Expert Systems With Applications 135 (2019) 388–409

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

CD-CARS: Cross-domain context-aware recommender systems

Douglas Véras[a,b,*], Ricardo Prudêncio[a], Carlos Ferraz[a]

[a] *Universidade Federal de Pernambuco (UFPE), Centro de Infromática, Av. Jornalista Anibal Fernandes, s/n, Cidade Universitária, CEP, Recife-PE, 50740-560, Brazil*

[b] *Universidade Federal Rural de Pernambuco (UFRPE), Department of Statistics and Informatics, Rua Dom Manoel de Medeiros, s/n, Dois Irmãos, CEP, Recife-PE, 52171-900, Brazil*

A R T I C L E I N F O

Article history:
Received 17 October 2018
Revised 18 May 2019
Accepted 9 June 2019
Available online 10 June 2019

Keywords:
Cross-domain recommendation
Context-aware recommendation
Collaborative filtering recommendation
Cross-domain context-aware recommendation

A B S T R A C T

In this paper, we address two research topics in Recommender Systems (RSs) which have been developed in parallel without a deeper integration: Cross-Domain RS (CDRS) and Context-Aware RS (CARS). CDRS have emerged to enhance the quality of recommendations in a target domain by leveraging sources of information in different domains. CDRS are especially useful to address cold-start, sparsity and diversity problems in target domains with scarce information. CARS, on its turn, have been proposed to consider contextual information for recommendations. Such systems are suitable when the users' interests change according to factors like time, location, among others. By combining these two approaches, better RSs can be developed, considering both the availability of useful data from multiple domains and the use of contextual information. In this paper, we formalize the combination of CDRS and CARS, which represents a more systematic integration of these approaches compared to previous work. Based on this formulation, we developed novel RSs techniques, named CD-CARS. To evaluate the developed CD-CARS techniques, we performed extensive experimentation through real datasets taking into account several scenarios. The recommendations were evaluated in terms of predictive and ranking performance, respectively achieving up to 62.6% and 45%, depending on the scenario, in comparison to traditional cross-domain collaborative filtering techniques. Therefore, the experimental results have shown that the integration of techniques developed in isolation can be useful in a variety of situations, in which recommendations can be improved by information gathered from different sources and can be refined by considering specific contextual information.

1. Introduction

A large number of Web sites and applications, such as Amazon,[1] Netflix,[2] Youtube,[3] Last.fm,[4] among many others, have adopted recommender systems (RS) (Adomavicius & Tuzhilin, 2005; Park, Kim, Choi, & Kim, 2012; Ricci, Rokach, Shapira, & Kantor, 2015) to provide their users with more relevant items. In the RS area, collaborative filtering (CF) is the most popular and widely implemented approach, since its implementation is relatively easy in different domains, and its quality is generally higher than other approaches, such as content-based filtering (CBF) (Nilashi, Ibrahim, & Bagherifard, 2018; Ricci et al., 2015). However, a common criticism of CF recommenders is that they tend to be biased toward popularity, constraining the degree of diversity (Fernández-Tobías, Cantador, Kaminskas, & Ricci, 2012). Furthermore, CFs are not able to recommend new items for which no ratings are available (a.k.a. *cold-start* problem) resulting in a low user satisfaction (Cantador, Fernández-Tobías, Berkovsky, & Cremonesi, 2015).

In order to minimize these problems, *Cross-Domain* Recommender Systems (CDRS) (Cantador et al., 2015; Cremonesi, Tripodi, & Turrin, 2011; Fernández-Tobías et al., 2012; Gao et al., 2013; Taneja & Arora, 2018; Winoto & Tang, 2008) have been developed to use knowledge or user preferences acquired in a source domain to improve recommendation in a target domain where data is scarce (e.g. using a consolidated database of book preferences to recommend in a new movie recommendation application). Instead of handling each domain independently, CDRS recommend items of a target domain by exploring similarities between users considering ratings from source and target domains (Cremonesi et al., 2011). CDRS commonly lead to a higher user satisfaction by addressing cold-start, sparsity, and diversity problems, although they

* Corresponding author.
E-mail addresses: douglas.veras@ufrpe.br (D. Véras), rbcp@cin.ufpe.br (R. Prudêncio), cagf@cin.ufpe.br (C. Ferraz).

[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 online radio, http://www.last.fm.

https://doi.org/10.1016/j.eswa.2019.06.020
0957-4174/

may not be necessarily more accurate than traditional CF recommender systems (Cantador et al., 2015).

In parallel, *Context-Aware* Recommender Systems (CARS) takes into account the user's context (e.g., location, time, mood, etc.) (Adomavicius & Tuzhilin, 2015; Lee & Park, 2007; Unger, Bar, Shapira, & Rokach, 2016; Viktoratos, Tsadiras, & Bassiliades, 2018; Villegas, Sánchez, Díaz-Cely, & Tamura, 2018; Wang, Li, Zhao, & Chen, 2017; Zarka, Cordier, Egyed-Zsigmond, Lamontagne, & Mille, 2016). In many applications, such as recommending a vacation package or a TV program, it may not be sufficient to consider only users and items ratings. It is also important to incorporate contextual information into the recommendation process in order to recommend relevant items to users under certain circumstances (Adomavicius & Tuzhilin, 2015).

Although developed in isolation, CDRS and CARS have related features that can be exploited to produce even better RSs. For instance, Fernández-Tobías et al. (2012) and Cantador et al. (2015) highlighted that context can be treated as a bridge between different domains. Additionally, one can use contextual information to complement CDRS recommendations (e.g. a movie can be recommended based on book preferences but also considering when the user will watch it). Despite these opportunities, just a few works have considered context-aware techniques in CDRS (Richa & Bedi, 2018; Ji & Shen, 2015; Tekin & van der Schaar, 2015).

According to Fernández-Tobías et al. (2012), and Cantador and Cremonesi (2014), no previous work had addressed CDRS by deploying contextual features in the past. More recently, however, other proposals have been published (Richa & Bedi, 2018; Braunhofer, Kaminskas, & Ricci, 2013; Ji & Shen, 2015; Kaminskas, Fernández-Tobías, Ricci, & Cantador, 2014; Moe & Aung, 2014a; Tekin & van der Schaar, 2015; Zhang, Yuan, & Yu, 2014), adopting diverse approaches, from semantic techniques to supervised learning, for instance. However, the majority of these works relies on a knowledge-based approach (Colombo-Mendoza, Valencia-García, Rodríguez-González, Alor-Hernández, & Samper-Zapater, 2015). We consider that approach as "ad-hoc" because it demands a great amount of knowledge about users and/or items (for each particular domain) and its acquisition is a very difficult process in general (e.g. a knowledge engineer is required) (Felfernig, Friedrich, Jannach, & Zanker, 2015). Our work, in turn, relies on the use of systematic context-aware paradigms (Adomavicius & Tuzhilin, 2015), which have been successfully adopted for CARS and, in general, require little domain knowledge as they are based on user ratings with contextual information (Adomavicius & Tuzhilin, 2015; Villegas et al., 2018).

In this work, we address CF under both the cross-domain and context-awareness recommendation perspectives. We propose a novel framework for recommender systems named as Cross-Domain Context-Aware Recommender System (CD-CARS). In order to verify the usefulness of the proposal, we implemented different versions of CD-CARS algorithms and evaluated them by performing experiments through real datasets. The obtained results have shown that the use of context-aware techniques can be considered a good approach in order to improve the cross-domain recommendation accuracy in comparison to traditional CDRSs. Other contributions of this work are: (1) the formalization of the CD-CARS problem from two relevant research fields, CDRS and CARS; (2) the definition of real datasets for evaluating CD-CARS, taking into account different domains and contextual information. Such datasets are scarce in the literature and we make them available; and (3) a framework useful to generate cross-selling or bundle recommendations for items from multiple domains (e.g. the recommendation of music accompanied by a movie to watch or a book to read).

The remainder of this paper is structured as follows. Section 2 reviews the literature about RSs focusing on cross-domain RS and context-aware RS, respectively. Section 3 describes some related works. Section 4 presents the proposed CD-CARS, including the formalization of the CD-CARS problem, how the contextual information is modeled, the proposed recommendation algorithms, and cross-domain CF-based algorithms adopted in combination with the CD-CARS algorithms. Section 5 presents the results of an experimental evaluation of the implemented CD-CARS, as well as a discussion about the findings of this research. Section 6 draws the conclusions and future works.

2. Recommender systems

In this section, we provide a brief background in Recommender Systems, focusing on important aspects of CDRS (Section 2.1) and CARS (Section 2.2), which are the relevant topics in our work.

2.1. Cross-domain recommender systems

There is a variety of CDRS approaches, which can be distinguished by certain aspects, such as the definition of "domain" itself. Regarding the differences between the attributes of the recommended items, a domain can be defined in four levels (Cantador et al., 2015):

- (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 domains, since they have some attributes in common (title, genre, etc.), but they also have different ones (e.g., airtime, channel, etc.) (Loni, Shi, Larson, & Hanjalic, 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.) (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[5] as well as in Netflix[6] (Pan & Yang, 2013).

Another important aspect is the domain in which items are recommended. The cross-domain recommendation task generally aims to exploit knowledge from a source domain D_S to recommend items in a target domain D_T. In this sense, Cantador et al. (2015) identified three cross-domain recommendation tasks (consider U_S and U_T as the sets of users, while I_S and I_T as the sets of items):

- *Multi-domain recommendation*: items are recommended in both source and target domains by exploiting knowledge from both domains (Carmagnola, Cena, & Gena, 2011), i.e., items are recommended in $I_S \cup I_T$ to users in U_S (or U_T, or even $U_S \cup U_T$);
- *Linked-domain recommendation*: items are recommended only in the target domain by exploiting knowledge from the source and target domains (Moreno, Shapira, Rokach, & Shani, 2012), i.e., items are recommended in I_T to users in U_S by exploiting knowledge about $U_S \cup U_T$ and/or $I_S \cup I_T$;
- *Cross-domain recommendation*: items are recommended only in the target domain by exploiting knowledge only from the

[5] https://movielens.org/.
[6] https://www.netflix.com.

source domain (Tiroshi & Kuflik, 2012), i.e., items are recommended in I_T to users in U_S by exploiting knowledge about U_S and/or I_S.

Despite the task that we aim to perform is classified as a *Linked-domain recommendation*, we refer to the recommender system proposed in this work as a *Cross-domain recommender system*. This is done as a matter of simplicity and is based on the survey of cross-domain RS (Cantador et al., 2015), in which the majority of the papers that perform *Linked-domain* and *Multi-domain* recommendation tasks refer themselves as *Cross-domain* RS.

Regarding the cross-domain recommendation goals, according to Cantador et al. (2015), the most common ones are: alleviating the cold-start problem (Shapira, Rokach, & Freilikhman, 2013); alleviating the new user problem (dos Santos, Marcelino, Bezerra, de Amorim, & do Nascimento, 2012); improving accuracy (Moreno et al., 2012); and increasing diversity (Winoto & Tang, 2008). The aim in this work is how to improve the quality of cross-domain collaborative filtering recommender systems. 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.

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

- *No overlap*, when each item belongs to only one domain, and each user only has preferences for items of one domain (Abel, Araújo, Gao, & Houben, 2011), i.e., $U_S \cap U_T = \oslash$ and $I_S \cap I_T = \oslash$;
- *User overlap*, when some users have preferences for items of, at least, two domains (source and target), but each item belongs to a single domain only (Sahebi & Brusilovsky, 2013), i.e., $U_S \cap U_T \neq \oslash$;
- *Item overlap*, when there are items belonging to distinct domains (source and target) (Cremonesi et al., 2011), i.e., $I_S \cap I_T \neq \oslash$;
- *User and item overlap*, when there is overlap between the users as well as between the items (Tiroshi & Kuflik, 2012), i.e., $U_S \cap U_T \neq \oslash$ and $I_S \cap I_T \neq \oslash$.

As stated in the problem of this work, it is necessary a *User overlap* among source and target domains, whereas an *Item overlap* is not. Besides, the performance of a cross-domain recommender is mainly affected by three parameters (Cantador et al., 2015): overlap between the source and target domains; density of the target domain data; and size of the target user's profile. Thus, it is important to consider the sensitivity of the cross-domain algorithms regarding these three parameters (Hwangbo & Kim, 2017).

