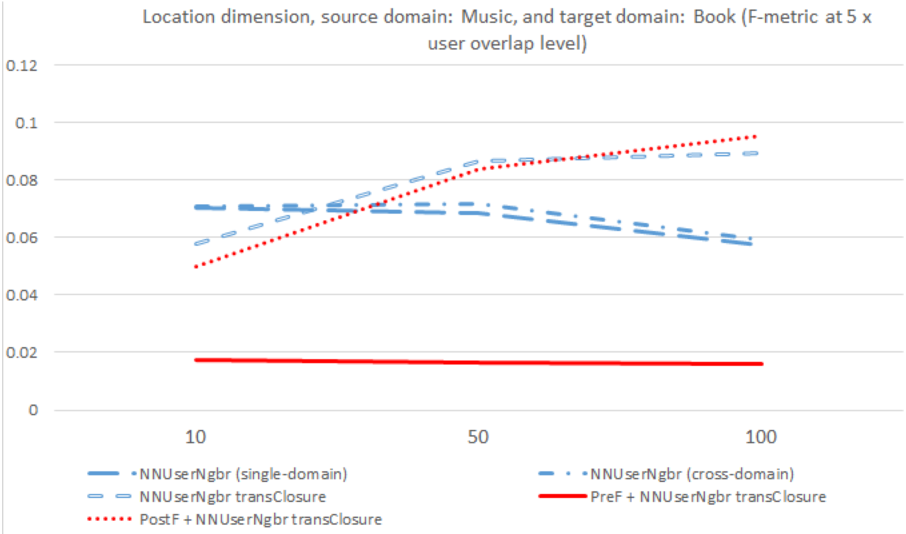


Figure 92 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the location dimension (target domain: book, and source: Music).

by fixing the top 'N' value to five. The statistical significance tests verified that the PostF F-metric value was greater than the *NNUserNgbr-transClosure* baseline algorithm for 100% of user overlap[84], whereas the opposite from this was observed for 10% and 50% of user overlap levels[85]. The applied tests also verified that the *NNUserNgbr-transClosure* F-metric values were greater than the PreF ones for all user overlap levels[86].

5.2.2.2.3 Companion Dimension

Table 35 shows the overall predictive performance of the recommender algorithms, considering all contextual values from the *Companion* dimension and different user overlap levels for the *Book* domain as target. As it can be seen from the table, the addition of user ratings from other domain (*Music*), by using the same algorithm for cross-domain recommendation (corresponding to the two first rows of the table), improved the predictive performance in, approximately, 2% (MAE) and 1% (RMSE) for 10% of user overlap.

Also, Table 35 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 33–74% (MAE) and 23–52% (RMSE) depending on the user overlap level. The PostF predictive performance was better than the *NNUserNgbr-transClosure* one for all user overlap levels with an improvement that varied in, approximately, 5–10% (MAE) and 5–11% (RMSE) depending on the user overlap level.

[84] W=97 and p-value=0.037

[85] W=97 and p-value=0.005413 for both cases

[86] W=99 and p-value=0.003914 for all tests

Table 35 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual values from the Companion dimension (source domain: Music, and target domain: Book).

Algorithm	**10% overlap**		**50% overlap**		**Full overlap**	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.671 ± 0.024	1.002 ± 0.048	0.505 ± 0.008	0.792 ± 0.020	0.419 ± 0.006	0.735 ± 0.012
NNUserNgbr (cross-domain)	0.653 ± 0.022	0.994 ± 0.044	0.513 ± 0.007	0.814 ± 0.018	0.461 ± 0.005	0.785 ± 0.010
NNUserNgbr-transClosure	0.168 ± 0.035	0.470 ± 0.085	0.290 ± 0.013	0.576 ± 0.017	0.306 ± 0.004	0.604 ± 0.136
PreF with NNUserNgbr-transClosure	0.984 ± 0.251	1.232 ± 0.154	0.693 ± 0.011	1.035 ± 0.030	0.721 ± 0.014	1.041 ± 0.016
PostF with NNUserNgbr-transClosure	**0.151 ± 0.028**	**0.417 ± 0.119**	**0.264 ± 0.017**	**0.522 ± 0.032**	**0.289 ± 0.006**	**0.573 ± 0.019**

In addition, the predictive performance of the PreF algorithm was worse than all other algorithms in the *Companion* dimension, as showed in Table 35. Figure 93 illustrates the predictive performance (MAE) of the proposed algorithms over different user overlap levels.

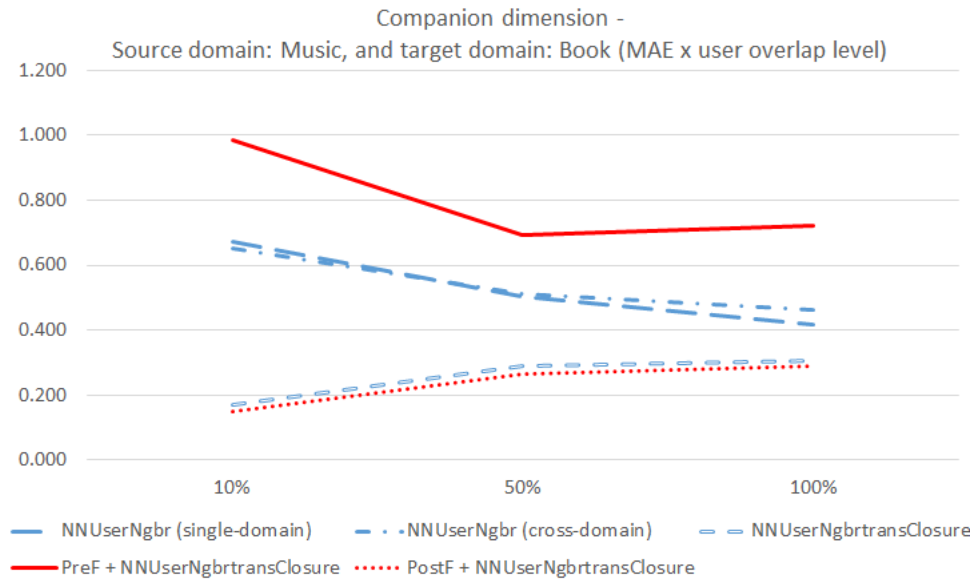


Figure 93 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the companion dimension (source domain: Music, and target domain: book).

The statistical significance tests verified that the PostF predictive error (MAE) was less than the *NNUserNgbr-transClosure* algorithm for 100% of user overlap[87]. For

[87] W=99 and p-value=0.003968

10% and 50% of user overlap, the *NNUserNgbr-transClosure* predictive errors (MAE) were statistically similar to the PostF ones[88]. On the other hand, the applied tests also verified that the *NNUserNgbr-transClosure* predictive errors (MAE) were less than the PreF ones for all user overlap levels[89]. Figure 94a, Figure 94b and Figure 94c show the boxplots with the prediction performance (MAE) of the algorithms, respectively, for the 10%, 50%, and 100% user overlap levels.

Figures 95a, 95b, and 95c present the results of the F-metric at different top 'N' values (between one and twenty), respectively, in 10%, 50%, and 100% of user overlap level for the *Book* domain as target, considering the *Companion* dimension. As it can be seen, the proposed algorithms were outperformed by the *NNUserNgbr-transClosure* baseline algorithm for all user overlap levels and any values of top 'N'. Figure 96 shows the variation of the F-metric value in different user overlap levels by fixing the top 'N' value to five.

The statistical significance tests verified that the *NNUserNgbr-transClosure* F-metric values were greater than the both proposed algorithms for all user overlap levels[90].

5.2.2.2.4 Combining Contextual Dimensions

In the previous sections, we presented the evaluation results regarding the contextual dimensions separately. In this section, we report the results for a combination of two contextual dimensions considering the same evaluation metrics and methodology described before.

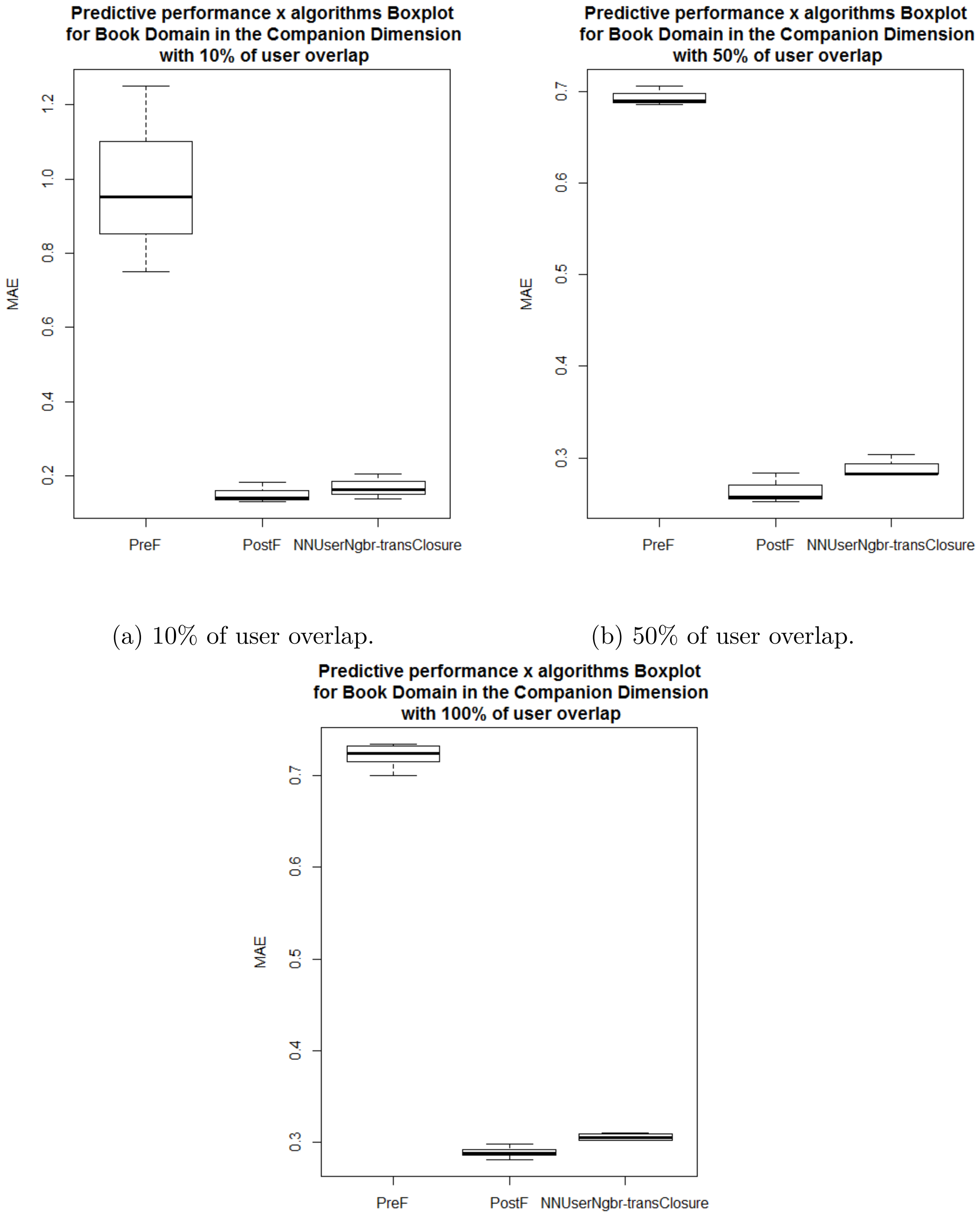
Table 36 reports the overall predictive performance of the recommender algorithms, considering *all contextual value combinations* from the *Temporal* and *Location* dimensions with different user overlap levels for the *Book* domain as target. As it was observed in the previous sections, the addition of user ratings from the *Music* domain also improved the predictive performance of the *NNUserNgbr* in, approximately, 10–12% (MAE) and 6–10% (RMSE) depending on the user overlap level (rows 1 and 2 from the table).

Also, Table 36 presents the overall performance of the *NNUserNgbr-transClosure*, PreF and PostF algorithms. The *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one (performed for cross-domain purposes) by achieving an improvement that varied in, approximately, 46–81% (MAE) and 31–59% (RMSE) depending on the user overlap levels. As it can be seen from table, the PostF predictive performance was better than the *NNUserNgbr-transClosure* algorithm in all user overlap levels, with an

[88] For 10% of user overlap, W=60 and p-value=0.35, while for 50% of user overlap, W=71 and p-value=0.2

[89] W=99 and p-value=0.003968 for all tests

[90] p-value=0.005814 and W=97 for all tests between the PostF and baseline algorithms, while p-value=0.003913 and W=100 for all tests between the PreF and baseline algorithms



(a) 10% of user overlap.

