Knowledge-Based Systems 140 (2018) 173–200

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Characterizing context-aware recommender systems: A systematic literature review

Norha M. Villegas*, Cristian Sánchez, Javier Díaz-Cely, Gabriel Tamura

Universidad Icesi, Calle 18 No. 122-135, Cali, 760031, Colombia

ARTICLE INFO

Article history:
Received 14 March 2017
Revised 31 October 2017
Accepted 2 November 2017
Available online 3 November 2017

Keywords:
Recommender systems
Context-aware recommender systems
Pre-filtering
Post-filtering
Context modeling
Recommender systems evaluation

ABSTRACT

Context-aware recommender systems leverage the value of recommendations by exploiting context information that affects user preferences and situations, with the goal of recommending items that are really relevant to changing user needs. Despite the importance of context-awareness in the recommender systems realm, researchers and practitioners lack guides that help them understand the state of the art and how to exploit context information to smarten up recommender systems. This paper presents the results of a comprehensive systematic literature review we conducted to survey context-aware recommenders and their mechanisms to exploit context information. The main contribution of this paper is a framework that characterizes context-aware recommendation processes in terms of: i) the recommendation techniques used at every stage of the process, ii) the techniques used to incorporate context, and iii) the stages of the process where context is integrated into the system. This systematic literature review provides a clear understanding about the integration of context into recommender systems, including context types more frequently used in the different application domains and validation mechanisms—explained in terms of the used datasets, properties, metrics, and evaluation protocols. The paper concludes with a set of research opportunities in this field.

1. Introduction

With the proliferation of big data & data analytics technologies, recommender systems (RS) are now crucial in seeking customer satisfaction through personalization [1]. RS aim at selecting and proposing the most relevant items, services and offers for their users, by considering their profiles, purchase history, preferences, opinions, interactions with offered products and services, as well as their relationships with other clients. At the same time, the generalization of smart-phones and ubiquitous computing has given RS access to context information [2]. Context-aware recommender systems (CARS) go one step further from traditional RS by exploiting context information such as time, location, and user activity to understand user situations and their influence on user preferences. The incorporation of context information into RS [2,3] leverages the value of these systems by improving the relevance of possible recommendations with respect to changing user needs [4,5].

The value of context information to improve the quality of recommendations has been demonstrated and supported by different researchers [6–11]. Nevertheless, RS as well as context-awareness researchers and practitioners interested in combining the two areas still lack a guide that helps them understand how to exploit context information to smarten up RS. Evidence of this is the absence of comprehensive and domain-independent surveys, particularly systematic literature reviews, that not only consolidate the state of the art of the field, but also explain the most common techniques used to integrate context into the recommendation process. After a rigorous revision of the state of the art, we found that none of the available surveys comprehensively characterize recommendation processes from the perspective of the exploitation of context information. In the best cases, existing surveys focus only on the identification of used context types, and most of them address the problem from the perspective of a particular domain.

This paper presents the findings of a systematic literature review (SLR) [12] on CARS that we conducted with the goal of helping practitioners and researchers understand how context information can be effectively combined with recommendation mechanisms. To this end, we studied a final set of 87 CARS papers that were classified as content-based, collaborative filtering and hybrid approaches. For each paper, we identified recommendation techniques, means to exploit context information, context types, application domains, validation mechanisms including the used datasets, the improvements obtained through the exploitation of context (when measured quantitatively), and research opportuni-

* Corresponding author.
E-mail addresses: nvillega@icesi.edu.co (N.M. Villegas), cesanchez@icesi.edu.co (C. Sánchez), jgdiaz@icesi.edu.co (J. Díaz-Cely), gtamura@icesi.edu.co (G. Tamura).

https://doi.org/10.1016/j.knosys.2017.11.003
0950-7051/

ties. The main results of our study are reported in this paper in the form of a framework that characterizes recommendation processes in terms of: i) the recommendation techniques used at every stage of the process, ii) the techniques used to incorporate context, and iii) the stages of the process where context is integrated into the system. This manuscript aims at providing a clear understanding about where context information is usually integrated into the system, what techniques are available to exploit context information depending on the underlying recommendation approach and the phase of the process where context is included, what context types are more frequently exploited in the different application domains, and what validation mechanisms—explained in terms of the used datasets, properties, metrics and evaluation protocols—are generally used to evaluate the proposed approaches. Last, but not least, the paper discusses research opportunities relevant to CARS.

This paper is structured as follows. Section 2 explains foundational concepts on recommender systems and context information. Section 3 visits related work by analysing the contributions of our SLR with respect to other surveys published on CARS. Section 4 explains the methodology we followed to conduct the SLR. Sections 5–8 constitute the contributions of this manuscript: Section 5 presents the findings of our SLR and the characterization framework for CARS; Section 6 reports on the validation methods and datasets identified in the studied approaches; Section 7 presents quantitative data, reported in the studied papers, on the improvements obtained from the exploitation of context information; and Section 8 summarizes and classifies research opportunities. Finally, Section 9 concludes the paper.

2. Background

This section briefly presents the fundamentals of RS, and context information as an enabler to improve the quality of recommendations.

2.1. Recommender systems (RS)

Dating back to the mid 1990s, the first recommender systems emerged by following two well differentiated paths. On the one hand, *content-based recommenders* drew from the fields of document retrieval [13,14] and user profiling [15] to define a common representation space for describing items and users. User profiles result from the aggregation of items that have been favorably or unfavorably qualified in the past. For a given user, items similar to the user's profile are recommended, without taking into account information from other users. On the other hand, *collaborative filtering recommenders* evolved from contributions in human computer interaction [16,17], where the preferences and choices of similar users are used as the basis for recommendation.

Each of these two types of systems has advantages and disadvantages. Content-based recommenders are easy to explain and understand, prove a good starting point for item navigation, and allow recommendations for new users and/or items (cold start problem). However, they imply the cumbersome task of thoroughly and explicitly describing all items using a common set of features, do not work with implied content, can only handle complementary item recommendation and, being centered on a single user, do not allow the recommendation of serendipitous items. In contrast, collaborative filtering recommenders are based on the common preferences of crowds of users. Thus, these systems cannot only recommend complementary as well as substitute items, but also surprise users by recommending unusual items. Nevertheless, they are not as transparent on their recommendations, need substantially more user data to work well, and do not provide a way to deal with the cold start problem.

Hybrid recommenders, a third type of RS, provide a middle ground between content-based and collaborative filtering systems, by leveraging their strengths and mitigating their drawbacks.

This categorization of RS was proposed by Adomavicius and Tuzhilin in [6]. Other authors have proposed other types of systems [1,18]. In particular, we consider case-based and knowledge-based systems to be subtypes of the content-based family, community-based systems to be subtypes of the collaborative filtering family, and demographic recommenders to be either content-based or collaborative filtering systems following a pre-filtering stage where data are partitioned in subsets according to user characteristics.

RS use information from items, users, and preferences. The main source of information is the item by user matrix that stores user preferences for individual items. These preferences can be explicitly stated (e.g., in the form of ratings or likes), or implicitly inferred from the interactions of the user with the system (e.g., purchases, accesses or reads). Content-based recommenders consider additional sources of information in the form of feature vectors describing different characteristics of each item (e.g., category, size, age, brand, author).

The characterization of CARS presented in this paper is driven by the stages of the processes followed by content-based and collaborative filtering systems.

2.2. Context information

Abowd et al. define *context* as "*any information useful to characterize the situation of an entity (e.g., a user or an item) that can affect the way users interact with systems*"[2]. The precision of recommendations may result highly affected by context information [7,8]. For example, a costumer could be more or less interested in a particular restaurant depending on the day of the week. Contextual information can be defined as static or dynamic [3]. When context is static, recommender applications assume that this information is immutable over time. An example of static context is the birthday of a user. On the contrary, dynamic context changes over time thus highly affecting user current needs. Instances of dynamic context are location, time, and user activity [5].

2.2.1. Context categories

Villegas et al. [5] characterize context along five general categories: individual, location, time, activity, and relational. Other characterizations, which can be instantiated from these general categories, have been proposed for domain specific CARS (e.g., the one proposed by Verbert et al. in [19] for CARS in the learning realm). To identify the context types exploited by the CARS studied in this SLR, we based on the classification of context information proposed by Villegas et al., which is summarized as follows:

- **Individual context:** Corresponds to information observed from independent entities (e.g., users or items) that may share common features. This category can be sub-classified into *natural, human, artificial*, or *groups of entities. Natural context* represents characteristics of living and non-living entities that occur naturally, that is, without human intervention (e.g. weather information). *Human context* describes user behavior and preferences (e.g., user payment preferences). *Artificial context* describes entities that result from human actions or technical processes (e.g., hardware and software configurations used in e-commerce platforms). The last subcategory, *groups of entities*, concerns groups of independent subjects that share common features, and that might relate each other (e.g., preferences of users in the user's social network).
- **Location context:** Refers to the place associated with an entity's activity (e.g., the city where a user lives). This category is sub-classified as *physical* (e.g., the coordinates of the user's location, a movie theater's address, or the directions to reach the

movie theater from the costumer's current location), and *virtual* (e.g., the IP address of a computer that is located within a network).

- **Time context:** Corresponds to information such as time of the day, current time, day of the week, and season of the year. Time context can be categorized as *definite* and *indefinite. Definite* context indicates time frames with specific begin and end points. *Indefinite* context refers to recurrent events that occur while another situation takes place, so it does not have a defined duration (e.g. a user's session in an e-commerce application).
- **Activity context:** Refers to the tasks performed by entities (e.g., shopping, the task a user does at a particular time).
- **Relational context:** Refers to entity relationships that arise from the circumstances in which the entities are involved [20]. Relational context can be defined as *social* (i.e., interpersonal relations such as associations or affiliations), *functional* (i.e. the usage than an entity makes of another).

2.2.2. Integrating Context into Recommender Systems

Traditional recommender systems rely on information about users and items. In contrast, CARS rely also on context information that is relevant for the recommendation. Therefore, recommendation tasks in context-aware recommender systems can be seen as a function of users, items and context information [8]:

$$f : Users \times Items \times Context \rightarrow R \tag{1}$$

There exists three paradigms to integrate context information into recommender systems, depending on the phase of the recommendation process at which context is processed [8]:

- **Contextual pre-filtering:** Context information is used as a filtering mechanism applied to the data, before the application of the recommendation model.
- **Contextual post-filtering:** Context information is initially ignored, and preferences are computed by applying traditional recommender algorithms on the entire data. The resulting set of recommendations is then filtered according to context information that is relevant to the user.
- **Contextual modeling:** Context information is directly integrated into the recommendation model, for example as part of the preference computation process.

This SLR characterizes CARS by considering these three paradigms to incorporate context into the recommendation process, and the techniques used for this integration.

3. Related work

We found 15 RS surveys published in relevant venues and journals between 2004 and 2016. However, only 7 out of these 15 surveys, published between 2012 and 2014, relate to the improvement of RS through the incorporation of context information. Aiming at providing a comprehensive understanding of the state of the art of this field, our SLR not only follows a well defined research methodology, but also characterizes CARS along all application domains, context types, and techniques reported in the studied literature. Most importantly, we documented the recommendation processes followed by content-based and collaborative filtering CARS, to characterize how these systems exploit context information along all phases of the process. The characterization includes recommendation techniques, paradigms for incorporating context, context types, application domains, and a detailed explanation of the mechanisms used to exploit context. We also compiled a catalog of datasets and validation methods used in the studied approaches, as well as a list of open challenges.

Table 1compares our literature review (last row) with the most relevant CARS surveys we found in the state of the art. This comparison is based on seven criteria that we define as follows: *i) SLR*, the literature review follows a systematic methodology; *ii) not focused on particular domains or techniques*, the survey reviews the state of the art across all identified domains and techniques; *iii) not focused on particular context types*, the survey reports the exploitation of different context types; *iv) identifies context exploitation techniques*, the survey reports the ways how context was exploited in the studied RS; *v) context in the stages of the recommendation process*, the literature review documents how context is exploited along the stages of the recommendation process; *vi) datasets*, the survey lists the datasets used by the studied systems; and *vii) validation techiques*, the review reports the techniques used to evaluate the studied approaches. The plus sign in a cell indicates that the survey is compliant with the corresponding criterion, whereas the absence of the sign indicates that it is not.

According to Table 1, four surveys focus on particular domains: learning processes [19], music services [21], digital libraries [23], and mobile applications [22]. All surveys identify the different types of context exploited in the studied RS, except the one by Campos et al. [24] that focuses on time context only. Furthermore, this survey does not provide insights on the exploitation of context into RS (context exploitation techniques are not identified), but on the evaluation methods used to evaluate the effectiveness of CARS. The surveys conducted by Verbert et al. [19], and Kaminskas and Ricci [21] describe the techniques used to exploit context in the studied systems and the means used to validate them. However, they focus on particular domains. The survey by Liu et al. [22] focuses only on methods to identify the relevant context and the context types exploited in mobile systems. Thus, besides being domain specific, it does not report on techniques used to take advantage of context. As our literature review, the survey conducted by Inzuza et al. [25] follows a systematic approach and does not relate to a particular application domain, technique or context type. However, it does not report on context exploitation techniques. Also similarly to our work, the work conducted by Seifu and Mogalla [26] aims at characterizing the process followed by CARS in the form of what they call *"a framework of CARS."* Nevertheless, their focus is not the way how context is incorporated and exploited, and the explanation of the framework in their six page paper is not as comprehensive as our characterization. Finally, none of the studied surveys report on the used datasets or relate context and its means to exploit it to the concrete phases of the recommendation process.

4. Methodological aspects

We conducted this study by following the guidelines proposed by Kitchenham and Charters in [12]. With our long-term research goal in mind—*to look for innovative and more effective ways of exploiting context information to improve the effectiveness of recommender systems*, we defined the set of research questions that would allow us to understand the state of the art of CARS. These questions are stated as follows:

- RQ1: How is context information exploited along the recommendation process?
- RQ2: What are the existing techniques used to incorporate context information into RS? For each technique, what are the most common application domains?
- RQ3: Is there any correlation between techniques used to incorporate context into RS and any of the traditional recommendation approaches (i.e., content-based, collaborative filtering and hybrid)?

Table 1
Related work—Comparing our SLR with other surveys on CARS.

Author/Year	SLR	Not focused on particular domains or techniques	Not focused on particular context types	Identifies context exploitation techniques	Context in the stages of the recommendation process	Datasets	Validation techniques
Verbert et al., 2012 [19]			+	+			+
Kaminskas and Ricci, 2012 [21]			+	+			+
Liu et al., 2013 [22]			+				
Champiri et al, 2014 [23]			+	+			
Campos et al., 2014 [24]		+					+
Inzunza et al., 2016 [25]	+	+	+				
Seifu and Mogalla., 2016 [26]		+	+				
Our literature review	+	+	+	+	+	+	+

- RQ4: What are the types of context more commonly exploited by RS? What techniques apply in each case?
- RQ5: What evaluation methods have been used to validate the effectiveness of CARS? What are the most common metrics used by these methods?

To answer these research questions and understand the way how context information is integrated into recommender systems, it was important first to characterize the processes that are followed by these systems, in particular by content-based and collaborative filtering approaches. That is, to understand the data that constitute the inputs, and the stages implemented by each type of recommender system to generate recommendations. This process-oriented characterization allowed us not only to report the techniques and context used by the studied RS, but also to map them to specific phases of the recommendation process, with the goal of leveraging the usefulness of this SLR for understanding the state of the art of this field.