2.2. Context-aware recommender systems

Likewise the cross-domain approach, context-aware recommender systems (CARS) is a challenging and emergent field of recommender systems (Adomavicius & Tuzhilin, 2015). In that field, there is not a standard definition of "context". However, some authors (Adomavicius & Tuzhilin, 2015; Palmisano, Tuzhilin, & Gorgoglione, 2008) have a similar point of view about "context" for recommender systems, which is the focus of this work. These authors consider "context" as dimensions (location, time, mood, etc.) and their attributes (country, city, year, day, sadness, happiness, etc.), which can be used to adapt the recommendations. We model contextual information in our CD-CARS based on this definition (Section 4.2).

An important aspect of CARS is how to obtain contextual information (Sundermann, Domingues, da Silva Conrado, & Rezende, 2016). Adomavicius and Tuzhilin (2015) mention three of the most common methods:

- *Explicitly*, when the contextual information is obtained directly from users (Colombo-Mendoza et al., 2015);
- *Implicitly*, when users are not aware of the contextual information gathering process by the CARS (e.g. sensors can be used) (Pham, Jung, & Le Anh Vu, 2014);
- *Inferring*, when 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 (Wang, Li, & Xu, 2015).

Once that the contextual information is obtained is also important to determine the relevance of it (Adomavicius & Tuzhilin, 2015). For that, there are several approaches, from automatic (e.g., by using several existing feature selection methods from machine learning, data mining, statistics, and so on.) (Chatterjee & Hadi, 2015; Liu & Motoda, 2012) to, even, manual ones (e.g., by using domain knowledge of an expert for a given application domain) (Brézillon, 2007). In the same way that some contextual dimensions (e.g., Temporal x Location) can be more relevant than others in a given domain, there may exist contextual attributes (e.g., country x city) more relevant than others, since a contextual dimension can be modelled as a hierarchical tree, for instance Adomavicius and Tuzhilin (2015).

Regarding techniques for CARS, according to Adomavicius and Tuzhilin (2015), there are three systematic paradigms found in the literature:

- *Contextual pre-filtering*, in which information about the current context is used for selecting the relevant set of data (i.e., pre-filtered user ratings). Then, ratings can be predicted using any traditional collaborative-filtering recommender system on the pre-filtered data (Panniello, Tuzhilin, & Gorgoglione, 2014; Panniello, Tuzhilin, Gorgoglione, Palmisano, & Pedone, 2009);
- *Contextual post-filtering*, in which contextual information is initially ignored and the ratings are predicted using any traditional collaborative-filtering recommender system on the entire data, and then the resulting recommendations (or predictions) are adjusted (or filtered) depending on the contextual information of the users (Panniello et al., 2014; Panniello et al., 2009);
- *Contextual modeling*, in which the contextual information is used directly in the recommendation or predictive process (neither before or after it) (Kim & Yoon, 2014). 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 (Oku, Nakajima, Miyazaki, & Uemura, 2006), matrix (or tensor) factorization (Kim & Yoon, 2014), heuristic calculations (Adomavicius, Sankaranarayanan, Sen, & Tuzhilin, 2005), among others.

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. In Véras, Prota, Bispo, Prudêncio, and Ferraz (2015), we describe some of these ad-hoc approaches: *Contextual rules*, in which there are all kinds of rules that allow recommender systems to sense and to react based on their context (Moon, Kim, Lee, & Kim, 2006) (e.g. "event-condition-action" rules, "Key-value" rules, etc.); *Contextual Ontology*, which are not algorithms but are crucial to other knowledge-based context-awareness techniques (Moe & Aung, 2014b); *Similarity-based*, in which algorithms compare contexts in order to recommend items (Alhamid et al., 2015) (for instance, the context can be represented in several ways such as tags, key-value pairs, among others); and *Supervised learning*, in which a set of labeled examples is produced, where each example is composed of features extracted from contextual attributes (Vildjiounaite, Kyllönen, Hannula, & Alahuhta, 2009) (e.g. time of the day, mood, etc.).

3. Related works

In this section, we describe related cross-domain recommender systems that use contextual information, highlighting their limitations in comparison to proposed CD-CARS according to the concepts previously described.

Braunhofer et al. (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 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 a recommendation with a certain degree of diversity. That work is domain-specific and does not take 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.

In Moe and 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 on ontological properties to manage personalization systematically. Ford-Fulkerson algorithm (Parameswaran, Venetis, & Garcia-Molina, 2011) was applied to build the bridge of the semantic concepts between the source and target domains. Finally, a technique for making recommendations according to the users' contexts was adopted, called TOPSIS (Jadidi, Firouzi, & Bagliery, 2010). The accuracy of the developed system was measured by means of ranking metrics such as Precision, Recall, and F-measure in a simple dataset without user overlap and with information about cosmetics and skin care problems. The work presented in Moe and 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.).

By extending the work proposed in Braunhofer et al. (2013) and 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[7] without user overlap, the authors evaluated the framework for recommending music (target domain) related to POIs (source domain) according to location and time as contextual information explicitly defined by the users. Similarly to Braunhofer et al. (2013), the authors evaluated the knowledge-based framework by means of empirical experimentation with some users, and again the same recommendation may be made for different users located in that POI.

Tekin and van der Schaar (2015) proposed a content aggregation algorithm, called DIStributed COntent Matching (DISCOM), capable of learning which content to gather and performing matching between it and users' preferences, by exploiting similarities between types of users. In that system, each user is represented together with its context, which is considered as the user type. Hence, the 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 others. In addition, it may also be represented by the device type that the user is using (e.g., computer, mobile phone). The authors adopted two datasets without user or item overlap for evaluation purposes: one on the TV domain and another for recommending 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 and Shen (2015) proposed an improved group-aware CF-based algorithm[8] that 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, 2009). 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 predictive measure (MAE). The same limitation mentioned for Tekin and van der Schaar (2015) can be verified on that work, i.e., recommendations are not personalized to a single user.

Hsieh, Yang, Wei, Naaman, and Estrin (2016) proposed a user-centric recommendation model that incorporates cross-platform and diverse personal digital traces in order to improve the recommendations. Their context-aware topic modeling algorithm, systematically, profiles users' interests based on their traces from different contexts with user overlap. They evaluated the model in the news and event domains through offline experiments by leveraging users' public Twitter traces. Besides, the authors also conducted a small study with 33 participants using Twitter, Facebook, and email traces. The main contribution of that work relies on the profiling of the users' digital traces from different contexts. In that work, the platform in which the users rate an item is considered as context (e.g. email, Twitter, Facebook, among others). In this way, that context is only the single one treated by the authors (ignoring other interesting contextual dimensions such as temporal, location, mood, etc.).

Richa and Bedi (2018) have proposed a cross-domain RS which introduced a parallel approach by using a general-purpose GPU (Graphics Processing Unit) in order to improve the performance in terms of processing speed. The authors have developed a prototype of the RS on four domains (restaurant, tourist places, shopping places, and hotels). Besides, the proposed RS stores user preferences and contextual information (user's location and device) for all of those domains (with user overlap) and recommends items for anyone of them (Multi-Domain). Although that work is similar to ours, its main limitation relies on the recommendation algorithm adopted, which is a single-domain CF-based

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

[8] The authors claim that their work is a context-aware RS by considering that a group can be viewed as a user type (context).

Table 1
Main limitations of context-aware-based related works in comparison to proposed CD-CARS.

Paper	Systematic approach	Accuracy goal	Linked-domain task	User overlap	Neighborhood-based CF
Braunhofer et al. (2013)					
Moe and Aung (2013), Moe and Aung (2014b), Moe and Aung (2014a)		X			
Kaminskas et al. (2014)					
Tekin and van der Schaar (2015)	X				X
Ji and Shen (2015)	X	X	X	X	
Hsieh et al. (2016)				X	
Richa and Bedi (2018)	X	X		X	X
CD-CARS	X	X	X	X	X

collaborative filtering (Neighborhood-based) (Ricci et al., 2015). As we describe in Section 4.4.1, the majority of the traditional single-domain CF-based algorithms can also be used for the cross-domain recommendation, however, we can achieve better results with cross-domain CF-based algorithms (Cremonesi et al., 2011).

In summary, the majority of the cross-domain RS described and categorized above relies on ad-hoc approaches of CARS (e.g. knowledge-based), which may be difficult to customize to new situations, since they are usually designed for a specific domain and do not take into account context obtained from user ratings (Adomavicius & Tuzhilin, 2015). The proposed CD-CARS, in turn, relies on the use of systematic context-aware techniques. 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). The systematic approach, unlike those works, has allowed us to formalize the cross-domain recommendation problem under the context-aware perspective.

Table 1 summarizes the main features of the related works mentioned above in comparison to proposed CD-CARS. As mentioned in Section 2.1, CD-CARS performs a "Linked-domain" task, but we consider it as "Cross-domain" because that is a common practice in the CDRS literature.

Notice that the works proposed in Ji and Shen (2015) and Richa and Bedi (2018) are the most similar to ours, because they adopt systematic approaches, have the accuracy goal and take user overlap into account. However, Ji and Shen (2015) is not based on the *Neighborhood-based* approach of collaborative filtering algorithms, and it is intended for making recommendations for a group of users instead of recommending items to a single user. Richa and Bedi (2018) has the limitation of using a single-domain CF-based algorithm adapted to cross-domain purposes, and it is intended for multi-domain recommendation besides not formalizing the cross-domain context-aware recommendation task.

4. CD-CARS

The proposed CD-CARS recommends items from target domain by exploring the similarities between users considering ratings, and also their contexts, from source and target domains, as illustrated in Fig. 1. 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.

In this section, we formalize the cross-domain context-aware recommendation problem (Section 4.1) and model the contextual information (Section 4.2). In Section 4.3, we describe the proposed CD-CARS algorithms while in Section 4.4 we present the base cross-domain algorithms adopted in combination with the proposed CD-CARS.

4.1. CD-CARS problem formalization

We initially address the CD-CARS under CF similarly to Adomavicius and Tuzhilin (2015), by considering user ratings as a function of three dimensions:

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

User ratings can be stored in a user-rating-context tensor for each item domain (e.g. books, movies, music, among others). Notice that the notion of domain adopted in this work is based on the "Item level" definition (described in Section 2.1). This way, movies and books, for example, are considered as belonging to different domains. Formally, consider the following definitions for a set of 'n' source domains ($S_1, S_2, \ldots, S_n$) and just one target domain (T):

- $U_{S_1}, U_{S_2}, \ldots, U_{S_n}, U_T$: sets of users for each domain;
- $I_{S_1}, I_{S_2}, \ldots, I_{S_n}, I_T$: 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) for each domain;
- $U_{S,T} = (U_{S_1} \cup U_{S_2} \cup \ldots \cup U_{S_n}) \cap U_T \neq \oslash$: at least one user must have preferences for items in the target domain and, at least, a source domain (user overlap);
- $I_{S,T} = I_{S_1} \cap I_{S_2} \cap \ldots \cap I_{S_n} \cap I_T = \oslash$: there is no item overlap between domains;
- $C_{S,T} = C_{S_1} = C_{S_2} = \ldots = C_{S_n} = C_T \neq \oslash$: the same set of possible contexts is observed for user ratings in all domains (contextual overlap).

Hence, the problem to be solved in this work is how 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 $i = 1, 2, \ldots, n$ 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 representation 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 (dos Santos et al., 2012). For a matter of simplicity, in this work, we consider that all user-ratings are on a five-star scale.

