



## UvA-DARE (Digital Academic Repository)

### The recommender canvas: A model for developing and documenting recommender system design

van Capelleveen, G.; Amrit, C.; Yazan, D.M.; Zijm, H.

**DOI**

[10.1016/j.eswa.2019.04.001](https://doi.org/10.1016/j.eswa.2019.04.001)

**Publication date**

2019

**Document Version**

Final published version

**Published in**

Expert Systems With Applications

**License**

Article 25fa Dutch Copyright Act (<https://www.openaccess.nl/en/policies/open-access-in-dutch-copyright-law-taverne-amendment>)

[Link to publication](#)

**Citation for published version (APA):**

van Capelleveen, G., Amrit, C., Yazan, D. M., & Zijm, H. (2019). The recommender canvas: A model for developing and documenting recommender system design. *Expert Systems With Applications*, 129, 97-117. <https://doi.org/10.1016/j.eswa.2019.04.001>

**General rights**

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

**Disclaimer/Complaints regulations**

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

*UvA-DARE is a service provided by the library of the University of Amsterdam (<https://dare.uva.nl>)*



Contents lists available at ScienceDirect

## Expert Systems With Applications

journal homepage: [www.elsevier.com/locate/eswa](http://www.elsevier.com/locate/eswa)

# The recommender canvas: A model for developing and documenting recommender system design



Guido van Capelleveen<sup>a,\*</sup>, Chintan Amrit<sup>b</sup>, Devrim Murat Yazan<sup>a</sup>, Henk Zijm<sup>a</sup>

<sup>a</sup> Department of Industrial Engineering and Business Information Systems, University of Twente, PO box 217, AE Enschede 7500, the Netherlands

<sup>b</sup> Faculty of Economics and Business Section Operations Management, University of Amsterdam, Postbus 15953, NL Amsterdam 1001, the Netherlands

## ARTICLE INFO

### Article history:

Received 22 October 2018

Revised 1 April 2019

Accepted 1 April 2019

Available online 1 April 2019

### Keywords:

Recommender canvas

Recommender system

Design decisions

Requirement specification

## ABSTRACT

The task of designing a recommender system is a complex process. Because of the many technological advancements that may be included in a recommender system, engineers are faced with a fast growing number of design related decisions to be taken. Unfortunately, there is no general approach yet for decision makers that can act as a framework guiding the design of a recommender system. The rich collection of literature on recommender systems, though, offers a great source to identify the key areas where these decisions need to be taken. In this paper, we survey existing literature with the aim of building a recommender system model inspired by Osterwalder's canvas theory. The result of our semi-structured synthesis is a novel design approach in the form of a canvas for designing recommender systems. This work provides a better understanding and can serve as a guide for decision making in recommender system design.

© 2019 Elsevier Ltd. All rights reserved.

## 1. Introduction

The typical purpose of a recommender system is to actively suggest items of interest to users. Thereby, recommender systems facilitate the exploration of available options while also filtering a set of items in an attempt to decrease the users' information load (Jugovac, Jannach, & Lerche, 2017). As a practical application, recommender systems are found today in many applications (Lu, Wu, Mao, Wang, & Zhang, 2015). Its popularity both in research and practice is evident in the various literature surveys that provide a comprehensive summary of the recommender system field (Adomavicius & Tuzhilin, 2005b; Bobadilla, Ortega, Hernando, & Gutiérrez, 2013; Jannach, Zanker, Ge, & Gröning, 2012; Lü et al., 2012; Park, Kim, Choi, & Kim, 2012; Pazzani, 1999).

The design of a recommender system is a challenging proposition due to numerous factors. A few of these include, data quality (Gunes, Kaleli, Bilge, & Polat, 2014), the prioritization of recommender system goals (e.g., precision, recall, accuracy, and novelty of item suggestions) (Bobadilla et al., 2013) (this is an ongoing debate among recommender system researchers), the increase in size (Bobadilla et al., 2013), more complex problem domains (Bouzekri et al., 2018) and more diverse applications across various

domains (Fernández-Tobías, Cantador, Kaminskas, & Ricci, 2012). In addition, new concepts such as trust derived from social networks (Ma, Zhou, Liu, Lyu, & King, 2011), context-awareness enabled through increased mobile phone use (Gavalas, Konstantopoulos, Mastakas, & Pantziou, 2014), and the need for configurable human-recommender interaction (He, Parra, & Verbert, 2016), have also added design considerations when building recommender systems. As a result, engineers have begun to design more distinctive and advanced (hybrid) filtering techniques.

These burgeoning concepts in recommender system development, the variety of choices, and the alternative system designs can be overwhelming for practitioners to consider. Furthermore, coordinating recommender system design tasks, such as eliciting the requirements and engineering issues for a recommender system, can become more imperative, owing to this increased project complexity (Bouzekri et al., 2018). The researchers of van Capelleveen, Amrit, Yazan, and Zijm (2018b) have had a similar experience with specifying requirements while designing recommenders for industrial symbiosis networks.

Models can provide a generic structured and uniform approach to decision making for complex problems (Azevedo, Santos et al., 2008). For example, business models (e.g., the business model ontology Osterwalder et al., 2004) can be used to capture, understand, communicate, design, analyze and change the business logic of a firm. While process models, e.g., CRISP-DM (Chapman et al., 2000), KDD (Fayyad, 1997), SEMMA (Sas Institute inc., 2013) and PID-CD (van der Spoel, 2016), serve as a descriptive set of steps for

\* Corresponding author.

E-mail addresses: [g.c.vanCapelleveen@utwente.nl](mailto:g.c.vanCapelleveen@utwente.nl) (G. van Capelleveen), [c.amrit@uva.nl](mailto:c.amrit@uva.nl) (C. Amrit), [d.m.yazan@utwente.nl](mailto:d.m.yazan@utwente.nl) (D.M. Yazan), [w.h.m.zijm@utwente.nl](mailto:w.h.m.zijm@utwente.nl) (H. Zijm).

approaching a (data mining) project. Similarly, building a model of recommender system design following an ontological approach can provide a basis for new management tools that can help to outline the design aspects relevant to structure the specific design complexities in their application context in a generic way.

To the best of our knowledge, the current literature only marginally addresses models for the use of requirement specification when designing recommender systems. In particular, there is a dearth of models that provide an overview and explanation of the design considerations engineers face when developing recommender systems that go beyond the traditional webshop application and address more complex recommender system applications. Therefore, to support the designer in this task of making a well-considered recommender system design, we go beyond simply reviewing filtering techniques. In this paper we structurally review all the aspects of effective recommender design in relation to the context and process the different dependencies. This involves eliciting requirements, reviewing the characteristics of the application domain, outlining the optional and required design choices prior to framing the techniques into design patterns, designing the user interface of a recommendation and applying evaluation mechanisms. Modeling all the recommender system concepts in a canvas provides structure to these recommender system design aspects and will be helpful to guide complex decision making in recommender systems projects. Therefore, the findings of this research are expressed in a theoretical model encompassing six major design aspects that act as a reference guide for developing recommender systems in complex domains (e.g., recommendation in industrial symbiosis networks [van Capelleveen, Amrit, & Yazan, 2018a](#)).

The remainder of the paper is organized as follows: [Section 2](#) explains the methodology undertaken in this research. [Section 3](#) presents our theoretical model that has been constructed with the support of the surveyed literature. [Section 4](#) illustrates the developed recommendation canvas in an application of a recommendation design for the Sharebox platform. [Section 5](#) reflects on the use of the canvas. In addition, we evaluate the limitations based on assumptions related to the applicability and generalizability of the model. The concluding [Section 6](#) summarizes our findings and lists some open issues that require future work.

## 2. Methodology

The aim of this research (that addresses the proposition aspect of design science [Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007](#)) is to build an ontological model (i.e., the recommender canvas) to support engineers in designing recommender systems for more complex domains. Our recommender canvas is inspired by the Business Model Canvas by [Osterwalder et al. \(2004\)](#). The applied methodology is based on the hermeneutic literature review approach ([Boell & Cecez-Kecmanovic, 2014](#)), focused on the process of developing an understanding, while respecting the creativity of design in a large body of literature relevant to the target problem. This study's approach consists of the following steps.

1. We review the literature of recommender systems to identify the areas relevant to the design of a recommender system, and retrieve the corresponding design concepts (See [Fig. 1](#)). We study each concept through a synthesis of the literature on the subject. For each concept, we search the literature with keywords related to the particular concept and apply snowballing techniques ([Boell & Cecez-Kecmanovic, 2014](#)) to papers, if a deeper understanding of the problem area is required. Based on our search ([Fig. 1](#)), we have identified six areas of research that relate to recommender design, and 22 different design concepts. The six areas are:

- (a) The goals of recommender systems: *what do we try to achieve with the recommender?*
  - (b) The domain characteristics in which recommendation takes place: *what characteristics may influence the design?*
  - (c) The functional design considerations of recommender systems: *what functionality does the user expect in the design?*
  - (d) The filtering techniques for creating recommendations and the techniques for soliciting data to create a sustainable basis for recommender system to recommend upon: *what techniques best apply to this case?*
  - (e) The interface of a recommender system: *how to present the recommendations?*
  - (f) The evaluation and optimization mechanisms for a recommender system: *how to test the recommendations and make sure that they remain relevant to users?*
2. Second, we convert all aspects into a model (the recommender canvas) that informs the design of a recommender system.
  3. Thirdly, we interview managers and software developers who have implemented recommender systems in various domains in order to validate the completeness of our model.
  4. Finally, we illustrate the application of the recommender canvas with a case design constructed for the Sharebox platform ([Sharebox, 2019; van Capelleveen et al., 2018b](#)).

## 3. The recommender system design model

The recommender model is a formalization of recommender system concepts synthesized from our literature review. We identified six main areas that constitute the essential recommender design issues. These areas are further broken down into a set of 22 related components that help in the design of a recommender model.

Inspired by the work of [Osterwalder et al. \(2004\)](#), we developed the idea to structure the findings of the literature review in a canvas presentation format. A canvas is a visual template for developing or documenting conceptual structures to serve as support for addressing design problems. Presenting a difficult concept with its attributes in a canvas structure has a number of supportive functions. As [Osterwalder et al. \(2004\)](#) argues, a canvas for modeling the business logic of a firm supports five main functions: (1) to understand and share the design concepts, (2) to analyze the design, (3) to manage the design, (4) to evaluate the concepts in the design for future prospects, and (5) to patent a design. While Osterwalder's canvas is focused to capture the business logic of firm, the goal of the recommender canvas is to capture the essential requirement specifications of a recommender system design. The contribution of a recommender system canvas is expected to support mainly in analyzing the requirements for each recommender system concept, in addition to the creation of common understanding about each concept of the recommender system (relating to function 1 and 2). Furthermore, when evaluating a design it is expected that a canvas could structure a discussion on how a recommender system design, and interrelated concepts of the recommender system design may be improved (related to function 5).

[Fig. 2](#) presents the recommender model in a canvas structure. This canvas has an ontological nature. The term ontology is used to refer to the shared understanding of some domain of interest by means of conceptualization to be used as a unifying framework for solving problems in the particular domain ([Uschold & Gruninger, 1996](#)). Therefore the canvas' concepts may have attributes and inter-relationships with other concepts. Although one might expect that a recommender system design follows the order presented here, these tasks do not necessarily occur in that order.

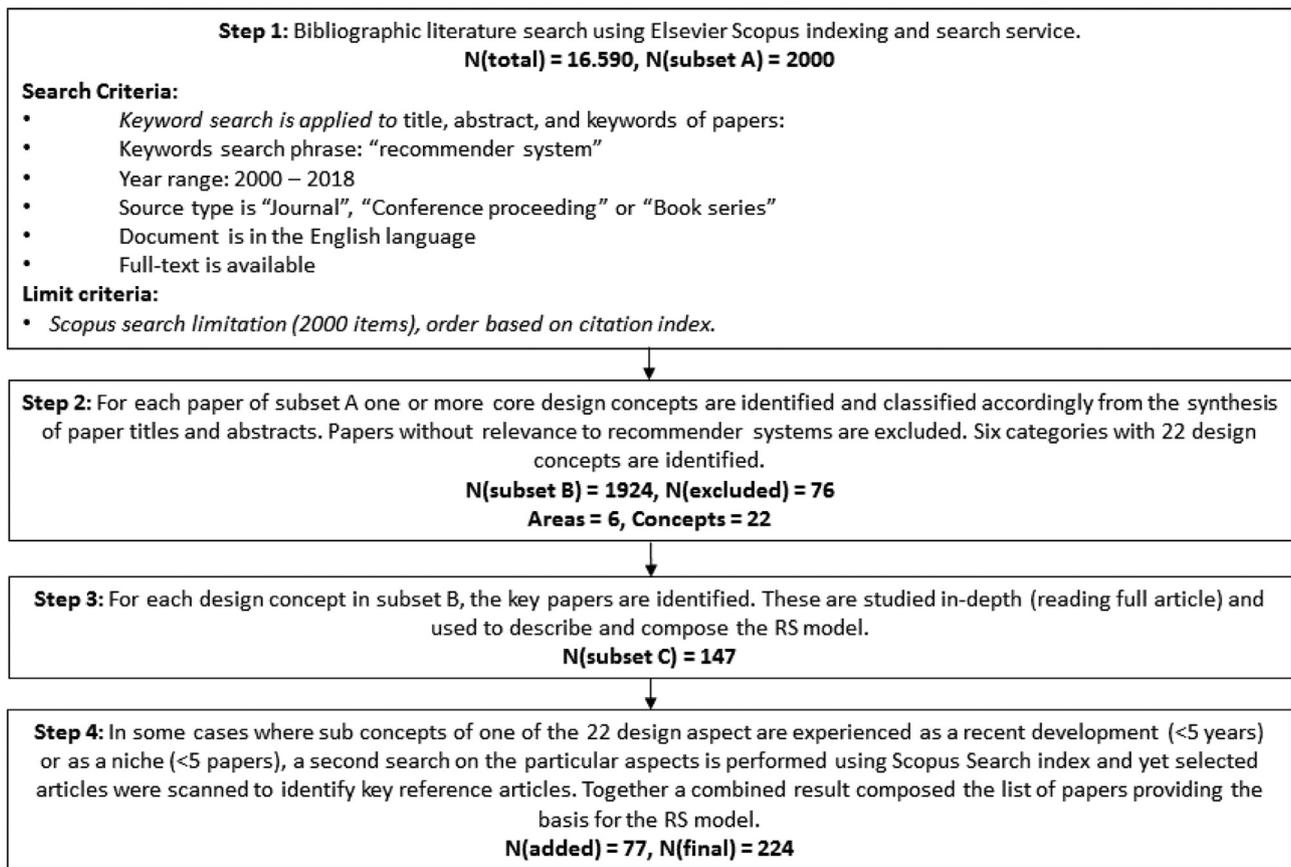


Fig. 1. Steps of the hermeneutic literature review.