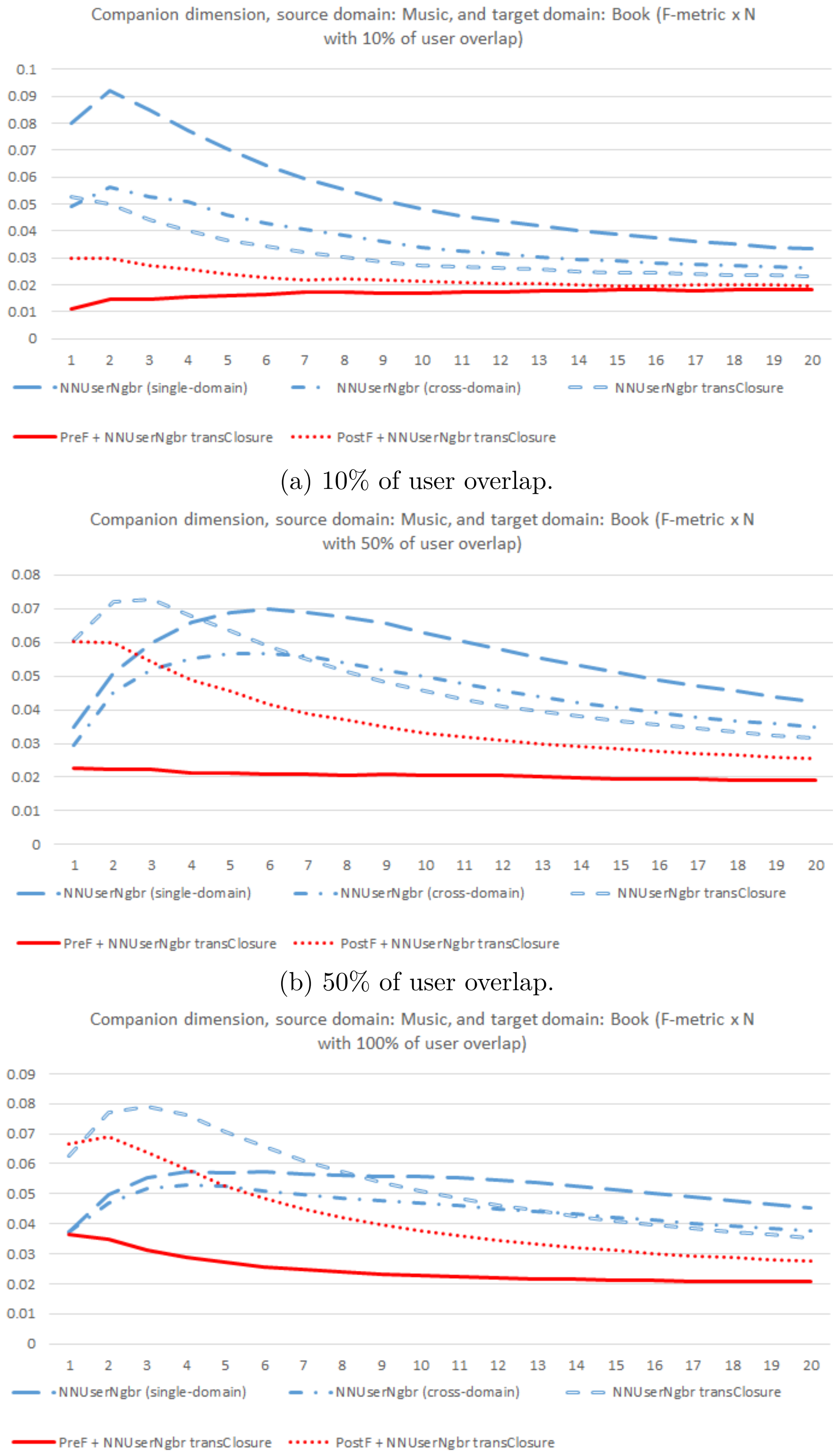
(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 94 – Overall prediction performance (MAE) boxplots for book domain in the companion dimension with different user overlap levels (source domain: Music).

improvement that varied in, approximately, 7–23% (MAE) and 8–18% (RMSE) depending on the user overlap level.

In addition, if we do not consider the standard deviation of the PreF and PostF algorithms showed in Table 32, then we can say that the PreF predictive performance



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 95 – F-metric performance x top 'N' items for the book domain in the companion dimension with different user overlap levels (source domain: Music).

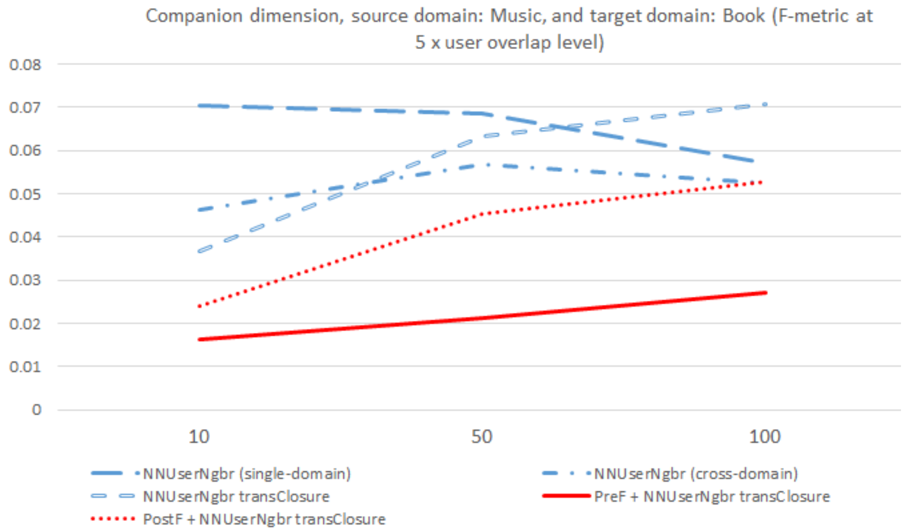


Figure 96 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the companion dimension (target domain: book, and source: Music).

Table 36 – Overall predictive performance (MAE/RMSE) with standard deviation (std) by varying the user overlap level for all contextual value combinations from the temporal and location dimensions (source domain: Music, and target domain: Book).

Algorithm	10% overlap		50% overlap		Full overlap	
	MAE±std	RMSE±std	MAE±std	RMSE±std	MAE±std	RMSE±std
NNUserNgbr (single-domain)	0.671 ± 0.024	1.002 ± 0.048	0.505 ± 0.008	0.792 ± 0.020	0.419 ± 0.006	0.735 ± 0.012
NNUserNgbr (cross-domain)	0.587 ± 0.022	0.897 ± 0.044	0.450 ± 0.007	0.739 ± 0.018	0.369 ± 0.005	0.678 ± 0.010
NNUserNgbr-transClosure	0.111 ± 0.009	0.361 ± 0.034	0.199 ± 0.002	0.434 ± 0.005	0.197 ± 0.003	0.464 ± 0.006
PreF with NNUserNgbr-transClosure	0.180 ± 0.226	0.480 ± 0.465	0.209 ± 0.122	0.510 ± 0.296	**0.140** ± **0.060**	**0.405** ± **0.161**
PostF with NNUserNgbr-transClosure	**0.103** ± **0.014**	**0.295** ± **0.030**	**0.175** ± **0.003**	**0.396** ± **0.007**	**0.150** ± **0.002**	**0.378** ± **0.006**

(measured by the MAE metric) was better than the PostF one when the user overlap level was 100%. Figure 97 (MAE) and Figure 98 (RMSE) illustrate the predictive performance of the proposed algorithms over different user overlap levels. Note that for this case we showed the figures for both predictive metrics, since we observed a difference in the PreF performance depending on the predictive metric used in the evaluation.

The statistical significance tests verified that the PostF predictive errors (MAE) were less than the *NNUserNgbr-transClosure* algorithm for the 50% and 100% user overlap

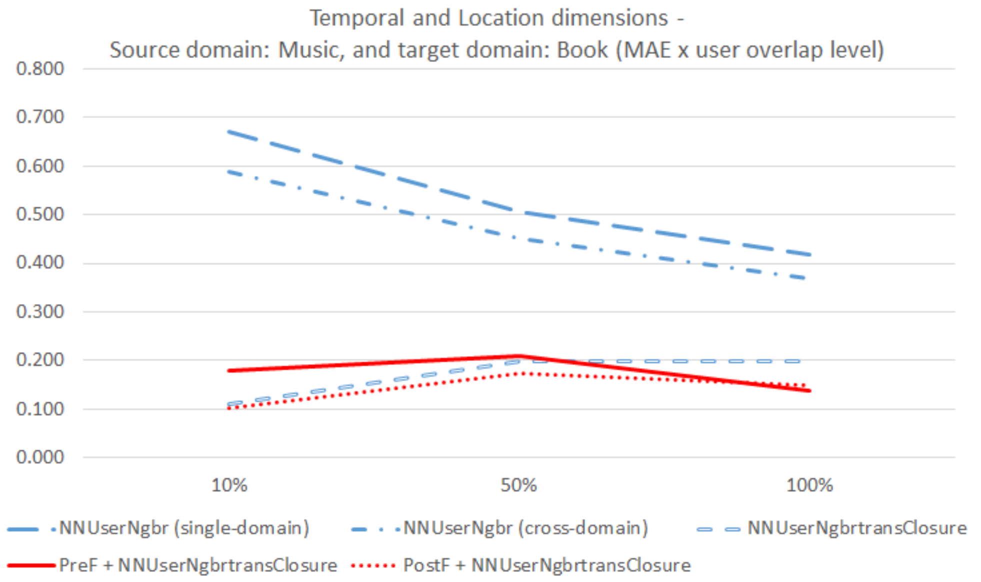


Figure 97 – Overall prediction error (MAE) for cross-domain algorithms by varying user overlap level in the temporal and location dimensions (source domain: music, and target domain: book).

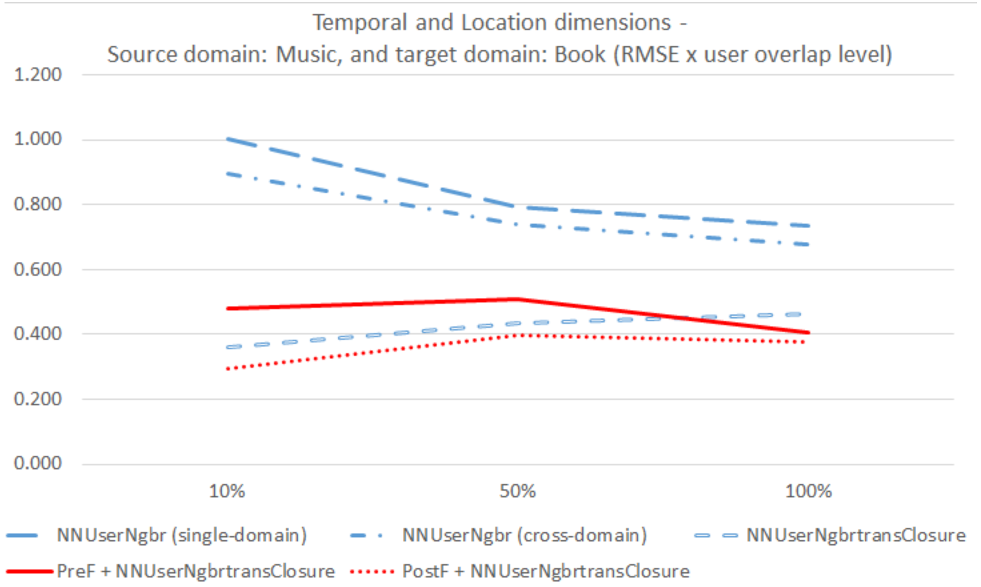


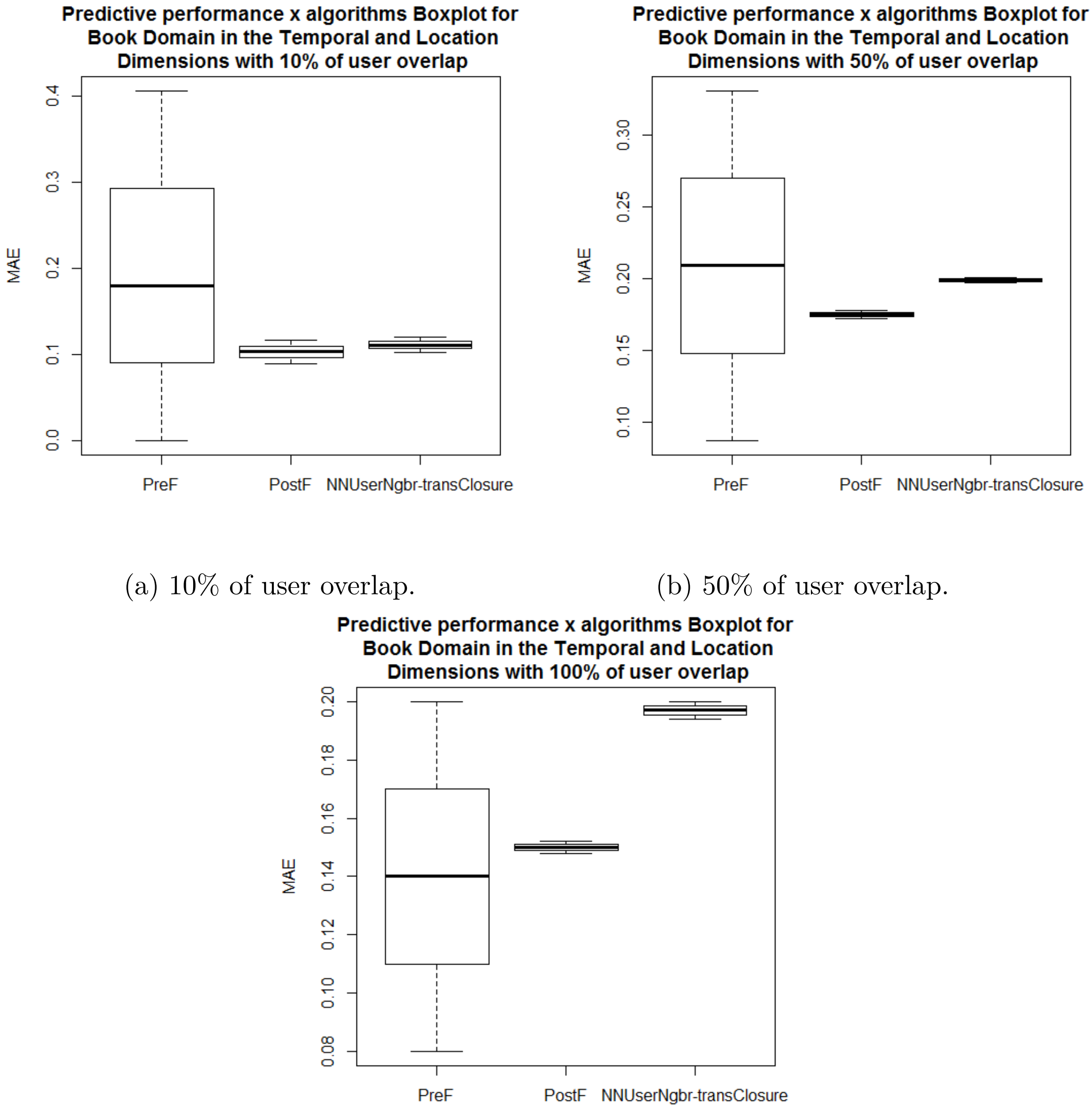
Figure 98 – Overall prediction error (RMSE) for cross-domain algorithms by varying user overlap level in the temporal and location dimensions (source domain: music, and target domain: book).

levels[91]. When the user overlap levels was 10%, the applied tests could not verify a statistical difference between the performance of the PostF and *NNUserNgbr-transClosure*[92] algorithms. On the other hand, the applied tests verify that the *NNUserNgbr-transClosure* predictive errors (MAE) were less than the PreF ones for 10% and 50% of user overlap

[91] In both cases, with W=100 and p-value=0.003968

[92] W=51 and p-value=0.35

levels[93], whereas the opposite from this was observed for 100% of user overlap[94].