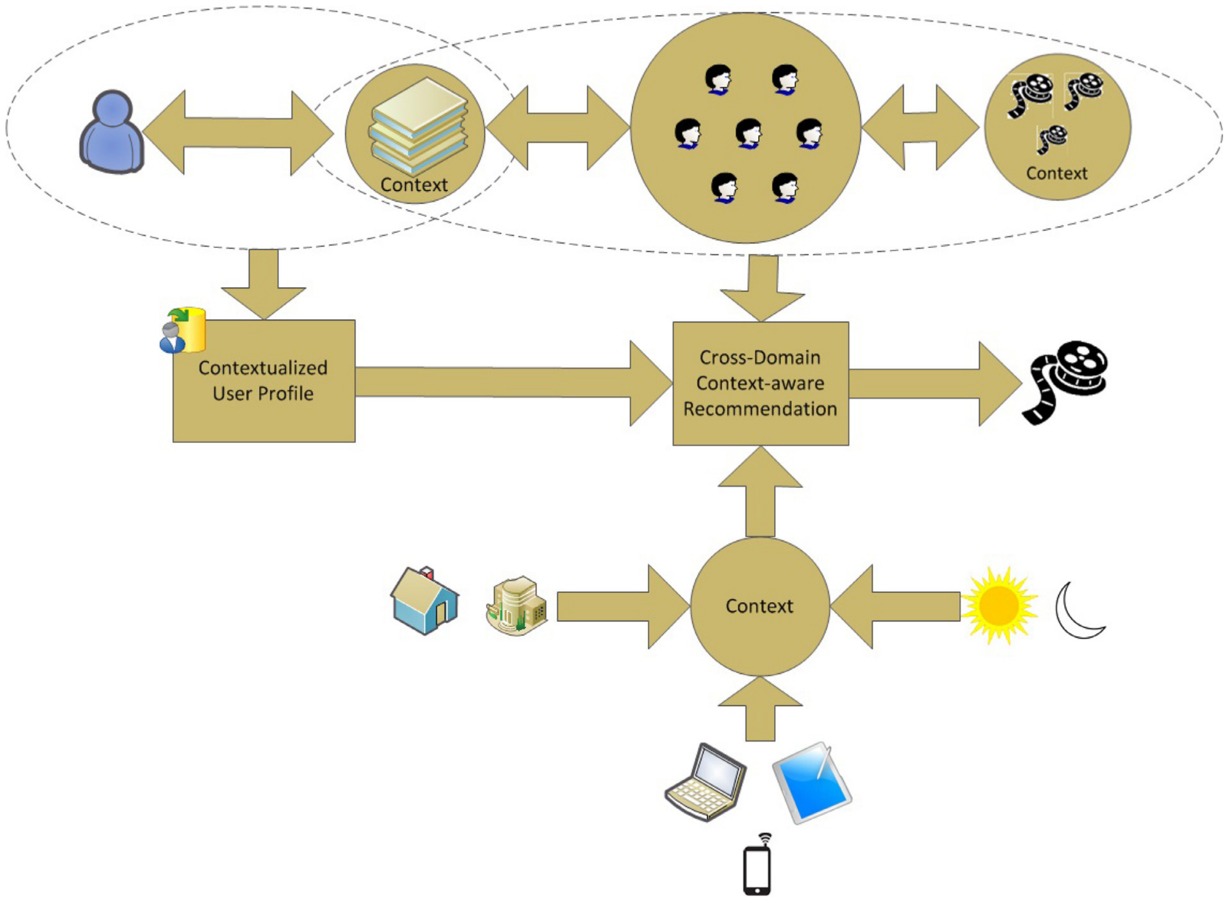

Fig. 1. Cross-domain context-aware recommendation.

4.2. Contextual features formalization

According to the CD-CARS problem formalization described before, we modeled a set of contextual features, for each domain $(C_{S_1}, C_{S_2}, \ldots, C_{S_n}, C_T)$, as a Cartesian product of k contextual dimensions: $C_d = D_1 \times D_2 \times \ldots \times D_k$ (where $d = S_1, S_2, \ldots, S_n, T$ are domains) (Adomavicius et al., 2005). Each dimension D_j (j = 1,2,...,k) can be represented by l contextual attributes $(A_1, A_2, \ldots, A_l)$. 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.

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): $A_1 = \{v_1 = Unknown, v_2 = Aberdeen, \ldots, v_{2839} = Zurich\}$ with 2839 possible values ($m = 2839$), $A_2 = \{v_1 = Unknown, v_2 = Alabama, \ldots, v_{381} = Wisconsin\}$ with 381 possible values ($m = 381$), $A_3 = \{v_1 = Unknown, v_2 = Australia, \ldots\ v_{113} = Zambia\}$ with 113 possible values ($m = 113$);
- Companion dimension (D_3): $A_1 = \{v_1 = Unknown, v_2 = Alone, v_3 = Accompanied, v_4 = Family, v_5 = Friends, v_6 = Partner, v_7 = Colleague\}$ with seven possible values ($m = 7$).

Thus, the contextual information of a rating (c') can be represented as a tuple of w values from different contextual attributes and/or dimensions, i.e., a possible context of a set of contextual features can be denoted as $c' = (v_1, v_2, \ldots, v_w)$, where each value v_s (s=1,2,...,w) belongs to a different contextual attribute $A_z(z = 1, 2, \ldots, l)$ and/or dimension $D_j(j = 1, 2, \ldots, k)$.

It is important to mention that the proposed contextual feature modeling is based on the "Key-Value" model (Véras, Prudencio, Ferraz, Bispo, & Prota, 2015). In this case, the matching between the context of the recommendation (c), which is called "contextual criteria", and the contextual information (c'), represented by this model in the user-ratings, is made in a linear way, as described in Algorithm 1. 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., the context of the recommendation). The contextual criteria is also represented as a tuple of w contextual values with the same contextual attributes and dimensions than 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,

Algorithm 1 Matching between the contextual criteria and the contextual information from ratings.

Input: c, c' (where c is the contextual criteria array of w contextual values, and c' is the contextual information array of w contextual values).
Output: isMatched (a boolean value determining if there is a match between the contextual criteria and the contextual information).

procedure CONTEXTUALMATCHING(c, c')
  for v=1 to w **do**
    if $c[v] \neq$ "*Unknown*" *and* $c[v] \neq c'[v]$ **then return** false
    end if
  end for
return true
end procedure

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 only the values different from "Unknown" on the contextual criteria. This mechanism is used for the proposed CD-CARS algorithms.

Taking into account the described contextual information model and the formalization described in Section 4.1, we can define a tensor ($CP(u, c, g)$) representing the set of item category preferences (e.g. comedy, action, religion) g from users (u) according the context of the recommendation (c). In this way, $\hat{CP}(u, c, g_i)$ is the number of "good" rated items for an item category g_i observed in a context c for a user u. The definition of a "good" item is made according to an α value, which can vary depending on 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 (on a five-star scale). In this way, we define that a "good" item must have, at least, a rating "four" on a five-star scale, i.e., $\alpha = 4$.

Thus, considering that initially $\hat{CP}(u, c, g_i) = 0$, and $j = S_1, S_2, \ldots, S_n, T$, we can build $CP(u, c, g)$ from the contextual user-rating tensors of the source and target domains ($CR_{S_1}, CR_{S_2}, \ldots, CR_{S_n}, CR_T$) as follows.

$$\hat{CP}(u, c, g_i) = \left\{\hat{CP}(u, c, g_i) + 1, \text{ if } \hat{CR}_j(u, i, c) >= \alpha \right. \tag{1}$$

However, CP tensor could not represent the most preferred item categories for a user u in a context c. Suppose that u has eleven "good" rated items of two categories (e.g. just one for "comedy" items and ten for "religion" items) both in c, i.e., $\hat{CP}(u, c, g_{i'}) = 1$ and $\hat{CP}(u, c, g_{i''}) = 10$. Thus, we can say that "religion" items are the most preferred for u in c.

Now, suppose that a user has 104 "good" rated items (thirty religion items, twenty-five educational items, twenty comedy items, nineteen romance items, and ten action items) in the same context c. The most preferred category may be "religion", but there are other important categories that could be considered as "most" preferred too. Thus, in order to determine which are the most preferred item categories for each user/context, we define a simple algorithm (Algorithm 2) based on a θ parameter, which can be adjusted through experiments depending on the adopted dataset.

As we can see in Algorithm 2, θ parameter is set to "2/3" of the number of the most preferred category. Taking into account the example mentioned before (a user with 104 "good" rated items, $\theta = 2/3 * 30 = 20$), only religion (30 occurrences), comedy (20 occurrences), and educational items (25 occurrences) would be considered the most preferred categories. Thus, items of other categories could be ignored in the recommendation (in that example, romance and action items, respectively with 19 and

Algorithm 2 Checking if the item's category is one of the most preferred by the user in the context of the recommendation.

Input: u, i, c
Output: isMostPreferredCategory (a boolean value determining if the item's category is one of the most preferred).

procedure ISMOSTPREFERREDCATEGORY(u, i, c)
  $m \leftarrow 0$
  for v=1 to n **do**
    if $\hat{CP}(u, c, g_{[v]}) > m$ **then**
      $m \leftarrow \hat{CP}(u, c, g_{[v]})$
    end if
  end for
  $\theta \leftarrow (2/3) * m$
  if $\hat{CP}(u, c, g_i) >= \theta$ **then return** true
  end ifreturn false
end procedure

10 occurrences). Note that the higher the θ value, the less the number of categories included in the users' preferred categories.

4.3. CD-CARS algorithms

The algorithms proposed in our work rely on the use of a base cross-domain recommender system (described in Section 4.4), 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:

$$\hat{R}(u, i) = CD(u, i, R_{S_1}, R_{S_2}, \ldots, R_{S_n}, R_T), i \in I_T \tag{2}$$

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 our CD-CARS problem, we consider contextual user-rating tensors and we need a function ($\mathcal{F}$) to make rating predictions of items ($i \in I_T$) 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 Eq. (3).

$$\hat{CR}(u, i, c) = \mathcal{F}(u, i, c, CR_{S_1}, CR_{S_2}, \ldots, CR_{S_n}, CR_T) \tag{3}$$

The function ($\mathcal{F}$) can be implemented using any of the proposed CD-CARS algorithms described in the following. We designed the algorithms according to different context-aware RS paradigms (see Section 2.2): *Pre-filtering* (PreF) and *Post-filtering* (PostF). These paradigms are usually adopted in single-domain RS, but we extended their logic for the cross-domain recommendation task by taking into account the contextual user-rating tensors from different domains.

4.3.1. 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 (R^c_T). On the other hand, the contextual user-rating tensors from the source domains ($CR_{S_1}, CR_{S_2}, \ldots, CR_{S_n}$) are collapsed into a two-dimension (2D) user-rating matrix ($R^c_{S_1}, R^c_{S_2}, \ldots, R^c_{S_n}$) by aggregating ratings for the same user-item pair in different contexts, prioritizing the user-ratings with contextual information (c') matching the context of the recommendation (c). Then, the base cross-domain algorithm is applied to these matrices to produce the predicted ratings ($\hat{CR}(u, i, c)$). Fig. 2 illustrates the PreF technique, which is formalized as follows.