Furthermore, each of the presented concepts or sub-concepts of an aspect cannot be considered as mandatory parts of every recommender system design. Therefore, we think that a canvas (Fig. 2), much like the business model canvas (Osterwalder et al., 2004), is an appropriate framework to aid in the design of a recommender. In the forthcoming six sections (Section 3.1–3.6) we describe and discuss each of the recommender system concepts.

### 3.1. Goals

Recommender goals are a common starting point in the development of recommender systems and involve the elicitation of shared goals among all stakeholders (e.g., end-users, item vendors, etc.). Typically, these recommender system goals are formulated as user and organizational goals. Then, the goals can be translated into system functionality which are generally described with the support of use cases in order to create shared understanding about the functional design of the recommender system (Bouzekri et al., 2018).

#### 3.1.1. Recommender goals

Setting goals and outlining the desired effects are part of almost every project that includes the design of recommenders. The traditional goal for e-commerce recommender systems is to support the product purchase decision of consumers by suggesting items they are likely to buy, along with the relevant item information (Xiao & Benbasat, 2007). Indirectly, of course, these user goals somehow align with organizational goals such as profit growth that typically result from the increased direct sales, stimulation of user

activity, and through creating a community bounded to the organizations' e-commerce platform. Although high accuracy metrics would be ideal (McNee, Riedl, & Konstan, 2006a), the quality of actual recommendation is influenced and balanced by more factors, including prediction accuracy, coverage, confidence, trust, novelty, serendipity, diversity, utility, risk, robustness, privacy, adaptability, and scalability (Shani & Gunawardana, 2011). Some recommenders (e.g., in an environmental domain) even aim for less measurable contextual goals such as user behavioral change (van Capelleveen et al., 2018b). Hence, the engineers' task in the design of a recommender system is to first define these goals in collaboration with the main system stakeholders. Thereafter, goals may be prioritized or weighted in order to provide a balanced focus (Bobadilla et al., 2013).

A recurring debate concerns the question whether what is measured reflects what a user prefers, i.e., whether scoring well on previously defined goals is not the result of over-optimization based on the user's instruments, or is achieved by the application of steering (or persuading) techniques (Chen, 2008; Cremonesi, Garzotto, & Turrin, 2012b; Nanou, Lekakos, & Fouskas, 2010). Example of these include a bias resulting from herd behavior (Chen, 2008), and the use of psychological presentation techniques (Adomavicius, Bockstedt, Curley, & Zhang, 2013). Identifying the reasons behind what is measured may be as important as the independent metric. For example, although news recommender systems appear to be highly accurate in suggesting the right news items to users, it does not reflect the actual satisfaction of user's objective needs but mostly the behavior of users browsing and clicking all the recommendations when they are bored (Pielot, Dingler, Pedro, & Oliver, 2015).

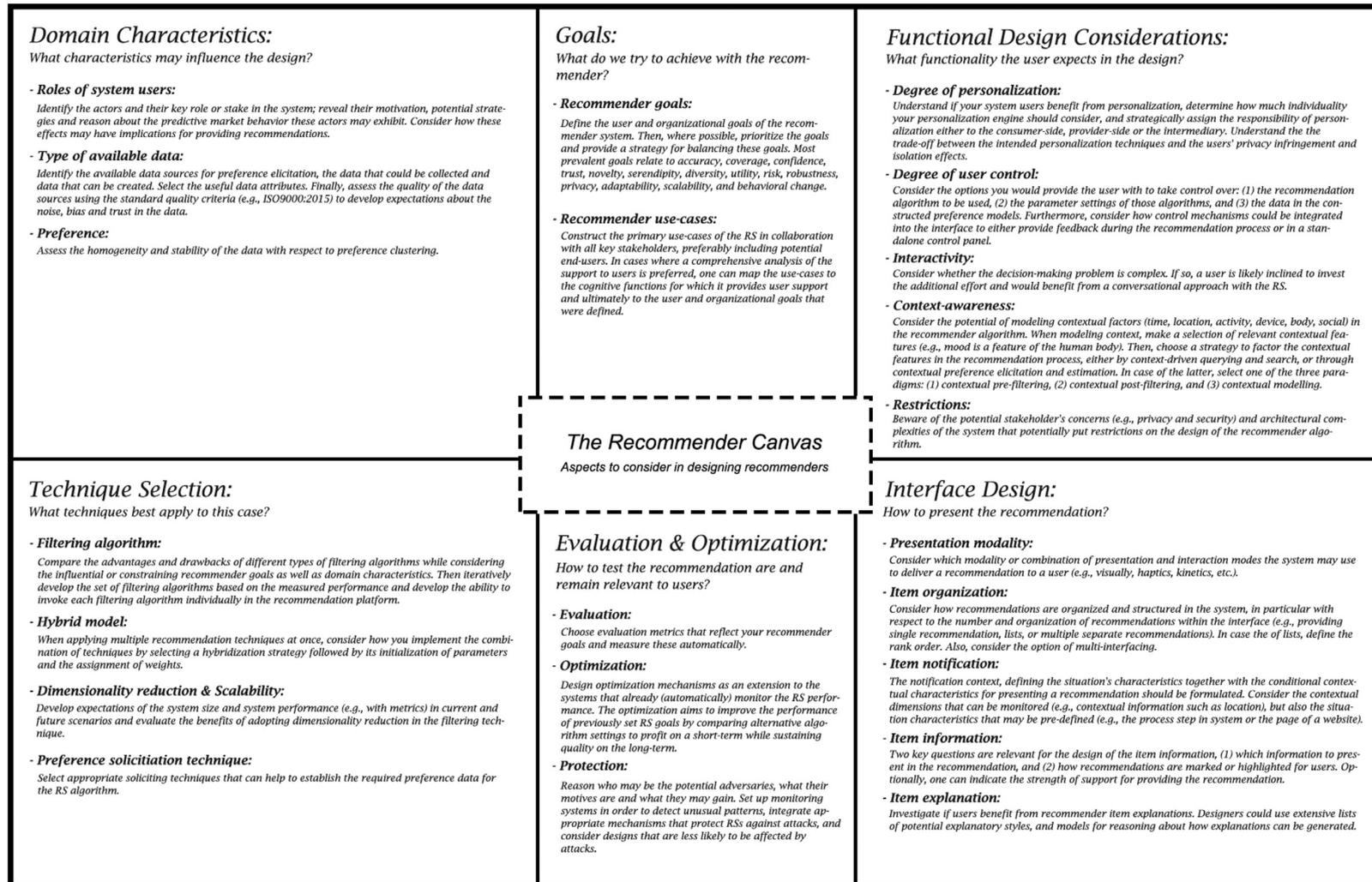


Fig. 2. The recommender canvas.

- Finding: Define the user and organizational goals of the recommender system. Then, where possible, prioritize the goals and provide a strategy for balancing these goals. Most prevalent goals relate to accuracy, coverage, confidence, trust, novelty, serendipity, diversity, utility, risk, robustness, privacy, adaptability, scalability, and behavioral change.

### 3.1.2. Recommender use cases

A second step in the requirement elicitation for recommender systems is to translate the earlier defined goals into practical use-cases. This can clarify and disseminate the expected actions and behaviors associated with the intended goals of the system (Cremonesi, Donatucci, Garzotto, & Turrin, 2012a). Scenarios are a common tool used in building a use-case (Cockburn, 1997). Input collected from all stakeholders about their expectations and how potential end-users interact with the recommender can serve as a basis to formulate these scenarios. Finally, there is the option to gain insight into how a recommender system supports a user's current task by connecting the user's goals and use-cases to a set of related cognitive functions of the decision maker (e.g., Zachary's taxonomy Zachary, 1986).

- Finding: Construct the primary use-cases of the recommender system in collaboration with all key stakeholders, preferably including potential end-users. In cases where a comprehensive analysis of the support to users is preferred, one can map the use-cases to the cognitive functions for which it provides user support and ultimately to the user and organizational goals that were defined.

## 3.2. Domain characteristics

The second aspect, domain characteristics, defines the possibilities, conditions and constrains for the application of a recommender system in a particular context. These are considered important, as the performance of filtering algorithms often depends on the contextual factors. Therefore, a recommender design cannot always be transferred to other domains and may first require adaptations, or in some cases may even have to be replaced by a completely new design (Fernández-Tobías et al., 2012). To understand if and how a recommender system can be developed for a particular domain, one should first analyze the domain characteristics. Three characteristics are considered essential to understand a domain, (1) the actors and their roles in a system, (2) the type of data available to the recommender system that can be used to generate item suggestions, and (3) the demographics of user preference in a system community.

### 3.2.1. Roles of system users: strategy, incentives and their behavior

The primary reason to identify system users (or actors) and their roles is to detect potentially conflicting preferences, such that a recommender may individualize or align recommendation based on the preferences of all actors. Recommendations may be provided in two types of configurations. In a single configuration, a recommender system suggests directly to a user or a group of users (e.g., items directly sold to one customer). In a network-based configuration, a recommender system suggests to multiple users based on their common preferences or by considering their

individual preferences (e.g., in match-making markets (Chamoso, Rivas, Rodriguez, & Bajo, 2018; van Capelleveen et al., 2018a), in group recommendation (Amer-Yahia, Roy, Chawlat, Das, & Yu, 2009) or in recommending group formations (Basu Roy, Lakshmanan, & Liu, 2015; Boratto & Carta, 2011)). Typically, in network-based configurations, the effort involved before establishing transactions or relationships among partners is much higher than in single configuration recommendation systems. Moreover, such a formation process tends to be susceptible to negotiation, and co-operations may be incentivized by the recommender system acting as a facilitator. Naturally, organizations tend to form procurement strategies and competing market behavior (Horling & Lesser, 2004), that may affect or even undermine the effectiveness of the different types of recommendations. Identifying the role of actors is a first step, in such cases, to reason about the actor's motivations. Knowing whether the motivations align and if the actors pursue a common preference, could help to predict and prevent recommender systems from potentially undesirable market cooperations. A second, less critical, reason for identifying the actors is the issue of providing recommendations to users with completely different motivations and goals; e.g., for industries the goal may be to identify profitable transactions while governmental bodies might aim to detect undesirable market phenomena in order to review their policy.

- Finding: Identify the actors and their key role or stake in the system; reveal their motivation, potential strategies and reason about the predictive market behavior these actors may exhibit. Consider how these effects may have implications for providing recommendations.

### 3.2.2. Types of available data

Each domain of application has data characteristics that manifest well to deploy particular knowledge sources for recommendation (Adomavicius & Zhang, 2012; Burke & Ramezani, 2011). However, to build preference models, the sources needed to populate these models should be identified, checked for their availability, and tested for their quality as well as usefulness in relation to the selected filtering methods.

The first concern is the identification of data sources. Data may be collected within the system (e.g., from purchase transactions), derived from external sources (e.g., open data (Di Noia, Mirizzi, Ostuni, Romito, & Zanker, 2012)), linked data (Pereira, Campos, Ströele, David, & Braga, 2018)), or created through combining knowledge from different data sets (Li & Zaïane, 2004). Sourced data can either be explicit, i.e., preference data created by the user (e.g., item ratings), or implicit, i.e., preference data inferred from the behavior of users (e.g., monitoring user's item page visits) (Bobadilla et al., 2013). Some data, such as ratings, may be composed of multiple aspects, therefore allowing the use of multi-criteria filtering algorithms (Adomavicius & Kwon, 2007; Manouselis & Costopoulou, 2007). The type of data that can be used depends on the set of system activities considered to elicit users' preferences from. These include:

- System information (Ekstrand, Riedl, & Konstan, 2011) (e.g., user ratings, transactions of purchased products).
- Social information, information gathered from activities on social networks (Liu, Dolan, & Pedersen, 2010; Ma et al., 2011; Ma et al., 2015), or crowd-based data (Colombo-Mendoza, Valencia-Garca, Rodriguez-Gonzalez, Alor-Hernandez, & Samper-Zapater, 2015; Su et al., 2014), e.g., the type of friends you made, the messages you posted or liked, tags. These networks also provide insights into trust, i.e., the role of a social relation

may indicate the strength of a recommendation (Massa & Avesani, 2007; Tang, Hu, & Liu, 2013; Yang, Guo, Liu, & Steck, 2014).