We conducted a bibliographic search of conference proceedings and journal papers published in IEEE, ACM, ScienceDirect, EBSCO and Springer. These databases were selected because of the quality of their publications, and their relevance to RS. We used the search string *((“recommendation systems” OR “recommender systems” OR “recommendation” OR “recommendations”) AND (“context aware” OR “context-aware” OR “context information” OR “contextual information” OR “location” OR “social” OR “time” OR “activity” OR “task” OR “environmental”)).*

To select the papers to be included in the study we applied four filters: i) *publication date*, we selected papers published between 2004 and 2016; ii) *publication type, number of citations and language*, we excluded workshop and symposium proceedings, papers with less than 10 citations (with some exceptions for papers recently published) and non-English papers; iii) *relevance*, we studied the abstracts to verify the relevance of each paper. After this third filter, we obtained a total of 286 articles, including surveys on RS.

We thoroughly analyzed all these 286 articles and characterized those proposing CARS according to seven criteria: i) *recommendation system approach*, whether it is content-based, collaborative filtering, or hybrid; ii) *recommendation techniques*, the mechanisms used at the different stages of the recommendation process; iii) *paradigm for incorporating context*, whether it is pre-filtering, post-filtering, or contextual modeling; iv) *context types*, the context categories that are exploited in the recommender system (based on the classification proposed by Villegas and Müller [5]); v) *application domain* (if applicable), the specific area targeted by the proposed RS; vi) *evaluation*, the methods and metrics used to validate the effectiveness of the proposed RS; and vii) *data sets* (when reported), the data used to evaluate the proposed approach.

The fourth and last filter consisted in excluding those papers for which we could not identify any of the mandatory criteria presented above. The final set of papers includes 87 manuscripts that propose CARS and 15 surveys, including four highly relevant papers that were published in 2017.

5. Characterization of Context-Aware RS (CARS)

This section summarizes, for each type of recommender system, the findings of our SLR. We consider that the differences between content-based, collaborative filtering, and hybrid recommenders are too profound to analyze them all together, thus we set to do it independently.

To characterize content-based and collaborative filtering CARS, we first represented their recommendation processes using flow diagrams (cf. Figs. 1 and 2) that allow us to distinguish the different phases they comprise, and identify the points where context information is exploited by the surveyed RS, following either the pre-filtering, post-filtering or contextual modeling paradigms.

Bold ovals labeled as “Context” indicate the points of the process where we consider that context information can be incorporated. Citations next to each oval correspond to the studied approaches that integrate context in that specific phase of the recommendation process. The absence of citations next to an oval indicates that although we consider context can be exploited at that phase of the process, we found no approaches that do so. Furthermore, each brace depicted in the diagrams groups the stages of the process associated with each of the three paradigms commonly used to incorporate context into a CARS (i.e., pre-filtering, contextual modeling and post-filtering).

It is important to stress out that we focus on the ways in which context can be incorporated and exploited in the recommendation process. Even though on the diagrams we illustrate that process as a whole, we are mainly interested in showing the specific points where the reviewed papers (their references are placed accordingly on the diagrams) decided to adapt the recommendation process to exploit context. While we rely on some of the reviewed papers to illustrate the characterization and findings presented in this section, a more detailed retelling on how each paper implements their system and incorporates context can be found on Tables 2 and 3.

5.1. Content-based approaches

We found 15 papers associated with content-based CARS. Table 2 summarizes the characterization of these papers (cf. Col-

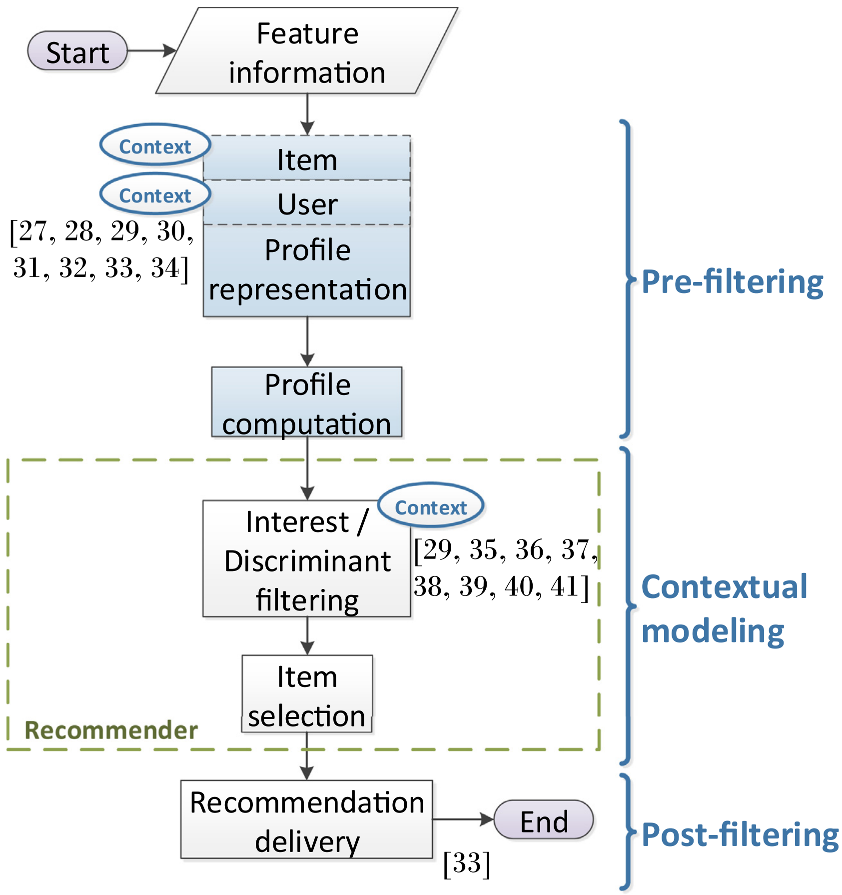

Fig. 1. Process followed by content-based CARS.

umn *Appr.*), which is driven by the process depicted in Fig. 1. Columns *Profile representation, Profile computation*, and *Discr. filter* indicate the techniques implemented by the studied systems to realize the main phases of the content-based recommendation process. Column *Paradigm* denotes the strategy used to incorporate context information: pre-filtering, contextual modeling, post-filtering. Column *Context Types* corresponds to the context categories exploited by the corresponding approach. Column *Domains* lists the application domains for which the RS was proposed. The last column explains the means used by the studied CARS to exploit context information.

5.1.1. The beginning of the process

The process implemented by content-based CARS (cf. Fig. 1) begins with the identification of the features in the available data that will define the common dimensional space used to describe item characteristics and user preferences (cf. *User profile definition* and *Item profile definition* in Fig. 1).

Pre-filtering strategies are applicable through the incorporation of contextual factors in the definition of item and/or user profiles. These strategies reduce significantly the search space for the discriminant filter by initially discarding a part of the information available. However, they require the inclusion of redundant user or item profiles for different contextual situations.

All content-based reviewed papers defined the features used as the basis for their recommendation, but only about half of them included contextual information as features. CARS proposed in [27–34] exploit context using a pre-filtering strategy to generate different contextual *user* profiles for the same user, with different preferences for different situations (see Table 2 for more details regarding the four papers that apply pre-filtering as the paradigm to incorporate context). For instance, [29] proposes a movie CARS where contextual variables of different types such as time (weekday, weekend), location (theater, home), and social context (companion, friends, family) are taken into account to consider or ignore past user ratings, by building several context-aware (micro) profiles that are used to generate context-aware recommendations. As a result, the same user can have different profiles.

None of the surveyed papers associate contextual information with items. We assume that this is because it is easier to think in terms of contextual user profiles than in terms of contextual item profiles, probably because user preferences naturally vary according to context situations. Still, it is completely possible to have different *item* profiles for different situations. Nevertheless, since very often the number of items is many times larger than the number of users, it would mean increasing the complexity of the recommendation process given that a considerably larger number of items must be handled by the system.

5.1.2. The core of the process

The next phase is the core of the recommendation process. In general, a discriminant filter working as a utility function between user and item profiles is responsible for generating a recommendation score from the item and user vectors. This can be done through several strategies: i) by applying some similarity measure such as *Cosine Similarity* (since items and users are represented on the same dimensional feature space, it is possible to compute the distances or similarities between them, with the goal of selecting the items closer to the user's preferences [27–31,34–36]; ii) by obtaining a given classification score by applying a supervised learning technique [37–39]; or iii) by applying a heuristic approach (context information can be considered into a discriminant filter,

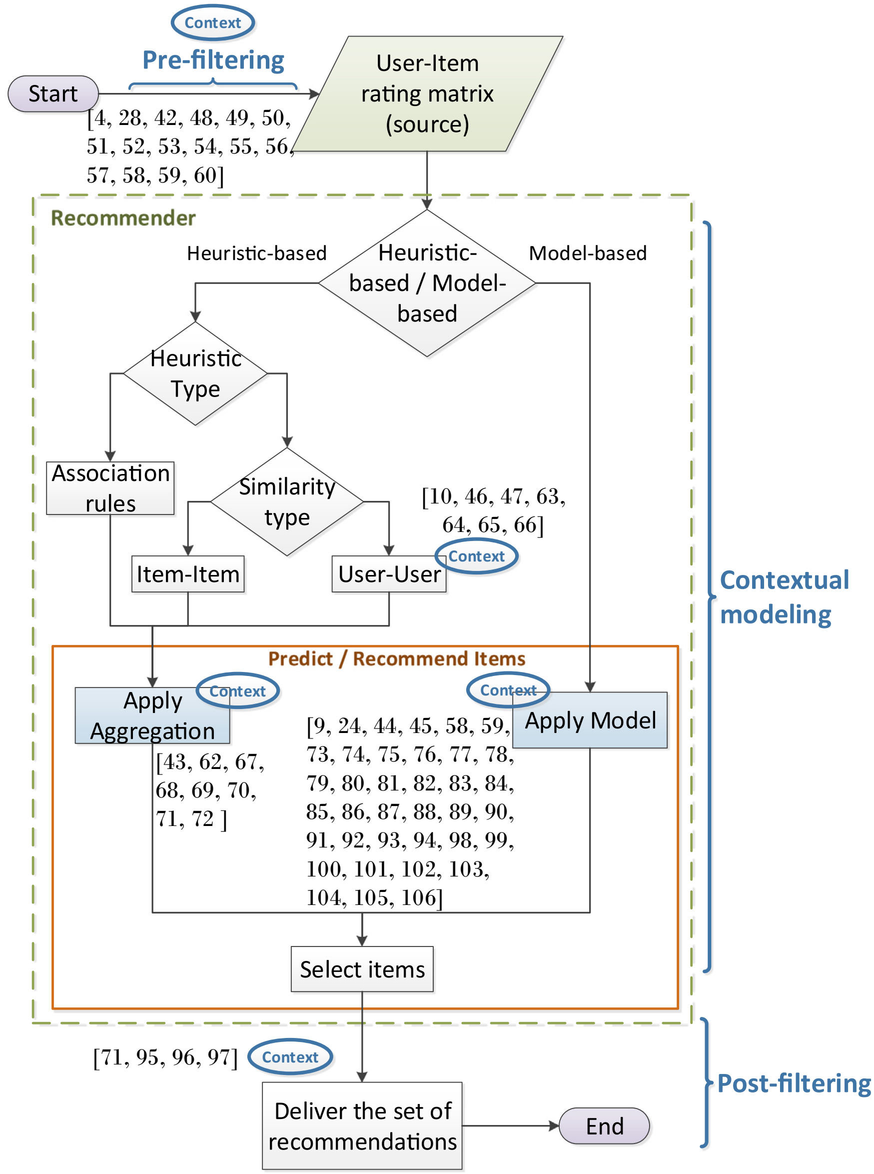

Fig. 2. Process followed by collaborative filtering CARS.

not as additional profile dimensions, but as an integral part of the function definition [32,33,35,40,41]. Either way, the recommender engine will associate a numeric value to each item, order the items accordingly, and select the ones that appear at the top or that surpass a specified threshold.

At this stage of the process, contextual information can be incorporated by influencing the similarity or distance between items and users. For example, Shin et al. [36] proposes a music recommendation system that incorporates the time at which users accessed different items (songs) in order to provide more relevant recommendations. In their system, users are described by a vector of their correlations to the considered time-related contexts (dawn, morning, monday, tuesday, spring, christmas), items are described as a vector of their correlations to the domain features (e.g., band, genre) as computed by a TF-IDF measure, and the historical accesses to items by users are kept as a collection of pairs of vectors as previously described. To perform a recommendation, the cosine similarity measure is applied to the user's current context and the historical accesses, the similarity of the historical accesses to the items is computed, and an aggregation of both measures allows the scoring of every available songs, so that the top five songs are presented to the user.

5.1.3. The end of the process

Finally, the selected recommendations are organized and delivered to the user. Post-filtering strategies apply at this stage to eliminate the recommendations that are irrelevant to the user's current context. We found that only the RS presented in [33] applied this paradigm to filter out movie recommendations that did not correspond to the current time and location.

5.1.4. Findings

Regarding the paradigm used to incorporate context into the RS (cf. Section 2.2.2), findings show that content-based approaches use contextual modeling as much as pre-filtering (one of those combining both strategies); both paradigms being followed by 53% of the papers. Only one of the studied content-based CARS [33] incorporates context information using post-filtering, combined with pre-filtering. We hypothesize that this may be in part because post-filtering strategies may result in wasting time and compu-

Table 2
Characterization of content-based approaches.

Appr.	Profile representation	Profile computation	Discr. filter	Paradigm	Context types	Domains	Means to incorporate context
[28]	Item features	Case based reasoning (CBR)	Cosine similarity	Pre-filtering	Time, Location, Activity, Artificial (environment)	Movies, Music, News	Generates a contextual user profile by revising the user's consumption behavior. Then, it uses cosine correlation to measure the similarity between the user contextual profile and the item profile.
[37]	Item features	Heuristic approach	Decision tree algorithm	Cont. Model.	Activity, Human (age, gender)	Indoor Shopping, Points of interest	Proposes a framework where the relationship between user profiles and services under the same context situation are analyzed to infer user preference rules, using the decision tree algorithm.
[27]	Item features structured by a reference ontology	Heuristic approach	Cosine similarity	Pre-filtering	Activity, Time, Location	Movies	Tracks user browsing behavior, and understands user preferences in each particular context. Then, it performs recommendations by means of an aggregation agent that selects the top *N* items with the highest inferred values.
[30]	Tag-based features	Heuristic approach	Cosine similarity	Pre-filtering	Time, Location, Activity, Natural (Weather)	Points of interest	Uses a relational Markov network to match the features of Points of Interest (POI) with the current context. POI's features (e.g. outdoor seating, waiter service, dinner) are taken as the inputs to a neural network used to classify the appropriate level of interest (5 categories) of the user for the POI, under the given context situation. The resulting vector that characterizes the POI is then compared to the user vector using cosine similarity.
[29]	Item features	Heuristic approach	Cosine similarity	Pre-filtering, Cont. Model.	Time, Social, Location	Movies	Pre-filtering: Splits user ratings according to the contextual situation in which the preference is expressed, then builds several context-aware (micro) profiles used to infer preferences for new products. Contextual Modeling: Considers context as a weighting factor that influences the recommendation score of a user for a certain item. It combines the non-contextual vector space representation of user preferences with a vector space representation of context, which is built using the pre-filtering approach.
[35]	Latent semantic features	Term frequency inverse document frequency (TF-IDF)	Cosine similarity	Cont. Model.	Location	News	User is defined by the articles read in the past along with his/her location. The system seeks to rank a set of articles that satisfy the geographical location of the user. The preference score is determined by a cosine function ($f(a, l)$) that measures the appropriateness of each article a to a location l.
[38]	Item features	Heuristic approach	Joint probabilistic distribution	Cont. Model.	Activity	Music	Formulates the context-aware recommendation of songs as a two-step process: i) infers the user's current situation category given some contextual features sensed from a mobile phone, and ii) finds a song that matches the given situation. The first part computes a probability distribution using the Bayes' rule. The second part computes a prior probability that captures the history of user preferences.