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 99 – Overall prediction performance (MAE) boxplots for book domain in the temporal and location dimensions with different user overlap levels (source domain: Music).

Taking into account the classification performance, Figures 61a, 61b, and 61c present the results of the F-metric at different top ‘N’ values, respectively, in 10%, 50%, and 100% of user overlap level for the *Music* domain as target, considering the combination between the *Temporal* and *Location* dimensions. For all user overlap levels and top ‘N’ values, the PreF classification performance was worse than the all other algorithms.

[93] For 10% of user overlap, W=99 and p-value=0.003968. For 50% of user overlap, W=97 and p-value=0.005413

[94] W=99 and p-value=0.003968

On the other hand, the PostF classification performance was better or similar than the *NNUserNgbr-transClosure* baseline algorithm in the majority of the user overlap levels (50% and 100%) for any top 'N' values.

Figure 101 shows the variation of the F-metric value in different user overlap levels by fixing the top 'N' value to five. The statistical significance tests verified that the PostF F-metric values were greater than the *NNUserNgbr-transClosure* baseline algorithm for 50% and 100% of user overlap levels[95], whereas the opposite from this was observed for 10% of user overlap[96]. On the other hand, the applied tests also verified that the *NNUserNgbr-transClosure* F-metric values were greater than the PreF ones for all user overlap levels[97].

5.2.2.3 Summary

In this section, we provide a summary of the results from the evaluation of the "book-music dataset". Figure 102 shows a dispersion diagram illustrating the predictive performance (MAE) for the algorithms by varying target domain (*Book* and *Music*), contextual dimension and user overlap levels, whereas Figure 103 shows the same, but considering the RMSE metric. It is important to mention that these figures do not take into account the standard deviation and the statistical significance of the results.

Table 37 presents the predictive performance (MAE) achieved by the PreF and PostF algorithms in comparison to the best baseline algorithm (*NNUserNgbr-transClosure*), by taking into account their statistical significance[98] and different target domain, contextual dimension and user overlap levels.

Regarding the classification performance, Figure 104 presents a dispersion diagram illustrating the F-metric performance (with N=5) for the algorithms by varying target domain (*Book* and *Music*), contextual dimension and user overlap levels. Once again, it is important to mention that we are not considering the standard deviation and the statistical significance of the results in that figure.

Table 38 shows the classification performance improvement (F-metric with N=5) obtained by the PreF and PostF algorithms in comparison to the best baseline algorithm (*NNUserNgbr-transClosure*), by taking into account their statistical significance[99] and different target domain, contextual dimension and user overlap levels.

As it can be seen, at least one proposed algorithm (PreF or PostF) achieved the best predictive performance among the algorithms (or it was similar to the best one) in almost all scenarios (with distinct target domains, contextual dimensions, and user overlap

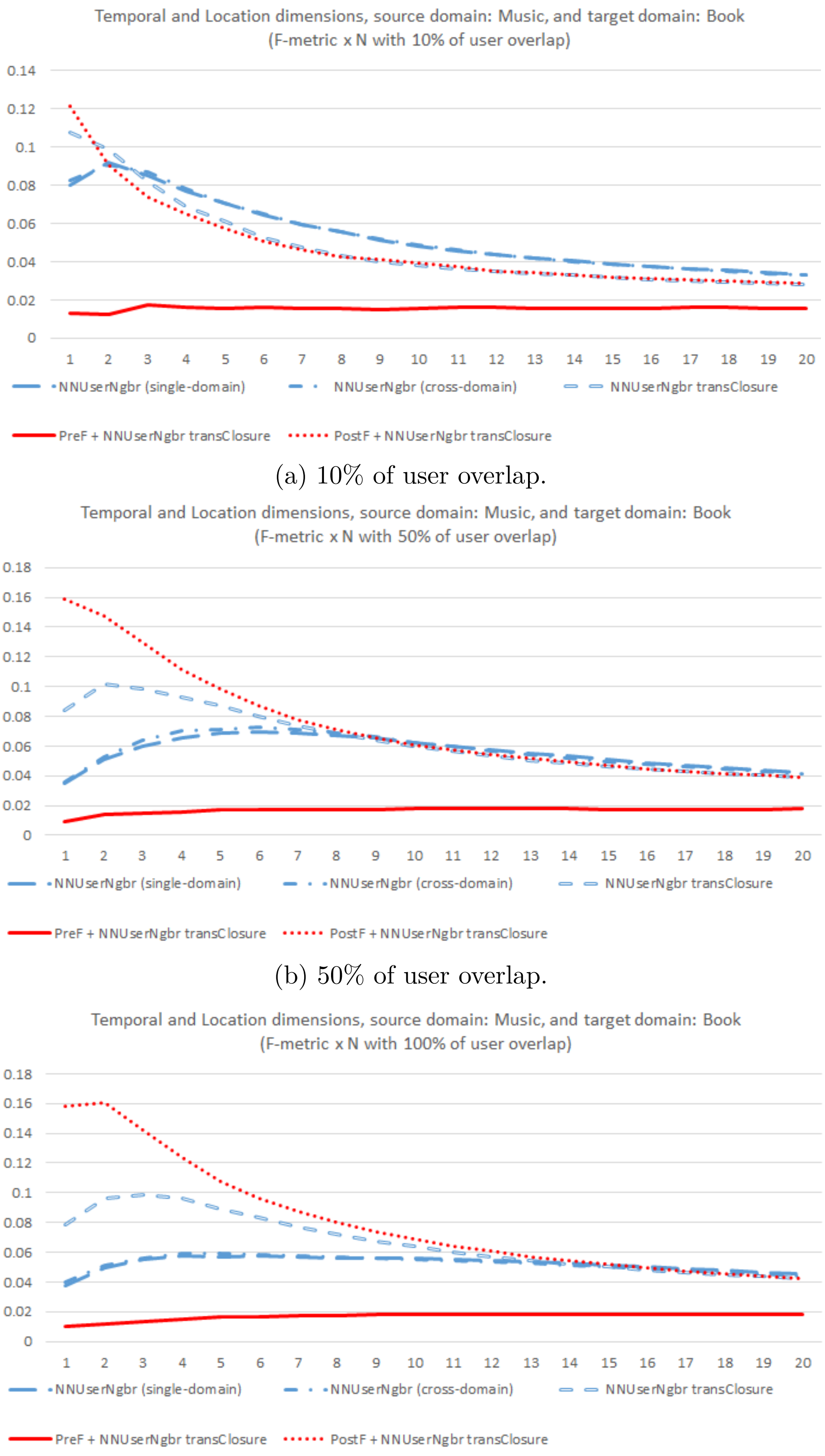
[95] p-value=0.005814 and W=97 for all tests

[96] p-value=0.035 and W=75

[97] p-value=0.003913 and W=99 for all tests

[98] In the table, "**" means that the result could not be considered statistically significant.

[99] In the table, "**" means that the result could not be considered statistically significant.



(a) 10% of user overlap.

(b) 50% of user overlap.

(c) 100% of user overlap.

Figure 100 – F-metric performance x top ‘N’ items for the book domain in the temporal and location dimensions with different user overlap levels (source domain: Music).

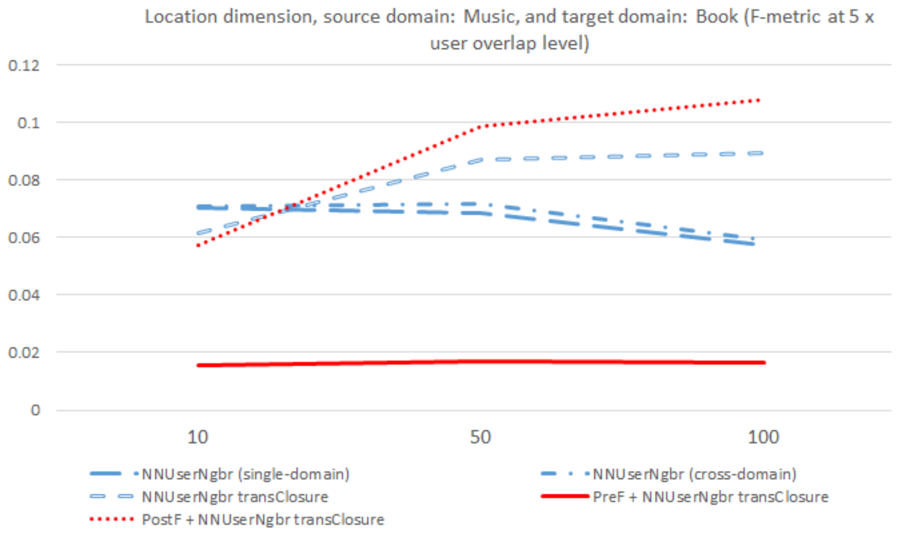


Figure 101 – Overall classification performance (F-metric at 5) for the algorithms by varying user overlap level in the temporal and location dimensions (target domain: book, and source: Music).

levels). By considering the classification metric, the PostF algorithm achieved the best performance among the algorithms (or it was similar to the best one) in the majority of the scenarios.

Most of the findings mentioned in the summary of the evaluation results for the “book-television dataset” (see Section 5.2.1.3) can also be mentioned in this summary (“book-music dataset”). In this way, we only highlight the main differences found in this summary in comparison to those findings:

- The addition of user ratings from an auxiliary (source) domain also improved the predictive performance of the *NNUserNgbr* algorithm, but in this dataset this fact has occurred in less scenarios than in the “book-television dataset”. This can also be observed for the classification performance of that algorithm. Likewise the results in the “book-television dataset”, that improvement occurred even when a source domain had less ratings than the target domain.

- Likewise the results in the “book-television dataset”, the proposed algorithms (PreF and PostF) had better predictive and classification performances in the *Temporal* dimension than others dimensions. The same findings mentioned for the predictive and classification performances of the PostF algorithm can be observed in this summary, as well as the PreF predictive performance. On the other hand, the PreF algorithm outperformed the *NNUserNgbr-transClosure* one in less scenarios (user overlap levels and target domains) than in that dataset by considering the classification performance. The PreF algorithm outperformed the *NNUserNgbr-transClosure* one only when the *Music* was the target domain with 100% of user overlap.

- Likewise the results in the "book-television dataset" regarding the combination of contextual dimensions (*Temporal* and *Location*), the PostF predictive and classification performances in that combination were close to their own performances using only the *Temporal* dimension as single source of contextual information, whereas the PreF predictive and classification performances were similar to their own performances using only the *Location* dimension. In addition, the predictive and classification performances of the PreF algorithm was also decreased with the addition of contextual information from other contextual dimension. Besides, the PostF predictive performance was also increased, however, its classification performance was increased with the addition of contextual information from other contextual dimension in opposite to the results from that dataset.

Table 37 – Overall predictive performance (MAE) of the proposed algorithms in comparison to the best baseline one by varying target domain (book and music), contextual dimension and user overlap levels.

Contextual dimension	**Target Domain**	**User Overlap Level**	**PreF Improvement**	**PostF Improvement**
Temporal	Music	10%	39.7%	16.2%
	Book	10%	16.1%**	12.4%
	Music	50%	62.6%	12.8%
	Book	50%	56%	10.8%
	Music	100%	55.8%	11.9%
	Book	100%	55.4%	13.3%
Location	Music	10%	24.5%**	16.1%**
	Book	10%	-2.9%**	8.9%**
	Music	50%	39.2%	6.7%
	Book	50%	25.5%**	3.7%
	Music	100%	57.4%	2.9%
	Book	100%	52%	3.8%
Companion	Music	10%	-113.5%	-85.7%**
	Book	10%	-484%	10.3%**
	Music	50%	-39.1%	2.9%**
	Book	50%	-139.3%	8.7% **
	Music	100%	-46.5%	11.2%
	Book	100%	-136%	5.4%
Temporal and Location	Music	10%	-0.2%**	-0.9%**
	Book	10%	-62.1%	7.3%**
	Music	50%	-2.3%	28.6%
	Book	50%	-4.9%	12.1%
	Music	100%	45.9%	23.1%
	Book	100%	29.3%	24%