- Demographic information (Wang, Chan, & Ngai, 2012) (e.g., age, gender, and place of residence).
- Internet of things, information or context-based data (Adomavicius & Tuzhilin, 2015) (e.g., GPS movements (Verbert et al., 2012), health data (Alhamid, Rawashdeh, Al Osman, Hossain, & El Saddik, 2015), emotion (Gonzalez, de la Rosa, Montaner, & Delfin, 2007), and eye-tracking (Zhao, Chang, Harper, & Konstan, 2016)).

The second concern is to source the data. The two key challenges are sparse data and the cold start problem. The sparse data problem (Koren, Bell, & Volinsky, 2009; Popescul, Pennock, & Lawrence, 2001) occurs in systems having a large data set (many users, many items) but where users only provide few ratings or transaction history. Often, dimensionality reduction techniques are applied to compensate for this sparse data problem, mostly based on matrix factorization (e.g., singular value decomposition) (Koren et al., 2009). In addition, when a market or community expands in terms of the number of users, items, ratings, or transactions, one can expect also a larger data size to process. Then, dimensionality reduction techniques (Section 3.4) may apply as well. The cold start problem (Lika, Kolomvatsos, & Hadjiefthymiades, 2014) addresses the lack of data as a result of: (a) a new community, (b) a new user, or (c) a new item (Bobadilla et al., 2013). It may be effective to identify the key users as a major proportion of the data is typically created by only a small number of users (Zeng, Zeng, Liu, Shang, & Zhou, 2014).

The final concern is about the quality of the data. Data quality issues in recommender systems often relate to noise, bias, and trust. Noise is often found in implicit data, although it provides a stronger basis for representing long-term user preference models, as collecting implicit preference data is less difficult (O'Mahony, Hurley, & Silvestre, 2006). Noise may be generated as ratings come in varying scales (Sparling & Sen, 2011), or are the result of duplicates, as items potentially are re-rated (Amatriain, Pujol, Tintarev, & Oliver, 2009b). Another form of noise is bias, which often occurs in ratings, either caused by natural user behavior (Amatriain, Pujol, & Oliver, 2009a; O'Mahony et al., 2006; Steck, 2010), or by undetected attacks that influence which recommendations are suggested (e.g., shilling attacks discussed in Section 3.6) (Gunes et al., 2014). Furthermore, halo-effects are a typical bias found in multi-component ratings (Sahoo, Krishnan, Duncan, & Callan, 2012). Therefore, data quality may be assessed by evaluating the data using quality attributes from ISO standard ISO9000:2015, i.e., completeness, validity, accuracy, consistency, availability and timeliness (International Organization for Standardization, 2017).

- Finding: Identify the available data sources for preference elicitation, the data that could be collected and data that can be created. Select the useful data attributes. Finally, assess the quality of the data sources using the standard quality criteria (e.g., ISO9000:2015) to develop expectations about the noise, bias and trust in the data.

### 3.2.3. Preference

Linked to data is the preference of individuals and groups. How preference is modeled depends on which predictive data characteristics of preference are considered (e.g., the degree of homogeneity in the preference among users, or the assumption that user or group preference is related to item-attributes), and how sta-

ble the preference is over time. Small niches of user preference are for example a decisive pattern for collaborative filtering algorithms (Burke, 2002), but there are also many data containing a large set of items, while the similarity between user's preference is less present (Zhang & Zeng, 2012). Note that not every user in every domain will show a high degree of agreement or correlation of user preference. Often, there is only partial agreement between various users that may require more specific approaches to detect the similarity of the user preferences of these so called 'gray-sheep' (Ghazanfar & Prgel-Bennett, 2014). Clear boundary conditions, within which preferences apply, are needed when designing a recommender system for a particular environment. This is because the ability to transfer a recommender algorithm across domains is highly dependent on the similarity of goals, user models and specific conditions of the domain (Drachler, Hummel, & Koper, 2009). In some cases overlap between domains exists which allows to build cross-domain recommendations. However, these often require special mechanisms in the filtering algorithm that integrate the single-domain user models and relate domain independent profile characteristics to the different domains (Fernández-Tobías et al., 2012).

The stability of preference is another concern. An assumption of many of the algorithms is that a preference of a user or group today is still valid tomorrow, expressed in either short or long-term preference stability (Burke & Ramezani, 2011). Multiple researchers have shown that algorithms are affected by trends, hypes and constant popular bias (Celma & Cano, 2008; Yin, Cui, Li, Yao, & Chen, 2012; Zhang & Zeng, 2012). As preferences may change over time, recommender systems are challenged to compensate for these temporal dynamics (Koenigstein, Dror, & Koren, 2011). This problem is also widely known as concept drifting in the field of data mining (Tsymbal, 2004). These drifts often either appear in data as spikes, caused by sudden changes in behavior or as trends, caused by gradual changes in behavior. Although user preference in some situations may change abruptly (e.g., in the case of unemployment the person's grocery preference may move to cheaper products), it is mostly experienced as a gradual process (e.g., a moving preference from classic fiction novels to more fantasy novels).

Various adaption techniques exist that compensate for these temporal dynamics (Gama, Žliobaitė, Bifet, Pechenizkiy, & Bouchachia, 2014). In the past, temporal dynamics were mostly compensated by using preference decay factors (Ding & Li, 2005; Li, Zheng, & Li, 2011), or by applying time windowing techniques, i.e., only learning from the newest  $t$  observations (Koychev & Schwab, 2000). But these approaches severely affect the algorithms, by a reduction of data instances for model learning. Moreover, short and long-term preference do not necessarily benefit from being modeled in the same way (Billsus & Pazzani, 2000). A possible alternative is to differentiate between the transient and long-term effects of user preference and propose to separate these patterns in the data (Koren, 2010; Xiang et al., 2010). For example, through the inclusion of time drifting parameters, temporal effects can be separately assessed and combined in the recommendation algorithms with weighting factors (Koren, 2010). Another approach is micro profiling (Baltrunas & Amatriain, 2009), in which user data is divided into small time-frames (e.g., a day, month or season), in order to capture patterns of user behavior during such time spans. A potential concern is that preference dynamics may occur both in preference data as well as in auxiliary user-item interaction data (e.g., comments) and do not necessarily evolve in the same way, therefore, benefit from being captured separately (Rafailidis, Kefalas, & Manolopoulos, 2017).

It is the task of the designer to assess the data in order to reveal what kind of spikes, trends and potential biases can be detected that influences the decision to incorporate adjustment factors compensating for the assumed concept drifts. They should

further assess the data for potential temporal dynamics and consider to include parameters in the recommendation algorithm that can compensate for these short and long-term concept drifts.

- Finding: Assess the homogeneity and stability of the data with respect to preference clustering.

### 3.3. Functional design considerations

The third aspect is the set of considerations a system developer has regarding the functional expectations of system users. These concepts relate to the use and extent of implementation of particular functionality in the filtering technique, or the recommender system as a whole. These concepts are independent design considerations in relation to the domain characteristics or the filtering techniques selected, or one of the other aspects of the model, and are presumed to be applicable in every recommender design. The concepts related to functional design considerations solely reflect the functional relationship between the user and the recommender algorithm. The physical relationship is part of the fifth aspect 'interface design'.

#### 3.3.1. The degree of personalization

Most recommender systems are built with the intent to provide recommendations to many different types of users or groups. Personalization is the ability to tailor those recommendations to individuals or groups based on the knowledge acquired about their preferences (Wu, Im, Tremaine, Instone, & Turoff, 2003). The first concern is what requires personalization, i.e., both the content and interface of a system can be personalized. The second concern is the degree of personalization that can be categorized as (a) personalized to individuals, (b) targeted to grouped individuals (e.g., sector-based recommendation), and (c) non-personalized (provide identical recommendations for everyone). A third concern is to appoint the architectural entity responsible for personalization. The personalized types of recommendation (respectively a and b) are delivered to users by using one out of three architectural approaches for personalization based on the entity point that is responsible for deciding about how to one should personalize. Here the architecture can be: (a) provider centric, (b) consumer centric, and (c) market centric (Adomavicius & Tuzhilin, 2005a). Finally, a design concern for recommendation to groups is whether the recommendation is personalized for each individual or a recommendation is aimed at advising the group as a whole (Jameson & Smyth, 2007).

While users generally benefit from personalization techniques in the presence of a clear item preference with each individual user, it can also negatively influence the recommendation process, isolating users from challenging perspectives (i.e., the filter-bubble effect) (Beam, 2014). Another negative effect of personalization for designers is that personalization often leads to a privacy debate; i.e., that personalization impinges on the privacy of users (Knijnenburg & Kobsa, 2013; Xu, Luo, Carroll, & Rosson, 2011).

- Finding: Understand if your system users benefit from personalization, determine how much individuality your personalization engine should consider, and strategically assign the responsibility of personalization either to the consumer-side, provider-side or the intermediary. Understand the trade-off between the intended personalization

techniques and the users' privacy infringement and isolation effects.

#### 3.3.2. Degree of user-control

The second consideration for the recommender design is defining the degree of system control a user is provided with that enables the user to influence the operation of the recommender engine. Theory suggests that users provided with controls positively enhance the recommendations (Harper et al., 2015). A balance has to be found between serving users effectively (at minimal user-effort), while also providing the users with the control they desire (Konstan & Riedl, 2012). Providing users with control over a recommender can be achieved in a number of ways. The first branch of literature suggests that one needs to let users select the algorithm they would like to receive recommendations from (Ekstrand, Kluver, Harper, & Konstan, 2015). In addition, the system might have options for users to adjust the algorithmic settings (e.g., changing the balance of the optimization goals) (Jugovac et al., 2017). Another means of user-control is to enable users to maintain their preference profile. This allows them to make corrections in the user-profile generated by adding or removing historical items, or previously created preference concepts from the user-preference model (Knijnenburg, Bostandjiev, O'Donovan, & Kobsa, 2012a). Also, control can be provided in terms of data privacy policies (Mun et al., 2010). Feedback mechanisms are also used as a form of indirect control over user profiles. These mechanisms allow users, for example, to click on links attached to a recommendation scrutinizing the system when it is wrong (Tintarev & Masthoff, 2011), or communicating through predefined reasons about why the users may, or may not have liked different types of items and may update a profile accordingly (Knijnenburg, Reijmer, & Willemsen, 2011a). Allowing users to take control over the recommender algorithm is closely related to interactivity. While we discussed various means for users to take control over their profile (data), the algorithm (type) and parameter settings, there is also the possibility to enable some form of user control during the recommendation process, referred to as interactive recommender systems (Jugovac & Jannach, 2017). Here one needs to note that interactivity and user control do not necessarily mutually exclude one another, and often may be applied together (Bostandjiev, O'Donovan, & Höllerer, 2012).

- Finding: Consider the options you would provide the user with to take control over: (1) the recommendation algorithm to be used, (2) the parameter settings of those algorithms, and (3) the data in the constructed preference models. Furthermore, consider how control mechanisms could be integrated into the interface to either provide feedback during the recommendation process or in a standalone control panel.

#### 3.3.3. Interactivity

A third design consideration is the interactivity (sometimes called conversational, dialog or critique-based approach (Chen & Pu, 2012; He et al., 2016)) between user and the recommender system, to influence the recommendation process. Most recommender algorithms rely on a preference model that require the user's preferences to be specified upfront, but in many cases, these user models are incomplete and quickly out of date (Mahmood & Ricci, 2009). Moreover, in complex decision settings users often encounter difficulties in expressing their preference, though

they are able to incrementally construct preference specification during their decision making in a contextual setting (Carenini & Poole, 2002). Also, recommender systems do not necessarily understand why users prefer recommendation (McNee, Riedl, & Konstan, 2006b). A recommender system can facilitate such preference development by adopting an interactive approach. Following this approach, users provide direct feedback on particular recommended items (e.g., I like this item but it must be cheaper), by which the set recommendations is directly recalculated or narrowed based on the given criteria (Chen & Pu, 2008). The advantage of such quick feedback loops is the increase of prediction accuracy over the conversational cycle. On the other hand, the process of system-interaction is more labor intensive. This type of recommendation technique shares concepts with the closely related field of information retrieval, in which a more explicit information need is translated into search queries (Belkin & Croft, 1992), which in the case of a recommender is the criteria communicated within a dialog between the user and the recommender system. A special type of interaction is the inclusion of constraints in the feedback process of recommender conversation (Felfernig & Burke, 2008). This type of recommender system can only recommend items that meet a particular property. Thus, a designer of a recommender may choose between a static recommendation design that learns the preference through historical instance data, and an interactive recommendation design through which the recommendations are created iteratively, refining the resulting set by using user-provided explicit criteria.