(*continued on next page*)

Table 2 (*continued*)

Appr.	Profile representation	Profile computation	Discr. filter	Paradigm	Context types	Domains	Means to incorporate context
[40]	Item features	Heuristic approach	Heuristic approach	Cont. Model.	Location	Indoor shopping	Focuses on mobile recommender systems for assisting indoor shopping by considering location-context. User preferences are calculated through a heuristic approach that integrates three factors: i) time spent in a brand store, ii) frequency of visits to the store, and ii) the matching between the special offers or promotional activities done in the brand store and the user's preferences.
[36]	Item features	Term frequency inverse document frequency (TF-IDF)	Cosine similarity	Cont. Model.	Time	Music	Context refers to the time at which the user listens to a song. The approach predicts user preferences by: i) computing the similarity between the user's current and historical contexts, ii) computing the correlation between historical context and an item, and iii) deriving the expected preference by multiplying measures obtained in i) and ii).
[39]	Latent semantic features	Heuristic approach	Joint probabilistic distribution	Cont. Model.	Activity, Location	Music	Implements a recommendation model where a set of latent topics is used to associate music content with a user's music preferences under certain location. It is based on the joint probability distribution of user, place, song and lyrics. The latent topics are the intrinsic factors that explain why users prefer certain pieces of music in a particular location and during a specific time period.
[31]	Item features	TF-IDF	Cosine similarity, Jacard similarity	Pre-filtering	Human (user-interest)	Web services	Infers user preferences from the description of the web services that have been accessed by the user.
[32]	Item features	Heuristic	Heuristic	Pre-filtering	Social (followers)	Multimedia	Utilizes Social context (followers) as the basis to decide on user-similarity.
[41]	Item features	Heuristic	Heuristic	Contextual modeling	Location	Points of Interest	Considers context as a weighting factor that influences the recommendation score of a user for a certain item.
[33]	Item property	Heuristic	Heuristic	Pre-Filt, Post-Filt.	Location, Time	Movies	Recommends items with a composite structure (movie theater + movie + showtime). This approach first computes a similarity metric that concerns to the relation between the composite item (theater, movie, showtime) -Pre-Filtering. Then, this similarity measure is incorporated into the discriminant filter -Post-Filtering.
[34]	Item feature	Heuristic	Euclidian Distance	Pre-filtering	Activity	General application	Utilizes a sequential patterns method to find rules from data records on users' smart-phones. Then, by detecting and matching the user's current situation to the rules, which consider his current context and the events in which he has participated, the system determines the most suitable rules for making just-in-time recommendations.

Table 3
Characterization of collaborative filtering approaches.

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[4]	**Heuristic-based, User sim:** Cosine similarity, **Aggr.:** Sum of products	Pre-filt.	Time, Social, Location	Movies	
[48]	**Heuristic-based, User sim:** Cosine similarity, **Aggr.:** Sum of products	Pre-filt.		Music	
[49]	**Heuristic-based, User sim:** Cosine similarity, **Aggr.:** Top N (most important users)	Pre-filt.		Movies	
[50,51]	**Heuristic-based, User sim:** Pearson correlation, **Aggr.:** Sum of products	Pre-filt.	Time, Location	Movies	Filter information according to the current context. A rating is computed for the given user and item, as an aggregation of the ratings of other similar users.
[52]	**Heuristic-based, User sim:** Pearson correlation, **Aggr.:**Sum of products	Pre-filt.	Location	Points of interest	
[28]	**Heuristic-based, User sim:** Pearson correlation, **Aggr.:**Sum of products	Pre-filt.	Location, Activity, Artificial (environment)	Movies, Music, News	
[53]	**Heuristic-based, User & Item sim:** K-medians, **Aggr.:** Maximum **Heuristic-based, User & Item sim:** Graph theory, **Aggr.:** Maximum	Pre-filt.	Time	E-retailing	
[54]	**Heuristic-based, User sim:** Graph Theory, **Aggr.:** Maximum	Pre-filt.	Location, Social	Points of Interest	
[60]	**Heuristic-based, User & Item sim:** Cosine similarity, **Aggr.:** Sum of products	Pre-filt.	Time	Movies, Music	The authors propose a neighbor-based collaborative filtering approach. A similarity measure over human and time contextual factors provides the basis for estimating the neighborhood of both users and items that will be considered in the recommendation process.
[42]	**Model-based, Tech.:** Matrix Fact.	Pre-filt.	Time, Social	Movies	Splits items that have been rated under different context situations. This split is performed only if there is statistical evidence that under these context situations users rate items differently.
[55]	**Model-based, Tech.:** Markov Chains	Pre-filt.	Time, Activity	General application	Processes user historical logs to extract contextual features such as day, time range, and location. Then, it identifies common preferences under different contextual conditions. Finally, it makes recommendations based on distributions of user preferences.
[56]	**Heuristic-based, User Sim:** Graph theory, **Aggr:** Sum of products	Pre-Filt.	Social	Music, E-retailing	Examines the context-aware recommendation as a search problem in the contextual graph.

(continued on next page)

Table 3 (*continued*)

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[57]	**Heuristic-based, User sim:** Pearson correlation, **Aggr:** Sum of products	Pre-Filt.	Different types	General Application	Context information associated with users is exploited to infer individual user profiles and from these, the profiles of the groups.
[58]	**Model-based, Tech:** Matrix Fact.	Pre-Filt., Cont. Model	Location, Time	Hotels & Tourism	The original user-item rating matrix is divided into sub-matrices according to the temporal states. Then, each sub-matrix is factorized by considering location characteristics.
[59]	**Model based:, Tech:** Matrix Fact.	Pre-Filt., Cont. Model.	Location	Web services	Users and services are clustered into groups according to their location. These are then characterized according to their particular QoS features into a local user-service matrix. There is also a global user-service matrix where location is not considered. Matrix factorization is performed on the local and global matrices in a step-wise hierarchical linear process
[46]	**Heuristic-based, User sim:** Pearson correlation, **Aggr.:** Sum of products	Cont. Model.	Location, Time	Points of interest	Adopts an adjusted Pearson coefficient that computes similarities between users in different contexts. In order to do so, the approach defines a context similarity matrix that includes the coefficient between two users' current contexts for using an item. This coefficient is then incorporated into the aggregation function that computes the missing ratings.
[62]	**Heuristic-based, User sim:** Pearson correlation, **Aggr.:** Maximum	Cont. Model.	Location, Time	Points of interest	Recommends restaurants by computing the approximate time in reaching it, and considering distance, speed and road conditions. This approximation is included into the aggregation function.
[43]	**Heuristic-based, Item sim:** Cosine similarity, **Aggr.:** Maximum **Heuristic-based, As. Rules:** Apriori, **Aggr.:** Maximum	Cont. Model.	Location, Time	Points of interest, Music	Transforms the initial user-item matrix by integrating contextual factors as virtual items.
[67]	**Heuristic-based, Item sim:** Pearson correlation / Cosine Similarity, **Aggr.:** Sum of products	Cont. Model.	Human (mood), Time	E-learning	
[10]	**Heuristic-based, User sim:** Cosine similarity, **Aggr.:** Sum of products	Cont. Model.	Time, Human (intent of purchase: Personal-work, Gift Partner, Friend, Parent)	E-retailing	Considers virtual users under different contexts and finds neighbors of contextually similar users to infer recommendations.
[47]	**Heuristic-based, User sim:** Jaccard Similarity, **Aggr.:** Sum of products	Cont. Model.	Location, Time	Points of interest	Modifies the Jaccard similarity measure to incorporate context.
[63]	**Heuristic-based, User sim:** Pearson Coefficient, **Aggr.:** Sum of products	Cont. Model.	Social	General application	Integrates the strength of the relationships between telecom users into the similarity measure. This strength is modeled taking into account context information associated with phone calls such as duration, time of day and day of the week.
[9]	**Model-based, User sim:** Graph theory, **Tech.:** Matrix Factorization	Cont. Model.	Social	Movies	Combines the user-item rating matrix with user-user social contextual information from a trust network to generate a modified rating matrix. This last matrix is then factorized.

(*continued on next page*)

Table 3 (*continued*)

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[84]	**Model-based, Tech.:** Matrix Factorization	Cont. Model.	Social	General application	
[85]	**Model-based, Tech.:** Matrix Factorization	Cont. Model.	Time	Movies	Consider context information to add biases on users and items into the recommendation model. Rating values are then influenced by context changes.
[45]			Time	Points of interest	
[86]			Time, Location	Movies	
[87]			Location, Time, Activity	Points of interest	
[91]			Social	Books, Music, Movies	
[92]			Social	General application	
[88]	**Model-based, Tech.:** Matrix Factorization	Cont. Model.	Time, Human (Hunger level, mood)	Food, Movies	Clusters items into groups according to the context of their consumption and treats them as virtual items associated with users in a new matrix that is then factorized. Missing ratings are inferred taken into account contextual information.
[89]	**Model-based, Tech.:** Matrix Factorization	Cont. Model.	Social	Books, Music, Movies	Considers context information to add biases on users and items into the recommendation model. Through matrix factorization, it creates a common latent factor space for users and items. In this representation space, users and items are clustered independently, so that they can then be brought back to a user-item rating matrix, where missing ratings can be inferred for groups of users.
[90]	**Model-based, Tech.:** Matrix Factorization	Cont. Model.	Human (age, gender)	Movies	Constructs several prediction models based on matrix factorization. Each model is then refined by taking into account the predictions from other models. Context information is considered to add biases on users and items into the recommendation model. Rating values are then influenced by context changes.
[73,74]	**Model-based, Tech.:** Tensor Factorization	Cont. Model.	Time, Human, Social	Movies	Perform context-aware recommendations using tensor factorization, which considers the latent features of users and items, and the interaction of the user with an item under a given context. The latent feature of users, items and context types are stored in three matrices. Thus, the inference of preferences is computed as the inner product of the latent feature vectors of the matrices.
[75]			Time	Movies	
[76]			Location, Activity	E-retailing	
[77]			Social, Time	Movies, Food	
[78]			Social, Time	E-retailing, Movies	
[79]			Human (hunger level), Time, Location	Food	
[80]			Social, Time	E-retailing, Movies	
[81,82]	**Model-based, Tech.:** Support Vector Machine (SVD)	Cont. Model.	Time, Social, Natural (weather), Location	Points of interest	Apply SVD to the ratings as represented in a user-item-context space to discriminate between recommended and not recommend items.
[83]	**Model-based, Tech.:** Support Vector Machine (SVD)	Cont. Model.	Location	Points of interest	
[94]	**Model-based, Tech.:** Bayesian Model	Cont. Model.	Time, Location, Human (mood)	Movies	By adopting a binary particle-swarm optimization technique, identifies the relevant contextual factors for user and item classes, and incorporates them into a latent probabilistic model.

(*continued on next page*)

Table 3 (*continued*)

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[24]	**Model-based, Tech.:** Naïve Bayes	Cont. Model.	Time	Movies	Identifies which members of a household made some specific unidentified ratings of movies by considering time-context conditions such as hour of the day, day of the week and date of rating, as well as number of ratings given by a user. To do this, it analyses temporal trends using probability models.
[98]	**Model-based, Tech.:** Sparse Linear Method	Cont. Model.	Time, Location, Social	Movies	Models the contextual rating deviations of items, by assuming that there is a rating deviation for each $<$ item, context condition $>$ pair. This deviation is represented in a matrix, where each row represents an item, and each column represents an individual contextual condition. Then, the ranking score is estimated by an aggregation of user ratings on other items in the same context.
[99]	**Model-based, Tech.:** Linear Regression	Cont. Model.	Social, Time	Hotels & Tourism	Predicts user preferences using a linear regression model, which includes a value that represents the user context preference. This value can be computed by means of three different probabilistic methods: i) mutual information based method, ii) information gain based method, and iii) chi-square statistic based method.
[100]	**Model-based, Tech:** Matrix Fact.	Cont. Model.	Location, Social	Points of interest, Hotels & Tourism	Location of venues and user social network information are integrated into the matrix factorization model.
[64]	**Heuristic-based, Item & User sim:**Pearson correlation, **Aggr:** Weighted ad-hoc	Cont. Model.	Social	Web services	The level of trust among users (social context) is included in the weighted aggregation
[65]	**Heuristic-based, User sim:** Ad hoc, **Aggr:** Ad hoc	Cont. Model.	Location, Social	Points of interest, Hotels & Tourism	The social (relationships) and location context of the user is integrated into the process to measure the similarity between users.
[44]	**Model-based, Tech:** Matrix Fact.	Cont. Model.	Time, Activity, Location, Artificial	General application	Context-aware preferences as dimensions of the matrix
[93]	**Model-based, Tech:** Matrix Fact.	Cont. Model	Social	E-retailing	Social context is considered in order to define groups of users with particular hyper-parameters used by the matrix factorization model
[68]	**Heuristic-based, User sim:** Cosine similarity, **Aggr:** Sum of Products	Cont. Model	Time, Social	General application	The prediction of user's preference is affected by the user-similiraty , which is computed by considering the context (i.e, the social taggins)
[69]	**Heuristic-based, User sim:** Pearson correlation, **Aggr:** Sum of Products	Cont. Model	Time	Movies	Adds a time dimension to the original input data. It is defined in a new table which shows item ratings for an active user at different time-frames.
[70]	**Heuristic-based, User sim:** Cosine similarity, **Aggr:** Sum of products	Cont. Model	Time	Music	Infers user's preference by considering a context score, which is computed for each item in the recommendation list which shows the suitability of that item for the current context of the user.
[101]	**Model-based, Tech:** Random walk	Cont. Model.	Social	Social Networks	Tags from social networks are the basis for user similarity (Jaccard). Posts from users are compared by applying an ad-hoc similarity measure. A random walk algorithm is applied in order to estimate weights relating users to users in the social domain and users to items on auxiliary domains (web posts, videos, labels)

(*continued on next page*)

Table 3 (*continued*)