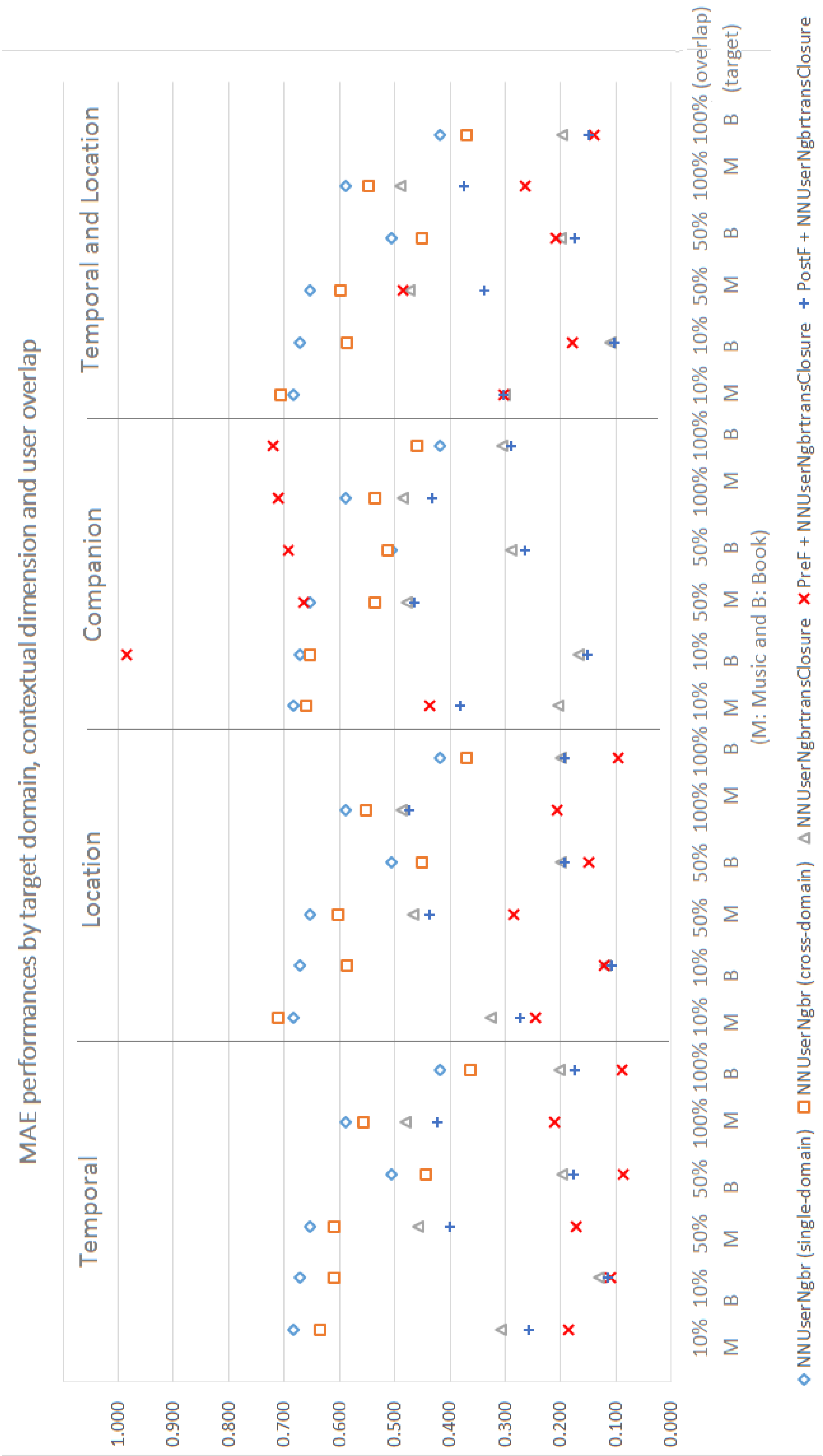


Figure 102 – Predictive performance (MAE) for the algorithms by varying target domain (book and music), contextual dimension and user overlap levels (dispersion diagram).

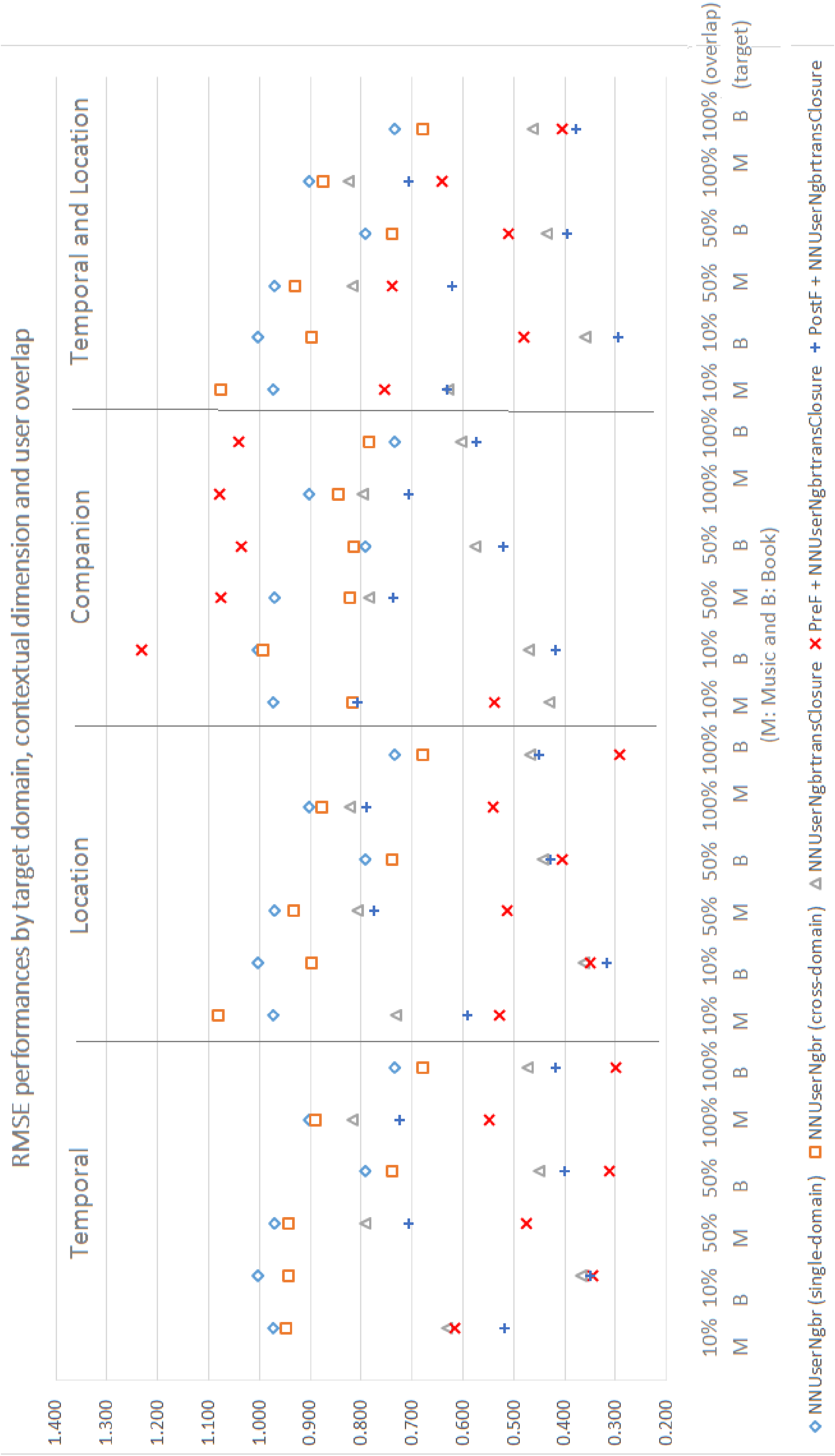


Figure 103 – Predictive performance (RMSE) for the algorithms by varying target domain (book and music), contextual dimension and user overlap levels (dispersion diagram).

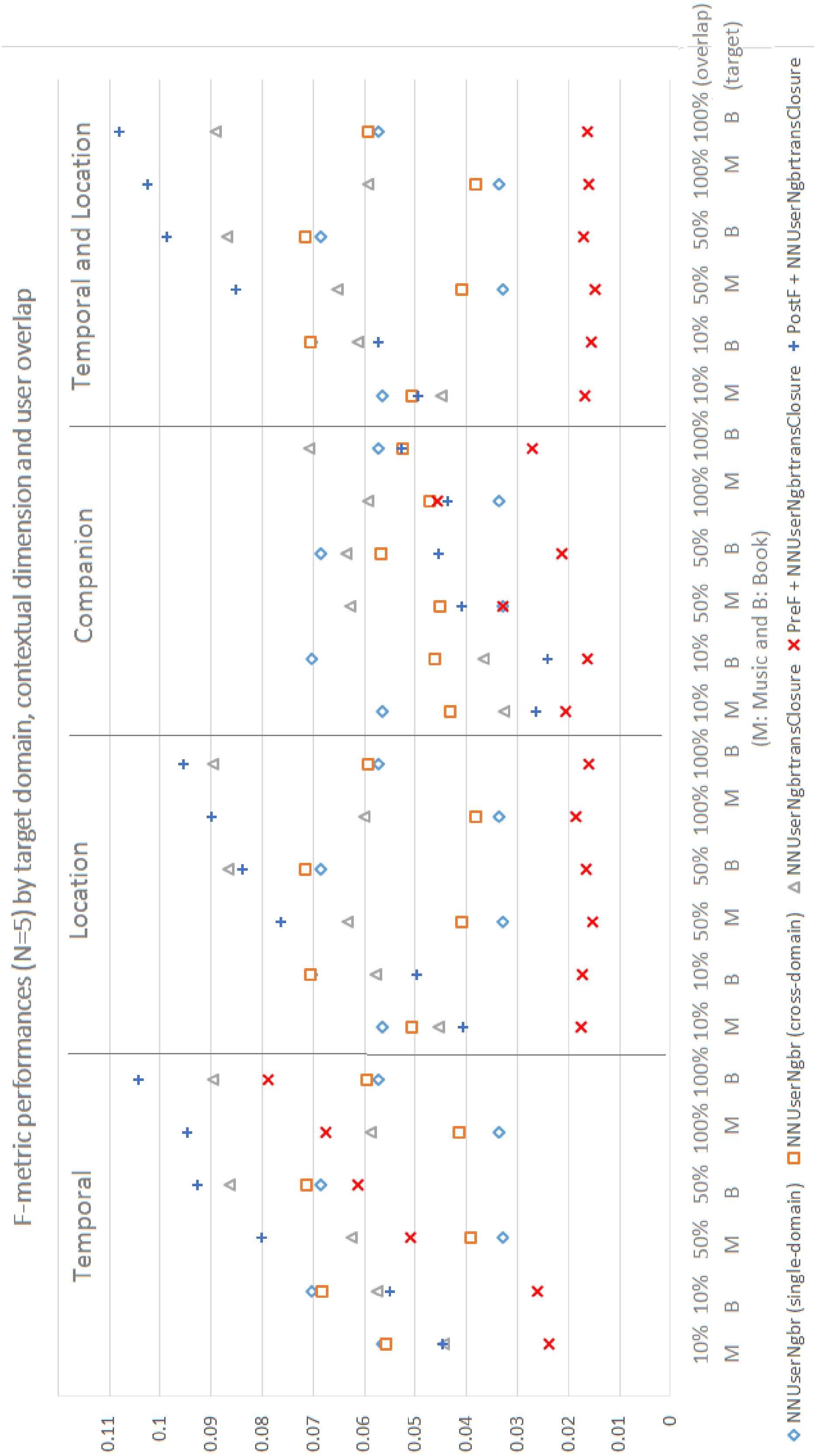


Figure 104 – Classification performance (F-metric with N=5) for the algorithms by varying target domain (book and music), contextual dimension and user overlap levels (dispersion diagram).

Table 38 – Overall classification performance (F-metric with N=5) of the proposed algorithms in comparison to the best baseline one by varying target domain (book and music), contextual dimension and user overlap levels.

Contextual dimension	**Target Domain**	**User Overlap Level**	**PreF Improvement**	**PostF Improvement**
Temporal	Music	10%	-86.6%	0.5%**
	Book	10%	-121.6%	-4.6%**
	Music	50%	-22.4%	22.1%
	Book	50%	-41.1%	6.9%
	Music	100%	13%	37.9%
	Book	100%	-13.8%	13.9%
Location	Music	10%	-157.4%	-11.3%
	Book	10%	-234%	-15.9%
	Music	50%	-311%	17.1%
	Book	50%	-420.5%	-3.4%
	Music	100%	-223.9%	33.3%
	Book	100%	-455%	6%
Companion	Music	10%	-58.7%	-23.3%
	Book	10%	-126.4%	-51.9%
	Music	50%	-92%	-54.1%
	Book	50%	-197%	-39.6%
	Music	100%	-29.7%	-35.2%
	Book	100%	-162.2%	-34.3%
Temporal and Location	Music	10%	-168%	9.1%
	Book	10%	-292.3%	-7.1%
	Music	50%	-339.3%	23.3%
	Book	50%	-408%	12%
	Music	100%	-270.7%	42.1%
	Book	100%	-444%	17.2%

5.2.3 Discussion

Given the evaluation results presented in the previous sections, we can say that the use of context-aware techniques has proven to be a good approach in order to improve the cross-domain recommendation quality in comparison to traditional cross-domain recommender systems based on collaborative filtering techniques, which do not take contextual information into account. This finding was observed in the presented experiments, in which we evaluated two CD-CARS algorithms performed in two different datasets ("Book-television" and "Book-music") by varying their target domains (*Television*, *Music* and *Book*), contextual dimensions (*Temporal*, *Location* and *Companion*), and user overlap levels (10%, 50%, and 100%).

As we could see in the experiments, *Temporal* was the contextual dimension in which the proposed algorithms had a better performance for all datasets, target domains and user overlap levels. This may have happened due to the great amount of contextual information obtained in that contextual dimension (100% of the ratings had temporal information) in comparison to other ones (*Location* with, approximately, a half of ratings, and *Companion* with, approximately, 20% from the ratings, as described in Section 4.1.3). This fact contrasts to the information gain verified in Section 4.1.2, where the *Location* dimension with the *City* attribute had the greater value for all target domains. In this way, more studies and experiments may be made in the future in order to determine the best contextual dimensions, attributes and values before evaluating the proposed algorithms, especially in the combination of contextual dimensions. In addition, in these studies we could verify the quality of recommendation of the proposed algorithms by reducing the number of temporal information present in the user ratings ("contextual sensitivity").

In addition, the quality of the contextual information may also have influenced on the recommendation quality for the proposed algorithms. As we have seen in Section 4.1.1.3, the *Companion* dimension has a poor quality of contextual information. Especially for the PreF algorithm, which filters ratings out from the target domain that are not from the recommendation context, the recommendation quality was more impaired than the PostF one, which uses the same set of ratings of the baseline algorithm for prediction calculations, and only in the end of the recommendation process ignores predictions (instead of initial ratings).