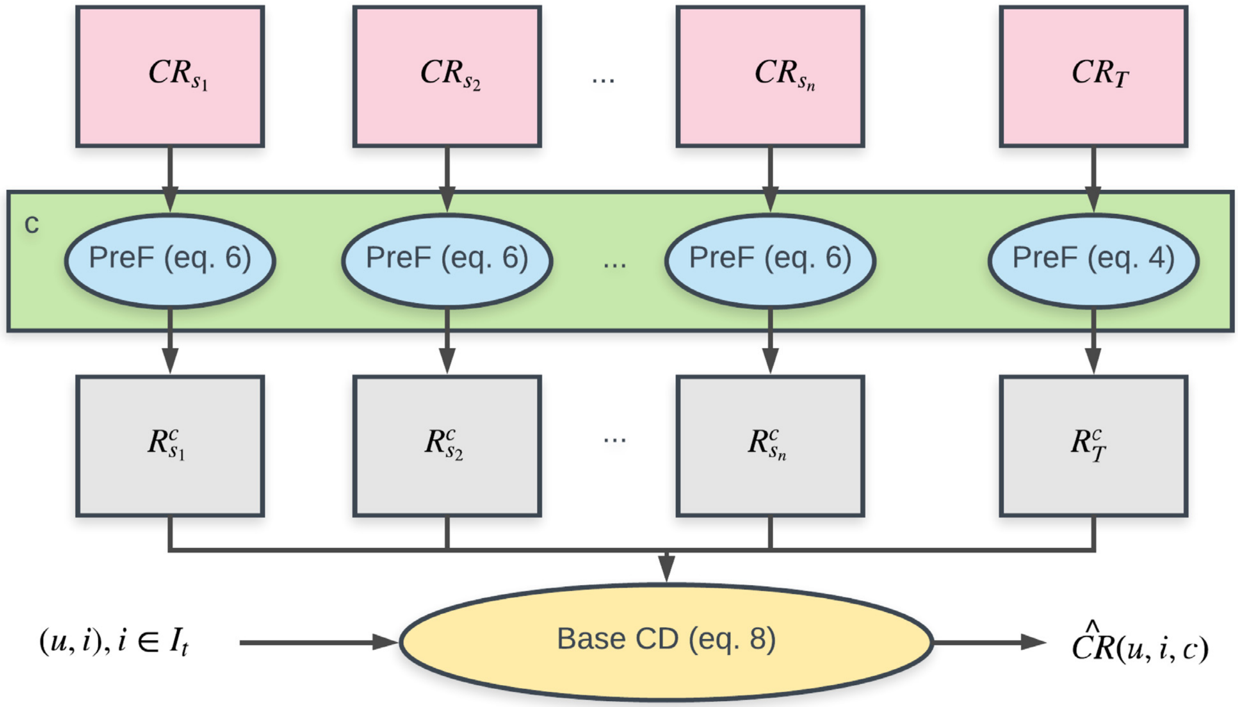

Fig. 2. The pre-filtering cross-domain recommendation is made by filtering the target contextual user-rating tensor for a given context.

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

$$R^c_T(u, i) = CR_T(u, i, c) \tag{4}$$

The context-filtered matrix only has ratings according to:

$$\hat{R^c_T}(u, i) = \begin{cases} \hat{CR}_T(u, i, c'), & \text{if } contextualMatching(c, c') \text{ (see Algorithm 1)} \\ not\ available, & \text{otherwise} \end{cases} \tag{5}$$

- Step (2): Define the 2D aggregated matrices, prioritizing the user-ratings with contextual information (c') matching the recommendation (c), for the source domains:

$$R^c_j(u, i) = CR_j(u, i, c), \tag{6}$$

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

$$\hat{R^c_j}(u, i) = \begin{cases} \hat{CR}_j(u, i, c'), & \text{if } contextualMatching(c, c') \text{ (see Algorithm 1)} \\ \frac{\sum_{c' \in C_j} CR_j(u, i, c')}{|C_j|}, & \text{otherwise} \end{cases} \tag{7}$$

- Step (3): Apply the base cross-domain technique (Eq. (2)) using the reduced matrices:

$$\hat{CR}(u, i, c) = CD(u, i, R^c_{S_1}, R^c_{S_2}, \ldots, R^c_{S_n}, R^c_T), i \in I_T \tag{8}$$

4.3.2. Cross-domain PostF algorithm

PostF algorithm initially collapses the contextual user-rating tensors from the source and target domains ($CR_{S_1}, CR_{S_2}, \ldots, CR_{S_n}, CR_T$) into two-dimension (2D) user-rating matrices ($R_{S_1}, R_{S_2}, \ldots, R_{S_n}, R_T$) by aggregating ratings for the same user-item pair in different contexts without considering the context of the recommendation. The base cross-domain is then applied using as input the aggregated rating matrices. Finally, the context of the recommendation (c) 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 (g) by the user (u) in context c. Fig. 3 illustrates this algorithm, which is formalized in the following.

- Step (1): Define the 2D aggregated matrices for the source and target domains:

$$R_j(u, i) = CR_j(u, i, c), \tag{9}$$

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

$$R_j(u, i) = \left\{ \frac{\sum_{c' \in C_j} CR_j(u, i, c')}{|C_j|} \right. \tag{10}$$

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

$$\hat{R}(u, i) = CD(u, i, R_{S_1}, R_{S_2}, \ldots, R_{S_n}, R_T), i \in I_T \tag{11}$$

- Step (3): a rating produced for an item ($\hat{R}(u, i)$) in Step (2) is maintained if the item's category is preferred by the user in the context of the recommendation. Otherwise, that rating is discarded:

$$\hat{CR}(u, i, c) = \begin{cases} \hat{R}(u, i), & \text{if } isMostPreferredCategory(u, i, c) \text{(see Algorithm 1)} \\ not\ available, & \text{otherwise} \end{cases} \tag{12}$$

4.4. Base cross-domain algorithms

In this work, we propose the adoption of single-domain and cross-domain algorithms as a base of the proposed PreF and PostF algorithms. Section 4.4.1 describes the single-domain CF-based algorithms, whereas Section 4.4.2 describes the cross-domain CF-based ones.

4.4.1. Single-domain as cross-domain algorithms

As stated by Cremonesi et al. (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

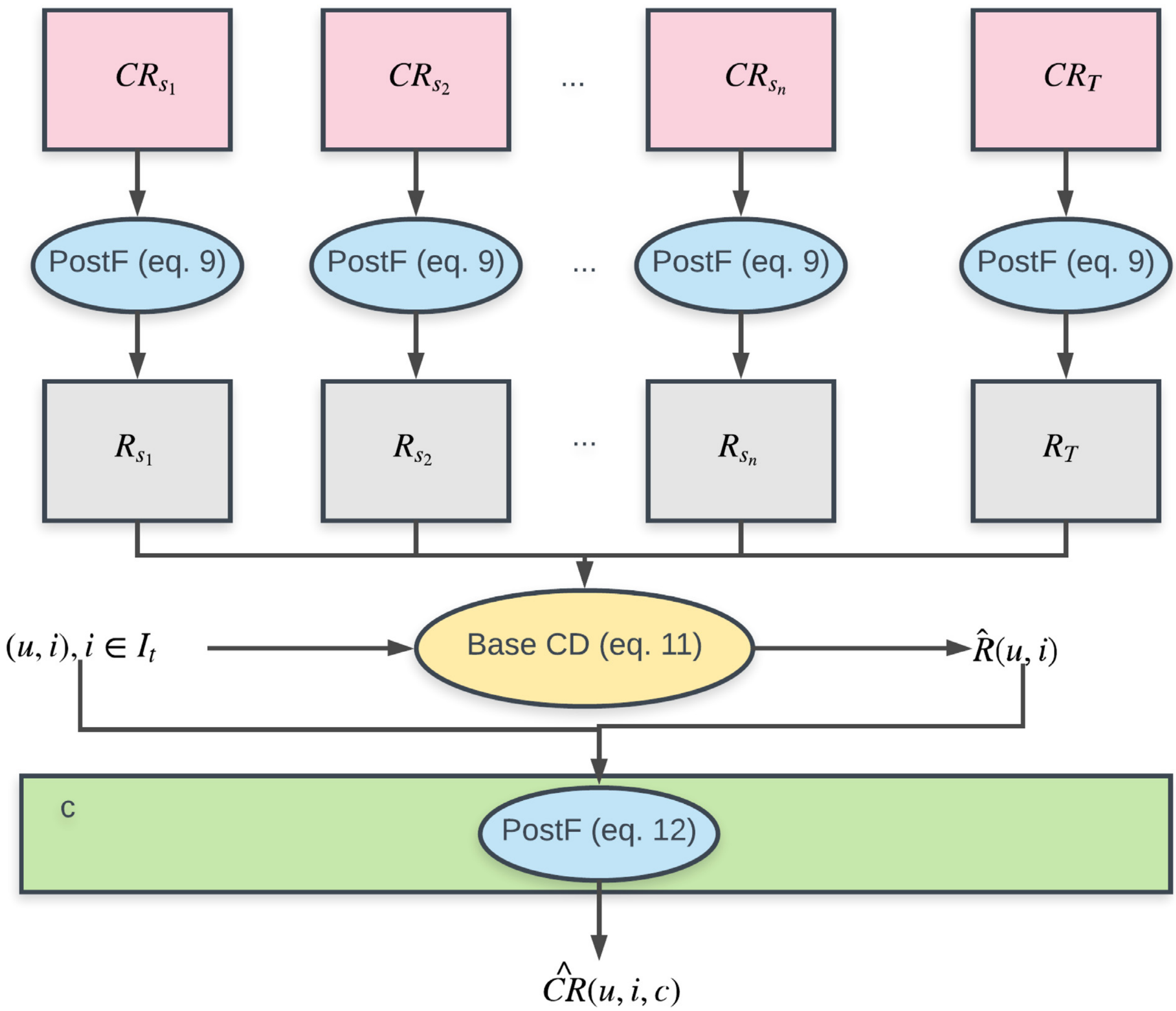

Fig. 3. The cross-domain recommendation is made over the aggregated user-rating matrices and then post-filtered according to contextual user preferences.

these matrices are normalized (dos Santos et al., 2012). Thus, these algorithms can also 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 4.1). Therefore, those CF-based algorithms can be used as a base cross-domain algorithm together with proposed CD-CARS algorithms. In this way, we can apply single-domain collaborative filtering algorithms as a cross-domain (CD) technique in Eqs. (8) and (11), for PreF and PostF algorithms, respectively.

Neighborhood-based algorithms (Koohi & Kiani, 2017; Ricci et al., 2015) 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. For example, the *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, recommendations are derived from a neighborhood of the n most similar users. The optimal value of n can be defined through experiments.

A crucial aspect of these algorithms is the similarity computation for items or users. The similarity in user-based and item-based CF algorithms can be computed by means of traditional similarity metrics, such as Euclidean distance, Cosine similarity, Pearson correlation, among others (Ricci et al., 2015). The recommendation of these single-domain neighborhood-based algorithms, used for cross-domain purposes, is made without modifications (Cremonesi et al., 2011), except by the fact that only items from the target domain are recommended.

4.4.2. 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 algorithm which is originally intended to perform cross-domain recommendations. Thus, we can directly apply it as a base cross-domain algorithm in combination with proposed CD-CARS algorithms.

The cross-domain algorithm adopted is proposed by Cremonesi et al. (2011), named *NNUserNgbr-transClosure*, and is a neighborhood-based CF one chosen due to its simplicity. It enhances the *NNUserNgbr* algorithm (described in Section 4.4.1), 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). 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 et al. (2011), this transitive closure procedure is described as follows. Given a binary relation **S**, where s_{ij} is equal to either 1 or 0, the algebraic *transitive closure* of **S** is the union of successive powers of the original matrix, i.e.:

$$\mathbf{S}_{trans} = \bigcup_{n \in \mathbb{N}} \mathbf{S}^n \tag{13}$$

Matrix **S** is represented by the weighted connections between items of a set. However, since this matrix does not represent a binary relation, Eq. (13) has been adapted as follows. The "union" operator $\bigcup$, which is defined for binary relations, has been replaced by the "maximum" operator, $\mathbf{Z} = \max(\mathbf{X}, \mathbf{Y})$, where the maximum matrix **Z** between two similarity matrices **X** and **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 et al. (2011) have limited the transitive closure to only two steps. Experiments showed that a transitive closure with more than two steps did not provide any notable improvement in the recommendation accuracy while increasing computational requirements. Thus, the enhanced item-to-item similarity matrix is computed as:

$$\mathbf{S}^* = max(\mathbf{S}, \mathbf{S}^2) \tag{14}$$

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.

5. CD-CARS evaluation

This section presents the experimental evaluation of the proposed CD-CARS algorithms in comparison to cross-domain CF-based ones. For that, we describe two datasets adopted in the experiments taking into account three contextual dimensions and three different target domains (Section 5.1). In Sections 5.2 and 5.3, we describe, respectively, how the contextual information is obtained in those datasets and how its relevance is verified. Section 5.4 describes the evaluation methodology adopted and the algorithms' settings. Section 5.5 presents the obtained results for each dataset used in the experiments as well as a discussion about our findings.