- Finding: Consider whether the decision-making problem is complex. If so, a user is likely inclined to invest the additional effort and would benefit from a conversational approach with the recommender system.

### 3.3.4. Context-awareness

A substantial amount of research is dedicated to the design consideration of modeling context in recommender systems, better known as context-based recommenders (Liu, Ma, Chen, & Xiong, 2013; Verbert et al., 2012). Context, as defined in Dey, Abowd, and Salber (2001, p. 106) is “any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves”. A system is presumed context-aware, as it aims to rely on many useful contextual signals to become more human-centered (Fischer, 2012). Adopting context in recommenders implies exploiting the notion of user behavior in order to be associated with suggestive interactions. An example of a context-aware system is a recommender system that uses GPS sensors to detect the presence of a user near a restaurant, and therefore can pro-actively suggest the user to have lunch at that place.

When creating the context-model for a recommender, three design aspects come forward, (1) which input to consider to create context-awareness and how to include the contextual information in the generation of a recommendation, (2) what utilization context to select (i.e., the situation in which to provide the recommendation), and (3) which contextual conditions can be applied within the utilization context. These three aspects are often linked, although separately addressed.

The recommendation based on that model can be generated following two approaches, (a) through context-driven querying and search, or (b) through contextual preference elicitation and estimation (Verbert et al., 2012). The latter has three contextual paradigms differentiated by the stage at which the context is used

in the recommendation process. There is the option of, (a) contextual pre-filtering, (b) contextual post-filtering, and (c) contextual modelling (Adomavicius & Tuzhilin, 2015; Champiri, Shahamiri, & Salim, 2015; Verbert et al., 2012). Of course, hybrid forms combining multiple approaches do exist as well and are believed to be capable of substantially increasing performance (Adomavicius & Tuzhilin, 2015).

Obtaining the awareness of a user's context can substantially be attributed to (but not limited by) a number of constituent dimensions.

- *Time-awareness* Exploiting the dimension of time to refine a preference model in more distinct patterns. The underlying assumption of time is that there exists either periodicity of preference that can be detected, or that models should consider users that change preferences over time (e.g., Baltrunas et al., 2011; Lee, Park, & Park, 2008).
- *Location-awareness*: The knowledge of having the user's physical location, access to information about objects or other users in their nearby environment (e.g., Bellotti et al., 2008; Colombo-Mendoza et al., 2015; Gavalas et al., 2014), and motion patterns revealing traveling behavior (Barranco, Noguera, Castro, & Martínez, 2012). This location can be exploited by preference locality (by being spatially close to something a user or group prefers), travel locality (by the preferred distance to visit things), as a boundary for similarity detection between users (Matyas & Schlieder, 2009), and to increase the trustworthiness of ratings (Gavalas et al., 2014).
- *Activity-awareness*: The notion or assumption about what task or goal a user has, and how far the user has progressed in achieving it (e.g., Champiri et al., 2015).
- *Device-awareness*: Detecting the current or preferred device (e.g., laptop, mobile phone, smart-watch, traffic information sign) a user interacts with for receiving a recommendation, in addition to the associated preference bound to the use of a device or linked to characteristics of the device (e.g., the shopping behavior of users of a mobile phone is different from those using a desktop computer (Abbar, Bouzeghoub, & Lopez, 2009)).
- *Body-awareness*: Sensing a user's physical condition of the human body and mental state to make a recommendation (e.g., Baltrunas et al., 2011; Guzmán et al., 2018).
- *Social-awareness*: Analyzing the social relations a user has established with other users or active interactions the user has or can have with people in the group or environment a user currently resides in (e.g., Macedo, Marinho, & Santos, 2015). The underlying assumption is that social relationships and culture of a group reflect the user's preference and behavior in that context.

- Finding: Consider the potential of modeling contextual factors (time, location, activity, device, body, social) in the recommender algorithm. When modeling context, make a selection of relevant contextual features (e.g., mood is a feature of the human body). Then, choose a strategy to factor the contextual features in the recommendation process, either by context-driven querying and search, or through contextual preference elicitation and estimation. In case of the latter, select one of the three paradigms: (1) contextual pre-filtering, (2) contextual post-filtering, and (3) contextual modelling.

### 3.3.5. Design restrictions (privacy, security and architecture)

Whereas previously presented functional aspects are enablers of recommender systems, aimed at improving accuracy or user

satisfaction, the following concerns may constrain the effectiveness or use of recommender systems. Privacy is one of the major concerns raised in the literature (Canny, 2002; Kaur, Kumar, & Batra, 2018; Knijnenburg & Kobsa, 2013). The risk of compromising the identity of users is relevant to the case when a database of ratings is shared with third parties for mining, statistical reporting or testing. The user of a rating may be traced by combining information or inferring from statistical outliers. A further privacy risk results from the use of explanations that reveal the user's connections (Ramakrishnan, Keller, Mirza, Grama, & Karypis, 2001). Some studies are dedicated to this issue and attempt to design privacy enhancing algorithms and cryptographically secure the storage and computation of recommendations (McSherry & Mironov, 2009) but often has a negative effect on the computational performance. Another strategy or enforced policy is to make users aware of the privacy concerns by using notification mechanisms and disclosure disclaimers. When users make information disclosure decisions, they consider help from these justification messages (e.g., "our users told us/allowed us to use"), but also are influenced by the justification to disclose (Knijnenburg & Kobsa, 2013). In addition to privacy, the success of recommender systems increasingly depends on data security, which can create a burden on the use of a system harvesting privacy related data (Beel, Gipp, Langer, & Breiting, 2016). Encryption related filtering mechanisms can partly secure user data (Erkin, Veugen, Toft, & Legendijk, 2012). Some recommender systems even advise the users on considering aspects of privacy and security (Zhu, Xiong, Ge, & Chen, 2014). Finally, architecture can be a concern for a recommender system design. For example, in Yang and Hwang (2013), Ruffo and Schifanella (2009) and Kim, Kim, and Cho (2008), recommendations take place in a peer to peer environment enforcing the algorithms to communicate and store data in different ways. Furthermore, architecture restrictions may derive from the kind of device used in case of decentralized calculation of recommendations, e.g., limited processing capabilities, CPU sharing, and restricted connection to Internet (Barragns-Martnez, Costa-Montenegro, & Juncal-Martnez, 2015).

- Finding: Beware of the potential stakeholder's concerns (e.g., privacy and security) and architectural complexities of the system that potentially put restrictions on the design of the recommender algorithm.

### 3.4. Technique selection

The fourth aspect considers the techniques required to design the recommendation algorithm. A filtering type, or the combination of filtering types should be selected and adapted to the requirements and characteristics of the domain in question. In addition, to cases where filtering types or recommender algorithms are combined, a hybrid model needs to be selected, and the required parameter values and weightings should be assigned. Finally, techniques are deployed to collect preference data supplying the filtering algorithms with initial data and feedback.

#### 3.4.1. Filtering algorithm

The recommendation techniques discussed in the literature often are classified by the filtering algorithm, providing multiple types of filtering techniques, including: (a) collaborative filtering, (b) content-based filtering, (c) demographic and context-based filtering, (d) knowledge-based filtering, and (e) hybrid filtering (Bobadilla et al., 2013; Burke, 2007; Schafer, Frankowski, Herlocker, & Sen, 2007). These types are shortly introduced hereafter. For a detailed explanation including the mathematical models

of the filtering methods see (a) Desrosiers and Karypis (2011), Herlocker, Konstan, and Riedl (2000), Lin, Alvarez, and Ruiz (2002), (b) Lops, de Gemmis, and Semeraro (2011), Pazzani and Billsus (2007), (c) Verbert et al. (2012), Adomavicius and Tuzhilin (2015), (d) Burke (2000), Felfernig, Friedrich, Jannach, and Zanker (2011), and (e) Burke (2007).

- *Collaborative filtering (CF)*: The term collaborative filtering refers to a class of models that predicts items using the rating history of multiple users utilizing a form of similarity between items or users to predict the likelihood users that would prefer an item they have not yet rated. The best-known variants of the CF algorithm assess the similarity between users (user-user CF) or between items (item-item CF). CF algorithms tend to be personalized, but non-personalized collaborative approaches exist, e.g., association rule mining which exploits the similarity of purchase behavior aggregated over time as a basis for the recommendation. It is arguable whether, in the purest perspective, these belong to the class of CF algorithms.
- *Content-based filtering (CB)*: The class of content-based filtering recommenders primarily extracts the content or item-attributes as the basis for item prediction. Similar to CF, CB algorithms mostly utilize the historical ratings of the user for an item but attempt to build a preference profile from items that a user has rated using the item-attributes as preference indicators. CB algorithms also exist both in personalized as well as non-personalized form.
- *Demographic filtering (DF) and Context-based filtering*: Demographic-based recommendation is based on a class of algorithms which extract user-related features reflecting the demographic background that can be mapped to ratings or purchasing behavior. Thus, in contrast to CB filtering, the user profile does consist of demographical user information such as age, gender, and residence. In addition to demographical data, the recommender system can also use contextual properties to guide item prediction.
- *Knowledge-based filtering (KB)*: This class of filtering algorithms interacts with users specifying their preference, in order to provide recommendation based upon domain knowledge. The domain knowledge can be interpreted as a containment of the functional translation of product features to preference. Case-based reasoning (Smyth, 2007) and constraint-based reasoning (such as in rule-based systems) (Felfernig & Burke, 2008) are typical examples of KB recommender systems.
- *Hybrid filtering (HF)*: Hybrid recommendation is a combination of different types of algorithms aforementioned for the same recommendation task. The hybridization strategy is to take advantage of the strengths of one or more algorithms and the data sources which they exploit. For example, demographic techniques are often combined with knowledge-based techniques to make them more robust.

It is noteworthy that each of the algorithms relies on different sources of input and all have their strengths and weaknesses. For a comparison of recommendation techniques that considers the advantages and drawbacks of each, see e.g., Burke (2002, Table II). However, what makes an algorithm particular well equipped for a given context along with the trade-offs to be made, is a task for the system designer. Generally, system designers follow the heuristic Occam's razor principle (William of Occam, c. 1320), to begin with a simple technique, such as user-based CF, rather than more recent or sophisticated algorithms and test whether the desired effect can be achieved before increasing the complexity (Cacheda, Carneiro, Fernández, & Formoso, 2011). Then depending on the success of the recommender systems, they are monitored on different performance metrics, and are improved and extended with additional features and algorithms iteratively.

- Finding: Compare the advantages and drawbacks of different types of filtering algorithms while considering the influential or constraining recommender goals as well as domain characteristics. Then iteratively develop the set of filtering algorithms based on the measured performance and develop the ability to invoke each filtering algorithm individually in the recommendation platform.

### 3.4.2. Hybrid model.

Each recommendation technique has its strengths and weaknesses. For example, content and collaborative systems are known to have cold-start problems, while knowledge-based approaches bootstrap more easily (Burke, 2002). Combining multiple recommender methods, termed hybrids, can strengthen often the performance of recommender output with fewer drawbacks than when individually applied. A hybrid recommender is any system combining multiple filtering techniques in order to generate a recommendation (Burke, 2007). The hybrid model determines how the filtering techniques are integrated to compose one set of recommendations based on two or more filtering algorithms. Hybrids should not be confused with the application of multiple separate algorithms at once, in one interface. The work in Burke (2002) provides an off-set of hybrid models that could be used to optimize multiple trade-offs when combining recommender filtering methods. Later work Burke (2007) shows an analysis to understand the different combinations of filtering algorithms that are likely to be successful in general, or under specific domains characteristics. The most considered hybrid models, described with strengths and drawbacks in Burke (2002, 2007), are:

- Switching: Based on internally set criteria, a switching model decides to select one from a set of recommendation techniques in a particular situation to generate one or more sets of item predictions.
- Weighted: The resulting items scores produced by the different recommender algorithms are multiplied by a weighting factor and subsequently combined in one new item prediction.
- Mixed: Item recommendations generated by multiple algorithms are mixed and presented as one set of recommendations.
- Feature combination: The different underlying knowledge sources for the filtering algorithms are all converted into features to be processed by a single recommendation algorithm.
- Feature augmentation: The ratings or classifications generated by one type of filtering algorithm are converted to features used by the second type of filtering algorithm to make item predictions.
- Cascade: A staged process of recommendation, in which one recommendation algorithm is applied after another one, thereby iteratively refining the recommendation set.
- Meta-level: An algorithm first generates a model rather than item predictions (e.g., preference values based on key-words stored in vectors) and takes this model as input for a second algorithm to predict (e.g., collaboratively compare these models).

- Finding: When applying multiple recommendation techniques at once, consider how you implement the combination of techniques by selecting a hybridization strategy followed by its initialization of parameters and the assignment of weights.