The combination of two contextual dimensions (*Temporal* and *Location*) in the recommendation process generated controversial results for the proposed algorithms. Independently of the dataset used, while the PreF had worse results in that combination than using only one contextual dimension (*Temporal* or *Location*), the PostF recommendation quality was improved for some situations, as it could be seen in the result summaries (Section 5.2.1.3 and Section 5.2.2.3). Again, this may have happened given the PreF feature, in which might be more susceptible to problems in a situation with just a few number of ratings, generated by the contextual specialization from the combination of two contextual dimensions.

It is important to remember that all contextual information used in the CD-CARS was obtained implicitly or by inference (see Section 4.1.1). In this way, there is no assurance that the contextual information acquired reflects the actual contextual information of the ratings. For instance, a user could watch a movie on Saturday and rate it only on Sunday, when the rating timestamp was observed. Thus, the actual temporal information of that rating might have been compromised. However, even considering this issue, it was possible to verify that the proposed algorithms had a good performance in the *Temporal* dimension.

Besides, the *Location* context is extracted from the users' accounts through their original IDs (i.e. static and single location of the users are obtained from their website accounts). In this way, the contextual information is little exploited when a user receives the recommendation of items for a location different from his/her location (e.g. a user that has all ratings in the United States and receives recommendations in Brazil), especially for the PreF algorithm. For it, the recommendation would be fully based on the user similarities from the source domain, since the user would not have any rating in the target domain (pre-filtered in a location that the user does not have any information). So, it would not be possible to calculate the similarity between the user and other users with ratings in the target domain. On the other hand, for the PostF algorithm, no recommendation would be possible without the association rules once that the user would not have any contextual preferences in that location.

Taking into account the PostF and PreF recommendation performances, we could observe that they had distinct results depending on the evaluation metric adopted. For example, considering the *Temporal* dimension, the PostF classification performance (F-metric) was better than the PreF one, whereas the opposite was observed when they were evaluated by means of predictive error metrics (MAE and RMSE). In other contextual dimensions, independently of the dataset, we have seen that the PostF had a better recommendation quality than the PreF algorithm, which had worse results than the baseline algorithms. In fact, besides the PreF's feature of filtering ratings in a preliminary way before its model training, it also differs from the PostF algorithm in relation to the use of information about item categories, once that the PostF uses a category preference tensor in its recommendation process, whereas the PreF does not. For alleviating this disparity, we could combine both algorithms in order to try having the best of their features in a single hybrid algorithm, for example (as described in Section 3.3.1.4).

By varying the target domain in the two datasets used in the experiments, we studied the impact of the density of the target domain data in comparison to the density of the source domain data. In both cases, we have seen that the addition of ratings from a source domain improved the recommendation quality of the cross-domain based algorithms, independently of its amount of ratings in relation to the target domain. In addition, even for domains less *related* among themselves (*Book* and *Music*), we could see a improvement on the recommendation quality of the cross-domain based algorithms.

Considering distinct user overlap levels (10%, 50% and 100%), we could see in the experiments that the proposed algorithms had a better recommendation quality as the user overlap level was higher. For the PreF algorithm, more user overlap may significate more ratings in filtered contexts, expanding the similarities among users in these contexts, whereas for the PostF, more user overlap level may expand the category preference tensor, with more contextual information about item category preferences of users. On the other hand, the baseline algorithms, especially the *NNUserNgbr-transClosure*, had a similar performance independently of the user overlap level.

As mentioned in Section 5.1.1, the proposed algorithms used the baseline ones with the same recommendation settings (e.g. n=475 for the CF-based algorithm as base). However, note that more tests can be done in order to achieve a "optimal" setting for each algorithm, especially in the PreF algorithm, which uses a contextual sub-dataset with smaller amount of data in relation to other algorithms. Besides, the PostF's threshold can be adjusted to an "optimal" value for other datasets and its minimal rating value for an item to be considered "good" can also be changed depending on the datasets, as described in Section 5.1.1. For the two datasets used in the experiments, we considered this value as "four" in a five-star scale.

However, it may be possible that for other datasets the ratings from the contextual user-rating tensors can have different scales or forms in distinct domains. For example, ratings of music could be represented as a binary form such as "Like" and "Dislike" while the ratings of movie and books could be represented, respectively, by five-star and ten-star scales. As mentioned before, the base recommendation algorithms have to deal with this issue. For instance, an algorithm could normalize the different scales from ratings among distinct domains (SANTOS et al., 2012).

Due to the lack of publicly real datasets available with cross-domain and contextual information, a feasible alternative to our experiments without having to produce contextual synthetic data was extract contextual information implicitly or by inference for the three contextual dimensions used in the experiments. However, other contextual dimensions still can be extracted from the user reviews, as for example, the *Task* dimension. In addition, other contextual attributes of the same contextual dimensions can be used in order to verify the recommendation quality of the proposed algorithms (e.g. by using "countries" instead of "cities" in the *Location* dimension).

Finally, other important aspect of recommender systems is the execution performance. Although we did not evaluate the proposed CD-CARS by considering this aspect, intuitively, we can say that the PreF algorithm may be capable of recommending items consuming less resources (e.g. time and memory consumption) than the baseline algorithms, once that the PreF uses these algorithms as base for a small set of ratings in comparison to them. On the other hand, the PostF algorithm may demand more resources, since it initially uses the baseline algorithms as base, and then, applies an addition step to them by using an additional data structure (category preference tensor).

5.3 Final Remarks

In this chapter, we presented and discussed experimental evaluations of two proposed CD-CARS algorithms in comparison to cross-domain CF-based ones. The experimental evaluation was made by considering two distinct datasets (described in Section 4.1.3) with three different contextual dimensions (*Temporal*, *Location* and *Companion*), target domains (*Television*, *Music* and *Book*) and user overlap levels (10%, 50%, and 100%).

The algorithms were evaluated regarding their predictive and classification performances, which were analyzed by means of statistical significance tests.

Finally, the conclusions and future works of this thesis are described in the next chapter.

6 Conclusion

In this thesis, we have found that context-aware techniques can be used in order to improve the *accuracy* of cross-domain recommendations. A traditional cross-domain CF-based algorithm provided better recommendations when used in combination with the implemented CD-CARS algorithms (*Pre-Filtering* and *Post-Filtering*).

By considering contextual information from three dimensions (*Temporal*, *Location* and *Companion*), experimental evaluations conducted in two real datasets, one with two more related domains (*Book* and *Television*) and another with two less related domains (*Book* and *Music*), showed that generating predictions exploiting knowledge from a source domain improved predictive and classification performances in the target domain. For both datasets, we made experiments by swapping source and target domains and, regardless of these domains evaluated as source or target, the proposed algorithms achieved better results in comparison to the baseline ones, especially by using contextual information from the *Temporal* and *Location* dimensions.

For the *Companion* contextual dimension, only one of the implemented algorithms had a good predictive performance, whereas its classification performance was not as good. As discussed in Chapter 5, the low quality and quantity of contextual information from that contextual dimension may have influenced on the negative classification performance, mainly for the *Pre-Filtering* algorithm, which also had a bad predictive performance by taking into account the *Companion* dimension.

With respect to the user overlap level variation (10%, 50% and 100%), we conclude that it influenced the proposed algorithms that, in general, are more accurate as higher is the user overlap level.

Finally, through a novel approach, we expect that the findings from this study contribute to the cross-domain RS area towards future research in cross-domain context-aware recommendations. In the following sections, we describe the contributions, limitations and future works of this thesis.

6.1 Contributions

One of the contributions of this thesis is the novel study about the successful integration of two emergent and relevant approaches of the recommender system (RS) area: *cross-domain* and *context-awareness*. This integration can lead to further research in the area in order to improve the quality of recommender systems. While the proposed CD-CARS algorithms address, mainly, the *accuracy* aspect of RSs quality, the cross-domain collaborative filtering algorithms, which are adopted in combination with the proposed ones, may address other aspects such as *cold-start* and *sparsity*. Therefore, the proposed CD-CARS takes the best aspects of those RS approaches into account.

Other contributions of this thesis are:

- The formalization of the cross-domain context-aware recommendation problem from the survey of two relevant research fields: cross-domain and context-aware RS. For that, we considered user ratings as a function of three dimensions (ADOMAVICIUS; TUZHILIN, 2015): *User*, *Item* and *Context*. Thus, the user ratings can be stored in multidimensional user-rating-context tensors for each item domain (e.g. books, movies, music, among others). In addition, it is necessary that there are user and contextual overlap among distinct domains. At least, the proposed contextual feature modelling is based on the "Key-Value" model, since it is simple and relatively easy to implement and use (VIEIRA; TEDESCO; SALGADO, 2009)(BETTINI et al., 2010);

- Proposal of novel CD-CARS algorithms based on three distinct and systematic paradigms of context-aware recommendation (*Pre-Filtering*, *Post-Filtering* and *Modelling*), which were chosen rather than ad-hoc context-aware approaches. One of the advantages of the proposed CD-CARS algorithms is the possibility of using traditional single-domain and cross-domain CF-based algorithms as a base algorithm, which is used in combination with the proposed ones. In addition, the proposed algorithms can be directly combined such as *Pre-Filtering* and *Post-Filtering*, or *Modelling* and *Post-Filtering*, generating hybrid versions of the proposed CD-CARS algorithms;

- Providing systematic CD-CARS algorithms that can be useful to recommend items for several domains (e.g. books, music, movies, etc.) in a simple way, since little information about users and items is required. It allows generating cross selling or bundle recommendations for items from multiple domains (e.g. the recommendation of a music accompanied of a movie to watch or a book to read);

- Provision of two real datasets for evaluating CD-CARS[1], taking into account different domains and contextual information. One of them for evaluating CD-CARS algorithms in two more related domains (*Book* and *Television*) and another considering two less related domains (*Book* and *Music*). These datasets were adapted and extracted from (LESKOVEC; ADAMIC; HUBERMAN, 2007), which contains ratings (five-star scale), product metadata and review information about different Amazon products[2]. In addition to these data, we included contextual information

[1] https://github.com/douglasveras/cd-cars-datasets

[2] https://snap.stanford.edu/data/amazon-meta.html

regarding three contextual dimensions: *Temporal*, *Location* and *Companion*, respectively, inferred from the ratings' *dates*, users' static addresses (obtained from their account on Amazon), and users' rating reviews.

6.2 Limitations

The main limitations of this thesis are:

- Absence of a mechanism in the proposed CD-CARS to handle ratings from the contextual user-rating tensors that have different scales or forms in distinct domains. For example, ratings of music could be represented as a binary form such as "Like" or "Dislike" while the ratings of movies and books may be represented, respectively, by five-star or ten-star scales. As mentioned in the CD-CARS proposal, the proposed CD-CARS algorithms let the responsibility of dealing with this issue with the cross-domain algorithms used as base. For instance, the base algorithm could normalize the different scales from ratings among distinct domains (SANTOS et al., 2012). Another solution is to normalize the ratings in these domains before the recommendation process.
- A deeper experimentation in the combination of different contextual dimensions (*Temporal*, *Location* and *Companion*) could be made. As previously mentioned, we just performed experiments in the combination of *Temporal* and *Location* dimensions, since the quality of the *Companion* dimension is low.
- Lack of realization of a concrete CD-CARS for evaluating the satisfaction of real users, and its capabilities in terms of execution time and memory required.

6.3 Lines for Further Work

The CD-CARS proposed in this thesis allows further investigation in multiple research lines such as:

1. Improvement of the implemented algorithms and implementation of other proposed CD-CARS algorithms such as *Modelling* or the combination between *Pre-Filtering* and *Post-Filtering*, for example, as well as other state-of-the-art CD-CFRS algorithms, which could be based on different cross-domain approaches in order to expand the findings of this thesis (e.g. *Linking and transferring knowledge* instead of the *Aggregating knowledge*, adopted in this thesis). For instance, we could compare the proposed CD-CARS with algorithms from the *Linking and transferring knowledge* approach, such as: *CodeBook Transfer* (CBT)(LI; YANG; XUE, 2009a), (GAO et al., 2013), among others;

2. Exploration of further contextual information in different domains:

    a) Other algorithms could be used, or proposed, to infer more precise contextual information from user reviews (e.g. use of supervised text mining techniques for a better inference quality of the *Companion* contextual dimension) (CHEN; CHEN, 2015)(DOMINGUES et al., 2014)(LAHLOU et al., 2013), which may lead to a better recommendation quality and information gain calculated;

    b) Creation of techniques for inference of other contextual dimensions (e.g. *Task*, *Mood*, etc.) from user reviews, independently on the item domain (e.g. music, books, games, etc.);

    c) Building mechanisms to explicitly collect contextual information from user ratings over multiple domains;

    d) Making the contextual modelling (adopted in the CD-CARS) more representative in order to describe semantic relations between contexts and domains, for example.

3. CD-CARS evaluation:

    a) Building a concrete CD-CARS for evaluating the satisfaction of real users taking into account the use of real resources (e.g. time, memory, etc.);

    b) Development of a benchmark for a rapid and efficient evaluation of the CD-CARS;

    c) Improving the current evaluation methodology used in this thesis taking into account different evaluation metrics (e.g. Breese score, Normalized Discounted Cumulative Gain, etc.) and partitioning of training and test sets, for example;

    d) Investigating and providing data mining techniques in order to select the most relevant contextual dimensions, attributes and values (or their combination) before performing recommendation or evaluation, once that the verification of all possible situations is costly;

    e) Combining other domains (e.g. *Music* and *Television*) or contextual dimensions (e.g. *Location* and *Companion*), as well as evaluating the algorithm performances by considering different user overlap levels (e.g. 0%, 25% and 75%);

    f) To verify the impact of the amount of ratings with contextual information in distinct contextual dimensions ("contextual sensitivity"), by reducing the ratings from the *Temporal* dimension for that it have a similar number of ratings in comparison to the *Location* dimension, for example;

    g) In order to verify the use of association rules by the PostF algorithm, we can made an evaluation for specific situations where the use of these rules is required by it. For instance, when a user receives a recommendation in the target domain and the category preferences tensor (used by the PostF) does not have information about his/her rated item categories in that domain yet. In this case, the association rules are used for enhancing the category preferences tensor, as described in Section 3.3.1.2.