5.1. Cross-domain datasets description

In this section, we describe the properties of two datasets[9] adopted for distinct objectives: *i)* for evaluating the CD-CARS in two more related domains (*Book* and *Television*, named as "book-television dataset", described in Section 5.1.1); and *ii)* for evaluation considering two less related domains[10] (*Book* and *Music*, named as "book-music dataset", described in Section 5.1.2). Both datasets were built from the Amazon dataset provided by Leskovec, Adamic, and Huberman (2007). This dataset contains metadata and users' reviews for different types of products (Books, music CDs, DVDs, and so on),[11] from which we extracted sets of users, items, ratings (on a five-star scale) and contextual information (rating date, location, and companion) for the three domains: Book, Music and Television.

It is important to mention that the contextual information was not available directly in the Amazon dataset. So, we had to obtain the contextual information implicitly and by inferring. We summarize that acquisition process in Section 5.2. Finally, we describe the adopted method for selecting relevant contextual information in Section 5.3.

5.1.1. Book-television dataset

We summarize the properties of the "book-television dataset" with a full overlap of users, as below:

- Cross-domain (both domains): 15341 users, 194,615 items, 1,249,949 ratings, 81.47 ratings per user, and 6.42 ratings per item.
- Books (single-domain): 15341 users, 165,896 items, 805,102 ratings, 52.48 ratings per user, and 4.85 ratings per item.
- Television (single-domain): 15341 users, 28,719 items, 444,847 ratings, 28.99 ratings per user, and 15.48 ratings per item.

In this dataset, we considered three different kinds (dimensions) of contextual information as presented in Section 4.2: *Temporal, Location*, and *Companion*. Regarding *Temporal* context, this information is present in all ratings in that dataset. The *Location* context, in turn, is present in 557,018 ratings (approximately 45% of the total of ratings). Finally, *Companion* context was observed in almost 20% of the total of ratings. As it can be seen, the *Books* domain has more ratings ($\simeq$64% from total) than *Television* domain ($\simeq$36% from total). It is important to say that samples of that dataset are used with other user overlap levels (10% and 50%) in order to perform sensitivity and cold-start evaluations (see Section 5.4.4). Besides, we can also verify the sparsity issue by considering the number of ratings per item in the *Books* domain.

5.1.2. Book-music dataset

We summarize the properties of the "book-music dataset" with a full overlap of users, below:

- Cross-domain (both domains): 13189 users, 219,034 items, 1,031,386 ratings, 78.20 ratings per user, and 4.71 ratings per item.
- Books (single-domain): 13189 users, 162,449 items, 742,844 ratings, 56.32 ratings per user, and 4.57 ratings per item.
- Music (single-domain): 13189 users, 56,585 items, 288,542 ratings, 21.88 ratings per user, and 5.10 ratings per item.

In this dataset, all ratings have information about *Temporal* context, whereas almost 46% and 11% of them, respectively, have information about *Location* and *Companion* contexts. We can see that the *Books* domain has more ratings ($\simeq$72% from total) than *Music* domain ($\simeq$28% from total).

Notice that, as mentioned in Section 5.1.1, samples of the dataset also are used with other user overlap levels in order to perform a sensitivity evaluation (see Section 5.4.4). In addition, these overlap levels and the ratings per item of both domains (*Books* and *Music*) can be used to evaluate the cold-start and sparsity problems, respectively.

5.2. Obtaining contextual information

In this section, we describe how the contextual information used in CD-CARS was obtained for three different contextual dimensions (*Temporal, Location*, and *Companion*).

5.2.1. 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 paper, 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

[9] Available at https://github.com/douglasveras/cd-cars-datasets.

[10] We consider this relation among distinct domains according to the set of item genres of them. The more the domains have item genres in common the more related they are considered.

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

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".

5.2.2. Location dimension

All user-ratings in our datasets have information about the user IDs (and the real Amazon user IDs). 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. Since we only obtained static address information (country, state, and city), we could not extract contextual attributes related to the abstract locations, e.g. "place" attribute (at home, at work, in a movie theatre, etc.). Therefore, only contextual attributes related to geographical location could be extracted.

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. That static context could represent the users' "origin" and a dynamic context, such as the users' current location, could be more relevant. Both types of context can be used in CD-CARS according to its contextual modelling. 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.

5.2.3. Companion dimension

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 and Tuzhilin (2014) with some adaptations by considering different domains. This method is shortly described in the following. Initially, we separated reviews into specific and generic ones by using the measures proposed in the original method (*LogSentences, LogWords, VRatio*, etc.) (Bauman & Tuzhilin, 2014).

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 and Tuzhilin (2014). However, we applied the clustering separately for each domain (*Book, Television*, and *Music*). As a result, the vast majority of the reviews (99.8%) were classified as "specific" for all domains in our datasets. 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 the 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 and Tuzhilin (2014), since these methods rely on the separation of specific and generic reviews. So, we did not use the generic reviews in those methods, unlike they are originally. 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). In this way, we identified six contextual values (alone, accompanied,[12] family, friends, partner, and colleagues[13]) for only one contextual attribute of "companion", from the word groups and topics selected.

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 and 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, 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 2 reports the results of this empirical evaluation considering the different domains and contextual values.

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".

5.3. Selecting relevant contextual information

If we consider that only relevant dimensions (*Location, Temporal* and *companion*) are present in the datasets, we could determine the relevance of the contextual attributes of these dimensions. For that, we let each user-rating in the datasets with contextual information about all contextual dimensions and their attribute variations: *Temporal* dimension with "day" (with values: Sunday to Saturday) and "day type" (values: weekend and weekday) attributes; *Location* dimension with "country", "state" and "city" attributes; and Companion dimension with a single attribute, "companion type", as described in the previous section.

Given these contextual dimensions and their attributes, we applied a data mining method, *InfoGainAttributeEval* from Weka (Hall et al., 2009), to select only the most relevant contextual attributes of each contextual dimension. That method 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). Thus, we applied the *InfoGainAttributeEval*[14] with the user-rating value as a class (five possible values: 1–5) and six attributes (day, day type, country, state, city, companion type) for the two datasets used in this paper, considering all target domains separately. Tables 3 and 4 report the information gain values[15] of the contextual attributes in different target domains for these datasets.

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,

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

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

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

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

Table 2
Classification accuracy of the companion extraction.

Target domain	Overall accuracy	Contextual values					
		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%

Table 3
Information gain of contextual attributes in different target domains for the book-television dataset.

Target domain	Temporal dimension		Location dimension			Companion dimension
	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 4
Information gain of contextual attributes in different target domains for the book-music dataset.

Target domain	Temporal dimension		Location dimension			Companion dimension
	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**

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

5.4. Evaluation methodology

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 et al. (2014). Analogously to that work, we performed different evaluation tasks (prediction and ranking) to verify the accuracy of CD-CARS by using contextual information from distinct dimensions, including scenarios with cold-start and sparsity issues. 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.

5.4.1. 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 5.1 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 4.4.1, 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, since *Book* is a common domain to both datasets. We set the 'n' parameter of the *NNUserNgbr* algorithm to "475" and selected the *Euclidean distance* as the distance metric for it. The same configuration and distance 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 *NNUserNgbr-transClosure* ones, therefore, 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 on a five-star scale) were considered in computation of the frequency of the users' preferred categories (see Section 4.2).

5.4.2. Predictive performance

We measured the predictive performance of the algorithms by using the Mean Average Error (MAE) metric (Ricci et al., 2015). MAE is a measure of the deviation of recommendations from their actual user-rating values. It evaluates 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. The process of splitting training and test sets, considering the target domain and context under test, 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*, illustrated in Fig. 4.

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.

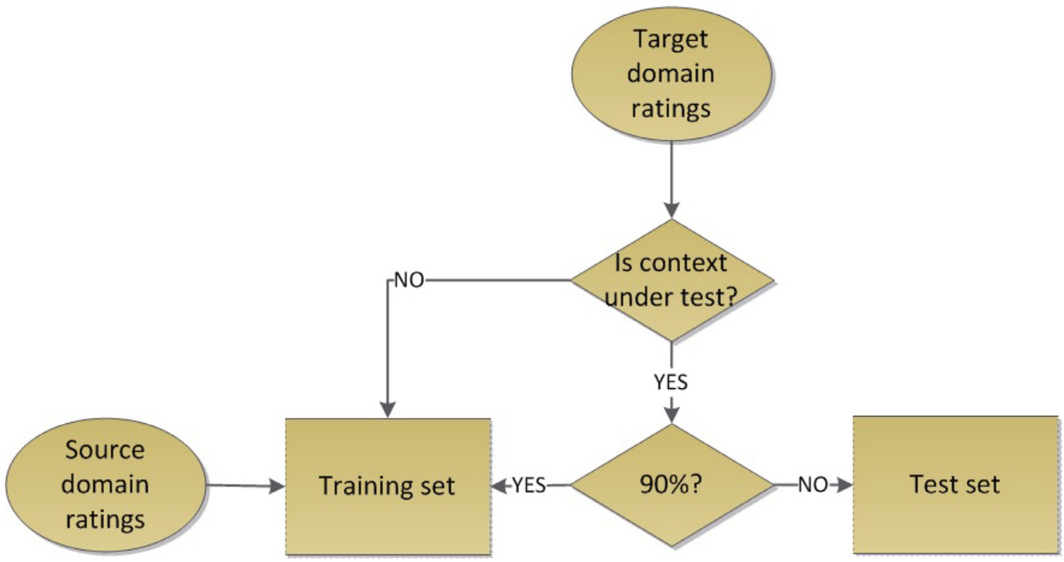

Fig. 4. Splitting training and test sets considering the target domain and context under test.

5.4.3. Ranking performance

Regarding the ranking performance of the algorithms, we adopted the F-metric, 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, and Turrin (2010).

Analogously to Cremonesi et al. (2010), we randomly extracted, approximately, 1.4% of the ratings ($\tilde{1}8,000$) 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.[16] In addition, 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 (Cremonesi et al., 2010).

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 that is made in the predictive evaluation, as illustrated in Fig. 4. 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 ranking performance is partitioned as *Hold-out*.

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[17] from the target domain randomly chosen from the ones unrated by the user ‘u’; and
- In decreasing order, we sort the list of 101 items[18] 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 et al. (2010), are defined as:

$$Recall(N) = \frac{\#hits}{|test\ set|} \tag{15}$$

$$Precision(N) = \frac{\#hits}{N * |test\ set|} \tag{16}$$

$$F\text{-}metric(N) = \frac{2 * Recall(N) * Precision(N)}{Recall(N) + Precision(N)} \tag{17}$$

Likewise the predictive 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 et al. (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 ranking performances through the F-metric curves by varying the number of top ‘N’ items (from one to twenty[19]). Besides, given that most of the online recommender systems usually recommend up to five items in their basic layout (Cremonesi et al., 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).

5.4.4. Sensitivity evaluation

We evaluated the quality of the cross-domain algorithms by varying the percentage of the user overlap (10%, 50%, and 100%). Especially for 10% of user overlap, we can verify the CD-CARS accuracy under a cold-start situation, because 90% of the users do not have any rating in the target domain.

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 (Fernández-Tobías et al., 2012; Sahebi & Brusilovsky, 2013).

[16] The adopted proportion among training set vs. probe set was the same empirically used in Cremonesi et al. (2010).

[17] The original method empirically adopts 1000 additional items, but the authors let this number free to be chosen depending on the dataset used.

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

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

5.4.5. Statistical significance analysis

In order to verify the statistical significance of the evaluation results, we adopted the nonparametric *MannWhitney U test* (De Winter & Dodou, 2010), also called *MannWhitney 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 (De 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.[20] For the tests of ranking performance, we verified if the F-metric values of the proposed algorithms were greater than the F-metric values of the baselines ones,[21] 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.

5.5. 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. Section 5.5.1 shows the evaluation results about two *related* domains (*Book* and *Television*), whereas Section 5.5.2 presents the evaluation results about two less *related* domains (*Book* and *Music*). Finally, we discuss the results in Section 5.5.3.

5.5.1. Book-television results

In this section, we provide a summary[22] of the results from the evaluation of the "book-television dataset". Fig. 5 shows a dispersion diagram presenting the predictive performance (MAE) for the algorithms by varying target domain (*Book* and *Television*), contextual dimension (*Temporal, Location, Companion* and the combination *Temporal and Location*) and user overlap levels (10%, 50%, and 100%). That figure does not take into account the standard deviation and the statistical significance of the results. Table 5 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[23] and different target domain, contextual dimension and user overlap levels.