### 3.4.3. Dimensionality reduction and scalability.

A growing community or market may result in an explosion of system users, items, and ratings, thereby testing the system's ability to scale. A first function of scalability techniques is the ability to perform timely recommendations in line with user expectations. The filtering algorithm design should be appropriate to the amount of data to process. The second function concerns the common issue of rating scarcity (a low number of rating data causing difficulty to generate recommendations) and is commonly addressed by reducing the dimensionality of the data. Dimensionality reduction techniques attempt to reduce the sparsity levels in preference matrices caused by this lack of data while simultaneously diminishing performance issues (Sarwar, Karypis, Konstan, & Riedl, 2000). Many dimensionality reduction techniques are based on the matrix factorization technique and offer a scalable solution to large recommender databases (Koren et al., 2009). Popular examples of techniques include the model-based technique Latent Semantic Index (LSI), the Singular Value Decomposition (SVD), Bayesian clustering, probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA) (Bobadilla et al., 2013; Lü et al., 2012). Often, SVD and LSI techniques are combined (Cacheda et al., 2011). The disadvantage of scalability in recommender systems is that it requires calculation of item predictions beforehand in a static offline setting. Furthermore, scaling recommender systems sometimes requires distributed system architectures that increase the complexity of the system design (Berkovsky, Kuflik, & Ricci, 2007; Palopoli, Rosaci, & Sarné, 2013). A prerequisite for these techniques is the assessment the potential size of the market in terms of number of users, items, ratings, transactions, etc.

- Finding: Develop expectations of the system size and system performance (e.g., with metrics) in current and future scenarios and evaluate the benefits of adopting dimensionality reduction in the filtering technique.

### 3.4.4. Preference soliciting technique

Recommender algorithms, and in particular rating-based recommender systems, have a continuous need to actively collect preference data. Recommender systems often experience a cold-start problem, characterized by the lack of data when it starts to recommend items to users. A specific instance of a cold-start problem is the item cold-start problem, requiring soliciting techniques which attempt to collect data for recurring new items (Saveski & Mantrach, 2014). The item cold-start problem also affects new users (user cold-start), who are affected by a lack of data (Bobadilla, Ortega, Hernando, & Bernal, 2012). A next set of techniques a designer therefore often needs, which forms an integral part of recommender systems, encompasses the acquisition of data for a recommender to operate. For example, employing a collaborative filtering system needs techniques which acquire ratings for items (Carenini, Smith, & Poole, 2003). The goal is to obtain ratings for items that have the highest predictive capacity to reveal a user's preference. Therefore, the process of rating items by users needs to be not only simple and quick, but also meaningful. The rated items also contribute directly to the quality of recommendation for a particular user, while also contributing to the overall system quality (Ekstrand et al., 2011). Timing of the elicitation is crucial, as a user needs to decide whether to invest time to provide feedback to the system (Carenini et al., 2003). The techniques thus are aimed at identifying which items and preferences should be elicited by the system (Elahi, Ricci, & Rubens, 2014).

The techniques vary. Some recommender systems use elicitation interfaces at sign up (Carenini et al., 2003; McNee, Lam, Konstan, & Riedl, 2003), either by creating a user profile or through a short interview (or personality quiz) through rating a selected seed set of items (Golbandi, Koren, & Lempel, 2010; Jugovac et al., 2017). Another approach for some particular domains is to let users select pictures and shape landscapes that appeal to them from which their preference could be extracted (tagged categories) (Neidhardt, Seyfang, Schuster, & Werthner, 2015). This type of elicitation is played in a game setting to make users more enthusiastic to provide profile information (Walsh & Golbeck, 2010). The analytic hierarchy process (AHP) is a common method to achieve this for group decisions (Jugovac et al., 2017). Other methods rely on direct rating requests, follow-up review requests (conversational approach), at moments the user is clearly motivated to provide ratings (Pommeranz, Broekens, Wiggers, Brinkman, & Jonker, 2012). Finally, there exist methods that extract ratings externally (Cheung, Kwok, Law, & Tsui, 2003). Bayesian and decision tree approaches could analyze the item set to identify the items that, once rated, are most useful for reducing the prediction error (Elahi et al., 2014). A special type of elicitation is the combination of multiple individual preferences such as in group elicitation (Garcia, Pajares, Sebastia, & Onaindia, 2012). Finally, indirect tracking behavior can also help to infer initial preferences (Hu, Koren, & Volinsky, 2008; Liu et al., 2010). This is a special type of technique that does not solicit data, but creates rating data from other sources, e.g., with filtering bots (Park, Pennock, Madani, Good, & DeCoste, 2006). An extensive list of elicitation techniques or 'active learning strategies' targeted for collaborative learning systems is provided in Elahi, Ricci, and Rubens (2016).

- Finding: Select appropriate soliciting techniques that can help to establish the required preference data for the recommender system algorithm.

### 3.5. Interface design

A challenging issue for many recommender systems is how, when, what and where to present an item recommendation. Interface design constitutes all characteristics of this physical presentation layer between the user and the recommender system. We have found five dimensions of the recommendation interface: (1) the presentation modality, (2) item organization, (3) the notification context (e.g., space and time), (4) the content (i.e., provided item information), and (5) the explanation (i.e., the provided meta-information). We now discuss each in detail.

#### 3.5.1. Presentation modality

The first branch of interface design considers the modality of recommender presentation together with the associated techniques used to create this presentation. Although recommendations are often visually represented in systems, there is a potential for other modalities, e.g., text-to-speech and nonverbal gesture generation (Azaria & Hong, 2016). Only a few cases describe recommender systems that provide auditory recommendations (Grasch, Felfernig, & Reinfrank, 2013) or use interaction modes with e.g., gestures, haptics or kinetics (Chen, Ma, Cerezo, & Pu, 2014; Katarya & Verma, 2016; Lee, Kaoli, & Huang, 2014b; Wörndl, Weicker, & Lamche, 2013). For example, the rise in sensing technologies enables the capture of behavioral data, providing a future prospect of adaptive recommendation based on the emotional state of the user (Calero Valdez, Ziefle, & Verbert, 2016). This

can be exploited as input or output in the entry, consumption and exit stage of the recommendation (Tkalcic, Kosir, & Tasic, 2011).

The composition of a visual design influences how users experience and act upon recommendations (Parra, Brusilovsky, & Trattner, 2014). Literature shows it may improve the usability of a system and helps achieve the underlying goal of introducing users to new items that interest them (Swearingen & Sinha, 2001). Trust, item information, and transparency are among the reasons why users indicate that recommendations are useful to them. Some design choices may even affect the opinion of users on an item (Cosley, Lam, Albert, Konstan, & Riedl, 2003). For example, by presenting recommendations in lists, serial position effects (i.e., the tendency of a person to mostly select the first and last items in a series) are likely to occur (Felfernig et al., 2007). The presentation modality used to communicate recommendations is found to influence the persuasion and satisfaction of users (Nanou et al., 2010).

Visualization of a recommendation can be in the form of text, icons, and images or by using combinations of these (Nanou et al., 2010). Although text or a combination of text and images are the most common, the visualization of recommendation by other means, e.g., through a combination of graphs, may enable innovative forms of exploration (O'Donovan, Smyth, Gretarsson, Bostandjiev, & Höllerer, 2008). Other forms of recommendation representation include relationship models, topic models, and hierarchical trees to visualize the underlying relations between users and items, social maps, or tags used in the system (Verbert, Parra, Brusilovsky, & Duval, 2013). Also, visual browsing may aid users in item discovery as an alternative to explicit search. Such a method offers the user to browse through images representing the item catalog, while eliciting preference on the type of items a user likes (Teo et al., 2016).

- Finding: Consider which modality or combination of presentation and interaction modes the system may use to deliver a recommendation to a user (e.g., visually, haptics, kinetics, etc.).

#### 3.5.2. Item organization

Structuring the recommendations in the interface affects where, how many and in which order recommendations are presented. Various methods are also available for adding structure to the recommended items, e.g., presenting a single top item, a list of top n-items, similar to top item(s), predicted ratings for all items or an overview displaying the trade-off between items (Tintarev & Masthoff, 2011). In the case of lists, how to rank and order these items is another concern for the designer (Cremonesi, Koren, & Turrin, 2010; Deshpande & Karypis, 2004; Vargas & Castells, 2011; Zhang & Hurley, 2008). However, presenting a larger number of well-chosen recommendations does not necessarily result in higher user satisfaction as increasing attractiveness may be counteracted by choice overload (Bollen, Knijnenburg, Willemsen, & Graus, 2010). Furthermore, eye-tracking studies reveal how users tend to fixate on recommended items, thus helping in positioning recommendations to increase the impact on the click and gaze of the presented items (Castagnos, Jones, & Pu, 2010). Different structures, such as lists or boxed areas of interest, also have different time patterns of gaze from users and anchor on different aspects (Chen & Pu, 2010). Designers can exploit heuristics, but they also should be aware of the potential effects in item organization. e.g., effects such as decoy, primacy, framing, priming and the use of defaults (Jannach, Zanker, Felfernig, & Friedrich, 2010).

Another challenging aspect for the presentation of recommendations is the ability of recommenders to adapt to diverse user needs. The engineer has the option of adopting a multi-interface design and visually aligning several hybrid algorithms within a single interface (Swearingen & Sinha, 2001). Furthermore, in group recommender systems the system is being controlled by multiple users at once, which also requires multi-interface design (Chen, 2011).

However, it is impossible to suggest the best single guideline for recommender interface design, as argued in Swearingen and Sinha (2002). The work of Swearingen and Sinha (2002) lists various interaction models in recommenders that have been considered successful. Moreover, the user interface design as a whole is only partly responsible for the user-satisfaction (Knijnenburg, Willemsen, & Kobsa, 2011b). Selecting appropriate filtering algorithms, available data and the domain context (discussed in Section 3.3) are also likely to influence the user satisfaction in recommendation (Konstan & Riedl, 2012; Pu, Chen, & Hu, 2012; Said, Fields, Jain, & Albayrak, 2013).

- Finding: Consider how recommendations are organized and structured in the system, in particular with respect to the number and organization of recommendations within the interface (e.g., providing single recommendation, lists, or multiple separate recommendations). In case of lists, define the rank order. Also, consider the option of multi-interfacing.

### 3.5.3. Item notification context

Every recommender system design includes the notification context (sometimes referred to as utilization context) and additionally the conditions for that context when to provide a recommendation. The process of discovery and acquisition of a context as a data-input is optional and is a case of modeling context in recommenders (see Section 3.3). However, it is mandatory to design the context of notification (data-output), i.e. what is the situation in which a recommendation is pulled by users or pushed to users (Fischer, 2012; Schafer, Konstan, & Riedl, 2001; Wang & Zhang, 2013). A designer has the option to conditionally link the modeling of the context with its utilization, thereby making recommender systems either dependent or independent on this perceived context.

The design of a notification context entails one to select a situation or to define the conditional constraints for a particular situation in which recommendations may be provided. The contextual dimensions sourced for modeling context (mentioned in 3.3), apply as well to the extent of control one has over these dimensions, for defining a recommender's notification context (e.g., we may define the time, location, device and activity, but not have control over a user's body or define the user's social context). The design choices for designers primarily relate to the time and space of a recommendation. The first concern is when to present a recommendation. In Woerndl, Huebner, Bader, and Gallego-Vico (2011), for example, a model for a navigation system is described that provides push notifications for gas stations based on a selection of factors including fuel level, amount of detour and the total length of the route. Such recommendations in general, can either be passive or pro-active, meaning that either the system pushes notifications or users are provided with recommendations upon user request (Gallego, Woerndl, & Huecas, 2013). The second concern is where to provide the recommendation, referring to both the current device and location of a user at that moment (Jugovac et al., 2017).

- Finding: The notification context, defining the situation's characteristics together with the conditional contextual characteristics for presenting a recommendation should be formulated. Consider the contextual dimensions that can be monitored (e.g., contextual information such as location), but also the situation characteristics that may be pre-defined (e.g., the process step in system or the page of a website).

### 3.5.4. Item information

A small but important detail of providing a recommendation relates to the item information provided in the recommendation. This can be critical to users considering whether to click on a recommended item. The minimum information a recommendation contains is an item title (e.g., product name), but it can be extended with descriptions, features or characteristics of any kind. For example, when recommending a new car to a user, it can be useful to include the unique selling characteristics of that car (e.g., "fuel efficient" or "often used in rough terrain"). In general, researchers have found that the presence of longer descriptions increase the perceived usefulness and ease of use of recommender systems (Swearingen & Sinha, 2002). Also, Cooke, Sujan, Sujan, and Weitz (2002) confirms that providing additional information can be of value. In their case, music clip samples were added to CD (Compact Disc) recommendations resulting in the increased likelihood of users considering unfamiliar CDs. A second design aspect of item information is that one may explicitly mark or highlight item descriptions included in a recommendation (instead of just a normal list of items). More specifically, one may consider using some form of classification or score to recommend an item (e.g., recommended, highly recommended, or 4 out of 5 stars).

- Finding: Two key questions are relevant for the design of the item information, (1) which information to present in the recommendation, and (2) how recommendations are marked or highlighted for users. Optionally, one can indicate the strength of support for providing the recommendation.

### 3.5.5. Item explanations

A point of critique for many recommender system evaluators is the lack of reasoning used in the algorithms (Tintarev & Masthoff, 2011). Systems are often treated as a 'black box' without providing transparency on how and why the recommendations were provided (Herlocker et al., 2000). Explanations provide such transparency and in many cases it is seen that users feel more confident about recommendations presented with transparency (Friedrich & Zanker, 2011; Papadimitriou, Symeonidis, & Manolopoulos, 2012; Sinha & Swearingen, 2002; Zhang & Curley, 2017). The objectives of an explanation are not limited to transparency, but also contribute to achieving higher validity, trustworthiness, persuasiveness, effectiveness, scrutability, efficiency, satisfaction, relevance, comprehensibility, and education of users (Jannach et al., 2010; Tintarev & Masthoff, 2007; 2011; 2012).