4. Development of CD-CARS applications. Since the proposed algorithms are not domain-specific, a myriad of cross-domain (or cross-selling) applications can be developed for multiple domains (e.g. a recommender system on the TV that, beyond TV shows, also could recommend books, sites, or other relevant information to the TV programs) (FERRAZ; SILVA; SILVA, 2015). Besides, these applications could be developed to be accessed in a ubiquitous way, depending on the users' contexts and their domains of interest (e.g. when a user watches TV in his/her bedroom, then the CD-CARS application could recommend movies, whereas when the user is in the living room with his/her friends, then the same application could recommend music for them).

References

ABBAR, S.; BOUZEGHOUB, M.; LOPEZ, S. Context-aware recommender systems: A service-oriented approach. In: *VLDB PersDB workshop*. [S.l.: s.n.], 2009. p. 1–6. Cited on page 36.

ABEL, F. et al. Analyzing cross-system user modeling on the social web. In: *Web Engineering*. [S.l.]: Springer, 2011. p. 28–43. Cited on page 42.

ABEL, F. et al. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, Springer, v. 23, n. 2-3, p. 169–209, 2013. Cited 2 many times on page 46 and 47.

ABOWD, G. D. et al. Towards a better understanding of context and context-awareness. In: SPRINGER. *Handheld and ubiquitous computing*. [S.l.], 1999. p. 304–307. Cited on page 34.

ADOMAVICIUS, G. et al. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, ACM, v. 23, n. 1, p. 103–145, 2005. Cited 7 many times on page 27, 47, 51, 54, 70, 80, and 81.

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, IEEE, v. 17, n. 6, p. 734–749, 2005. Cited 4 many times on page 23, 25, 32, and 33.

ADOMAVICIUS, G.; TUZHILIN, A. Context-aware recommender systems. In: *Recommender systems handbook (Second Edition)*. [S.l.]: Springer, 2015. p. 191–226. Cited 20 many times on page 9, 26, 27, 29, 30, 47, 48, 49, 51, 52, 53, 54, 55, 56, 66, 68, 70, 74, 82, and 223.

AGRAWAL, R.; IMIELIŃSKI, T.; SWAMI, A. Mining association rules between sets of items in large databases. In: ACM. *ACM SIGMOD Record*. [S.l.], 1993. v. 22, n. 2, p. 207–216. Cited 2 many times on page 79 and 121.

ALHAMID, M. et al. Recam: a collaborative context-aware framework for multimedia recommendations in an ambient intelligence environment. *Multimedia Systems*, Springer Berlin Heidelberg, online, p. 1–15, 2015. ISSN 0942-4962. Disponível em: <http://dx.doi.org/10.1007/s00530-015-0469-2>. Cited on page 55.

AMATRIAIN, X. et al. Data mining methods for recommender systems. In: *Recommender Systems Handbook*. [S.l.]: Springer, 2011. p. 39–71. Cited on page 60.

ANAND, S. S.; MOBASHER, B. *Contextual recommendation*. [S.l.]: Springer, 2006. Cited on page 53.

ANSARI, A.; ESSEGAIER, S.; KOHLI, R. Internet recommendation systems. *Journal of Marketing research*, American Marketing Association, v. 37, n. 3, p. 363–375, 2000. Cited on page 54.

AZAK, M. *CrosSing: A framework to develop knowledge-based recommenders in cross domains*. Dissertação (Mestrado) — MIDDLE EAST TECHNICAL UNIVERSITY, 2010. Cited 2 many times on page 25 and 44.

BALTRUNAS, L. et al. Incarmusic: Context-aware music recommendations in a car. In: SPRINGER. *EC-Web*. [S.l.], 2011. v. 11, p. 89–100. Cited on page 47.

BALTRUNAS, L.; LUDWIG, B.; RICCI, F. Matrix factorization techniques for context aware recommendation. In: ACM. *Proceedings of the fifth ACM conference on Recommender systems*. [S.l.], 2011. p. 301–304. Cited 2 many times on page 54 and 81.

BALTRUNAS, L.; MAKCINSKAS, T.; RICCI, F. Group recommendations with rank aggregation and collaborative filtering. In: *Proceedings of the fourth ACM conference on Recommender systems*. [S.l.: s.n.], 2010. p. 119–126. ISBN 9781605589060. Cited on page 37.

BAUMAN, K.; TUZHILIN, A. Discovering contextual information from user reviews for recommendation purposes. In: *CBRecSys*. [S.l.: s.n.], 2014. p. 1–8. Cited 4 many times on page 98, 99, 100, and 101.

BAZIRE, M.; BRÉZILLON, P. Understanding context before using it. In: *Modeling and using context*. [S.l.]: Springer, 2005. p. 29–40. Cited on page 48.

BELL, R.; KOREN, Y.; VOLINSKY, C. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In: ACM. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2007. p. 95–104. Cited on page 87.

BENNETT, J.; LANNING, S. The netflix prize. In: *Proceedings of KDD cup and workshop*. [S.l.: s.n.], 2007. v. 2007, p. 35–36. Cited 2 many times on page 32 and 38.

BERKOVSKY, S.; KUFLIK, T.; RICCI, F. Cross-domain mediation in collaborative filtering. In: *User Modeling 2007*. [S.l.]: Springer, 2007. p. 355–359. Cited 4 many times on page 42, 44, 57, and 58.

BERKOVSKY, S.; KUFLIK, T.; RICCI, F. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, Springer, v. 18, n. 3, p. 245–286, 2008. Cited on page 47.

BERNERS-LEE, T.; HENDLER, J. The semantic web. a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American Magazine*, v. 1, p. 34–43, maio 2001. Cited on page 35.

BETTINI, C. et al. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, Elsevier, v. 6, n. 2, p. 161–180, 2010. Cited 3 many times on page 49, 90, and 223.

BEZERRA, B. L.; CARVALHO, F. d. A. de. A symbolic approach for content-based information filtering. *Information Processing Letters*, Elsevier, v. 92, n. 1, p. 45–52, 2004. Cited on page 86.

BEZERRA, B. L. D.; CARVALHO, F. D. A. T. D. Symbolic data analysis tools for recommendation systems. *Knowledge and Information Systems*, Springer, v. 26, n. 3, p. 385–418, 2011. Cited on page 86.

BLANCO-FERNÁNDEZ, Y. et al. Tripfromtv+: Exploiting social networks to arrange cut-price touristic packages. In: IEEE. *IEEE International Conference on Consumer Electronics (ICCE)*. [S.l.], 2011. p. 223–224. Cited 3 many times on page 62, 63, and 67.

BLANCO-FERNÁNDEZ, Y. et al. Exploiting digital tv users' preferences in a tourism recommender system based on semantic reasoning. *Consumer Electronics, IEEE Transactions on*, IEEE, v. 56, n. 2, p. 904–912, 2010. Cited 3 many times on page 62, 63, and 67.

BLANCO-FERNÁNDEZ, Y. et al. Tripfromtv+: targeting personalized tourism to interactive digital tv viewers by social networking and semantic reasoning. *IEEE Transactions on Consumer Electronics*, IEEE, v. 57, n. 2, p. 953–961, 2011. Cited 7 many times on page 29, 35, 61, 62, 63, 64, and 67.

BLANCO-FERNÁNDEZ, Y.; PAZOS-ARIAS, J. An MHP framework to provide intelligent personalized recommendations about digital TV contents. *Software: Practice and Experience*, v. 38, n. October 2007, p. 925–960, 2008. Cited on page 35.

BLANCO-FERNáNDEZ, Y. et al. Exploiting synergies between semantic reasoning and personalization strategies in intelligent recommender systems: A case study. *Journal of Systems and Software*, Elsevier Inc., v. 81, n. 12, p. 2371–2385, dez. 2008. ISSN 01641212. Cited on page 37.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *the Journal of machine Learning research*, JMLR. org, v. 3, p. 993–1022, 2003. Cited on page 99.

BOUNEFFOUF, D. Situation-aware approach to improve context-based recommender system. *arXiv preprint arXiv:1303.0481*, 2013. Cited on page 52.

BOURKE, S.; MCCARTHY, K.; SMYTH, B. Power to the people: exploring neighbourhood formations in social recommender system. In: *Proceedings of the fifth ACM conference on Recommender systems*. [S.l.: s.n.], 2011. p. 337–340. ISBN 9781450306836. Cited on page 34.

BOYTSOV, A. et al. Situation awareness meets ontologies: A context spaces case study. In: SPRINGER. *International and Interdisciplinary Conference on Modeling and Using Context*. [S.l.], 2015. p. 3–17. Cited on page 52.

BRAUNHOFER, M.; KAMINSKAS, M.; RICCI, F. Location-aware music recommendation. *International Journal of Multimedia Information Retrieval*, Springer, v. 2, n. 1, p. 31–44, 2013. Cited 5 many times on page 61, 62, 63, 65, and 67.

BREESE, J. S.; HECKERMAN, D.; KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. In: MORGAN KAUFMANN PUBLISHERS INC. *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. [S.l.], 1998. p. 43–52. Cited on page 37.

BRÉZILLON, P. Context modeling: Task model and practice model. In: *Modeling and Using Context*. [S.l.]: Springer, 2007. p. 122–135. Cited 2 many times on page 49 and 52.

BRUSILOVSKY, P.; KOBSA, A.; NEJDL, W. *The adaptive web: methods and strategies of web personalization*. [S.l.]: Springer, 2007. v. 4321. Cited on page 35.

BURKE, R. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, Springer, v. 12, n. 4, p. 331–370, 2002. Cited on page 32.

BURKE, R. Hybrid web recommender systems. In: *The adaptive web*. [S.l.]: Springer, 2007. p. 377–408. Cited 2 many times on page 32 and 33.

CAMPOS, P. G.; DÍEZ, F.; CANTADOR, I. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, Springer, v. 24, n. 1-2, p. 67–119, 2014. Cited on page 56.

CANTADOR, I.; CREMONESI, P. Tutorial on cross-domain recommender systems. In: ACM. *Proceedings of the 8th ACM Conference on Recommender systems*. [S.l.], 2014. p. 401–402. Cited on page 68.

CANTADOR, I. et al. Cross-domain recommender systems. In: *Recommender Systems Handbook*. [S.l.]: Springer, 2015. p. 919–959. Cited 13 many times on page 9, 26, 30, 38, 39, 40, 41, 43, 44, 45, 46, 47, and 57.

CAO, B.; LIU, N. N.; YANG, Q. Transfer learning for collective link prediction in multiple heterogenous domains. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. [S.l.: s.n.], 2010. p. 159–166. Cited 2 many times on page 38 and 47.

CARMAGNOLA, F.; CENA, F. User identification for cross-system personalisation. *Information Sciences*, Elsevier, v. 179, n. 1, p. 16–32, 2009. Cited on page 40.

CARMAGNOLA, F.; CENA, F.; GENA, C. User model interoperability: a survey. *User Modeling and User-Adapted Interaction*, Springer, v. 21, n. 3, p. 285–331, 2011. Cited on page 40.

CHAARI, T. et al. A comprehensive approach to model and use context for adapting applications in pervasive environments. *Journal of Systems and Software*, Elsevier, v. 80, n. 12, p. 1973–1992, 2007. Cited on page 49.

CHATTERJEE, S.; HADI, A. S. *Regression analysis by example*. [S.l.]: John Wiley & Sons, 2015. Cited on page 52.

CHEN, G.; CHEN, L. Augmenting service recommender systems by incorporating contextual opinions from user reviews. *User Modeling and User-Adapted Interaction*, Springer, v. 25, n. 3, p. 295–329, 2015. Cited on page 225.

CHUNG, R.; SUNDARAM, D.; SRINIVASAN, A. Integrated personal recommender systems. In: ACM. *Proceedings of the ninth international conference on Electronic commerce*. [S.l.], 2007. p. 65–74. Cited on page 44.

CHURCH, K. et al. Mobile information access: A study of emerging search behavior on the mobile internet. *ACM Transactions on the Web (TWEB)*, ACM, v. 1, n. 1, p. 4, 2007. Cited on page 47.

COLOMBO-MENDOZA, L. O. et al. Recommetz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Systems with Applications*, Elsevier, v. 42, n. 3, p. 1202–1222, 2015. Cited on page 51.

CREMONESI, P.; GARZOTTO, F.; TURRIN, R. Investigating the Persuasion Potential of Recommender Systems from a Quality Perspective. *ACM Transactions on Interactive Intelligent Systems*, v. 2, n. 2, p. 1–41, jun. 2012. ISSN 21606455. Cited on page 33.

CREMONESI, P.; KOREN, Y.; TURRIN, R. Performance of recommender algorithms on top-n recommendation tasks. In: ACM. *Proceedings of the fourth ACM conference on Recommender systems*. [S.l.], 2010. p. 39–46. Cited 3 many times on page 59, 129, and 130.

CREMONESI, P.; TRIPODI, A.; TURRIN, R. Cross-domain recommender systems. In: IEEE. *IEEE 11th International Conference on Data Mining Workshops (ICDMW)*. [S.l.], 2011. p. 496–503. Cited 20 many times on page 9, 24, 25, 41, 42, 43, 47, 58, 59, 61, 68, 84, 86, 88, 89, 90, 123, 124, 129, and 130.

CREMONESI, P.; TURRIN, R. Controlling Consistency in Top-N Recommender Systems. In: *IEEE International Conference on Data Mining Workshops*. [S.l.]: Ieee, 2010. p. 919–926. ISBN 978-1-4244-9244-2. Cited on page 37.

DAS, A. S. et al. Google news personalization: scalable online collaborative filtering. In: ACM. *Proceedings of the 16th international conference on World Wide Web*. [S.l.], 2007. p. 271–280. Cited on page 32.