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[102]	**Model-based, Random walk**	Cont. Model.	Time	Web services	Making time-aware personalized QoS prediction is important for high-quality web service recommendation because their performance is highly correlated with invocation time, since service status and network conditions are continuously changing. Time is integrated into a modified Pearson correlation similarity measure (similarities between users and between web services); time is also considered when making the final QoS prediction.
[103]	**Model-based, Tech:** Matrix Fact.	Cont. Model.	Social, Time	E-retailing	Social networking features of users (demographics, user posts, groups of related users, temporal activity preferences) that also interact with an unrelated e-commerce site can be transformed into latent factors that can be used for product recommendation, particularly for unknown new users of the e-commerce site.
[104]	**Model-based, Tech:** Matrix Fact.	Cont. Model.	Social	Retailing	The authors propose Social Poisson Factorization (SPF) probabilistic model that incorporates social network information into a traditional factorization method, assuming that each user's clicks are driven by their latent preferences for items and the latent influence of their friends (modeled as conditional probabilities). SPF also allows for generating explanations of recommendations based on the social relationships of users.
[105]	**Model-based, Tech:** Matrix Fact.	Cont. Model.	Social	Retailing	A probability based matrix factorization is proposed, taking into account trust relationships in a social network in the item recommendation process for retailing purposes. Users and items are then clustered using a Gaussian Mixture Model to enhance the recommendation performance.
[106]	**Model-based, Tech:** Matrix Fact.	Cont. Model.	Location, Social	Points of interest	The authors propose a probabilistic matrix factorization method which considers contextual information taken from a location-based social network, where each point of interest is described using a topic model, geographical and social correlations.
[66]	**Heuristic-based, User sim:** Jaccard similarity, **Aggr:** Heuristic graph based.	Cont. Model.	Social	Social Networks	The social features of folksonomies are used to provide a user with recommendations of similar users and resources. User profiles consider social contexts, by incorporating information of actions performed by the user on neighboring users' tags, and of other neighboring users on the user's tags. User neighborhoods are defined based on the social network friend relationships according to a specified length of the minimum path linking two users.
[71]	**Heuristic-based, User-Sim:**Ad-hoc, **Aggr:** Sum of products	Cont. Model.	Social	E-retailing	Adopts an ad-hoc similarity measure that computes similarities between users in different social context. This measure is then incorporated into the aggregation function that computes the missing ratings
[72]	**Heuristic-based, User sim:** Graph theory, **Aggr.:** Probability	Cont. Model., Post-Filt.	Time, Location, Natural (weather), Social	Movies, Hotels & Tourism	t Proposes a graph-based contextual model framework. It examines the context-aware recommendation as a search problem in the contextual graph. It also includes a probabilistic-based post-filtering strategy to improve the recommendation results giving contextual factors.
[97]	**Model-based, Tech:** Matrix Fact.	Post-Filt.	Time	Movies	The authors propose two successive SVD matrix factorizations to further refine the latent factors for users and items independently, while using time context to filter out unfit items.

(*continued on next page*)

Table 3 (*continued*)

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[95]	**Heuristic-based, User sim:** Cosine Similarity, **Aggr.:** Sum of products	Post-Filt.	Location, Time, Natural (weather)	Hotels & Tourism	Keeps track of contextual features of past user travels to each location. Context aware recommendations are inferred by finding the most similar users, calculating a score for each location, and filtering locations that do not meet contextual conditions.
[96]	**Heuristic-based, Users sim:** Pearson correlation, **Aggr.:** Sum of products	Post-Filt.	Time, Location	Points of interest	Adjusts inferred ratings to deliver contextual recommendations.

tational resources, since the obtained recommendations may become useless after evaluating them with respect to the current context of the user, which is taken into account only at the end of the process. Indeed, pre-filtering approaches provide more benefits in what respects to computational complexity, and contextual-modeling solutions have proven to be more effective for the accuracy of recommendations [4].

With respect to the types of contextual information commonly used in the reviewed systems, and their application domains, we found that time context is commonly used in application domains such as movies and news; location context, in domains associated with movies, music and points of interest; activity context in domains related to movies, music and points of interest; social context in multimedia applications and human context in web services recommendations. It is of particular interest that none of the reviewed content-based CARS target the e-retailing domain, an otherwise popular application domain in traditional RS.

5.2. Collaborative filtering approaches

Fig. 2 depicts the general process followed by collaborative filtering CARS. Based on this process, we characterized the 69 collaborative filtering CARS studied in our SLR. This characterization is summarized in Table 3. Column *Recommendation strategy* presents the techniques implemented by the studied approaches, which can follow different paths of the recommendation process, as explained later in this section. As in the characterization of content-based CARS (cf. Table 2), the characterization of collaborative filtering CARS includes the paradigm used to incorporate context into the system (cf. column *Paradigm*), the types of context information exploited by the studied approaches (cf. column *Context Types*), the application domain (cf. column *Domains*) and the mechanisms used to exploit context (cf. column *Means to incorporate context*).

5.2.1. The beginning of the process

The input of the collaborative filtering process is a user-item rating matrix, where usually rows represent users, and columns represent items. This matrix can include additional dimensions to represent contextual information in the form of synthetic columns or rows, as in the case of the systems presented in [4,42–44]. For example, Baltrunas and Ricci [42] extend the user-item rating matrix into a user-item-context matrix, where contextual information consists in categorical tags (e.g. sunny, cloudy, raining) associated with a given rating.

Depending on the application domain, this matrix can be either obtained directly from the interactions of users with items (e.g., by capturing media accesses instantly [28,45–47], or inferred from historical interactions stored in transactional databases (e.g., by analyzing event logs of previous accesses to the recommended items [10,48]. This matrix can be very sparse and its processing can be computationally challenging when the number of users and items is considerable (several hundreds of thousands).

At the beginning of the process, *pre-filtering* strategies generate different contextual user-item rating matrices, independent of each other. On the one hand, pre-filtering strategies reduce computational complexity since only a portion of the rating matrix is considered; on the other hand, they imply an extra effort in the acquisition of information, since ratings must be generated for every contextual situation that remains relevant after applying the filter.

We identified 16 papers reporting on the application of context-based pre-filtering strategies to generate recommendations [4,28,42,48–60]. Pre-filtering is a simple strategy that discards a large part of the data to be analyzed, according to the user's current context. An instance is the process followed by the CARS proposed by Lee et al. [48], in which the authors analyze the access logs to songs, and extract context from the timestamps. Then, they define fuzzy membership functions to fuzzy sets for different contextual variables such as season, time of day, or day of the week, in such a way that the same song recommended at different moments is not considered to be the same item. Another example of collaborative-filtering pre-filtering CARS is the one proposed by Baltrunas and Ricci [42]: if a statistical test shows that context affects the consumption of an item, they split the item into several synthetic items according to the context situation. For instance, a movie could be split into the same movie associated with winter time, and another one associated with summer time.

5.2.2. The core of the process

To perform the actual recommendation, we identified that most systems apply one of two types of collaborative filtering approaches: *heuristic-based* and *model-based* methods. We found no relationships between any of these methods and particular application domains.

Heuristic-based methods. In the studied systems, heuristic-based approaches are realized through association rules, or the analysis of similarities between users or items. The Apriori algorithm [61] is a common technique for association rule learning. First, it identifies the frequent individual items in the database. Then, it extends them to larger itemsets as long as those appear often enough in the database. Finally, these itemsets are used to determine association rules that allow the discovery of hidden relationships in the data, based on the conditional probability existing between itemsets. The association rules approach is mainly applied to transactional data. However, it can also be applied to the user-item rating matrix, by considering each user row as a single transaction.

An interesting finding of our SLR is that despite approaches such as the one reported in [62] mine association rules, none of the studied systems exploit this technique to incorporate context. A reason for this could be that it would imply extra efforts to acquire the information required to generate a more comprehensive rating matrix, such that the extracted rules are meaningful enough in terms of support, and include context in rule antecedents.

Heuristic-based approaches based on similarity analysis consist in determining the distance between users or items. Each user can be seen as a vector in a feature space with an independent dimension associated with each item (and vice-versa). In general, these distances are determined using neighbourhood or clustering-based methods.

These methods work in two ways. The first one, user-user collaborative filtering, consists in inferring user preferences by determining the group of users that are more similar to the target user, and aggregating the items that are most popular among the members of the user group. The second one, item-item collaborative filtering, consists in determining the similarity among items rated by similar users. In either case, the method requires the computation of the distances between users or items, which can be computationally demanding when dealing with a considerable number of users or items.

Seven of the heuristic-based approaches included in this SLR incorporate context through user-user similarity matching; for instance, the approaches presented in [10,46,47,63–66] incorporate context to the analysis of user-user similarities (more details can be found on Table 3). On the other hand, none of the heuristic-based approaches use item-item collaborative filtering to incorporate context. As discussed previously, we hypothesize that this is because it results more natural to associate context with users than with items. Nevertheless, in some application domains (e.g., products that are mainly consumed in a particular time of the day), context can be effectively associated with items, in which case an item-item collaborative filtering method that incorporates context would be an appropriate strategy.

Continuing with the recommendation process based on heuristic methods, the information obtained from applying the selected method is aggregated to rank the items to be recommended. Eight of the reviewed papers correspond to collaborative filtering RS that incorporate context as additional factors in the aggregation function. In particular, by using a maximization function [43,62], a sum of products [67–71], and probabilities [72]. For instance, Khalid et al. [62] combine the approximated time required to reach a restaurant, the road speed conditions and the distance from the user into a defined metric. Then, the restaurant maximizing this metric is recommended to the user.

Model-based methods. Model-based approaches rely mostly on latent factor models applied to the user-item rating matrix. As we have said before, we can interpret this matrix as either a multi-dimensional representation space where each user is a vector with each item as a dimension, or a multi-dimensional representation space where each item is a vector with each user as a dimension.

The idea of latent factors RS is to obtain a single multi-dimensional space where both users and items can be represented, side by side, through matrix decomposition techniques. In this latent space (usually of smaller dimensionality than the user-item rating space), it is then possible to compute similarities and distances between users and users, users and items, and items and items.

We identified that some systems introduce contextual factors as additional dimensions of the original matrix (e.g., [44,73–83]), while some other include contextual information as additive biases on users and items, to affect the calculation of missing ratings (e.g., [9,45,84–92]). An example of the first group is presented in [73], where the authors perform contextual recommendations using tensor factorization. This technique stores the latent feature of users, items and context types in three different matrices. Then, ratings are calculated as the inner product of the latent feature vectors of the given matrices. As a case of the second group, we can consider the RS presented in [85], which performs context-aware recommendations by incorporating temporal changes into the matrix factorization technique. In particular, this approach seeks to capture past temporal patterns over products and items to predict future behaviour, and thus infer preferences. A particular case is the approach presented by Liu et al. [93], which incorporates social context from a social network into the recommendation model by considering that users belonging to different social groups should have different hyperparameters to be used during the matrix factorization process.

It is important to note that despite the collaborative filtering recommendation process indicates that heuristic-based and model-based techniques are not commonly used together, the authors of papers [9] and [88] propose CARS where model-based and heuristic-based techniques are combined. For instance, in [9] user interactions are represented in the form of a social network graph, where each node represents a user, and arc weights correspond to the trust existing between users represented by adjacent nodes (i.e., social context). This approach uses a heuristic-based technique (i.e., graph theory) along with a model-based method (i.e., matrix factorization).

We found a few papers reporting on the application of other approaches. In particular, machine learning techniques, where context information is usually incorporated by implementing probabilistic models such as the Bayesian model [24,94], or the usage of classifiers such as support vector machines [81–83].

5.2.3. The end of the process

Similarly to content-based CARS, at the end of the process a contextual filter can be applied to the resulting recommendations to eliminate those items that are irrelevant to the current context. We found four papers reporting on the incorporation of context as a post-filtering strategy to ignore [95], filter [72,96,97], or adjust [72,96] the inferred recommendations.

For example, the systems reported in [72,96] ignore context until a traditional collaborative filtering algorithm produces restaurant recommendations, which are then adjusted to the user's current context.

5.2.4. Findings

The information summarized in Table 3 suggests a correlation between the strategy used to generate recommendations and the paradigm used to incorporate context into the recommendation process of collaborative-filtering CARS.

In general, model-based approaches incorporate context using contextual modeling. This can be explained by the fact that models provide a more natural way to capture interactions between users, items and context. We also found papers reporting on the combination of model-based methods and pre-filtering strategies [42,55,58], or even the combination of the three strategies including contextual modelling [59]. However, these combinations may be risky since a pre-filtering strategy can cause loss of valuable information thus affecting accuracy [4].

Heuristic-based approaches are almost evenly distributed between the application of pre-filtering and contextual modeling strategies to realize context-aware recommendations. Regarding the application of pre-filtering, data sources are usually partitioned by context factors to improve data uniformity, which leads to stronger user/item similarities, as well as better confidence and support measures for association rules, thus improving the relevance of recommendations. In the case of contextual modeling, context information modifies how similarity is calculated.

With respect to contextual information, we found that most of the studied collaborative filtering systems have time, social, and location as the predominant factors. Furthermore, the application domains to which the surveyed systems are commonly applied are

Table 4
Characterization of hybrid approaches.

Appr.	Techniques	Paradigm	Context types	Domains	Means to incorporate context
[107]	**Content-based Profile representation** Item features	Pre-filt.	Time, Location	Movies, Music	Associate ratings with content-based attributes used to describe both user preferences and item features, and with the contextual factors gathered from the user experience (e.g., time of the day). Over the resulting vector space, the authors propose the application of several types of machine learning classification models.
	Collaborative filtering Model-based, Tech.: Naïve Bayes, Random forest, Multilayer Perceptron, and Support Vector Machine	Cont. Model.			
[108]	**Collaborative filtering User sim:** K-means	Pre-filt.	Location	Music, Points of interest	Takes into account user demographics: the geographical distance between the user and the event, and the subsequent time that it would take the user to arrive. It segments users into clusters, with every user having a probability of belonging to every cluster, and with each cluster having a probability distribution of liking every item. A discriminant filter evaluates the utility of the item for the user, considering a particular context.
	Content-based Profile representation Item features, **Discr. filter** Heuristic	Cont. Model.			
[28]	**Collaborative filtering User sim:**Pearson correlation, **Heuristic-based** Sum of products	Pre-filt.	Time, Location, Activity, Artificial (environment)	Movies, Music, News	Performs contextual recommendations by combining a discriminant filter with an a aggregation of the ratings of similar users. A similarity measure between users takes into account their contextual profile.
	Content-based Profile representation Item features, **Discr. filter** Cosine similarity	Pre-filt.			
[109]	**Content-based Profile representation:** Item features	Cont. Model.	Social	Web Services	Identifies a couple of reading “experts” whose opinions can be regarded as guidance for news recommendation to particular individuals. Further, integrates this “expert” model with the content information and collaborative filtering, and propose a hybrid recommendation framework.
	Collaborative filtering Model-based, Tech: Matrix Factorization	Cont. Model.			
[110]	**Collaborative filtering Model-based**	Cont. Model.	Social, Location, Time	Social Networks	Social context is taken into account by considering the groups to which users belong to on an events-based social network. Users and events are described by the hour at which users attend events (time), and are compared by applying cosine similarity. Geographical preference of events is modeled by obtaining a probability density per user, taking into account the densities of attended events.
	Content-based Profile representation Item features, **Profile comput.** TF-IDF **Discr. filter** Cosine similarity	Pre-filt.			

movies, restaurants, music, points of interests, social networks and e-retailing.