Item explanations come in different forms. Many of these explanations are in textual form, e.g., including the explanations in the recommendation text, as illustrated in "People like you liked. Oliver Twist by Charles Dickens" (Tintarev & Masthoff, 2007). Also, explanation styles sometimes are composed of simple graphical or statistical representations: examples include the use of histograms with groupings, tag clouds, overall average ratings, and tables of

neighbor's ratings (see Gedikli, Jannach, & Ge, 2014; Herlocker et al., 2000; Tintarev & Masthoff, 2011 for detailed descriptions of the explanation styles). Furthermore, there are recommender systems that use visual networks to provide recommendations, while simultaneously explaining the recommendations. For example, graph-based visualization (e.g., cluster maps and Venn diagrams) can help to explain relationships that have been used to generate the recommendation (Verbert, Parra, & Brusilovsky, 2014), or to provide an exploration interface for recommendations (Gretarsson, O'Donovan, Bostandjiev, Hall, & Höllerer, 2009). Other researchers have used word-cloud techniques to visualize the stored preference model used by a recommender to a user, providing insight on a higher level on the basis of the recommendations (Bogdanov et al., 2013). Furthermore, geographical maps can represent the recommendation and can be analyzed to make improvements to personalized recommendation algorithms (Gansner, Hu, Kobourov, & Volinsky, 2009). Reasoning with models can help to design explanations. For example, the model of Friedrich and Zanker (2011) supports the creation of an explanation by including three major dimensions of a recommendation: the reasoning model, the recommendation paradigm, and the exploited information categories.

- Finding: Investigate if users benefit from recommender item explanations. Designers could use extensive lists of potential explanatory styles, and models for reasoning about how explanations can be generated.

### 3.6. Evaluation and optimization

The final aspect is about methods and measures to test and maintain the recommendation quality. A first set of techniques attempts to evaluate the earlier set goals (see Section 3.1). Evaluation metrics show how well the recommendation performs on these measurable goals. Optimization considers the process of improving the algorithms by tuning the parameters in such a way that it can boost the performance of the recommender system. Methods, such as A/B testing (that can help to compare alternative sample sets, e.g., user groups), filtering techniques, or system factors. Finally, mechanisms exist which protect the system from adversaries that could harm the quality of recommendation data sources.

#### 3.6.1. Evaluation: measuring effectiveness.

Evaluation metrics, or performance metrics are a means to test how good the recommendation provided actually is Herlocker, Konstan, Terveen, and Riedl (2004). Common metrics used to evaluate recommender system goals include accuracy metrics (e.g., precision and recall) and error metrics (e.g., Mean Squared Error (MSE) and Root Mean Squared Error (RMSE)) (Cremonesi et al., 2010; Gunawardana & Shani, 2009). Another means of evaluation, but often omitted, is testing the user experience by means of opinion-based research methods, and user centric studies (e.g., surveys and interviews) (Barrington, Oda, & Lanckriet, 2009; Knijnenburg, Willemsen, Gantner, Soncu, & Newell, 2012b; Pu, Chen, & Hu, 2011; Shani & Gunawardana, 2011). The metrics used to measure recommender system goals is project specific. For example, the success of recommending news items can be measured both by click-through rates, as well as on dwell time (Yi, Hong, Zhong, Liu, & Rajan, 2014). Evaluation tests can either be performed online or offline. While online experiments offer real-world performance measuring, offline experiments are more popular as these tend to be less complex, less costly, and less risky (Gunawardana & Shani, 2009). The

(offline) evaluation tests can also be a justification to implement recommendation algorithms, these are also used to select the most promising design(s) for online implementation (Ekstrand et al., 2011). Recommendation can be casted as a ranking problem (of top-N recommendations). This concept of Learning to Rank (LtR) encompasses its own variety of evaluation metrics as ranking can be learned through point-wise (e.g., classic metrics MSE and RMSE), pair-wise (e.g., Bayesian personalized ranking) or list-wise evaluation techniques (e.g., the mean average precision (MAP), Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (NDCG)) (Karatzoglou, Baltrunas, & Shi, 2013). Note that in LtR point-wise approaches are likely to portray a skewed performance caused by distinctive outliers, therefore accuracy metrics that directly measure top-N recommender quality are usually preferred (Cremonesi et al., 2010).

When evaluating recommender systems consider that goals are always contextualized by defining which set of recommendations are part of an evaluation, i.e., which algorithm, group and time-frame (short v.s. long-term effects) (Shi, Luo, Shang, & Gu, 2017). Furthermore, careful interpretation is required when using frameworks such as Apache Mahout, Lenskit, and MyMediaLite (discontinued) as performance metrics may show disparities among the same evaluation algorithms based on how they are implemented (Said & Bellogín, 2014).

- Finding: Choose evaluation metrics that reflect your recommender goals and measure these automatically.

#### 3.6.2. Optimization

Optimization of recommender systems is concerned with the maximization of expected reward while minimizing the expected costs. Evaluation and optimization are closely related but serve a different purpose. Evaluation methods serve to monitor performance and support recommender design choices, while optimization addresses the techniques that actively manage the recommender system by adjusting the parameters of the recommendation engine in order to (temporarily) increase recommender performance and maintain the quality of recommendations. In an offline setting we may perform a sensitivity analysis on performance measures, such as accuracy, by plotting the Receiver Operating Characteristic (ROC) curves to determine the optimal balance between the parameters of the function (e.g., precision and recall). In online environments, these optimizations may be dynamically monitored and adjusted (Ekstrand et al., 2011). In the case of multiple or hybrid algorithms, the optimization can be tested using a black-box testing approach (popular in online environments). Black-box testing assumes that the internal design of the recommender is not known to the testing system (Asmus, Borchmann, Sbalzarini, & Walther, 2014; Lee, Jung, Selamat, & Hwang, 2014a; Pham, Luong, & Jung, 2013). Methods such as A/B test (Davidson et al., 2010; Gomez-Urbe & Hunt, 2015; Kohavi et al., 2013), or multi-armed bandits (Barraza-Urbina, 2017; Glowacka, 2017; Li, Chu, Langford, & Schapire, 2010) can be used either to select the algorithm, determine the weightings (for hybrid algorithms) (Burke, 2007), create a multi-objective recommender (Rodriguez, Posse, & Zhang, 2012), or optimize the parameters of the algorithms (Chan, Treleaven, & Capra, 2013; Manouselis, Vuorikari, & Van Assche, 2010). The purpose of testing is to find the patterns that could lead to the exploitation of local performance optimization but also balance with other utility goals of recommenders (e.g., exploration). Recommendation performance is likely to vary in evolving systems and cause temporal dynamics that needs to be considered in the evaluation as well as in the

optimization algorithms to address preferences on the changing recommender system functionality (Shi et al., 2017).

- Finding: Design optimization mechanisms as an extension to the systems that already (automatically) monitor the recommender system performance. The optimization aims to improve the performance of previously set recommender system goals by comparing alternative algorithm settings to profit on a short-term while sustaining quality on the long-term.

### 3.6.3. Protection

Recommender systems are vulnerable to attacks that affect the system's recommendation behavior leading to biased recommendations. The most common attack is a profile injection attack, also known as shilling attack, which involves the creation of fictive profiles from which biased rating data can be sent with the intent of promoting or dissuading particular items (Williams, Mobasher, & Burke, 2007). The attack models are typically developed to target a specific type of filtering algorithm or data source and can exploit the system using a variety of fraudulent schemes (Gunes et al., 2014; Lam & Riedl, 2004; Mobasher, Burke, Bhaumik, & Williams, 2007). A robustness analysis assesses the effect of attacks on a recommendation algorithm in order to secure the recommender system against such attacks (Burke, O'Mahony, & Hurley, 2015; O'Mahony, Hurley, Kushmerick, & Silvestre, 2004). This analysis tests both the performance shift (measured by a change in accuracy) and the stability (measured by the change in different product recommendation). Hybrid algorithms tend to be less susceptible to profile attacks and system checks for registration prevent or limit the creation of bogus accounts (Mobasher et al., 2007). A number of detection techniques exist that reveal unusual or malicious patterns of profile creation and rating data, indicating an attack (Burke, Mobasher, Williams, & Bhaumik, 2006). Classical classification techniques (Hurley, Cheng, & Zhang, 2009) such as Bayesian classifier (Wu, Wu, Cao, & Tao, 2012), peer comparison (Chirita, Nejdil, & Zamfir, 2005; O'Mahony, Hurley, & Silvestre, 2005), but also neural networks are used for detecting deep level features that indicate shilling attacks (Tong et al., 2018). However, the success of classifying the suspicion is dependent on what is defined as malicious in the attack model used to build the classification.

- Finding: Reason who may be the potential adversaries, what their motives are and what they may gain. Set up monitoring systems in order to detect unusual patterns, integrate appropriate mechanisms that protect recommender systems against attacks, and consider designs that are less likely to be affected by attacks.

## 4. An example case of the recommender canvas: Sharebox

Here we present an example of an application of the canvas for designing a recommender for an online platform facilitating industrial symbiosis. Industrial symbiosis (IS) is a cooperation among industries, wherein waste streams as a secondary output of an industry are utilized as (part of) resources for the production processes of other industries (Chertow, 2000). Sharebox (Sharebox, 2019) is the selected online platform that functions both as a management tool for IS, as well as an anonymous marketplace for trading industrial waste and byproducts. Fig. 3 shows an application of

the canvas model to a real case of a recommender system design (based on van Capelleveen et al., 2018b) intended for the Sharebox platform. The canvas for the Sharebox case is constructed with the support of feedback from a consortia with expertise in field of: (a) the process industry (22 industrial cluster representatives), (b) industrial symbiosis facilitation (2 facilitator representatives), and (c) recommender system design and implementation (6 recommender system experts)). The triangulation of results from the semi-structured interviews (2 process industry representatives), a short survey (10 respondents), and a(n) (interactive) feedback session with members of the Sharebox consortia on the elicited high-level requirements (30 participants present)). The feedback data consists of discussion results on the design principles and end-user preferences regarding each of the aspects of the canvas in the context of designing a recommender for Sharebox. These have been aggregated and processed into the canvas presented in Fig. 3. This case illustrates the potential application of a recommendation canvas as a tool for developing a recommendation design. The following Section 5 discusses how such canvas may support the design process.

## 5. Discussion

### 5.1. Use of recommender system models

The research on the design of recommender systems from a requirement specification perspective is a rather nascent field of research. Yet, only a few concepts independently aim to support requirement analysts and recommender designers (typically the product owner and the developers in scrum teams) in their task to understand, communicate, design and analyze recommender systems from a coordinating development point of view. In the following section, we outline the potential roles a recommender canvas model can play during recommender system development. We identify two categories of functions. In addition we show the potential relevance for a canvas as an educational tool.

**(I) A tool for analyzing, sharing ideas, and creating common understanding when building initial designs of recommender systems.** The first application area where a recommender system model can contribute is in the analysis of building a recommender system from scratch. A generic framework may facilitate to develop ideas, elicit requirements, share ideas, and create a common understanding in the project. The benefit of using an (ontological) model is that it stimulates engineers to formulate their design thoughts, hence initiating a structured discussion on the set of design aspects thereby making the design decisions explicit, similar to how business model canvas facilitates collaborative discussions among stakeholders when creating new businesses (Osterwalder et al., 2004). The strength of using the model is that it is a simplified representation of the recommender system design, therefore it helps in communicating only the essential design decisions in an accessible way.