DEY, A. K.; ABOWD, G. D.; SALBER, D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-computer interaction*, L. Erlbaum Associates Inc., v. 16, n. 2, p. 97–166, 2001. Cited on page 48.

DIDAY, E.; BOCK, H.-H. *Analysis of symbolic data: Exploratory methods for extracting statistical information from complex data*. [S.l.]: Springer-Verlag, 2000. Cited on page 86.

DOMINGUES, M. A. et al. Exploiting text mining techniques for contextual recommendations. In: IEEE. *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*. [S.l.], 2014. v. 2, p. 210–217. Cited on page 225.

DOURISH, P. What we talk about when we talk about context. *Personal and ubiquitous computing*, Springer, v. 8, n. 1, p. 19–30, 2004. Cited on page 53.

ENRICH, M.; BRAUNHOFER, M.; RICCI, F. Cold-start management with cross-domain collaborative filtering and tags. In: *E-Commerce and Web Technologies*. [S.l.]: Springer, 2013. p. 101–112. Cited 2 many times on page 39 and 44.

EYKE, J. W. *Temporal Problems, with a Focus on Mood, in Music Recommendation Within Last. FM*. Tese (Doutorado) — University of Sheffield, Department of Information Studies, 2009. Cited 2 many times on page 32 and 38.

FERNÁNDEZ-TOBÍAS, I. et al. Cross-domain recommender systems: A survey of the state of the art. In: *Spanish Conference on Information Retrieval*. [S.l.: s.n.], 2012. Cited 10 many times on page 24, 25, 26, 30, 38, 43, 61, 68, 90, and 131.

FERRAZ, C. A.; SILVA, D. V. e; SILVA, J. S. da. A collaborative tv-internet application model to enrich tv viewing experience in a pervasive way. In: IEEE. *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. [S.l.], 2015. p. 148–153. Cited on page 226.

FREYNE, J.; BERKOVSKY, S. Evaluating recommender systems for supportive technologies. In: *User Modeling and Adaptation for Daily Routines*. [S.l.]: Springer, 2013. p. 195–217. Cited on page 45.

GAO, S. et al. Cross-domain recommendation via cluster-level latent factor model. In: *Machine Learning and Knowledge Discovery in Databases*. [S.l.]: Springer, 2013. p. 161–176. Cited 3 many times on page 39, 44, and 224.

GIVON, S.; LAVRENKO, V. Predicting social-tags for cold start book recommendations. In: ACM. *Proceedings of the third ACM conference on Recommender systems*. [S.l.], 2009. p. 333–336. Cited on page 44.

GOGA, O. et al. Exploiting innocuous activity for correlating users across sites. In: INTERNATIONAL WORLD WIDE WEB CONFERENCES STEERING COMMITTEE. *Proceedings of the 22nd international conference on World Wide Web*. [S.l.], 2013. p. 447–458. Cited on page 46.

GU, T.; PUNG, H. K.; ZHANG, D. Q. A service-oriented middleware for building context-aware services. *Journal of Network and computer applications*, Elsevier, v. 28, n. 1, p. 1–18, 2005. Cited on page 49.

GUPTA, K. M. Taxonomic conversational case-based reasoning. In: *Case-Based Reasoning Research and Development*. [S.l.]: Springer, 2001. p. 219–233. Cited on page 64.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, JMLR. org, v. 3, p. 1157–1182, 2003. Cited on page 52.

HALL, M. et al. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, ACM, v. 11, n. 1, p. 10–18, 2009. Cited 2 many times on page 103 and 126.

HAN, X. et al. Alike people, alike interests? inferring interest similarity in online social networks. *Decision Support Systems*, Elsevier, v. 69, p. 92–106, 2015. Cited on page 34.

HENRICKSEN, K.; INDULSKA, J. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and mobile computing*, Elsevier, v. 2, n. 1, p. 37–64, 2006. Cited on page 49.

HERLOCKER, J. L. et al. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, ACM, v. 22, n. 1, p. 5–53, 2004. Cited on page 37.

HIDASI, B.; TIKK, D. Fast als-based tensor factorization for context-aware recommendation from implicit feedback. In: *Machine Learning and Knowledge Discovery in Databases*. [S.l.]: Springer, 2012. p. 67–82. Cited 2 many times on page 54 and 82.

HILL, W. et al. Recommending and evaluating choices in a virtual community of use. In: ACM PRESS/ADDISON-WESLEY PUBLISHING CO. *Proceedings of the SIGCHI conference on Human factors in computing systems*. [S.l.], 1995. p. 194–201. Cited on page 23.

HIPP, J.; GÜNTZER, U.; NAKHAEIZADEH, G. Algorithms for association rule mining—a general survey and comparison. *ACM sigkdd explorations newsletter*, ACM, v. 2, n. 1, p. 58–64, 2000. Cited on page 79.

HOPFGARTNER, F.; JOSE, J. Semantic user profiling techniques for personalised multimedia recommendation. *Multimedia systems*, v. 16, p. 255–274, 2010. Cited on page 37.

HU, L. et al. Personalized recommendation via cross-domain triadic factorization. In: INTERNATIONAL WORLD WIDE WEB CONFERENCES STEERING COMMITTEE. *Proceedings of the 22nd international conference on World Wide Web*. [S.l.], 2013. p. 595–606. Cited on page 39.

HU, Y.; KOREN, Y.; VOLINSKY, C. Collaborative filtering for implicit feedback datasets. In: IEEE. *Eighth IEEE International Conference on Data Mining (ICDM)*. [S.l.], 2008. p. 263–272. Cited on page 36.

JADIDI, O.; FIROUZI, F.; BAGLIERY, E. Topsis method for supplier selection problem. *World Academy of Science, Engineering and Technology*, Citeseer, v. 47, p. 956–958, 2010. Cited on page 64.

JAIN, A. K. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, Elsevier, v. 31, n. 8, p. 651–666, 2010. Cited on page 99.

JAIN, P.; KUMARAGURU, P.; JOSHI, A. @ i seek'fb. me': Identifying users across multiple online social networks. In: INTERNATIONAL WORLD WIDE WEB CONFERENCES STEERING COMMITTEE. *Proceedings of the 22nd international conference on World Wide Web companion*. [S.l.], 2013. p. 1259–1268. Cited on page 46.

JI, K.; SHEN, H. Making recommendations from top-n user-item subgroups. *Neurocomputing*, Elsevier, v. 165, p. 228–237, 2015. Cited 5 many times on page 61, 62, 63, 66, and 67.

JOJIC, O.; SHUKLA, M.; BHOSAREKAR, N. A probabilistic definition of item similarity. In: *Proceedings of the fifth ACM conference on Recommender systems*. New York, New York, USA: ACM Press, 2011. p. 229–236. ISBN 9781450306836. Cited on page 37.

KAMAHARA, J. et al. A Community-Based Recommendation System to Reveal Unexpected Interests. In: *11th International Multimedia Modelling Conference*. [S.l.]: IEEE, 2005. p. 433–438. ISBN 0-7695-2164-9. Cited on page 34.

KAMINSKAS, M. et al. Knowledge-based identification of music suited for places of interest. *Information Technology & Tourism*, Springer, v. 14, n. 1, p. 73–95, 2014. Cited 9 many times on page 29, 30, 51, 55, 61, 62, 63, 65, and 67.

KAMINSKAS, M.; RICCI, F. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, Elsevier, v. 6, n. 2, p. 89–119, 2012. Cited on page 47.

KARATZOGLOU, A. et al. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: ACM. *Proceedings of the fourth ACM conference on Recommender systems*. [S.l.], 2010. p. 79–86. Cited 2 many times on page 54 and 82.

KIM, S.; YOON, Y. Recommendation system for sharing economy based on multidimensional trust model. *Multimedia Tools and Applications*, Springer, p. 1–14, 2014. Cited 3 many times on page 54, 82, and 88.

KOREN, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: ACM. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2008. p. 426–434. Cited 2 many times on page 53 and 87.

LAHLOU, F. Z. et al. A text classification based method for context extraction from online reviews. In: IEEE. *8th International Conference on Intelligent Systems: Theories and Applications (SITA)*. [S.l.], 2013. p. 1–5. Cited on page 225.

LAROSE, D. T. k-nearest neighbor algorithm. In: _____. *Discovering Knowledge in Data*. John Wiley and Sons, Inc., 2005. p. 90–106. ISBN 9780471687542. Disponível em: <http://dx.doi.org/10.1002/0471687545.ch5>. Cited on page 60.

LEE, H.; KWON, J. Personalized tv contents recommender system using collaborative context tagging-based user's preference prediction technique. *International Journal of Multimedia & Ubiquitous Engineering*, v. 9, n. 5, p. 231–240, 2014. Cited on page 51.

LEE, H. J.; PARK, S. J. Moners: A news recommender for the mobile web. *Expert Systems with Applications*, Elsevier, v. 32, n. 1, p. 143–150, 2007. Cited on page 47.

LEE, W.; YANG, T.-H. Personalizing information appliances: a multi-agent framework for TV programme recommendations. *Expert Systems with Applications*, v. 25, n. 3, p. 331–341, out. 2003. ISSN 09574174. Cited on page 37.

LEKAKOS, G.; CARAVELAS, P. A hybrid approach for movie recommendation. *Multimedia Tools and Applications*, v. 36, n. December 2006, p. 55–70, 2008. Cited on page 37.

LEKAKOS, G.; GIAGLIS, G. A Lifestyle‚ÄêBased Approach for Delivering Personalized Advertisements in Digital Interactive Television. *Journal of Computer-Mediated Communication*, v. 6, n. 1, p. 00–00, 2004. Cited on page 37.

LESKOVEC, J.; ADAMIC, L. A.; HUBERMAN, B. A. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, ACM, v. 1, n. 1, p. 5, 2007. Cited 3 many times on page 61, 92, and 223.

LI, B.; YANG, Q.; XUE, X. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In: *IJCAI*. [S.l.: s.n.], 2009. v. 9, p. 2052–2057. Cited 2 many times on page 66 and 224.

LI, B.; YANG, Q.; XUE, X. Transfer learning for collaborative filtering via a rating-matrix generative model. In: ACM. *Proceedings of the 26th Annual International Conference on Machine Learning*. [S.l.], 2009. p. 617–624. Cited 3 many times on page 44, 46, and 47.

LI, L. et al. A contextual-bandit approach to personalized news article recommendation. In: ACM. *Proceedings of the 19th international conference on World wide web*. [S.l.], 2010. p. 661–670. Cited on page 66.

LIAW, A.; WIENER, M. Classification and regression by randomforest. *R news*, v. 2, n. 3, p. 18–22, 2002. Cited on page 60.

LINDEN, G.; SMITH, B.; YORK, J. Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, IEEE, v. 7, n. 1, p. 76–80, 2003. Cited on page 32.

LIU, H.; MOTODA, H. *Feature Selection for Knowledge Discovery and Data Mining.* [S.l.]: Springer Science & Business Media, 1998. Cited on page 73.

LIU, H.; MOTODA, H. *Feature selection for knowledge discovery and data mining.* [S.l.]: Springer Science & Business Media, 2012. v. 454. Cited on page 52.

LONI, B. et al. Cross-domain collaborative filtering with factorization machines. In: *Advances in Information Retrieval.* [S.l.]: Springer, 2014. p. 656–661. Cited 3 many times on page 39, 58, and 60.

LóPEZ-NORES, M. et al. MiSPOT: dynamic product placement for digital TV through MPEG-4 processing and semantic reasoning. *Knowledge and Information Systems*, v. 22, n. 1, p. 101–128, mar. 2009. ISSN 0219-1377. Cited on page 37.

LOVÁSZ, L. et al. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, v. 2, p. 353–398, 1996. Cited on page 59.

LOW, Y.; AGARWAL, D.; SMOLA, A. J. Multiple domain user personalization. In: ACM. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.* [S.l.], 2011. p. 123–131. Cited on page 40.

MAHMOOD, T.; RICCI, F.; VENTURINI, A. Improving recommendation effectiveness: Adapting a dialogue strategy in online travel planning. *Information Technology & Tourism*, Cognizant Communication Corporation, v. 11, n. 4, p. 285–302, 2009. Cited on page 47.

MARILLY, E. et al. Community-based applications. *Bell Labs Technical Journal*, v. 15, n. 4, p. 93–109, mar. 2011. ISSN 10897089. Cited on page 35.

MCJONES, P. Eachmovie collaborative filtering data set. *DEC Systems Research Center*, v. 249, 1997. Cited on page 57.

MILLER, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, ACM, v. 38, n. 11, p. 39–41, 1995. Cited on page 100.

MOE, H. H.; AUNG, W. T. Building ontologies for cross-domain recommendation on facial skin problem and related cosmetics. *International Journal of Information Technology and Computer Science (IJITCS)*, v. 6, n. 6, p. 33, 2014. Cited 4 many times on page 61, 62, 63, and 67.

MOE, H. H.; AUNG, W. T. Context aware cross-domain based recommendation. In: *International Conference on Advances in Engineering and Technology.* [S.l.: s.n.], 2014. Cited 7 many times on page 29, 55, 61, 62, 63, 64, and 67.

MOE, H. H.; AUNG, W. T. et al. Cross-domain recommendations for personalized semantic services. *International Journal of Computer Applications Technology and Research*, v. 2, n. 1, p. 72–76, 2013. Cited 4 many times on page 61, 62, 63, and 67.

MOON, A. et al. Designing CAMUS based context-awareness for pervasive home environments. In: *International Conference on Hybrid Information Technology.* [S.l.: s.n.], 2006. v. 1, p. 666–672. ISBN 0769526748. Cited on page 54.

MOON, A. et al. Two-step recommendation based personalization for future services. In: *International Conference on Advanced Communication Technology.* [S.l.: s.n.], 2009. v. 03, p. 2268–2272. ISBN 9788955191394. Cited 2 many times on page 34 and 55.