5.3. Hybrid approaches

Since hybrid approaches combine collaborative filtering and content-based recommendation methods in many different ways, there is not a unique abstract process that can characterize hybrid solutions the way we previously did for the non-hybrid processes depicted in Figs. 1 and 2. Table 4 presents the characterization of hybrid approaches, emphasizing on the way context is exploited.

As we found only five papers documenting hybrid RS, it is impossible to generalize their findings. Each approach follows its own strategy.

5.4. Findings in the exploitation of context information

Fig. 3 summarizes general findings related to the exploitation of context by the systems described in the surveyed articles.

Fig. 3.a presents the overall distribution of context types. According to this chart, time is the most used context factor followed by location and social information, whereas artificial is the less exploited context type followed by natural, human and activity. In the studied approaches, artificial context refers to data gathered from mobile sensors, natural context refers to weather conditions, and human context corresponds to user age, gender, mood, intent of purchase, preferences and hunger level. Only papers exploiting social context comment on the reasons why the exploited context type was selected. We hypothesize that, besides being relevant in

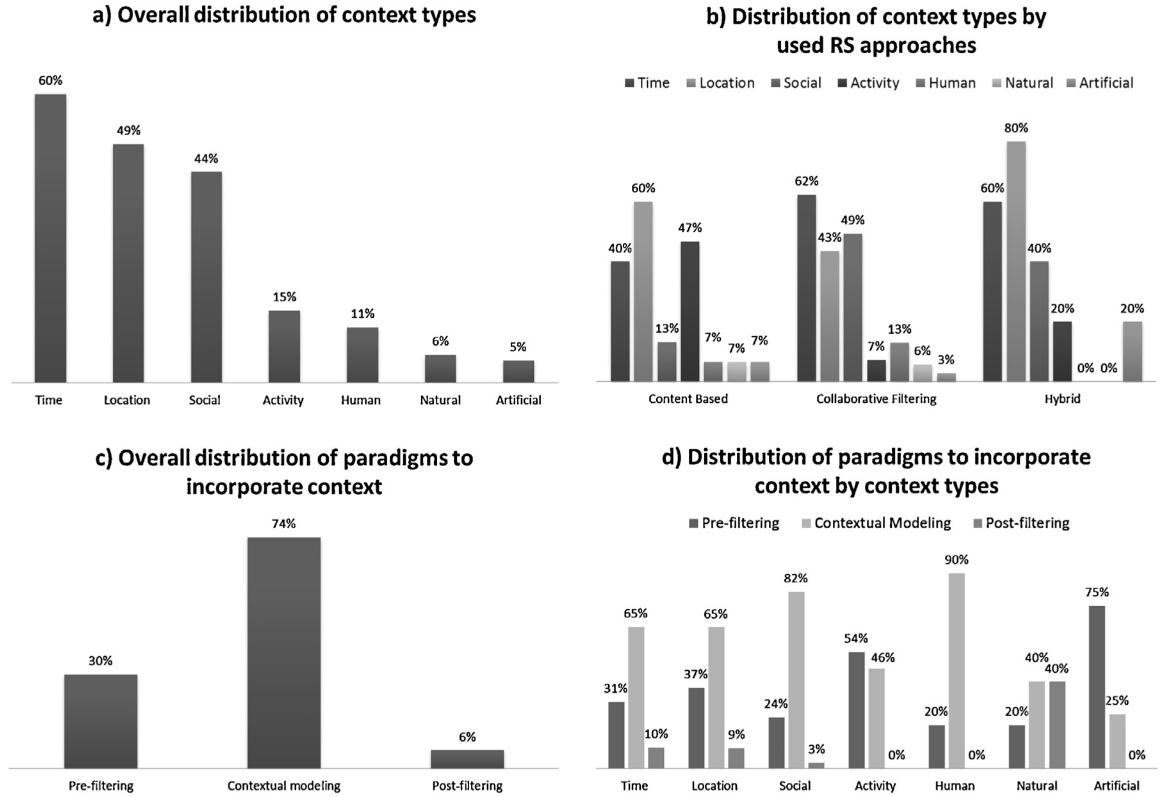

Fig. 3. Summary of findings in the exploitation of context information.

all application domains, the main reason why time is the most exploited context type is that it is the easiest one to acquire: every system records information about transaction dates, without requiring the explicit approval of users. As time context, location is also highly relevant and easy to acquire, however, its acquisition and usage, as in the case of social, activity and human context, requires user explicit approval. Artificial context does not necessarily compromise user privacy, however, its acquisition requires physical sensing infrastructures that are not always available.

Regarding the context types used with the different recommendation approaches (i.e., content-based, collaborative filtering and hybrid), it is important to highlight that (cf. Fig. 3.b): i) only 13% of the content-based RS exploit social context. This is expected since social context emerges from the relationships among users, which are less relevant in content-based approaches; ii) location and activity are the most used context types in content-based RS. A reason for this is that the relationships existing between users and items usually emerge from the place where the item is used or bought, and the activity the user is performing while using an item. In addition, items are easily associated with places and activities; iii) time is the most exploited context type in collaborative filtering systems. This is probably associated with its easy acquisition, which becomes more relevant in collaborative filtering where it is required to characterize users under similar context situations; and iv) as expected, human context is more relevant in collaborative filtering than in content-based approaches, probably because demographic information is highly used in the analysis of user similarities.

Without doubt, contextual modeling, recognized by its effectiveness in improving the performance of recommendations, is the most common paradigm used to incorporate context into RS (cf. Fig. 3.c). Post-filtering, as discussed in previous subsections, is the less used, since its application may result on the discarding of time and space wise costly recommendations. Concerning the distribution of paradigms to incorporate context by context types (cf. Fig. 3.d), it is worth pointing out that systems exploiting activity (13 papers) and artificial (4 papers) context have pre-filtering as the predominant paradigm to incorporate context.

Most popular application domains identified in the studied papers are movies (30 papers, 34%), points of interest (POI, 18 papers, 21%), music (15 papers, 17%), and e-retailing (11 papers, 13%). Other domains are hotels & tourism (6 papers, 7%), web services (5 papers, 6%), news (4 papers, 5%), food (3 papers, 3%), indoor shopping (2 papers, 2%), social networks (2 papers, 2%), and e-learning (1 paper, 1%). Seven of the studied papers do not report targeting particular application domains (general application).

Fig. 4 presents the distribution of context types by application domains. Movies is the only domain that exploits all context types, being time, social, and location the most exploited ones. As expected, location is the most common context type in the points of interest domain, followed by time. Concerning the music domain, location, time and activity are the most used context types. Activity is more predominant in this domain than in the others, probably because music genres are commonly associated with specific user activities. In the e-retailing domain, social is the predominant context type, followed by time. Here it is evident the influence of collaborative filtering as the predominant type of recommendation algorithm, particularly in this domain. Context types location, activity and human are equally exploited in e-retailing applications. Finally, it is worth also noticing that natural context, which in general refers to weather conditions, is more used in points of interest applications.

6. Characterization of validation methods

The improvement of user experience is the ultimate goal of a recommender system. In order to measure it, a series of properties, each with a set of metrics, have been proposed and used since the

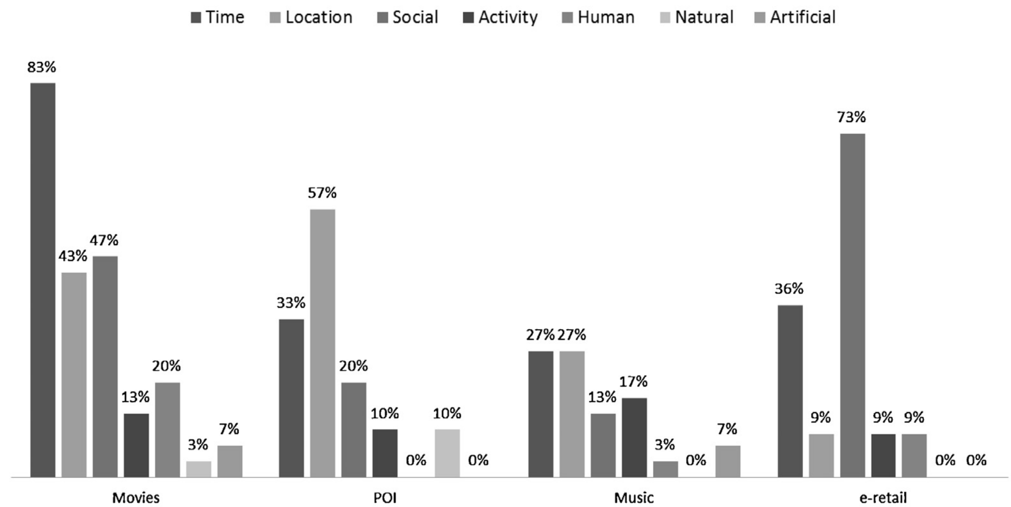

Fig. 4. Distribution of context types by most popular application domains.

Table 5
Metrics used to evaluate predictive power.

Class	Prediction metrics	#Approaches	Approaches' references
Rating prediction metrics	MAE	27	[4,9,42,46,47,50,51,59,63,69,73,77,79,80,84,88,89,91–93,96,100–103,105,106]
	RMSE	24	[9,10,47,56,71,77–80,84–94,100,101,105,106]
Usage prediction metrics	Precision	43	[4,10,24,27–29,32,33,36,39–44,47,51,53,54,56,58,60,62,63,65–67,70–72,75,80,82–84,95,97,98,101,103,106,107,109]
	Recall	28	[4,10,29,33,40,42–44,50,53,54,58,62,63,65,67,70–72,75,82,84,97,98,101,103,106,109]
	F-measure	10	[4,10,29,33,40,43,57,62,67,97]
	AUC	5	[24,60,74,103,107]
	MAP	8	[24,34,55,76,79,95,98,103]
	BR	1	[95]
Ranking metrics	NDCG or DCG	9	[30,31,35,47,57,72,74,87,110]
	Hit Ratio	6	[48,68,70,74,87,99]
	MRR or CRR	3	[99,103,104]
	Map@K	1	[101]
	Rt10	1	[35]
	None or other type reported	5	[37,45,49,52,81]

first developments in the field. These properties allow us to determine the pertinence of the recommendations being suggested. Instances of these properties are predictive power, confidence, diversity, learning rate, coverage, scalability and user evaluation [111].

In this section we summarize the properties that were considered to evaluate the recommendation systems documented in the surveyed papers, particularly predictive power, which is the most commonly used evaluation property. The first two parts of this section focus on prediction metrics and evaluation protocols identified in the studied articles. Then, we summarize other properties that were also used to assess the quality of recommendations in the studied CARS. Finally, we present the list of datasets that we identified in our survey.

6.1. Prediction metrics

Among the different metrics that can be considered to evaluate RS, the most commonly used is predictive power. This could relate to the information retrieval origins of RS. All but five of the papers we surveyed use some kind of prediction metric to assess the quality of their recommendations.

Table 5 presents the distribution of the reviewed articles with respect to prediction metrics. The first column represents the class of metric. The second column refers to the specific prediction metric techniques, grouped by their class. The third column presents the number of papers that use the metric to validate the proposal, which are listed in the last column. It is important to note that some articles may use more than one prediction metric to evaluate their approach. We borrowed the definitions of these metrics from [111] and [112].

Prediction metrics are based on different types of comparisons between the recommended items and the accessed or consumed items. As mentioned in [111], there are three classes of prediction metrics: rating prediction, usage prediction and ranking metrics (cf. first column of Table 5).

6.1.1. Rating prediction metrics

These metrics measure the correctness of the recommendations in terms of their error. The two metrics we identified in the studied articles are *root mean squared error* (RMSE) and *mean absolute error* (MAE). These metrics measure the distance between predicted and real ratings. So, lower values of RMSE and MAE indicate a higher predictive power. Since RMSE squares the error, it tends to penalize large errors more heavily. The choice between RMSE and MAE is at discretion of the developer. For instance, in the movies domain, while in [85] the RS is evaluated by measuring the quality of suggestions using RMSE, giving more importance to larger differences between the predicted and real ratings, in [73] the evaluation is based on MAE, considering a linear approach to measure the errors.

6.1.2. Usage prediction metrics

These metrics are based on different types of proportions between recommended and consumed items, as determined by the contingency table that compares them. The following are the usage prediction metrics that we identified in the surveyed papers:

- *Precision (or true positive rate)* measures the proportion of recommended items that result relevant to the users, that is, those recommended items that the user actually consumes. The CARS proposed in [39] is evaluated with respect to a context-free approach using this metric. This system exploits user location (i.e., a gym, the library, the office, the transportation system) to suggest appropriate songs. The results show that the proposed approach outperforms its baseline (e.g., a precision of 60% and 50%, respectively, in situations where the location context corresponds to the transportation system).
- *Recall (or sensitivity)* measures the proportion of consumed items that were correctly recommended, that is, the fraction of items relevant to the user that were suggested by the system. Recall and precision are usually considered together as two facets of the quality of the recommendation. An example is presented in [53], where precision and recall are used as the basis to show that the greater the cardinality of a set of recommended items is, the higher the value of recall is.
- *Specificity (or true negative rate)* measures the proportion of not recommended items that are irrelevant to the users. This metric was not directly used in any of the surveyed papers, but it is a basis for the definition of other metrics such as AUC, explained below.
- The *F-measure* family of metrics combines precision and recall, allowing for the comparison of different RS using a single metric. Adomavicius et al. [4] use this metric to compare the effects of taking into account independent context factors (i.e., social, time and location), or combinations of them, when predicting user ratings. The results showed that the segments theater-weekend (i.e., location-time), theater (i.e., location), and theater-friends (i.e., location-social) substantially outperform the standard methods in terms of F-measure. They also applied F-measure to show how their approach outperforms regular non-context RS.
- *AUC (or area under the curve*) is a more robust metric that considers the variations between the true positive rate (recall) and the true negative rate (1 - specificity). The movie CARS published in [74] is evaluated using this metric.
- Other usage prediction metrics are refinements of simpler ones, such as *mean average precision* (MAP) [55], or *benefit ratio* (BR) [95]. The latter is defined as the ratio between the number of users who get an improved prediction and the number of users who get a deteriorated prediction.

6.1.3. Ranking metrics

These metrics assume that the utility of a recommended item is proportional to its position in the ordered list of recommendations produced by the RS. The ranking metrics used to evaluate the CARS included in our survey are the following:

- *Normalized discounted cumulative gain (NDCG) and discounted cumulative gain (DCG)* consider that highly ranked relevant objects give more satisfaction than poorly ranked ones. Biancalana et al. [30] use NDCG to compare their CARS performance with the performance of other approaches. They also study the effect on the quality of recommendations, as measured by NDCG, by taking into account different contextual factors separately. Biancalana et al. [30] and Hong et al. [53] argue that CARS produce better results when the number of items to recommend increases.
- *Hit ratio* measures whether a user's target choice appears in the top-K recommendation list. Generally denoted as Hit@K, where K indicates the number of recommended items. Unger et al. [87] find that the use of latent context models provides a noticeable advantage over non-contextual models for almost every value of K. The advantage is greater with small values of K (i.e., ranging from 1 to 4), which means that the latent context model is highly capable of ranking a suggested recommendation according to the user's current context.
- *Mean reciprocal rank (MRR) and Cumulative reciprocal rank (CRR)* evaluate the ranking position of a user's target choice in the recommendation list. Chen and Chen [99] use CRR to evaluate recommendations that take into account location context.
- *Mean average precision (MAP@K)* considers the precision of the first K recommended ranked items. Every item on the list of ranked items contributes to the MAP@K measure of the recommendation proportionally to its position, if they were indeed accessed/consumed by the user for which the recommendations were made. Jiang et al. [101] use this metric, along with other metrics (MAE, RMSE, Precision, Recall, F1 measure) to evaluate the performance of different configurations of their proposed model.
- *Rt10* averages the ratings of the top 10 recommended items. It is used specially in information retrieval. Son et al. [35] show, using the Rt10 metric, that news article recommendations are more effective when considering their particular geographical location.