**(II) A tool for evaluating the current state of recommender system design and for reflecting on future prospects for recommender systems.** The second area of contribution takes place in the evaluation of existing recommender systems. In this regard, agile software development considers this a natural part of software evolution, where a state of the functionality or software design is documented to explore future requirements (where are we, where do we want to go, how to get there). Due to the constant change of domain characteristics and requirements, system engineers intermittently have to make adaptations to the recommender system design. Therefore, it may be constructive to understand which issues have changed over time and require a change or additional functionality in the recommender design. We think that the use of a canvas will make it easier to identify, plan and prioritize new

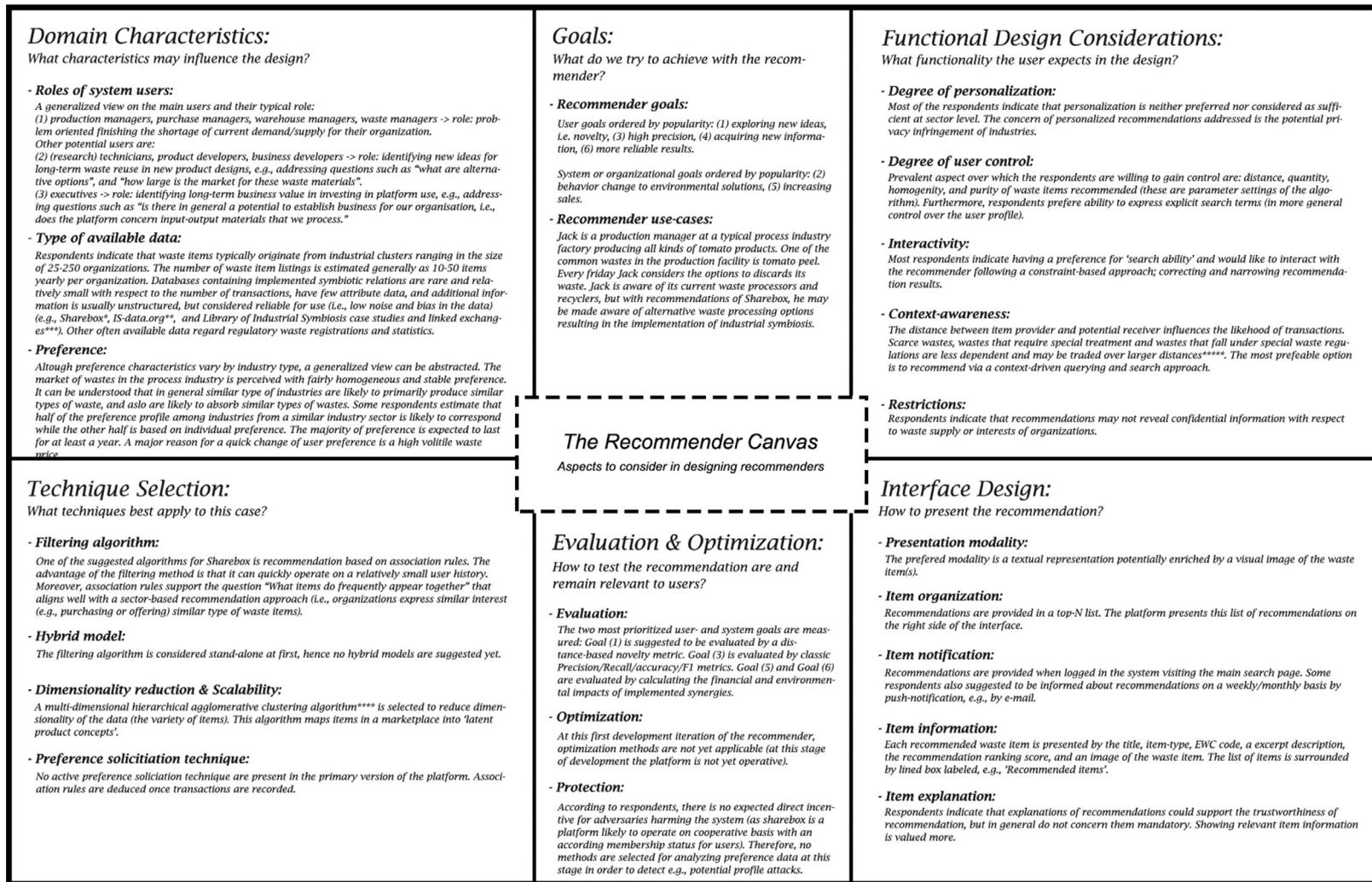


Fig. 3. The recommender canvas: The case of Sharebox \* (European Commission, 2017), \*\* (The ISDATA project, 2015), \*\*\* (Evans, Benedetti, & Holgado Granados, 2019), \*\*\*\* (van Capelleveen et al., 2018b), \*\*\*\*\* (Jensen, Basson, Hellowell, Bailey, & Leach, 2011).

design aspects for the existing recommender system and their fit in the current system architecture.

An additional benefit of the canvas could be to use it as a tool in education. Models in education are justified by the ability of students to learn principles, learn about doing science and being able to take part in conducting scientific research (Gilbert, Boulter, & Elmer, 2000; Justi & Gilbert, 2002). An example of a successful canvas in science education is the business model canvas (Osterwalder et al., 2004), which has been applied as a teaching tool in entrepreneurship classroom settings (Hixson & Paretti, 2014). The canvas function is to support the collection, organization, presentation of evidence, anchors participants in a common language, establishes conceptual relationships, helps in testing and refining hypotheses and in supporting social interactions and sense-making (Hixson & Paretti, 2014).

### 5.2. Internal & external validity

This recommender canvas is designed by studying the concepts, components, and issues listed in the recommender system literature. The extensive field of recommender systems lists numerous issues and challenges that affect recommender system design, making it difficult to include every detail within the limits of a single paper. However, we believe that with the structure presented, we have captured all essential elements needed for recommender design in sufficient detail. Secondly, one may derive the presented concepts as a sequential process to follow in recommender system design. However, one should note that the order originated from the synthesis recommender system design literature. Although this may align with well-practiced data mining approaches, the view of the authors is that agility in software development is not putting strict constraints on the order of acts, but rather is an interrelated set of concepts that is built together incrementally. Moreover, the order is less relevant as our main aim is to provide structure and explanatory guidelines for recommender system designers, not a step by step sequence of design considerations in the form of a guidance framework. Another threat to the internal validity of the canvas is its applicability. This requires extensive testing in future work, i.e., applying the canvas model in more real-life situations in order to evaluate the effectiveness of the canvas in facilitating recommender system development.

A threat to the external validity of this study is the ability to generalize the proposition aspect, being the recommender system canvas, to all cases of recommender system development projects. However, since our model is built upon a broad variety of cases found in the literature originating from all sorts of domain applications with an emphasis on different aspects of the design problem, it is arguable that our model could serve to describe and document recommender system designs independent of the problem space or domain of application. However, not necessarily every recommender system development project would benefit from the structured approach we present. In contrast, many software projects flourish on the basis of creativity rather than structural design (Dyb & Dingsyr, 2008; Highsmith & Cockburn, 2001). However, in projects with high complexity, we think the requirement specification approach we describe could be particular beneficial.

A further concern may be the limited temporal validity of the model, as new concepts may develop as a consequence of future technological developments in the field of recommender system. Models, however, have never had ample justification for remaining static (for an extended period), and many scientific models also grow through iterative development (Justi & Gilbert, 2002). Hence, this canvas may also be an initial design for ongoing recommender design models.

## 6. Conclusion

In this paper, we argue that the burgeoning concepts in recommender system development, the variety of choices, and the numerous alternative system designs may be overwhelming for practitioners to consider, and they would benefit from a structured ontological approach in coordinating design decisions. Our work resulted in a recommender system canvas developed based on a hermeneutic literature review (Boell & Cecez-Kecmanovic, 2014).

This canvas enables practitioners to create a high-level structured overview of recommender system designs while externalizing the relationships between interrelated concepts. Furthermore, it contributes a model to science illustrating the decision space of designing recommender systems. The canvas intends to serve as a requirement specification tool to analyze the fundamental questions of recommender system design to a broad audience of software engineers, software project managers, and education institutions. The tool may be used to support complex recommender system development projects, (1) the analysis, idea sharing, and creation of a common understanding when building initial designs of recommender systems, (2) the evaluation of the current state of recommender system design and the reflection of future prospects for recommender systems.

Although the work on modeling recommender system design is preliminary, there is a vast potential to explore this canvas further in different domain applications. Furthermore, the research may focus on understanding relationships among various canvas aspects and synthesizing guiding principles (i.e., 'best practices') for low-level decision making in each of the individual aspects. In conclusion the study provides a novel requirement specification perspective on recommender systems with the proposal of a canvas management tool to facilitate recommender system design.

### Author credit

Guido van Capelleveen (Phd Student, main author) Chintan Amrit (Daily supervisor) Devrim Murat Yazan (Daily supervisor) Henk Zijm (promotor).

### Acknowledgement

This research is funded by Horizon 2020 program under grant agreement No. 680843.

### References

- Abbar, S., Bouzeghoub, M., & Lopez, S. (2009). Context-aware recommender systems: A service-oriented approach. In *Very large data bases persdb workshop* (pp. 1–6).
- Adomavicius, G., Bockstedt, J. C., Curley, S. P., & Zhang, J. (2013). Do recommender systems manipulate consumer preferences? A study of anchoring effects. *Information Systems Research*, 24(4), 956–975. doi:10.1287/isre.2013.0497.
- Adomavicius, G., & Kwon, Y. (2007). New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems*, 22(3), 48–55. doi:10.1109/MIS.2007.58.
- Adomavicius, G., & Tuzhilin, A. (2005a). Personalization technologies: A process-oriented perspective. *Communications of the ACM*, 48(10), 83–90. doi:10.1145/1089107.1089109.
- Adomavicius, G., & Tuzhilin, A. (2005b). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749. doi:10.1109/TKDE.2005.99.
- Adomavicius, G., & Tuzhilin, A. (2015). Context-aware recommender systems. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender systems handbook* (pp. 191–226). Boston, MA: Springer US. doi:10.1007/978-1-4899-7637-6\_6.
- Adomavicius, G., & Zhang, J. (2012). Impact of data characteristics on recommender systems performance. *ACM Transactions on Management Information Systems*, 3(1), 3:1–3:17. doi:10.1145/2151163.2151166.
- Alhamid, M. F., Rawashdeh, M., Al Osman, H., Hossain, M. S., & El Saddik, A. (2015). Towards context-sensitive collaborative media recommender system. *Multimedia Tools and Applications*, 74(24), 11399–11428. doi:10.1007/s11042-014-2236-3.