MORENO, O. et al. Talmud: transfer learning for multiple domains. In: ACM. *Proceedings of the 21st ACM international conference on Information and knowledge management*. [S.l.], 2012. p. 425–434. Cited 2 many times on page 40 and 41.

MUKHERJEE, D. et al. A context-aware recommendation system considering both user preferences and learned behavior. In: *7th International Conference on Information Technology in Asia*. [S.l.: s.n.], 2011. p. 1–7. ISBN 9781612841304. Cited on page 36.

NAKATSUJI, M. et al. Recommendations over domain specific user graphs. In: *ECAI*. [S.l.: s.n.], 2010. p. 607–612. Cited 2 many times on page 58 and 59.

NETO, B.; FREITAS, R. de. *Um processo de software e um modelo ontológico para apoio ao desenvolvimento de aplicações sensíveis a contexto*. Tese (Doutorado) — Universidade de São Paulo, 2007. Cited on page 51.

OH, S. et al. Comparison of techniques for time aware tv channel recommendation. In: IEEE. *Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on*. [S.l.], 2014. p. 989–992. Cited on page 51.

OKU, K. et al. Context-aware svm for context-dependent information recommendation. In: IEEE COMPUTER SOCIETY. *Proceedings of the 7th international Conference on Mobile Data Management*. [S.l.], 2006. p. 109. Cited on page 54.

O'SULLIVAN, D.; SMYTH, B.; WILSON, D. Improving the quality of the personalized electronic program guide. *User Modeling and User-Adapted Interaction*, v. 14, p. 5–36, 2004. Cited on page 37.

OWEN, S. et al. *Mahout in action*. [S.l.]: Manning, 2011. Cited 3 many times on page 109, 111, and 126.

PALMISANO, C.; TUZHILIN, A.; GORGOGLIONE, M. Using context to improve predictive modeling of customers in personalization applications. *Knowledge and Data Engineering, IEEE Transactions on*, IEEE, v. 20, n. 11, p. 1535–1549, 2008. Cited 3 many times on page 48, 51, and 53.

PAN, W. et al. Transfer learning in collaborative filtering for sparsity reduction. In: *AAAI*. [S.l.: s.n.], 2010. v. 10, p. 230–235. Cited 2 many times on page 44 and 47.

PAN, W.; XIANG, E. W.; YANG, Q. Transfer learning in collaborative filtering with uncertain ratings. In: *AAAI*. [S.l.: s.n.], 2012. Cited on page 39.

PAN, W.; YANG, Q. Transfer learning in heterogeneous collaborative filtering domains. *Artificial intelligence*, Elsevier, v. 197, p. 39–55, 2013. Cited 2 many times on page 39 and 45.

PANNIELLO, U.; TUZHILIN, A.; GORGOGLIONE, M. Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and User-Adapted Interaction*, Springer, v. 24, n. 1-2, p. 35–65, 2014. Cited on page 56.

PANNIELLO, U. et al. Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems. In: ACM. *Proceedings of the third ACM conference on Recommender systems*. [S.l.], 2009. p. 265–268. Cited on page 54.

PARAMESWARAN, A.; VENETIS, P.; GARCIA-MOLINA, H. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems (TOIS)*, ACM, v. 29, n. 4, p. 20, 2011. Cited on page 64.

PARK, D. H. et al. A literature review and classification of recommender systems research. *Expert Systems with Applications*, Elsevier, v. 39, n. 11, p. 10059–10072, 2012. Cited 2 many times on page 32 and 34.

PAZZANI, M. J. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, Springer, v. 13, n. 5-6, p. 393–408, 1999. Cited on page 33.

PESSEMIER, T. D.; DOOMS, S.; MARTENS, L. Context-aware recommendations through context and activity recognition in a mobile environment. *Multimedia Tools and Applications*, Springer, v. 72, n. 3, p. 2925–2948, 2014. Cited on page 47.

PHAM, X. H.; JUNG, J. J.; VU, S.-B. P. L. A. Exploiting social contexts for movie recommendation. *Malaysian Journal of Computer Science*, v. 27, n. 1, p. 68–79, 2014. Cited on page 51.

QUEIROZ, S. R. d. M.; CARVALHO, F. d. A. de. Making collaborative group recommendations based on modal symbolic data. In: SPRINGER. *Brazilian Symposium on Artificial Intelligence*. [S.l.], 2004. p. 307–316. Cited on page 35.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2015. Disponível em: <http://www.R-project.org/>. Cited on page 131.

REICHLING, T.; WULF, V. Expert recommender systems in practice: evaluating semi-automatic profile generation. In: ACM. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. [S.l.], 2009. p. 59–68. Cited on page 36.

RENDLE, S. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM, v. 3, n. 3, p. 57, 2012. Cited on page 60.

RESNICK, P. et al. Grouplens: an open architecture for collaborative filtering of netnews. In: ACM. *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. [S.l.], 1994. p. 175–186. Cited on page 23.

RESNICK, P.; VARIAN, H. R. Recommender systems. *Communications of the ACM*, ACM, v. 40, n. 3, p. 56–58, 1997. Cited on page 32.

RICCI, F.; ROKACH, L.; SHAPIRA, B. *Introduction to recommender systems handbook*. [S.l.]: Springer, 2011. Cited 11 many times on page 23, 24, 33, 34, 35, 37, 81, 84, 85, 86, and 87.

SAHEBI, S.; BRUSILOVSKY, P. Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In: *User Modeling, Adaptation, and Personalization*. [S.l.]: Springer, 2013. p. 289–295. Cited 9 many times on page 42, 44, 45, 47, 57, 58, 60, 61, and 131.

SAHEBI, S.; COHEN, W. W. Community-based recommendations: a solution to the cold start problem. In: *Workshop on Recommender Systems and the Social Web, RSWEB*. [S.l.: s.n.], 2011. Cited on page 60.

SANTOS, V. dos et al. A recommender system architecture for an inter-application environment. In: IEEE. *12th International Conference on Intelligent Systems Design and Applications (ISDA)*. [S.l.], 2012. p. 472–477. Cited 12 many times on page 9, 25, 41, 58, 59, 61, 69, 73, 74, 90, 220, and 224.

SETTEN, M. V.; POKRAEV, S.; KOOLWAAIJ, J. Context-aware recommendations in the mobile tourist application compass. In: SPRINGER. *Adaptive hypermedia and adaptive web-based systems*. [S.l.], 2004. p. 235–244. Cited on page 47.

SHANI, G.; GUNAWARDANA, A. Evaluating recommendation systems. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 257–297. Cited 2 many times on page 37 and 128.

SHAPIRA, B.; ROKACH, L.; FREILIKHMAN, S. Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, Springer, v. 23, n. 2-3, p. 211–247, 2013. Cited 8 many times on page 38, 41, 44, 47, 57, 58, 60, and 61.

SHARDANAND, U.; MAES, P. Social information filtering: algorithms for automating "word of mouth". In: ACM PRESS/ADDISON-WESLEY PUBLISHING CO. *Proceedings of the SIGCHI conference on Human factors in computing systems*. [S.l.], 1995. p. 210–217. Cited on page 23.

SHEPSTONE, S.; TAN, Z.-H.; JENSEN, S. Using audio-derived affective offset to enhance tv recommendation. *Multimedia, IEEE Transactions on*, v. 16, n. 7, p. 1999–2010, Nov 2014. ISSN 1520-9210. Cited 2 many times on page 47 and 52.

SHI, Y.; LARSON, M.; HANJALIC, A. Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering. *User Modeling, Adaption and Personalization*, Springer, v. 6787, p. 305–316, 2011. Cited 2 many times on page 44 and 47.

SONG, S.; MOUSTAFA, H.; AFIFI, H. Enriched IPTV services personalization. In: *IEEE International Conference on Communications*. [S.l.]: Ieee, 2012. p. 1911–1916. ISBN 978-1-4577-2053-6. Cited on page 54.

SOUZA, D. et al. Towards a context ontology to enhance data integration processes. In: *Proceedings of the 4th Workshop on Ontologies-based Techniques for Databases (in VLDB'08)*. [S.l.: s.n.], 2008. Cited on page 49.

STEWART, A. et al. Cross-tagging for personalized open social networking. In: ACM. *Proceedings of the 20th ACM conference on Hypertext and hypermedia*. [S.l.], 2009. p. 271–278. Cited 3 many times on page 40, 41, and 44.

STRANG, T.; LINNHOFF-POPIEN, C. A context modeling survey. In: *Workshop Proceedings*. [S.l.: s.n.], 2004. Cited on page 49.

SZOMSZOR, M. et al. *Semantic modelling of user interests based on cross-folksonomy analysis*. [S.l.]: Springer, 2008. Cited on page 42.

TANG, J. et al. Cross-domain collaboration recommendation. In: ACM. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2012. p. 1285–1293. Cited on page 92.

TANG, X.; WAN, X.; ZHANG, X. Cross-language context-aware citation recommendation in scientific articles. In: ACM. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. [S.l.], 2014. p. 817–826. Cited 5 many times on page 61, 62, 63, 65, and 67.

TEKIN, C.; SCHAAR, M. van der. Contextual online learning for multimedia content aggregation. *Multimedia, IEEE Transactions on*, IEEE, v. 17, n. 4, p. 549–561, 2015. Cited 6 many times on page 61, 62, 63, 65, 66, and 67.

TIROSHI, A. et al. Cross social networks interests predictions based ongraph features. In: ACM. *Proceedings of the 7th ACM conference on Recommender systems*. [S.l.], 2013. p. 319–322. Cited 3 many times on page 58, 59, and 60.

TIROSHI, A.; KUFLIK, T. Domain ranking for cross domain collaborative filtering. In: *User Modeling, Adaptation, and Personalization*. [S.l.]: Springer, 2012. p. 328–333. Cited 2 many times on page 40 and 42.

TREWIN, S. Knowledge-based recommender systems. *Encyclopedia of library and information science*, v. 69, n. Supplement 32, p. 180, 2000. Cited 2 many times on page 24 and 25.

TUCKER, L. R. Some mathematical notes on three-mode factor analysis. *Psychometrika*, Springer, v. 31, n. 3, p. 279–311, 1966. Cited on page 64.

UBERALL, C.; MUTTUKRISHNAN, R. Recommendation index for DVB content using service information. In: *IEEE International Conference on Multimedia and Expo*. [S.l.: s.n.], 2009. p. 1–4. Cited 2 many times on page 35 and 36.

VÉRAS, D. et al. A literature review of recommender systems in the television domain. *Expert Systems with Applications*, Elsevier, v. 42, n. 22, p. 9046–9076, 2015. Cited 4 many times on page 33, 35, 36, and 54.

VERAS, D. et al. Context-aware techniques for cross-domain recommender systems. In: IEEE. *2015 Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.], 2015. p. 282–287. Cited 2 many times on page 40 and 54.

VIEIRA, V. et al. A context-oriented model for domain-independent context management. *Revue d'intelligence artificielle*, v. 22, n. 5, p. 609–627, 2008. Cited on page 49.

VIEIRA, V.; TEDESCO, P.; SALGADO, A. C. Towards an ontology for context representation in groupware. In: *Groupware: Design, Implementation, and Use*. [S.l.]: Springer, 2005. p. 367–375. Cited on page 49.

VIEIRA, V.; TEDESCO, P.; SALGADO, A. C. Modelos e processos para o desenvolvimento de sistemas sensíveis ao contexto. In: *Jornadas de Atualização em Informática*. [S.l.]: André Ponce de Leon F. de Carvalho, Tomasz Kowaltowski.(Org.), 2009. p. 381–431. Cited 6 many times on page 17, 48, 49, 50, 90, and 223.

VILDJIOUNAITE, E. et al. Unobtrusive dynamic modelling of tv programme preferences in a finnish household. *Multimedia systems*, v. 15, p. 143–157, 2009. Cited on page 55.

WANG, F.; LI, D.; XU, M. A location-aware tv show recommendation with localized sementaic analysis. *Multimedia Systems*, Springer Berlin Heidelberg, online, p. 1–8, 2015. ISSN 0942-4962. Disponível em: <http://dx.doi.org/10.1007/s00530-015-0451-z>. Cited 2 many times on page 52 and 55.

WINOTO, P.; TANG, T. If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? a study of cross-domain recommendations. *New Generation Computing*, Springer, v. 26, n. 3, p. 209–225, 2008. Cited 5 many times on page 24, 25, 41, 58, and 61.

WINTER, J. C. D.; DODOU, D. Five-point likert items: t test versus mann-whitney-wilcoxon. *Practical Assessment, Research & Evaluation*, Dr. Lawrence M. Rudner, v. 15, n. 11, p. 1–12, 2010. Cited on page 131.

YUAN, Z. et al. Structural context-aware cross media recommendation. In: *Advances in Multimedia Information Processing–PCM 2012*. [S.l.]: Springer, 2012. p. 790–800. Cited 5 many times on page 62, 63, 64, 66, and 67.

ZHANG, H.; ZHENG, S. Personalized TV program recommendation based on TV-anytime metadata. In: *IEEE International Symposium on Consumer Electronics*. [S.l.: s.n.], 2005. di, p. 242–246. ISBN 0780389204. Cited on page 37.

ZHANG, J.; YUAN, Z.; YU, K. Cross media recommendation in digital library. In: *The Emergence of Digital Libraries–Research and Practices*. [S.l.]: Springer, 2014. p. 208–217. Cited 4 many times on page 61, 62, 63, and 67.

ZHAO, L. et al. Active transfer learning for cross-system recommendation. In: CITESEER. *AAAI*. [S.l.], 2013. Cited on page 47.

ZHIWEN, Y.; XINGSHE, Z. Design, implementation, and evaluation of an agent-based adaptive program personalization system. In: *Fifth International Symposium on Multimedia Software Engineering*. [S.l.: s.n.], 2003. p. 140–147. ISBN 0769520316. Cited on page 37.

ZHUANG, F. et al. Cross-domain learning from multiple sources: a consensus regularization perspective. *Knowledge and Data Engineering, IEEE Transactions on*, IEEE, v. 22, n. 12, p. 1664–1678, 2010. Cited on page 44.