Finally, from the five papers that do not report the usage of a particular prediction metric, two of them use other mechanisms to evaluate their models. For instance, to evaluate user satisfaction, Hong et al. [37] measure effectiveness and usability, whereas Baltrunas et al. [45] use a standard usability questionnaire. The approach presented in [81] is compared to a baseline model in terms of accuracy without reporting any metrics. However, these authors published the same model in [82] including a quantitative evaluation.

6.2. Evaluation protocols

This subsection presents the different evaluation protocols applied by the authors of the surveyed papers. These protocols define the way data sets are handled and partitioned into training and test sets to evaluate the quality of the recommendations. We found that in all reported cases context was consistently considered as a data set partitioning criterion, and that the baseline approach is usually a context-free RS, or a CARS that follows a different approach than the one being proposed.

Table 6 presents the distribution of reviewed articles with respect to evaluation protocols. The first column lists the evaluation protocols, the second column shows the number of papers that use the protocol to validate the proposed CARS, and the third column specifies each of the corresponding surveyed papers. Papers [33,41,56,103] did not report on the used evaluation protocol.

6.2.1. Holdout or cross-validation

This is one of the most commonly used evaluation protocols. It consists in splitting the dataset into two sets: training (e.g. 70% of the data) and test (30%). The recommendation model/algorithm is trained using the first set, and evaluated using the second one. The training and test data can be obtained in different ways, depending on the application domain and the way context information affects the recommendations. For example, in [39], Cheng and Shen eval-

Table 6
Evaluation protocols.

Evaluation protocols	#Approaches	Approaches
Holdout or cross-validation	46	[9,10,28,31–33,39,45,48,54,57–60,62,63,65–72,75,76,78,79,81,82,84,87,91–94,96,97,100–102,104–106,109,110]
K-fold cross validation	21	[4,30,34,38,42–44,47,51,55,73,77,80,83,86,88–90,95,98,107]
Hypothesis test	5	[27,35,40,46,99]
Bootstrapping	2	[4,29]
Simulation	1	[64]
None reported	4	[33,41,56,103]

uate their music CARS by splitting the data set according to time and location context, before extracting the training and test sets.

6.2.2. K-fold cross-validation

This is a more sophisticated evaluation protocol that consists in partitioning the dataset into a *K* equally sized groups of items called folds, to then perform a cross-validation evaluation process. One of the folds is chosen as the test set and the union of the other folds as the training set. This process is repeated *K* times, each time changing the fold used as test set. This evaluation protocol is used to evaluate the CARS presented in [42]: for each fold, the authors compute the MAE, precision and recall metrics, and average their results to then estimate the quality of their recommendation model. The CARS proposed in [86] and [4] apply independent recommendation processes for each relevant context. The authors evaluate the performance of these systems using K-fold cross-validation. This allows them to compare the predicted ratings for each context, and establish the contexts for which the recommendation is more accurate.

6.2.3. Hypothesis test

This protocol uses statistical inference. It is based on the computation of the statistical significance of the differences between the compared CARS. In particular, it is useful to identify whether there is a significant difference between contextual and non-contextual recommendations. The CARS presented in [99] is evaluated using this protocol, where the hypothesis is that user preferences are influenced by contextual factors, and that the proposed recommendation algorithm is capable of capturing such influences. For example, user restaurant preferences may not be influenced solely by aspects such as food quality, value, and service, but also by contextual factors such as location.

6.2.4. Bootstrapping

This protocol relies on random sampling with replacement. That is, a subset of size *N* is taken from the original data set and then partitioned into training and test data. This process is repeated multiple times, considering always the whole original data set as the basis for the re-sampling. The estimation of the performances of the RS is finally aggregated from the results of each re-sample. For instance, Musto et al. [29] use a bootstrapping-based protocol proposed in [4]. This protocol consists in identifying different possibly overlapping subsets of the dataset based on context types (e.g., establishing a contextual segment composed of time context observations, or another one composed of location context observations). The authors extract 500 random re-samples from their dataset and split them by assigning 29/30th of the items to the training set and 1/30th to the test set. They use precision, recall and F1 as the metrics to evaluate the performance of their system with respect to the different contextual segments.

6.2.5. Simulation

When there is no dataset available upon which to perform the evaluation of the recommendation model, it is possible to generate an artificial synthetic dataset using simulation techniques, based on certain suppositions (e.g. normal distributions). Eirinaki et al [64] applied this method to generate a social network simulating trust relationships between users (social context), and the matrix relating users to items (in their case, web services).

6.3. Other properties

Predictive metrics measure how close predicted preferences are from user real preferences. However, predictive power is not enough to measure whether the recommendation was satisfactory, useful or effective to the users [112]. A recommendation system may be highly accurate, but only for those items for which a recommendation may result useless (e.g., products that the user buys very frequently).

Table 7 presents the approaches that consider properties other than predictive power to evaluate the proposed CARS. The plus sign in a cell indicates that the corresponding property is used to evaluate the CARS proposed in the paper represented by the row (cf. first column of the table). As in the case of the prediction metrics presented above, we borrowed the definition of these properties from [111] and [112].

6.3.1. Learning rate

This property measures how fast an algorithm produces good recommendations. Learning rate is also associated with the parameter that determines how fast or slow a recommendation model will converge towards an optimal solution. We found that all of the CARS evaluated through this property are based on model-based strategies (i.e., matrix and tensor factorization, and linear regression), and exploit context information by implementing the contextual modeling paradigm.

6.3.2. Confidence

This property refers to the trustworthiness of the system predictions, and the extend to which they help users make more effective decisions. The work published in [90] uses this property to evaluate, under specific contexts, the quality of several prediction models based on matrix factorization,

6.3.3. Diversity

This property measures how dissimilar are the recommended items among them. It is defined as the opposite of similarity. Zhang et al. evaluate the quality of their movie CARS in terms of diversity [90]. They argue that a good recommender system is the one that delivers considerable different recommendations, for example, films belonging to different genres.

6.3.4. Novelty

Based on the assertion that the relevance of a recommended item depends not only on its correctness, but also on its novelty. Nocera et al. [66] define an ad-hoc measure that takes into account whether the recommended items were already known to the user (e.g. accesed in the past).

Table 7
Other properties.

Appr.	Learning rate	Confidence	Diversity	Novelty	Coverage	Scalability	Usability
[76]	+						
[98]	+						
[86]	+						
[90]	+	+	+				
[62]		+				+	
[94]			+		+		
[66]				+			
[53]						+	
[79]	+					+	
[99]	+						+
[30]							+
[27]							+
[38]							+

6.3.5. Coverage

This property measures the proportion of items that the system recommends from the universe of available items. Not all of the available items are subject to be recommended. This is the case of collaborative-filtering RS for items that have not been yet consumed or rated by the users. Sitkrongwong et al. measure accuracy and coverage for different contextual factors [94]. They found that, since not every context applies to all items, it is possible to increase the coverage by ignoring some of the relevant contextual factors. Nevertheless, there is a trade-off between accuracy and coverage that can be mitigated by identifying the set of relevant contextual factors for each user and each item separately, instead of identifying the relevant contextual factors for the entire data set.

6.3.6. Scalability

This property refers to the computational capability of the recommender system to handle a growing amount of data. Khalid et al. address this property by storing and processing data on geographically distributed nodes [62]. Shi et al. measure scalability in terms of time complexity [79]. We did not find any relation between context and scalability.

6.3.7. Usability

This property measures the satisfaction of the user with respect to the ease of use of the RS. In [27], Hawalah and Fasli evaluate usability through a questionnaire that asks users to rate a set of statements, including some to evaluate the contextual nature of the system: i) *the items recommended to me matched my interests*, ii) *the items recommended to me took my personal context requirements into consideration*, and iii) *I was only provided with general recommendations*.

6.4. Data sets

Table 8 characterizes the 16 data sets that we identified as publicly available from 32 out of the 87 characterized papers. For each data set, we indicate the papers that use it, the domain, and the supported context types.

7. The effect of incorporating context into RS

When conducting an SRL on CARS, a natural question is the level of improvement of RS performance (e.g., in terms of accuracy) obtained from the inclusion of a particular context type into the recommendation process. Nevertheless, answering this question results impractical, given the wide spectrum of recommendation techniques that can be combined with the different context types, through any of the three existing paradigms to include context information into RS. Furthermore, the performance of these systems vary depending on the used dataset and evaluation metrics, which make the results incomparable. For this reason, questions such as *what is the context type that provides the best results for improving recommendations in a particular context domain?* were not included in the set of research questions that drove the development of this SLR.

Despite the limitations to compare the effectiveness of particular context types, we surveyed the impact of incorporating context information into the reported systems. We found that only 36 out of the 87 studied articles quantitatively evaluate the obtained improvements with respect to baseline approaches (cf. Table 9). This constitutes an opportunity for this research community—formal validations and benchmarks of CARS are of paramount importance to advance this field. The systems reported in these 36 papers were all evaluated with respect to at least one baseline approach in terms of accuracy, through any of the metrics listed in Table 5.

Table 9 presents the improvements reported by these papers. For each approach (cf. Column *Appr.*) the table includes the types of context exploited by the corresponding CARS, the application domain, and the improvement obtained for each of the used metrics. The table groups accuracy metrics according to the three metric categories (i.e., usage prediction, rating prediction and ranking prediction), explained in Section 6.1. The goal of this table is to report the surveyed information rather than to provide a basis for comparing the improvements obtained in RS when including the different context types.

8. Research opportunities

This section provides CARS researchers with a list of research opportunities, most of them borrowed from the studied articles. From each paper, we identified, categorized, and analyzed the challenges that authors defined as worthy of future work. Each subsection corresponds to one of the nine challenge categories that we identified: *dynamic context management, context gathering, context reasoning, contextual modeling, problems inherent in RS, CARS evaluation, users in the loop, self-adaptation* and *privacy and ethical considerations.*

8.1. Dynamic Context Management

Traditional CARS assume that context information is immutable over time, even when user situations continuously change. Evidence of this are deal recommendation systems that keep sending offers to the user for events currently happening in her home city, despite she is in a several day business trip that is scheduled in her agenda, and the user's agenda as well as her current location can be easily monitored by modern applications [7]. This static

Table 8
Data sets identified in the SLR.

Appr.	Domain	Brief description	Context types	URL
[73]	Movies	Information about movies, users and ratings.	Human (age, gender)	[9,50,57,60,75,78,80,90,97]
[9,50,57,60,75, 78,80,90,97]	Movies	MovieLens: information about ratings, users, and items (movies).	Human (age, gender, occupation), Time (day, month, year, hour, minute, second)	[72,86,88,94]
[72,86,88,94]	Movies	Data set collected for experiments using an on-line application for rating movies. Users fill in a simple questionnaire created to explicitly acquire the contextual information describing the situation during the consumption. It contains records of users, ratings and movies.	Time (season, day type), Location, Natural (weather), Social	[85]
[85]	Movies	Provided by the Netflix Prize. It contains records of ratings, users, and movies.	Time	[24]
[24]	Movies	CAMRa 2011s MoviePilot Dataset: contains ratings, users, and items.	Time	[36,48]
[36,48]	Music	Information about users, artists, bi-directional user-friend relations, and user-listened artist relations	Social, Time (day, month, year)	[54,58,62,65,100,106]
[54,58,62,65, 100,106]	Points of interest, Hotels & Tourism	Data set acquired from FourSquare. It contains information places.	Location, Social	[68]
[68]	General application	Information about users, tagged papers, and tags.	Time, Social	[69]
[69]	Movies	Provided by the Comaq Systems Research Center. Ratings given by users to movies.	Time	[65]
[65]	Points of interest, Hotels & Tourism	Friendship network with information about locations and user check-ins (user, check-in time, latitude, longitude, location)	Social, Location	[57]
[57]	General application	Information of ratings given by users to jokes	Human (user preferences)	[71,91,93,105]
[71,91,93,105]	E-retailing	Information about reviews of products done by users	Social	[70,93]
[70,93]	E-retailing, Music	Information about reviews of products done by users	Social, Time	[91–93,105]
[91–93,105]	E-retailing, Books, Music, Movies	Information about user reviews and recommendation services for movies, books, and music	Social	http://socialcomputing.asu.edu/datasets/Douban

vision of context information causes that RS deliver recommendations that are irrelevant to users, which has negative effects for businesses.

To deal with this dynamic nature of context, CARS must be equipped with runtime mechanisms to identify relevant context and integrate it into the recommendation process dynamically [3,5]. This implies also to enable RS to manage the life cycle of context information at runtime, for instance, to identify context variables that become relevant or irrelevant, and treat them accordingly. For example, by adapting the recommendation model according to new context variables that may become relevant while the user interacts with the system.

Dynamic context management research in RS includes investigating mechanisms to i) identify context changes that affect the relevance of recommendations; ii) characterize the life cycle and dynamics of context information; and iii) develop situation-aware and self-adaptation mechanisms to enable CARS with the ability to adjust recommendation models at runtime. Among the studied papers, [3,28,30,36,47,69,85] declare dynamic context and its management challenges as a future research area.

The following two categories of research opportunities, context gathering and context reasoning, are completely related to dynamic context management, since they are concrete phases of the context information life cycle [5].

8.2. Context Gathering

Context gathering refers to the process of acquiring context information from the user's environment. When the relevant context is dynamic (e.g., context that changes over time such as the purchase intent of a user), context acquisition requires automatic mechanisms to detect context sources that become available at runtime, and deploy the sensors required to gather this information. Context gathering challenges include: i) the acquisition of context information from non-explicit and non-traditional context sources (e.g., to identify user intents and motivations); and ii) the development of user interfaces that allow the acquisition of relevant context, without requiring user explicit inputing through traditional interfaces. The authors of the following papers highlight the importance of context gathering research [27,40,45,48,60,76,84,96].

8.3. Context reasoning

Context reasoning refers to the inference of implicit context facts from raw context [5]. When context is highly dynamic, context management mechanisms must support the addition of reasoning rules dynamically. Context reasoning challenges in RS include: i) inferring context facts from the combination of different context variables; ii) understanding, particularly at runtime, the re-

Table 9
The effect of incorporating context into the RS that were evaluated quantitatively.