Regarding the ranking performance, Fig. 6 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. Table 6 shows the ranking 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[24] 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 ranking 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 ranking performance of that algorithm. That occurred even when a source domain had fewer ratings than the target domain (*TV* lesser than *Books*) and even when there was a cold-start scenario (10% of user overlap).
- In all scenarios, the *NNUserNgbr-transClosure* algorithm outperformed the *NNUserNgbr* one by considering their predictive performances. This fact also occurred in almost all scenarios for their ranking performances.
- The proposed algorithms (PreF and PostF) had better predictive and ranking performances in the *Temporal* dimension than other dimensions. In this contextual dimension, the PostF outperformed the *NNUserNgbr-transClosure* algorithm in all scenarios by considering either their predictive or ranking performances. The PreF outperformed the *NNUserNgbr-transClosure* algorithm in all scenarios by considering the predictive performance. With respect to the ranking 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 ranking), we see distinct results between algorithms. While the PostF algorithm outperforms the PreF one by considering the ranking performance in almost all scenarios, the opposite happens when we take the predictive performance into account.
- The more is the user overlap level the better is the ranking 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. That fact did not seem to occur when we considered the predictive performance of the algorithms.
- The predictive and ranking 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 5.1). The more particular were the tested contexts the worse were the PreF performances (e.g. in the combination of *Location* and *Temporal* dimensions, the results were worse than in the *Location* dimension alone). 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 quantity of the contextual information obtained in the *Companion* dimension (see Section 5.1.1), we can

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

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

[22] Due to the great number of experiments, detailed results are fully described in the thesis (Véras, 2016).

[23] In the table, "**" means that the result could not be considered statistically significant.

[24] In the table, "**" means that the result could not be considered statistically significant.

Table 5
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%

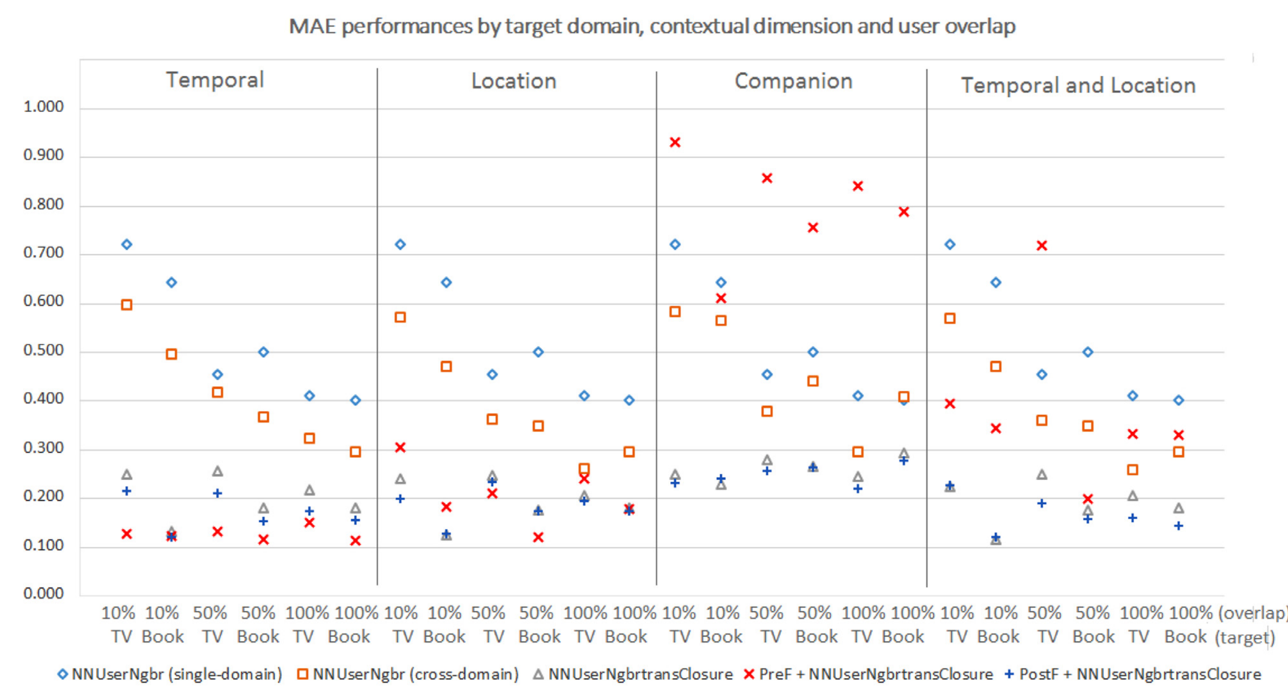

Fig. 5. Predictive performance (MAE) for the algorithms by varying target domain (book and TV), contextual dimension and user overlap levels (dispersion diagram).

see that both proposed algorithms had low predictive and ranking performances in comparison to other dimensions. In particular, the PostF algorithm achieved a good performance by considering only the MAE.

- By combining contextual dimensions (*Temporal* and *Location*), we can see the PostF predictive and ranking 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 ranking performances were similar to their own performances using only the *Location* dimension. The ranking performances of both algorithms were reduced with the addition of contextual information from another 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.
- With respect to the cold-start and sparsity problems, respectively represented in the scenarios with 10% of user overlap and *Books* domain as target (see Section 5.1.1), PreF and PostF algorithms outperformed the baseline ones in terms of predictive performance in these conditions, especially in the *Temporal* dimension. With regard to the ranking performance, only PostF outperformed the baseline algorithms in the same conditions.

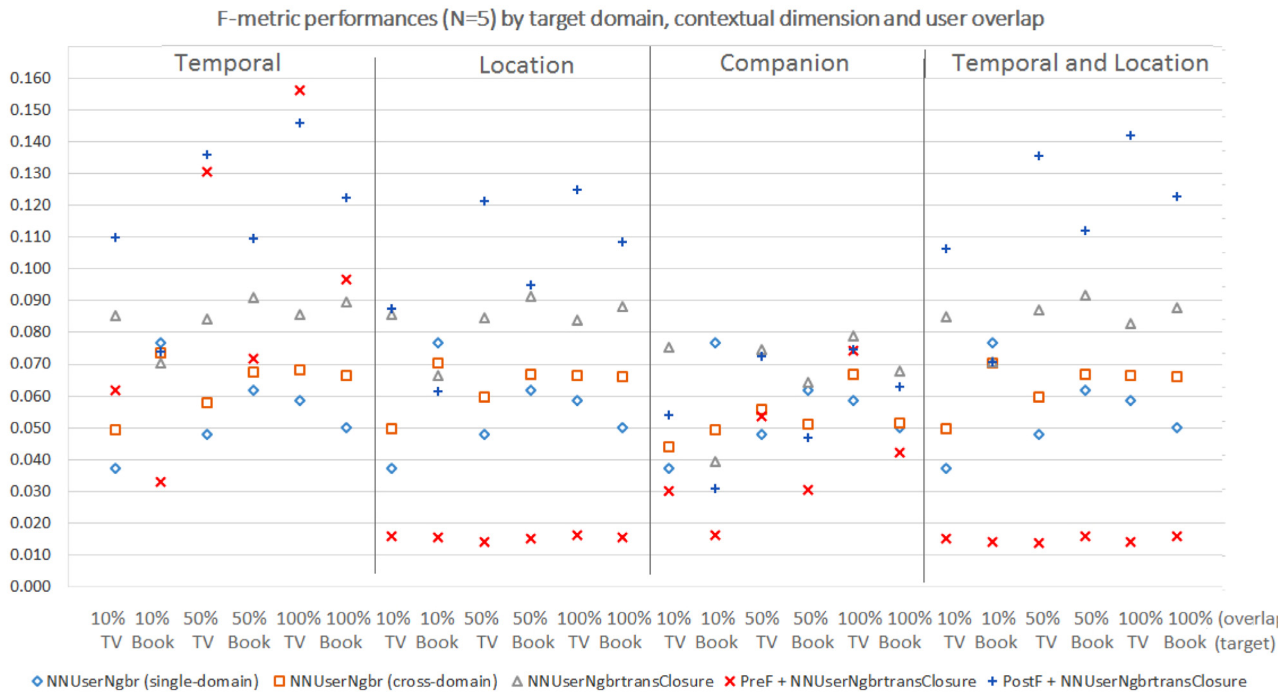

Fig. 6. Ranking performance (F-metric with N = 5) for the algorithms by varying target domain (book and TV), contextual dimension and user overlap levels (dispersion diagram).

Table 6
Overall ranking 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%

5.5.2. Book-music results

In this section, we provide a summary[25] of the results from the evaluation of the “book-music dataset”. Fig. 7 shows a dispersion diagram illustrating the MAE performance for the algorithms by varying target domain (*Book* and *Music*), contextual dimension and user overlap levels. That figure does not take into account the standard deviation and the statistical significance of the results. Table 7 presents the MAE performance achieved by the PreF and PostF algorithms in comparison to the best baseline algorithm (*NNUserNgbr-transClosure*), by taking into account their statistical

[25] Due to the great number of experiments, detailed results are fully described in the thesis (Véras, 2016).

Table 7
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%

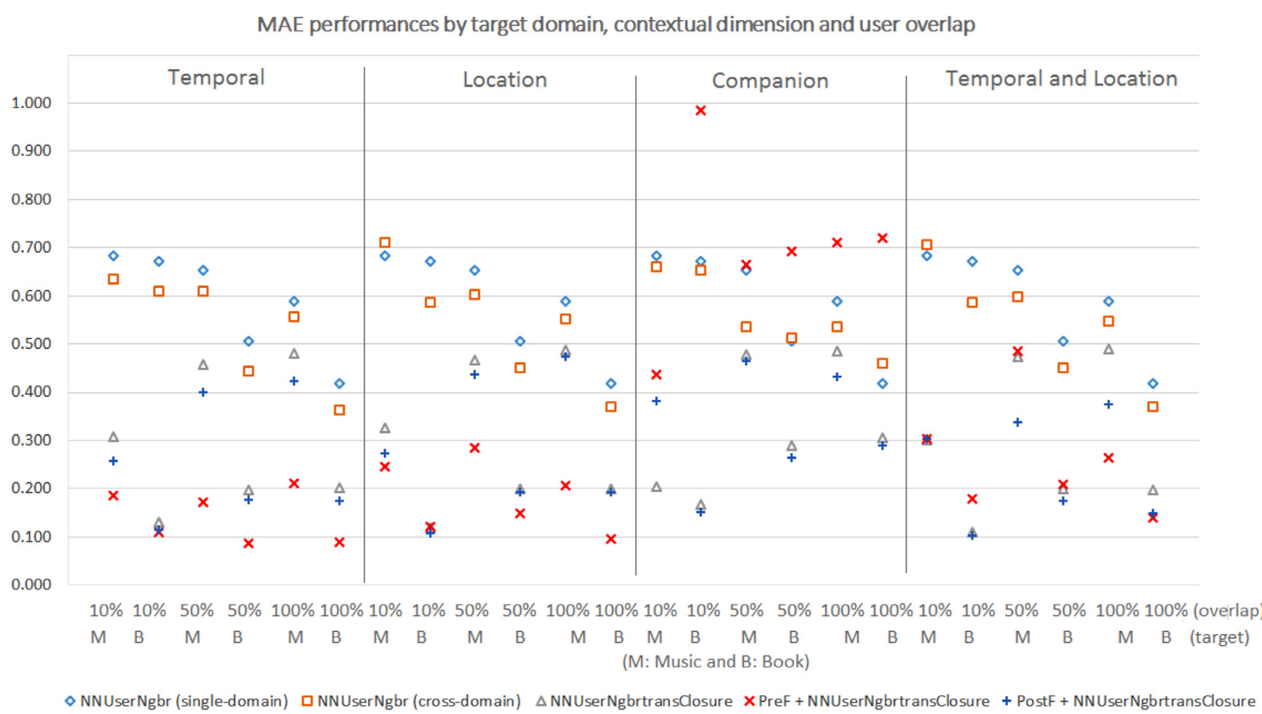

Fig. 7. MAE performance for the algorithms by varying target domain (book and music), contextual dimension and user overlap levels.

significance[26] and different target domain, contextual dimension and user overlap levels.

Regarding the ranking performance, Fig. 8 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, we are not considering the standard deviation and the statistical significance of the results in that figure. Table 8 shows the ranking 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[27] and different target domain, contextual dimension and user overlap levels.

[26] In the table, "**" means that the result could not be considered statistically significant.

[27] In the table, "**" means that the result could not be considered statistically significant.

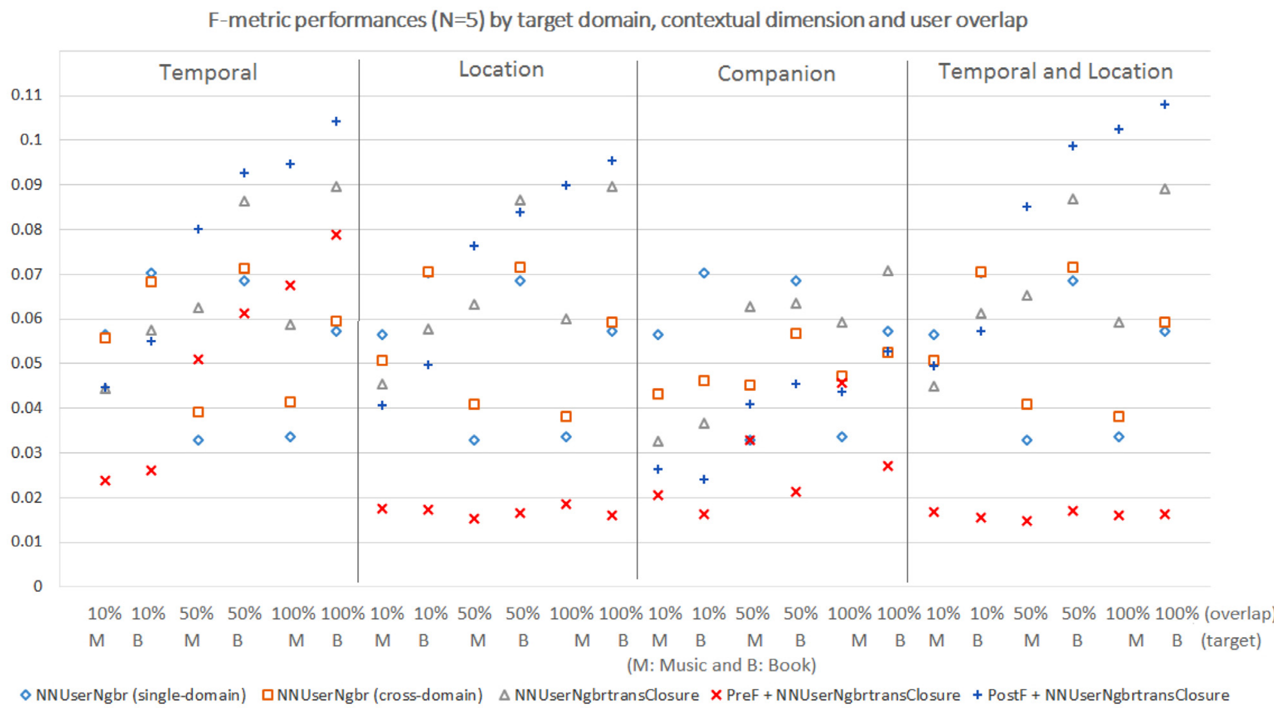

Fig. 8. F-metric performance (N = 5) for the algorithms by varying target domain (book and music), contextual dimension and user overlap levels.

Table 8
Overall ranking 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%

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 levels). By considering the ranking 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 5.5.1) 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 fewer scenarios than in the “book-television dataset”. This can also be observed for the ranking performance of that algorithm, including in the cold-start scenarios.
- The PreF algorithm outperformed the *NNUserNgbr-transClosure* one in fewer scenarios than in “book-television dataset” by con-

sidering the ranking 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", the predictive and ranking performances of the PreF algorithm was also decreased with the addition of contextual information from another contextual dimension. Besides, the PostF predictive performance was also increased, however, its ranking performance was increased with the addition of contextual information from other contextual dimension in opposite to the results from that dataset.
- In the *Temporal* dimension, PostF algorithm did not outperform the baseline algorithms in terms of ranking performance in the cold-start and sparsity scenarios (see Section 5.1.2), differently of its predictive performance, which was better than the baseline algorithms in the same conditions. PreF algorithm had similar results.

5.5.3. Discussion

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 5.1). This fact contrasts to the information gain verified in Section 5.3, 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 the recommendation of the proposed algorithms by reducing the number of temporal information present in the user ratings ("contextual sensitivity").

Unlike other contextual dimensions, the contextual information of *Companion* was obtained from inference and had a poor quality (see Section 5.2.3). That might have influenced on the recommendation accuracy for the proposed algorithms, especially for PreF one. Thus, a study focused in this problem may be done by experimenting more robust text mining algorithms in order to improve the quality of contextual information for that contextual dimension (e.g. supervised text mining techniques) (Lahlou, Benbrahim, Mountassir, & Kassou, 2013). In fact, contextual information of *Companion* may be difficult to obtain in real-world applications currently, because the majority of them are limited to obtain only user-ratings, but a few ones can improve their recommendation with that kind of information (e.g. Trip advisor[28]) (Inzunza, Juárez-Ramírez, & Jiménez, 2017).

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. 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.

In both datasets, 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 an 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 imply 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 we expected, the most of the experimented algorithms had better predictive and ranking performances in scenarios of lesser sparsity and with more information about users (higher user overlap levels), except for the *NNUserNgbr* algorithm performed without considering auxiliary domains (single-domain). Thus, we can say that the adopted cross-domain baseline algorithms as well the CD-CARS ones are able of making good recommendations under cold-start and sparsity conditions. However, one limitation of this work is that the diversity of the recommendations in a target domain was not evaluated. This and other important aspects of recommender systems will be evaluated in the future, as discussed next.

With respect to the evaluation of the proposed algorithms, we kept it as close as possible to the methodology adopted in baseline algorithms, i.e., evaluating the accuracy of the recommendations in terms of prediction and ranking (offline experiments with real datasets). Such evaluation methodology could be improved by considering different offline metrics (e.g. Breese score, Normalized Discounted Cumulative Gain, etc.) (McFee & Lanckriet, 2010) and partitioning of training and test sets, for example. However, we will concentrate future efforts in building a concrete CD-CARS for evaluating the suitability and usefulness of recommendations to users by using utility, coverage and novelty metrics, for instance (Gunawardana & Shani, 2015). Such online evaluation could prove how effective and useful would be the recommendations in real-world applications.

It is important to remember that all contextual information used in the CD-CARS was obtained implicitly or by inference (see Section 5.2). 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, especially, in the *Temporal* dimension.

Besides, the *Location* context is static and the same for all the ratings provided by a user, as mentioned in Section 5.2.2. Despite that, CD-CARS algorithms achieved good performance in several scenarios, especially for PostF one. In PreF algorithm, 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). 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). Similar behavior may occur in PostF algorithm, in which all ratings could be post filtered out since a user could not have item category preferences (e.g. comedy, action, religion) in the context

[28] https://www.tripadvisor.com.

of the recommendation (e.g. a specific location). In both cases, a dynamic location could improve the CD-CARS results in that contextual dimension.

As described in Section 4.2, 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 has category preferences in source domains only. 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 too. In order to alleviate this problem, some techniques could be used. For example, using association rule mining (Lazcorreta, Botella, and Fernandez-Caballero, 2008; Soysal, 2015) 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 could make enhancement of the category preferences tensor by using association rules to infer other item categories preferred by the users according to the possible contexts.

It is important to say that we could combine PreF and PostF algorithms in order to try having the best of their features in a single hybrid algorithm, for example. In addition, another algorithm could be implemented, based in a third CARS paradigm (*Modeling*, mentioned in Section 2.2). That algorithm could make “multidimensional” recommendations by considering contextual information beyond user-item ratings without (pre- or post-) filtering the contextual user-rating tensors. For instance, we could extend two single-domain context-aware *Modelling* approaches: heuristic calculations (Adomavicius et al., 2005) and matrix factorization (Baltrunas, Ludwig, & Ricci, 2011), in order to make cross-domain context-aware recommendations. The traditional single-domain heuristic calculations could be expanded for including other dimensions as context and item domain (e.g. an adapted Euclidian distance could be calculated[29]). On the other hand, a matrix factorization approach, such as the described in Baltrunas et al. (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).

Finally, notice that the item categories from contextual preference tensor could also be expressed as a set of attributes, which characterize an item (e.g. user tags), instead of being expressed as item genres (e.g. comedy, action, rock). It allows other datasets to be used and experimented in CD-CARS. Besides, we plan in future make the contextual information more representative (e.g. by using ontologies) which will also allow the integration of other datasets to our base, by making fuzzy/semantic matching of contexts, for instance.

6. Conclusions

In this work, we have shown that context-aware techniques can be used to improve the *accuracy* of cross-domain recommendations while maintaining the advantages of CD-CFRS in relation to cold-start and sparsity issues. A traditional cross-domain CF-based algorithm provided better recommendations when used in combination with the implemented CD-CARS algorithms. Experimental evaluations conducted in two real data sets, one with two more related domains (*Book* and *Television*) and the other with two less related domains (*Book* and *Music*), showed that considering contextual information of three dimensions (*Temporal, Local* and *Companion*), generating predictions exploring knowledge of a source domain improved the predictive and classification performances in the target domain (up to 62% and 45%, respectively) depending on the scenario (domain, context, and user-overlap level).

Other contributions of this work are the formalization of the cross-domain context-aware recommendation problem; the proposal of novel CD-CARS algorithms based on distinct and systematic paradigms of context-aware recommendation (*Pre-Filtering* and *Post-Filtering*). Both algorithms allow traditional single-domain and cross-domain CF-based algorithms to be used as a base; and the provisioning of two real datasets for evaluating CD-CARS taking into account different domains and contextual information.

Through the 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. Nevertheless, the CD-CARS proposed in this work allows further investigation in multiple research directions such as: improving the CD-CARS algorithms (e.g. combining PreF and PostF or implementing a Modeling approach) or adopting other cross-domain algorithms as a base; algorithms could be used or proposed to infer other, or more precise, contextual information from user reviews; investigating and providing data mining techniques to select the most relevant contextual dimensions, attributes, and values (or their combination) before performing recommendation or evaluation, given that the verification of all possible situations is costly; evaluating CD-CARS for other scenarios (user-overlap levels, contextual dimensions, domains, etc.) and by means of online experiments, which could prove that real-world applications would be useful for real users; among others.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Douglas Véras: Conceptualization, Funding acquisition, Formal analysis, Writing - original draft, Writing - review & editing. **Ricardo Prudêncio:** Conceptualization, Writing - review & editing. **Carlos Ferraz:** Conceptualization, Writing - review & editing.

Acknowledgments

This work has been supported by CAPES, CNPq, and Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE), Grant No. IBPG-0169-1.03/12.

References

Abel, F., Araújo, S., Gao, Q., & Houben, G.-J. (2011). Analyzing cross-system user modeling on the social web. In *Web engineering* (pp. 28–43). Springer.

Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS), 23*(1), 103–145.

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

Adomavicius, G., & Tuzhilin, A. (2015). Context-aware recommender systems. In *Recommender systems handbook (second edition)* (pp. 191–226). Springer.

Alhamid, M., Rawashdeh, M., Dong, H., Hossain, M., Alelaiwi, A., & El Saddik, A. (2015). Recam: A collaborative context-aware framework for multimedia recommendations in an ambient intelligence environment. *Multimedia Systems, online*, 1–15. doi:10.1007/s00530-015-0469-2.

Baltrunas, L., Ludwig, B., & Ricci, F. (2011). Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM conference on recommender systems* (pp. 301–304). ACM.

Bauman, K., & Tuzhilin, A. (2014). Discovering contextual information from user reviews for recommendation purposes. In *Cbrecsys* (pp. 1–8).