Appr.	Types of context	Application domains	Usage prediction				Rating prediction		Ranking Prediction
			Precision	Recall	F-Measure	MAP	MAE	RMSE	NDCG, Hit Ratio, MRR
[67]	Human(mood), Time	e-learning	2%	2%	5%				
[50]	Time, Location	Movies		22%			32%		
[90]	Human (age, gender)	Movies						3%	
[99]	Social, Time	Hotels and Tourism							
[79]	Human (hunger level), Time, Location	Food				15%	9%	9%	
[80]	Social, Time	E-retailing, Movies	6%				17%	14%	
[47]	Location, Time	Point of interest	Between 1,7% and 3,1%				9%	4%	
[51]	Time, Location, Social	Movies	10%						
[72]	Time, Location, Natural (weather), Social	Movies, Hotels and Tourism	Between 80% and 200%; and between 16% and 103%						
[29]	Time, Social, Location	Movies			About 10%				
[98]	Time, Location, Social	Movie	Between 2% and 42%			Between 2% and 6%			

(*continued on next page*)

Table 9 (*continued*)

[43]	Time, Location	Music, Point of interest	Between 5% and 33%	Between 5% and 33%	Between 5% and 33%				
[75]	Time	Movies		Between 30% and 35%					
[73]	Human(age, gender), Time, Social	Movies					Between 5% and 30%		
[84]	Social	Not Identified	Between 12% and 22%		About 21%			About 24%	
[76]	Location, Activity	e-retailing	About 53%			About 40%			
[87]	Location, Time, Activity	Point of interest							Hit ratio: About 25%
[68]	Time, Social	General application							Hit ratio: Between 34.56% and 35.91%
[93]	Social	E-commerce					About 10%	About 10%	
[91]	Social	Books, Music, Movies					Between 9% and 18%	Between 7% and 17%	
[58]	Social	Books, Music, Movies	Avg: 73.27 times better	Avg: 73.27 times better					
[69]	Time	Movies					About 5%		
[92]	Social	General application					Avg: 21%	Avg: 18%	
[31]	Human (user interest)	Web services							NDCG: 40%
[32]	Social	Multimedia	About 25%						
[100]	Social, Location	Points of interest, Hotels & Tourism					Best case: 22%	Best case: 35%	
[65]	Social, Location	Points of interest, Hotels & Tourism	Best case: 15%	Best case: 10%					
[110]	Social, Location, Time	Social networks & Tourism							NDCG: 60%
[101]	Social	General application					Between 10% and 27%		
[59]	Location	Web services					Between 2% and 3%		
[102]	Time	Web services					Between 5% and 20%		
[104]	Social	E-retailing							MRR: between 8% and 25%
[56]	Social	E-retailing	Best case: 78%						
[57]	Different types of context	General application	Best case: 78%						DCG: Between 2.5% and 5%
[106]	Social, Location	Points of interest					Best case: 12.6%	Best case: 14.5%	
[105]	Social	E-retailing					Best case: 16.24%	Best case: 16.09%	

lationships between context situations and user preferences; and iii) exploiting context available in user profiles effectively. Authors of papers [4,30,39,45,52,58,82,86,107,108] identify context reasoning as a relevant research topic.

8.4. Contextual modeling

Pre-filtering, contextual modeling and post-filtering are the three existing paradigms to incorporate context into RS. In contextual modeling, context information is directly integrated into the recommendation model, which, in many cases, has been proved to be more effective than pre- and post-filtering approaches. As a result, an important number of researchers investigate how to exploit context information through contextual modeling [4,24,30,41,43,51,56,68,73,79,81,86,91,105,106]. Contextual modeling challenges include the development of new techniques and mechanisms to: i) integrate context into traditional recommendation models; ii) improve rating estimation methods by exploiting context; and iii) identify the context variables that must be integrated into the recommendation model.

8.5. Problems inherent in RS

Context information can be also useful to solve specific problems in RS. Such is the case of the cold-start, self-biased recommendations, and sparsity problems. Concerning the cold-start problem, context provides information that allows the characterization of users, even when they are newcomers to the system [38,39,93,109]. Regarding the self-biased problem, an important challenges is to develop mechanisms to prevent the self-influence of frequently recommended items on future recommendations; the approach presented by Nocera et al. [66] deals with this problem using a novelty metric that considers social context. Concerning the sparsity problem, context-dependent matrices could help decrease sparsity by taking into account different subsets of dimensions under particular context situations [56,59,88,92,93,101,102,109]. For example, to infer user ratings in a department store, instead of taking into account all of the products the user has bought in the past, one could use only those products directly associated with the user's current purchase intent (e.g., vacation planning, back to school season).

8.6. CARS evaluation

The evaluation of new methods and techniques is crucial to advance the state of the art of CARS, and to confidently apply new developments in real life. Major evaluation challenges identified from the studied papers are [29,36,46,51,64,69,76,107]: i) the investigation of new properties and metrics; ii) the development of benchmarks that facilitate the understanding of approaches that perform better in particular circumstances; iii) the development and documentation of real life experiments in different application domains; and iv) the acquisition of contextual real data to improve the quality of validations.

8.7. Users in the loop

There is an increasing tendency to conceive users as part of software systems, instead of entities that simply interact with systems. This is commonly known as *the integration of users in the loop*. Users can be integrated in the recommendation process, at one or several of its phases, for example, through feedback that can be used to improve recommendations. Users in the loop are also valuable sources of relevant context. An important challenge is to achieve a seamless integration to avoid affecting the natural behavior of the user. This challenge category was explicitly addressed in [50].

8.8. Self-adaptation

Self-adaptive software systems adjust their structure or behavior at runtime to control the satisfaction of functional and non-functional requirements [113]. To achieve these dynamic capabilities, these systems are instrumented with feedback loops that measure outputs and compare them against reference inputs. If the measure output does not correspond with the desired value specified in the reference input, a controller adjusts the target system to obtain better results [114]. An interesting research direction for the advancement of recommender systems is to instrument them with feedback loop-based mechanisms that allow them to self-improve at runtime. Authors of paper [33] highlight self-adaptation as a promising research direction. In particular, they are interested in implementing a feedback mechanism that adjusts the semantic similarity metric at runtime with the goal of improving performance.

8.9. Privacy and ethical considerations

Privacy and ethics are important aspects to be considered in CARS. Several relevant challenges arise from the need to assure these aspects, which is particularly difficult at runtime. For example, whenever a new context source is identified as relevant, how to validate with the user that this information can be used by the system, that this usage is transparent to the user, and that this information will be used only for the purposes approved by the user. Privacy and ethical aspects are of paramount importance to develop confidence and trust in the use of personalization in CARS [27].

9. Conclusions

This paper presented a comprehensive characterization of context-aware recommendation processes and systems, based on the findings of a systematic literature review (SLR) we conducted to survey CARS that were published between 2004 and 2016. This study was conducted with the goal of helping practitioners and researchers understand how context information can be effectively combined with recommendation mechanisms. The main results provide a clear understanding about where context information is usually integrated into the recommendation process, the techniques available to exploit context information depending on the underlying recommendation approach and the phase of the process where context is included, the context types more frequently exploited in the different application domains, and the most common used evaluation mechanisms, including properties, metrics and protocols.

Despite the comprehensiveness of this study, it is unfeasible to conclude about the effectiveness of using particular context types in specific application domains. This is in part because the effect of including context into RS is difficult to generalize given that the results depend on the nature of the used data sets and recommendation approaches. Furthermore, validation methods must be improved to include quantitative measures that allow a more objective evaluation of the proposed approaches—36 out of the 87 studied papers evaluate their systems quantitatively by comparing, against other approaches used as baselines, the improvements obtained with the integration of context information into the recommender system.

Besides the need for improving validation methods, this survey exposes also several research challenges that deserve further investigation. In particular, those related to the need for i) instrumenting CARS with runtime mechanisms to manage context dynamically along its life cycle; ii) developing new techniques to exploit context directly into the recommendation model; iii) exploiting context to solve inherent RS problems, in particular, the cold-start, self-biased recommendations, and sparsity problems; iv) instrumenting RS with self-adaptation capabilities, and v) solving user-oriented issues such as their better integration in the recommendation loop, as well as the privacy and ethical considerations that arise.

Acknowledgment

This work was funded by Universidad Icesi through its institutional research support program.

References

[1] F. Ricci, L. Rokach, B. Shapira, P.B. Kantor, Recommender Systems Handbook, vol. 1, Springer, 2011.
[2] G.D. Abowd, A.K. Dey, P.J. Brown, N. Davies, M. Smith, P. Steggles, Towards a better understanding of context and context-awareness, in: Handheld and Ubiquitous Computing, Springer, 1999, pp. 304–307.
[3] N.M. Villegas, Context Management and Self-Adaptivity for Situation-Aware Smart Software Systems, Ph.D. thesis, University of Victoria, 2013.
[4] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, Incorporating contextual information in recommender systems using a multidimensional approach, ACM Trans. Inf. Syst. 23 (1) (2005) 103–145.
[5] N.M. Villegas, H.A. Müller, Managing dynamic context to optimize smart interactions and services, in: The Smart Internet, Springer, 2010, pp. 289–318.
[6] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
[7] S. Ebrahimi, N.M. Villegas, H.A. Müller, A. Thomo, SmarterDeals: a context-aware deal recommendation system based on the SmarterContext engine, in: Proc. 2012 Conf. of the Center for Advanced Studies on Collaborative Research, IBM Corp., 2012, pp. 116–130.
[8] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: Recommender Systems Handbook, Springer, 2011, pp. 217–253.
[9] H. Ma, T.C. Zhou, M.R. Lyu, I. King, Improving recommender systems by incorporating social contextual information, ACM Trans. Inf. Syst. 29 (2) (2011) 9.
[10] U. Panniello, M. Gorgoglione, Incorporating context into recommender systems: an empirical comparison of context-based approaches, Electron. Commerce Res. 12 (1) (2012) 1–30.
[11] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: a survey, Decis. Support Syst. 74 (2015) 12–32.
[12] B. Kitchenham, S. Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report, Keele University, 2007.
[13] B. Sheth, P. Maes, Evolving agents for personalized information filtering, in: Proc. 9th Conf. on Artificial Intelligence for Applications, IEEE, 1993, pp. 345–352.
[14] K. Lang, Newsweeder: learning to filter netnews, in: Proc. 12th Int. Conf. on Machine Learning, 1995, pp. 331–339.
[15] M. Pazzani, D. Billsus, Learning and revising user profiles: the identification of interesting web sites, Mach.Learn. 27 (3) (1997) 313–331.
[16] W. Hill, L. Stead, M. Rosenstein, G. Furnas, Recommending and evaluating choices in a virtual community of use, in: Proc. SIGCHI Conf. on Human Factors in Computing Systems, ACM Press/Addison-Wesley Publishing Co., 1995, pp. 194–201.
[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in: Proc. 1994 ACM Conf. on Computer supported cooperative work, ACM, 1994, pp. 175–186.
[18] R. Burke, Knowledge-based recommender systems, in: Encyclopedia of Library and Information Systems, Marcel Dekker, 2000, p. 2000.
[19] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, E. Duval, Context-aware recommender systems for learning: a survey and future challenges, IEEE Trans. Learn. Technol. 5 (4) (2012) 318–335.
[20] A. Zimmermann, A. Lorenz, R. Oppermann, An operational definition of context, in: Modeling and Using Context, Springer, 2007, pp. 558–571.
[21] M. Kaminskas, F. Ricci, Contextual music information retrieval and recommendation: State of the art and challenges, Comput. Sci. Rev. 6 (2) (2012) 89–119.
[22] Q. Liu, H. Ma, E. Chen, H. Xiong, A survey of context-aware mobile recommendations, Int. J. Information Technol. Decis. Mak. 12 (01) (2013) 139–172.
[23] Z.D. Champiri, S.R. Shahamiri, S.S.B. Salim, A systematic review of scholar context-aware recommender systems, Expert Syst. Appl. 42 (3) (2015) 1743–1758.
[24] P.G. Campos, F. Díez, A. Bellogín, Temporal rating habits: a valuable tool for rating discrimination, in: Proc. of the 2nd Challenge on Context-Aware Movie Recommendation, ACM, 2011, pp. 29–35.
[25] S. Inzunza, R. Juárez-Ramírez, A. Ramírez-Noriega, User and context information in context-aware recommender systems: a systematic literature review, in: New Advances in Information Systems and Technologies, Springer, 2016, pp. 649–658.
[26] S. Seifu, S. Mogalla, A comprehensive literature survey of context-aware recommender systems, Int. J. Adv. Res. Comput. Sci. Softw. Eng. 3 (4) (2016) 40–46.
[27] A. Hawalah, M. Fasli, Utilizing contextual ontological user profiles for personalized recommendations, Expert Syst. Appl. 41 (10) (2014) 4777–4797.
[28] A.M. Otebolaku, M.T. Andrade, Context-aware media recommendations for smart devices, J. Ambient Intell. Hum. Comput. 6 (1) (2015) 13–36.
[29] C. Musto, G. Semeraro, P. Lops, M. de Gemmis, Contextual eVSM: a content-based context-aware recommendation framework based on distributional semantics, in: E-Commerce and Web Technologies, Springer, 2013, pp. 125–136.
[30] C. Biancalana, F. Gasparetti, A. Micarelli, G. Sansonetti, An approach to social recommendation for context-aware mobile services, ACM Trans. Intell. Syst. Technol. 4 (1) (2013) 10.
[31] B. Cao, J. Liu, M. Tang, Z. Zheng, G. Wang, Mashup service recommendation based on user interest and social network, in: 2013 IEEE 20th International Conference on Web Services, 2013, pp. 99–106.
[32] Z. Wang, L. Sun, W. Zhu, S. Yang, H. Li, D. Wu, Joint social and content recommendation for user-generated videos in online social network, IEEE Trans. Multimed. 15 (3) (2013) 698–709.
[33] L.O. Colombo-Mendoza, R. Valencia-García, A. Rodríguez-González, G. Alor-Hernández, J.J. Samper-Zapater, Recommetz: a context-aware knowledge-based mobile recommender system for movie showtimes, Expert Syst. Appl. 42 (3) (2015) 1202–1222.
[34] W. Lee, K. Lee, Making smartphone service recommendations by predicting users' intentions: a context-aware approach, Inf. Sci. 277 (2014) 21–35.
[35] J.-W. Son, A. Kim, S.-B. Park, et al., A location-based news article recommendation with explicit localized semantic analysis, in: Proc. 36th ACM SIGIR Int. Conf. on Research and Development in Information Retrieval, ACM, 2013, pp. 293–302.
[36] D. Shin, J.-w. Lee, J. Yeon, S.-g. Lee, Context-aware recommendation by aggregating user context, in: Proc. 2009 IEEE Conference on Commerce and Enterprise Computing, IEEE, 2009, pp. 423–430.
[37] J. Hong, E.-H. Suh, J. Kim, S. Kim, Context-aware system for proactive personalized service based on context history, Expert Syst. Appl. 36 (4) (2009) 7448–7457.
[38] X. Wang, D. Rosenblum, Y. Wang, Context-aware mobile music recommendation for daily activities, in: Proc. 20th ACM Int. Conf. on Multimedia, ACM, 2012, pp. 99–108.
[39] Z. Cheng, J. Shen, Just-for-me: an adaptive personalization system for location-aware social music recommendation, in: Proc. Int. Conf. on Multimedia Retrieval, ACM, 2014, p. 185.
[40] B. Fang, S. Liao, K. Xu, H. Cheng, C. Zhu, H. Chen, A novel mobile recommender system for indoor shopping, Expert Syst. Appl. 39 (15) (2012) 11992–12000.
[41] M.-H. Kuo, L.-C. Chen, C.-W. Liang, Building and evaluating a location-based service recommendation system with a preference adjustment mechanism, Expert Syst. Appl. 36 (2) (2009) 3543–3554.
[42] L. Baltrunas, F. Ricci, Experimental evaluation of context-dependent collaborative filtering using item splitting, User Model. User Adap. Interact. 24 (1–2) (2014) 7–34.
[43] M.A. Domingues, A.M. Jorge, C. Soares, Dimensions as virtual items: Improving the predictive ability of top n recommender systems, Inf. Process. Manage. 49 (3) (2013) 698–720.
[44] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, J. Tian, Mining mobile user preferences for personalized context-aware recommendation, ACM Trans. Intell. Syst. Technol. 5 (4) (2014) 58:1–58:27.
[45] L. Baltrunas, B. Ludwig, S. Peer, F. Ricci, Context relevance assessment and exploitation in mobile recommender systems, Person. Ubiquit. Comput. 16 (5) (2012) 507–526.
[46] T.H. Dao, S.R. Jeong, H. Ahn, A novel recommendation model of location-based advertising: Context-aware collaborative filtering using ga approach, Expert Syst. Appl. 39 (3) (2012) 3731–3739.
[47] L. Hong, L. Zou, C. Zeng, L. Zhang, J. Wang, J. Tian, Context-aware recommendation using role-based trust network, ACM Trans. Knowl. Discov. Data 10 (2) (2015) 13.
[48] D. Lee, S.E. Park, M. Kahng, S. Lee, S.-g. Lee, Exploiting contextual information from event logs for personalized recommendation, in: Computer and Information Science 2010, Springer, 2010, pp. 121–139.
[49] Y.S. Lee, X.H. Pham, D.N. Trung, J.J. Jung, H.T. Nguyen, Social context-based movie recommendation: a case study on mymoviehistory, in: Context-Aware Systems and Applications, Springer, 2014, pp. 339–348.
[50] L. He, F. Wu, A time-context-based collaborative filtering algorithm, in: Proc. 2009 IEEE Int. Conf. on Granular Computing, IEEE, 2009, pp. 209–213.
[51] Z. Huang, X. Lu, H. Duan, Context-aware recommendation using rough set model and collaborative filtering, Artif. Intell. Rev. 35 (1) (2011) 85–99.
[52] A. Chen, Context-aware collaborative filtering system: predicting the users preference in the ubiquitous computing environment, in: Location-and Context-Awareness, Springer, 2005, pp. 244–253.
[53] W. Hong, L. Li, T. Li, Product recommendation with temporal dynamics, Expert Syst. Appl. 39 (16) (2012) 12398–12406.
[54] H. Bagci, P. Karagoz, Random walk based context-aware activity recommendation for location based social networks, in: Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on, IEEE, 2015, pp. 1–9.