Richa & Bedi, P. (2018). Parallel proactive cross domain context aware recommender system. *Journal of Intelligent & Fuzzy Systems, 34*(3), 1521–1533.

---

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

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research, 3*, 993–1022.
Braunhofer, M., Kaminskas, M., & Ricci, F. (2013). Location-aware music recommendation. *International Journal of Multimedia Information Retrieval, 2*(1), 31–44.
Brézillon, P. (2007). Context modeling: Task model and practice model. In *Modeling and using context* (pp. 122–135). Springer.
Cantador, I., & Cremonesi, P. (2014). Tutorial on cross-domain recommender systems. In *Proceedings of the 8th ACM conference on recommender systems* (pp. 401–402). ACM.
Cantador, I., Fernández-Tobías, I., Berkovsky, S., & Cremonesi, P. (2015). Cross-domain recommender systems. In *Recommender systems handbook* (pp. 919–959). Springer.
Cao, B., Liu, N. N., & Yang, Q. (2010). Transfer learning for collective link prediction in multiple heterogenous domains. In *Proceedings of the 27th international conference on machine learning (icml-10)* (pp. 159–166).
Carmagnola, F., Cena, F., & Gena, C. (2011). User model interoperability: A survey. *User Modeling and User-Adapted Interaction, 21*(3), 285–331.
Chatterjee, S., & Hadi, A. S. (2015). *Regression analysis by example*. John Wiley & Sons.
Colombo-Mendoza, L. O., Valencia-García, R., Rodríguez-González, A., Alor-Hernández, G., & Samper-Zapater, J. J. (2015). Recommetz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Systems with Applications, 42*(3), 1202–1222.
Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on recommender systems* (pp. 39–46). ACM.
Cremonesi, P., Tripodi, A., & Turrin, R. (2011). Cross-domain recommender systems. In *IEEE 11th international conference on data mining workshops (ICDMW)* (pp. 496–503). IEEE.
De Winter, J. C., & Dodou, D. (2010). Five-point likert items: *t* test versus mann-whitney-wilcoxon. *Practical Assessment, Research & Evaluation, 15*(11), 1–12.
Enrich, M., Braunhofer, M., & Ricci, F. (2013). Cold-start management with cross–domain collaborative filtering and tags. In *E-commerce and web technologies* (pp. 101–112). Springer.
Felfernig, A., Friedrich, G., Jannach, D., & Zanker, M. (2015). Constraint-based recommender systems. In *Recommender systems handbook* (pp. 161–190). Springer.
Fernández-Tobías, I., Cantador, I., Kaminskas, M., & Ricci, F. (2012). Cross-domain recommender systems: A survey of the state of the art. *Spanish conference on information retrieval*.
Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., & Guo, J. (2013). Cross-domain recommendation via cluster-level latent factor model. In *Machine learning and knowledge discovery in databases* (pp. 161–176). Springer.
Gunawardana, A., & Shani, G. (2015). Evaluating recommender systems. In *Recommender systems handbook* (pp. 265–308). Springer.
Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: An update. *ACM SIGKDD Explorations Newsletter, 11*(1), 10–18.
Hsieh, C.-K., Yang, L., Wei, H., Naaman, M., & Estrin, D. (2016). Immersive recommendation: News and event recommendations using personal digital traces. In *Proceedings of the 25th international conference on world wide web* (pp. 51–62). International World Wide Web Conferences Steering Committee.
Hwangbo, H., & Kim, Y. (2017). An empirical study on the effect of data sparsity and data overlap on cross domain collaborative filtering performance. *Expert Systems with Applications, 89*, 254–265.
Inzunza, S., Juárez-Ramírez, R., & Jiménez, S. (2017). User modeling framework for context-aware recommender systems. In *World conference on information systems and technologies* (pp. 899–908). Springer.
Jadidi, O., Firouzi, F., & Bagliery, E. (2010). Topsis method for supplier selection problem. *World Academy of Science, Engineering and Technology, 47*, 956–958.
Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters, 31*(8), 651–666.
Ji, K., & Shen, H. (2015). Making recommendations from top-n user-item subgroups. *Neurocomputing, 165*, 228–237.
Kaminskas, M., Fernández-Tobías, I., Ricci, F., & Cantador, I. (2014). Knowledge-based identification of music suited for places of interest. *Information Technology & Tourism, 14*(1), 73–95.
Kim, S., & Yoon, Y. (2014). Recommendation system for sharing economy based on multidimensional trust model. *Multimedia Tools and Applications*, 1–14.
Koohi, H., & Kiani, K. (2017). A new method to find neighbor users that improves the performance of collaborative filtering. *Expert Systems with Applications, 83*, 30–39. doi:10.1016/j.eswa.2017.04.027.
Lahlou, F. Z., Benbrahim, H., Mountassir, A., & Kassou, I. (2013). Inferring context from users reviews for context aware recommendation. In *International conference on innovative techniques and applications of artificial intelligence* (pp. 227–239). Springer.
Lazcorreta, E., Botella, F., & Fernández-Caballero, A. (2008). Towards personalized recommendation by two-step modified apriori data mining algorithm. *Expert Systems with Applications, 35*(3), 1422–1429. doi:10.1016/j.eswa.2007.08.048.
Lee, H. J., & Park, S. J. (2007). Moners: A news recommender for the mobile web. *Expert Systems with Applications, 32*(1), 143–150.
Leskovec, J., Adamic, L. A., & Huberman, B. A. (2007). The dynamics of viral marketing. *ACM Transactions on the Web (TWEB), 1*(1), 5.
Li, B., Yang, Q., & Xue, X. (2009). Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In *Ijcai: 9* (pp. 2052–2057).
Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining*: 454. Springer Science & Business Media.
Loni, B., Shi, Y., Larson, M., & Hanjalic, A. (2014). Cross-domain collaborative filtering with factorization machines. In *Advances in information retrieval* (pp. 656–661). Springer.
McFee, B., & Lanckriet, G. R. (2010). Metric learning to rank. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 775–782).
Moe, H. H., & Aung, W. T. (2014a). Building ontologies for cross-domain recommendation on facial skin problem and related cosmetics. *International Journal of Information Technology and Computer Science (IJITCS), 6*(6), 33.
Moe, H. H., & Aung, W. T. (2014b). Context aware cross-domain based recommendation. *International conference on advances in engineering and technology*.
Moe, H. H., & Aung, W. T. (2013). Cross-domain recommendations for personalized semantic services. *International Journal of Computer Applications Technology and Research, 2*(1), 72–76.
Moon, A., Kim, H., Lee, K., & Kim, H. (2006). Designing CAMUS based context-awareness for pervasive home environments. In *International conference on hybrid information technology: 1* (pp. 666–672).
Moreno, O., Shapira, B., Rokach, L., & Shani, G. (2012). Talmud: Transfer learning for multiple domains. In *Proceedings of the 21st ACM international conference on information and knowledge management* (pp. 425–434). ACM.
Nilashi, M., Ibrahim, O., & Bagherifard, K. (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications, 92*, 507–520. doi:10.1016/j.eswa.2017.09.058.
Oku, K., Nakajima, S., Miyazaki, J., & Uemura, S. (2006). Context-aware svm for context-dependent information recommendation. In *Proceedings of the 7th international conference on mobile data management* (p. 109). IEEE Computer Society.
Palmisano, C., Tuzhilin, A., & Gorgoglione, M. (2008). Using context to improve predictive modeling of customers in personalization applications. *Knowledge and Data Engineering, IEEE Transactions on, 20*(11), 1535–1549.
Pan, W., & Yang, Q. (2013). Transfer learning in heterogeneous collaborative filtering domains. *Artificial intelligence, 197*, 39–55.
Panniello, U., Tuzhilin, A., & Gorgoglione, M. (2014). Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and User-Adapted Interaction, 24*(1–2), 35–65.
Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., & Pedone, A. (2009). Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the third ACM conference on recommender systems* (pp. 265–268). ACM.
Parameswaran, A., Venetis, P., & Garcia-Molina, H. (2011). Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems (TOIS), 29*(4), 20.
Park, D. H., Kim, H. K., Choi, I. Y., & Kim, J. K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications, 39*(11), 10059–10072.
Pham, X. H., Jung, J. J., & Le Anh Vu, S.-B. P. (2014). Exploiting social contexts for movie recommendation. *Malaysian Journal of Computer Science, 27*(1), 68–79.
R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical ComputingVienna, Austria.
Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2015). *Recommender systems handbook*. Springer.
Sahebi, S., & Brusilovsky, P. (2013). 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* (pp. 289–295). Springer.
dos Santos, V., Marcelino, E. J., Bezerra, B. L., de Amorim, J., & do Nascimento, A. I. (2012). A recommender system architecture for an inter-application environment. In *12th international conference on intelligent systems design and applications (ISDA)* (pp. 472–477). IEEE.
Shapira, B., Rokach, L., & Freilikhman, S. (2013). Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction, 23*(2–3), 211–247.
Soysal, O. M. (2015). Association rule mining with mostly associated sequential patterns. *Expert Systems with Applications, 42*(5), 2582–2592. doi:10.1016/j.eswa.2014.10.049.
Sundermann, C. V., Domingues, M. A., da Silva Conrado, M., & Rezende, S. O. (2016). Privileged contextual information for context-aware recommender systems. *Expert Systems with Applications, 57*, 139–158.
Taneja, A., & Arora, A. (2018). Cross domain recommendation using multidimensional tensor factorization. *Expert Systems with Applications, 92*, 304–316. doi:10.1016/j.eswa.2017.09.042.
Tekin, C., & van der Schaar, M. (2015). Contextual online learning for multimedia content aggregation. *Multimedia, IEEE Transactions on, 17*(4), 549–561.
Tiroshi, A., & Kuflik, T. (2012). Domain ranking for cross domain collaborative filtering. In *User modeling, adaptation, and personalization* (pp. 328–333). Springer.
Unger, M., Bar, A., Shapira, B., & Rokach, L. (2016). Towards latent context-aware recommendation systems. *Knowledge-Based Systems, 104*, 165–178.
Véras, D. (2016). *CD-CARS: Cross-Domain Context-Aware Recommender Systems*. Federal University of Pernambuco (UFPE) Ph.D. thesis..
Véras, D., Prota, T., Bispo, A., Prudêncio, R., & Ferraz, C. (2015). A literature review of recommender systems in the television domain. *Expert Systems with Applications, 42*(22), 9046–9076.
Véras, D., Prudencio, R., Ferraz, C., Bispo, A., & Prota, T. (2015). Context-aware techniques for cross-domain recommender systems. In *2015 Brazilian conference on intelligent systems (bracis)* (pp. 282–287). IEEE.
Viktoratos, I., Tsadiras, A., & Bassiliades, N. (2018). Combining community-based knowledge with association rule mining to alleviate the cold start problem in context-aware recommender systems. *Expert Systems with Applications, 101*, 78–90.

Vildjiounaite, E., Kyllönen, V., Hannula, T., & Alahuhta, P. (2009). Unobtrusive dynamic modelling of tv programme preferences in a finnish household. *Multimedia Systems, 15*, 143–157. doi:10.1007/s00530-009-0157-1.

Villegas, N. M., Sánchez, C., Díaz-Cely, J., & Tamura, G. (2018). Characterizing context-aware recommender systems: A systematic literature review. *Knowledge-Based Systems, 140*, 173–200.

Wang, F., Li, D., & Xu, M. (2015). A location-aware tv show recommendation with localized sementaic analysis. *Multimedia Systems, online*, 1–8. doi:10.1007/s00530-015-0451-z.

Wang, S., Li, C., Zhao, K., & Chen, H. (2017). Learning to context-aware recommend with hierarchical factorization machines. *Information Sciences, 409*, 121–138.

Winoto, P., & Tang, T. (2008). 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, 26*(3), 209–225.

Zarka, R., Cordier, A., Egyed-Zsigmond, E., Lamontagne, L., & Mille, A. (2016). Trace-based contextual recommendations. *Expert Systems with Applications, 64*, 194–207.

Zhang, J., Yuan, Z., & Yu, K. (2014). Cross media recommendation in digital library. In *The emergence of digital libraries–research and practices* (pp. 208–217). Springer.