[55] K. Yu, B. Zhang, H. Zhu, H. Cao, J. Tian, Towards personalized context-aware recommendation by mining context logs through topic models, in: Advances in Knowledge Discovery and Data Mining, Springer, 2012, pp. 431–443.
[56] M. Mao, J. Lu, G. Zhang, J. Zhang, Multirelational social recommendations via multigraph ranking, IEEE Trans. Cybern. PP (99) (2017) 1–13.
[57] W. Wang, G. Zhang, J. Lu, Member contribution-based group recommender system, Decision Supp. Syst. 87 (2016) 80–93.
[58] H. Gao, J. Tang, X. Hu, H. Liu, Exploring temporal effects for location recommendation on location-based social networks, in: Proceedings of the 7th ACM Conference on Recommender Systems, ACM, 2013, pp. 93–100.
[59] P. He, J. Zhu, Z. Zheng, J. Xu, M.R. Lyu, Location-based hierarchical matrix factorization for web service recommendation, in: 2014 IEEE International Conference on Web Services, ICWS, 2014, 2014, pp. 297–304.
[60] L. Zheng, F. Zhu, S. Huang, J. Xie, Context neighbor recommender: Integrating contexts via neighbors for recommendations, Inf. Sci. 414 (Supplement C) (2017) 1–18.
[61] R. Agrawal, R. Srikant, et al., Fast algorithms for mining association rules, in: Proc. 20th Int. Conf. very large data bases, VLDB, vol. 1215, 1994, pp. 487–499.
[62] O. Khalid, M.U.S. Khan, S.U. Khan, A.Y. Zomaya, Omnisuggest: a ubiquitous cloud-based context-aware recommendation system for mobile social networks, IEEE Trans. Serv. Comput. 7 (3) (2014) 401–414.
[63] J. Qi, C. Zhu, Y. Yang, Recommendations based on social relationships in mobile services, Syst. Res. Behav. Sci. 31 (3) (2014) 424–436.
[64] M. Eirinaki, M.D. Louta, I. Varlamis, A trust-aware system for personalized user recommendations in social networks, IEEE Trans. Syst. Man Cybern. 44 (4) (2014) 409–421.
[65] J.-D. Zhang, C.-Y. Chow, Core: exploiting the personalized influence of two-dimensional geographic coordinates for location recommendations, Inf. Sci. 293 (2015) 163–181.
[66] A. Nocera, D. Ursino, An approach to providing a user of a social folksonomy with recommendations of similar users and potentially interesting resources, Knowl. Based Syst. 24 (8) (2011) 1277–1296.
[67] P. Dwivedi, K.K. Bharadwaj, A fuzzy approach to multidimensional context aware e-learning recommender system, in: Mining Intelligence and Knowledge Exploration, Springer, 2013, pp. 600–610.
[68] N. Zheng, Q. Li, A recommender system based on tag and time information for social tagging systems, Expert Syst. Appl. 38 (4) (2011) 4575–4587.
[69] S.-H. Min, I. Han, Detection of the customer time-variant pattern for improving recommender systems, Expert Syst. Appl. 28 (2) (2005) 189–199.
[70] N. Hariri, B. Mobasher, R. Burke, Context-aware music recommendation based on latenttopic sequential patterns, in: Proceedings of the sixth ACM conference on Recommender systems, ACM, 2012, pp. 131–138.
[71] P. Symeonidis, E. Tiakas, Y. Manolopoulos, Product recommendation and rating prediction based on multi-modal social networks, in: Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, 2011, pp. 61–68.
[72] H. Wu, K. Yue, X. Liu, Y. Pei, B. Li, Context-aware recommendation via graph-based contextual modeling and post-filtering, Int. J. Distrib. Sensor Netw. 2015 (2015) 16.
[73] A. Karatzoglou, X. Amatriain, L. Baltrunas, N. Oliver, Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering, in: Proc. 4th ACM Conf. on Recommender Systems, ACM, 2010, pp. 79–86.
[74] A. Rettinger, H. Wermser, Y. Huang, V. Tresp, Context-aware tensor decomposition for relation prediction in social networks, Social Netw. Anal. Mining 2 (4) (2012) 373–385.
[75] B. Hidasi, D. Tikk, Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2012, pp. 67–82.
[76] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, N. Oliver, Tfmap: optimizing map for top-n context-aware recommendation, in: Proc. 35th ACM SIGIR Int. Conf. on Research and Development in Information Retrieval, ACM, 2012, pp. 155–164.
[77] S. Rendle, Z. Gantner, C. Freudenthaler, L. Schmidt-Thieme, Fast context-aware recommendations with factorization machines, in: Proc. 34th ACM SIGIR Int. Conf. on Research and development in Information Retrieval, ACM, 2011, pp. 635–644.
[78] B. Zou, C. Li, L. Tan, H. Chen, GPUTENSOR: efficient tensor factorization for context-aware recommendations, Inf. Sci. 299 (2015) 159–177.
[79] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, Cars2: Learning context-aware representations for context-aware recommendations, in: Proc. 23rd ACM Int. Conf. on Information and Knowledge Management, ACM, 2014, pp. 291–300.
[80] C. Zheng, E. Haihong, M. Song, J. Song, Cmptf: Contextual modeling probabilistic tensor factorization for recommender systems, Neurocomputing (2016).
[81] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, Context-aware SVM for context-dependent information recommendation, in: Proc. 7th MDM Int. Conf. on Mobile Data Management, IEEE, 2006. 109–109
[82] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, H. Kato, Context-aware ranking method for information recommendation, in: Advances in Communication Systems and Electrical Engineering, Springer, 2008, pp. 319–337.
[83] Y. Omori, Y. Nonaka, M. Hasegawa, Design and implementation of a context-aware guide application for mobile users based on machine learning, in: Knowledge-Based and Intelligent Information and Engineering Systems, Springer, 2010, pp. 271–279.
[84] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, S. Yang, Social contextual recommendation, in: Proc. of the 21st ACM Int. Conf. on Information and Knowledge Management, ACM, 2012, pp. 45–54.
[85] Y. Koren, Collaborative filtering with temporal dynamics, Commun. ACM 53 (4) (2010) 89–97.
[86] A. Odić, M. Tkalčič, J.F. Tasič, A. Košir, Predicting and detecting the relevant contextual information in a movie-recommender system, Interact. Comput. 25 (1) (2013) 74–90.
[87] M. Unger, A. Bar, B. Shapira, L. Rokach, Towards latent context-aware recommendation systems, Knowl. Based Syst. 104 (2016) 165–178.
[88] P. Do, H. Le, V.T. Nguyen, T.N. Dung, A context-aware collaborative filtering algorithm through identifying similar preference trends in different contextual information, in: Advanced in Computer Science and its Applications, Springer, 2014, pp. 339–344.
[89] K. Ji, R. Sun, X. Li, W. Shu, Improving matrix approximation for recommendation via a clustering-based reconstructive method, Neurocomputing 173 (2016) 912–920.
[90] M. Zhang, J. Tang, X. Zhang, X. Xue, Addressing cold start in recommender systems: a semi-supervised co-training algorithm, in: Proc. 37th ACM SIGIR Int. Conf. on Research & development in Information Retrieval, ACM, 2014, pp. 73–82.
[91] H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, in: Proceedings of the Fourth ACM international Conference on Web Search and Data Mining, ACM, 2011, pp. 287–296.
[92] X. Liu, K. Aberer, Soco: a social network aided context-aware recommender system, in: Proceedings of the 22nd international Conference on World Wide Web, ACM, 2013, pp. 781–802.
[93] J. Liu, C. Wu, W. Liu, Bayesian probabilistic matrix factorization with social relations and item contents for recommendation, Decis. Supp. Syst. 55 (3) (2013) 838–850.
[94] P. Sitkrongwong, S. Maneeroj, P. Samatthiyadikun, A. Takasu, Bayesian probabilistic model for context-aware recommendations, in: Proc. 17th Int. Conf. on Information Integration and Web-based Applications & Services, ACM, 2015, p. 22.
[95] Z. Xu, L. Chen, G. Chen, Topic based context-aware travel recommendation method exploiting geotagged photos, Neurocomputing 155 (2015) 99–107.
[96] X. Ramirez-Garcia, M. García-Valdez, Post-filtering for a restaurant context-aware recommender system, in: Recent Advances on Hybrid Approaches for Designing Intelligent Systems, Springer, 2014, pp. 695–707.
[97] L. Cui, W. Huang, Q. Yan, F.R. Yu, Z. Wen, N. Lu, A novel context-aware recommendation algorithm with two-level svd in social networks, Future Gener. Comput. Syst. (2017).
[98] Y. Zheng, B. Mobasher, R. Burke, Deviation-based contextual slim recommenders, in: Proc. 23rd ACM Int. Conf. on Information and Knowledge Management, ACM, 2014, pp. 271–280.
[99] G. Chen, L. Chen, Augmenting service recommender systems by incorporating contextual opinions from user reviews, User Model. User Adap. Interact. 25 (3) (2015) 295–329.
[100] D. Yang, D. Zhang, Z. Yu, Z. Wang, A sentiment-enhanced personalized location recommendation system, in: Proceedings of the 24th ACM Conference on Hypertext and Social Media, in: HT '13, ACM, New York, NY, USA, 2013, pp. 119–128.
[101] M. Jiang, P. Cui, X. Chen, F. Wang, W. Zhu, S. Yang, Social recommendation with cross-domain transferable knowledge, IEEE Trans. Knowl. Data Eng. 27 (11) (2015) 3084–3097.
[102] Y. Hu, Q. Peng, X. Hu, A time-aware and data sparsity tolerant approach for web service recommendation, in: 2014 IEEE International Conference on Web Services, ICWS, 2014, 2014, pp. 33–40.
[103] W.X. Zhao, S. Li, Y. He, E.Y. Chang, J.-R. Wen, X. Li, Connecting social media to e-commerce: cold-start product recommendation using microblogging information, IEEE Trans. Knowl. Data Eng. 28 (5) (2016) 1147–1159.
[104] A.J. Chaney, D.M. Blei, T. Eliassi-Rad, A probabilistic model for using social networks in personalized item recommendation, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, ACM, New York, NY, USA, 2015, pp. 43–50.
[105] J. Li, C. Chen, H. Chen, C. Tong, Towards context-aware social recommendation via individual trust, Knowl. Based Syst. 127 (Supplement C) (2017) 58–66.
[106] X. Ren, M. Song, H. E, J. Song, Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation, Neurocomputing 241 (Supplement C) (2017) 38–55.
[107] I. Fernández-Tobías, P.G. Campos, I. Cantador, F. Díez, A contextual modeling approach for model-based recommender systems, in: Advances in Artificial Intelligence, Springer, 2013, pp. 42–51.
[108] S.V. Rodríguez, H.L. Viktor, A personalized location aware multi-criteria recommender system based on context-aware user preference models, in: Artificial Intelligence Applications and Innovations, Springer, 2013, pp. 30–39.
[109] C. Lin, R. Xie, X. Guan, L. Li, T. Li, Personalized news recommendation via implicit social experts, Inf. Sci. 254 (2014) 1–18.
[110] A.Q. de Macedo, L.B. Marinho, R.L.T. Santos, Context-aware event recommendation in event-based social networks, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, 2015, pp. 123–130.
[111] G. Shani, A. Gunawardana, Evaluating recommendation systems, in: Recommender Systems Handbook, Springer, 2011, pp. 257–297.
[112] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (1) (2004) 5–53.

[113] R. de Lemos, H. Giese, H.A. Müller, M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N.M. Villegas, T. Vogel, D. Weyns, L. Baresi, B. Becker, N. Bencomo, Y. Brun, B. Cikic, R. Desmarais, S. Dustdar, G. Engels, K. Geihs, K.M. Göschka, A. Gorla, V. Grassi, P. Inverardi, G. Karsai, J. Kramer, A. Lopes, J. Magee, S. Malek, S. Mankovskii, R. Mirandola, J. Mylopoulos, O. Nierstrasz, M. Pezzè, C. Prehofer, W. Schäfer, R. Schlichting, D.B. Smith, J.P. Sousa, L. Tahvildari, K. Wong, J. Wuttke, Software Engineering for Self-Adaptive Systems: A second Research Roadmap, vol. vol. 7475, Springer, pp. 1–32.

[114] M. Litoiu, M. Shaw, G. Tamura, N.M. Villegas, H.A. Müller, H. Giese, R. Rouvoy, E. Rutten, What can control theory teach us about assurances in self-adaptive software systems?, in: R. de Lemos, D. Garlan, C. Ghezzi, H. Giese (Eds.) Software Engineering for Self-Adaptive Systems III, Lecture Notes in Computer Science (LNCS), vol. 9640, Springer, Berlin, Heidelberg, 2017. (In press)