- Amatriain, X., Pujol, J. M., & Oliver, N. (2009a). I like it . . . i like it not: Evaluating user ratings noise in recommender systems. In G.-J. Houben, G. McCalla, F. Pianesi, & M. Zancanaro (Eds.), *User modeling, adaptation, and personalization Berlin, Heidelberg*: Springer Berlin Heidelberg (pp. 247–258).
- Amatriain, X., Pujol, J. M., Tintarev, N., & Oliver, N. (2009b). Rate it again: Increasing recommendation accuracy by user re-rating. In *Proceedings of the third ACM conference on recommender systems*. In RecSys '09 New York, NY, USA: ACM (pp. 173–180). doi:10.1145/1639714.1639744.
- Amer-Yahia, S., Roy, S. B., Chawlat, A., Das, G., & Yu, C. (2009). Group recommendation: Semantics and efficiency. *Proceedings of the Very Large Database Endowment*, 2(1), 754–765. doi:10.14778/1687627.1687713.
- Asmus, J., Borchmann, D., Sbalzarini, I. F., & Walther, D. (2014). *Towards an FCA-based recommender system for black-box optimization Workshop notes* p. 35.
- Azaria, A., & Hong, J. (2016). Recommender systems with personality. In *Proceedings of the 10th ACM conference on recommender systems*. In RecSys '16 New York, NY, USA: ACM (pp. 207–210). doi:10.1145/2959100.2959138.
- Azevedo, A., Santos, M. F., et al. (2008). Kdd, semma and crisp-dm: a parallel overview. In *IADIS European conference of data mining: 8* (pp. 182–185).
- Baltrunas, L., & Amatriain, X. (2009). Towards time-dependant recommendation based on implicit feedback. *Workshop on context-aware recommender systems (cars'09)*.
- Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., et al. (2011). Incarmusic: Context-aware music recommendations in a car. In C. Huemer, & T. Setzer (Eds.), *E-commerce and web technologies* (pp. 89–100). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-23014-1\_8.
- Barragns-Martnez, B., Costa-Montenegro, E., & Juncal-Martnez, J. (2015). Developing a recommender system in a consumer electronic device. *Expert Systems with Applications*, 42(9), 4216–4228. doi:10.1016/j.eswa.2015.01.052.
- Barranco, M. J., Noguera, J. M., Castro, J., & Martnez, L. (2012). A context-aware mobile recommender system based on location and trajectory. In J. Casillas, F. J. Martínez-López, & J. M. Corchado Rodríguez (Eds.), *Management intelligent systems* (pp. 153–162). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-30864-2\_15.
- Barraza-Urbina, A. (2017). The exploration-exploitation trade-off in interactive recommender systems. In *Proceedings of the eleventh ACM conference on recommender systems*. In RecSys '17 (pp. 431–435). New York, NY, USA: ACM. doi:10.1145/3109859.3109866.
- Barrington, L., Oda, R., & Lanckriet, G. R. G. (2009). Smarter than genius? Human evaluation of music recommender systems. In *Proceedings of the 10th international society for music information retrieval conference, ISMIR 2009* (pp. 357–362).
- Basu Roy, S., Lakshmanan, L. V., & Liu, R. (2015). From group recommendations to group formation. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. In SIGMOD'15 (pp. 1603–1616). New York, NY, USA: ACM. doi:10.1145/2723372.2749448.
- Beam, M. A. (2014). Automating the news: How personalized news recommender system design choices impact news reception. *Communication Research*, 41(8), 1019–1041. doi:10.1177/0093650213497979.
- Beel, J., Gipp, B., Langer, S., & Breiting, C. (2016). Research-paper recommender systems: A literature survey. *International Journal on Digital Libraries*, 17(4), 305–338. doi:10.1007/s00799-015-0156-0.
- Belkin, N. J., & Croft, W. B. (1992). Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12), 29–38. doi:10.1145/138859.138861.
- Bellotti, V., Begole, B., Chi, E. H., Ducheneaut, N., Fang, J., Isaacs, E., et al. (2008). Activity-based serendipitous recommendations with the magitti mobile leisure guide. In *Proceedings of the SIGCHI conference on human factors in computing systems*. In CHI '08 (pp. 1157–1166). New York, NY, USA: ACM. doi:10.1145/1357054.1357237.
- Berkovsky, S., Kuflik, T., & Ricci, F. (2007). Distributed collaborative filtering with domain specialization. In *Proceedings of the 2007 ACM conference on recommender systems*. In RecSys '07 (pp. 33–40). New York, NY, USA: ACM. doi:10.1145/1297231.1297238.
- Billsus, D., & Pazzani, M. J. (2000). User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2), 147–180. doi:10.1023/A:1026501525781.
- Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012). A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26, 225–238. doi:10.1016/j.knsys.2011.07.021.
- Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, 109–132. doi:10.1016/j.knsys.2013.03.012.
- Boell, S. K., & Cecez-Kecmanovic, D. (2014). A hermeneutic approach for conducting literature reviews and literature searches. *Communications of the Association for Information Systems*, 34, 12.
- Bogdanov, D., Haro, M., Fuhrmann, F., Xambó, A., Gómez, E., & Herrera, P. (2013). Semantic audio content-based music recommendation and visualization based on user preference examples. *Information Processing & Management*, 49(1), 13–33. doi:10.1016/j.ipm.2012.06.004.
- Bollen, D., Knijnenburg, B. P., Willemssen, M. C., & Graus, M. (2010). Understanding choice overload in recommender systems. In *Proceedings of the fourth ACM conference on recommender systems*. In RecSys'10 (pp. 63–70). New York, NY, USA: ACM. doi:10.1145/1864708.1864724.
- Boratto, L., & Carta, S. (2011). State-of-the-art in group recommendation and new approaches for automatic identification of groups. In A. Soro, E. Vargiu, G. Armano, & G. Paddeu (Eds.), *Information retrieval and mining in distributed environments* (pp. 1–20). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-16089-9\_1.
- Bostandjiev, S., O'Donovan, J., & Höllerer, T. (2012). Tasteweights: A visual interactive hybrid recommender system. In *Proceedings of the sixth ACM conference on recommender systems*. In RecSys '12 (pp. 35–42). New York, NY, USA: ACM. doi:10.1145/2365952.2365964.
- Bouzekri, E., Canny, A., Fayollas, C., Martinie, C., Palanque, P., Barboni, E., et al. (2018). Engineering issues related to the development of a recommender system in a critical context: Application to interactive cockpits. *International Journal of Human-Computer Studies*. doi:10.1016/j.ijhcs.2018.05.001.
- Burke, R. (2000). Knowledge-based recommender systems. In A. Kent (Ed.), *Encyclopedia of library and information systems*: 69. CRC Press.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), 331–370. doi:10.1023/A:1021240730564.
- Burke, R. (2007). Hybrid web recommender systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 377–408). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-72079-9\_12.
- Burke, R., Mobasher, B., Williams, C., & Bhaumik, R. (2006). Classification features for attack detection in collaborative recommender systems. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*. In KDD'06 (pp. 542–547). New York, NY, USA: ACM. doi:10.1145/1150402.1150465.
- Burke, R., O'Mahony, M. P., & Hurley, N. J. (2015). Robust collaborative recommendation. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender systems handbook* (pp. 961–995). Boston, MA: Springer US. doi:10.1007/978-1-4899-7637-6\_28.
- Burke, R., & Ramezani, M. (2011). Matching recommendation technologies and domains. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 367–386). Boston, MA: Springer US. doi:10.1007/978-0-387-85820-3\_11.
- Cacheda, F., Carneiro, V., Fernández, D., & Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web*, 5(1), 2:1–2:33. doi:10.1145/1921591.1921593.
- Calero Valdez, A., Ziefle, M., & Verbert, K. (2016). Hci for recommender systems: The past, the present and the future. In *Proceedings of the 10th ACM conference on recommender systems*. In RecSys'16 (pp. 123–126). New York, NY, USA: ACM. doi:10.1145/2959100.2959158.
- Canny, J. (2002). Collaborative filtering with privacy via factor analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*. In SIGIR'02 (pp. 238–245). New York, NY, USA: ACM. doi:10.1145/564376.564419.
- Carenini, G., & Poole, D. (2002). Constructed preferences and value-focused thinking: Implications for ai research on preference elicitation. In *Aaai-02 workshop on preferences in AI and CP: Symbolic approaches* (pp. 1–10).
- Carenini, G., Smith, J., & Poole, D. (2003). Towards more conversational and collaborative recommender systems. In *Proceedings of the 8th international conference on intelligent user interfaces*. In IUI'03 (pp. 12–18). New York, NY, USA: ACM. doi:10.1145/604045.604052.
- Castagnos, S., Jones, N., & Pu, P. (2010). Eye-tracking product recommenders' usage. In *Proceedings of the fourth ACM conference on recommender systems*. In RecSys'10 (pp. 29–36). New York, NY, USA: ACM. doi:10.1145/1864708.1864717.
- Celma, O., & Cano, P. (2008). From hits to niches?: Or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2nd KDD workshop on large-scale recommender systems and the Netflix prize competition*. In NETFLIX'08 (pp. 5:1–5:8). New York, NY, USA: ACM. doi:10.1145/1722149.1722154.
- Chamoso, P., Rivas, A., Rodríguez, S., & Bajo, J. (2018). Relationship recommender system in a business and employment-oriented social network. *Information Sciences*, 433–434, 204–220. doi:10.1016/j.ins.2017.12.050.
- Champiri, Z. D., Shahamiri, S. R., & Salim, S. S. B. (2015). A systematic review of scholar context-aware recommender systems. *Expert Systems with Applications*, 42(3), 1743–1758. doi:10.1016/j.eswa.2014.09.017.
- Chan, S., Treleaven, P., & Capra, L. (2013). Continuous hyperparameter optimization for large-scale recommender systems. In *2013 IEEE international conference on big data* (pp. 350–358). doi:10.1109/BigData.2013.6691595.
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). Crisp-dm 1.0 step-by-step data mining guide. SPSS. <https://www.the-modeling-agency.com/crisp-dm.pdf>.
- Chen, L., & Pu, P. (2008). Interaction design guidelines on critiquing-based recommender systems. *User Modeling and User-Adapted Interaction*, 19(3), 167. doi:10.1007/s11257-008-9057-x.
- Chen, L., & Pu, P. (2010). Eye-tracking study of user behavior in recommender interfaces. In P. De Bra, A. Kobsa, & D. Chin (Eds.), *User modeling, adaptation, and personalization* (pp. 375–380). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-13470-8\_35.
- Chen, L., & Pu, P. (2012). Critiquing-based recommenders: Survey and emerging trends. *User Modeling and User-Adapted Interaction*, 22(1), 125–150. doi:10.1007/s11257-011-9108-6.
- Chen, Y. (2011). Interface and interaction design for group and social recommender systems. In *Proceedings of the fifth ACM conference on recommender systems*. In RecSys'11 (pp. 363–366). New York, NY, USA: ACM. doi:10.1145/2043932.2044007.

- Chen, Y., Ma, X., Cerezo, A., & Pu, P. (2014). Empatheticons: Designing emotion awareness tools for group recommenders. In *Proceedings of the XV international conference on human computer interaction*. In *Interaccion'14* (pp. 16:1–16:8). New York, NY, USA: ACM. doi:10.1145/2662253.2662269.
- Chen, Y.-F. (2008). Herd behavior in purchasing books online. *Computers in Human Behavior*, 24(5), 1977–1992. Including the Special Issue: Internet Empowerment. doi: 10.1016/j.chb.2007.08.004.
- Chertow, M. R. (2000). Industrial symbiosis: Literature and taxonomy. *Annual Review of Energy and the Environment*, 25(1), 313–337. doi:10.1146/annurev.energy.25.1.313.
- Cheung, K.-W., Kwok, J. T., Law, M. H., & Tsui, K.-C. (2003). Mining customer product ratings for personalized marketing. *Decision Support Systems*, 35(2), 231–243. Web Data Mining. doi: 10.1016/S0167-9236(02)00108-2.
- Chirita, P.-A., Nejdil, W., & Zamfir, C. (2005). Preventing shilling attacks in online recommender systems. In *Proceedings of the 7th annual ACM international workshop on web information and data management*. In *WIDM'05* (pp. 67–74). New York, NY, USA: ACM. doi:10.1145/1097047.1097061.
- Cockburn, A. (1997). Structuring use cases with goals. *Journal of Object-Oriented Programming*, 10(5), 56–62.
- Colombo-Mendoza, L. O., Valencia-Garca, R., Rodriguez-Gonzalez, A., Alor-Hernandez, G., & Samper-Zapater, J. J. (2015). Recometz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Systems with Applications*, 42(3), 1202–1222. doi:10.1016/j.eswa.2014.09.016.
- Cooke, A. D., Sujian, H., Sujian, M., & Weitz, B. A. (2002). Marketing the unfamiliar: The role of context and item-specific information in electronic agent recommendations. *Journal of Marketing Research*, 39(4), 488–497. doi:10.1509/jmkr.39.4.488.19121.
- Cosley, D., Lam, S. K., Albert, I., Konstan, J. A., & Riedl, J. (2003). Is seeing believing?: How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI conference on human factors in computing systems*. In *CHI'03* (pp. 585–592). New York, NY, USA: ACM. doi:10.1145/642611.642713.
- Cremonesi, P., Donatucci, A., Garzotto, F., & Turrin, R. (2012a). Decision-making in recommender systems: The role of user's goals and bounded resources. In *Decisions@ recsys* (pp. 1–7).
- Cremonesi, P., Garzotto, F., & Turrin, R. (2012b). Investigating the persuasion potential of recommender systems from a quality perspective: An empirical study. *ACM Transactions on Interactive Intelligent Systems*, 2(2), 11:1–11:41. doi:10.1145/2209310.2209314.
- Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on recommender systems*. In *RecSys'10* (pp. 39–46). New York, NY, USA: ACM. doi:10.1145/1864708.1864721.
- Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., et al. (2010). The youtube video recommendation system. In *Proceedings of the fourth ACM conference on recommender systems*. In *RecSys'10* (pp. 293–296). New York, NY, USA: ACM. doi:10.1145/1864708.1864770.
- Deshpande, M., & Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1), 143–177. doi:10.1145/963770.963776.
- Desrosiers, C., & Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 107–144). Boston, MA: Springer US. doi:10.1007/978-0-387-85820-3\_4.
- Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2), 97–166. doi:10.1207/S15327051HCI16234\_02.
- Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., & Zanker, M. (2012). Linked open data to support content-based recommender systems. In *Proceedings of the 8th international conference on semantic systems*. In *I-SEMANTICS'12* (pp. 1–8). New York, NY, USA: ACM. doi:10.1145/2362499.2362501.
- Ding, Y., & Li, X. (2005). Time weight collaborative filtering. In *Proceedings of the 14th ACM international conference on information and knowledge management*. In *CIKM'05* (pp. 485–492). New York, NY, USA: ACM. doi:10.1145/1099554.1099689.
- Drachsler, H., Hummel, H. G., & Koper, R. (2009). Identifying the goal, user model and conditions of recommender systems for formal and informal learning. *Journal of Digital Information*, 10(2), 1–14.
- Dyb, T., & Dingsyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9), 833–859. doi:10.1016/j.infsof.2008.01.006.
- Ekstrand, M. D., Kluver, D., Harper, F. M., & Konstan, J. A. (2015). Letting users choose recommender algorithms: An experimental study. In *Proceedings of the 9th ACM conference on recommender systems*. In *RecSys'15* (pp. 11–18). New York, NY, USA: ACM. doi:10.1145/2792838.2800195.
- Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2), 81–173. doi:10.1561/1100000009.
- Elahi, M., Ricci, F., & Rubens, N. (2014). Active learning strategies for rating elicitation in collaborative filtering: A system-wide perspective. *ACM Transactions on Intelligent Systems Technology*, 5(1), 13:1–13:33. doi:10.1145/2542182.2542195.
- Elahi, M., Ricci, F., & Rubens, N. (2016). A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20, 29–50. doi:10.1016/j.cosrev.2016.05.002.
- Erkin, Z., Veugen, T., Toft, T., & Lagendijk, R. L. (2012). Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE Transactions on Information Forensics and Security*, 7(3), 1053–1066. doi:10.1109/TIFS.2012.2190726.
- European Commission (2017). Horizon2020 project sharebox. Unpublished workshop data <https://www.sharebox-project.eu/>.
- Evans, S., Benedetti, M., & Holgado Granados, M. (2019). Library of industrial symbiosis case studies and linked exchanges. Data retrieved from, <https://doi.org/10.17863/CAM.12608>.
- Fayyad, U. M. (1997). Data mining and knowledge discovery in databases: Implications for scientific databases. In *Proceedings of the ninth international conference on scientific and statistical database management*. In *SSDBM'97* (pp. 2–11). Washington, DC, USA: IEEE Computer Society. doi:10.1109/SSDM.1997.621141.
- Felfernig, A., & Burke, R. (2008). Constraint-based recommender systems: Technologies and research issues. In *Proceedings of the 10th international conference on electronic commerce*. In *ICEC'08* (pp. 3:1–3:10). New York, NY, USA: ACM. doi:10.1145/1409540.1409544.
- Felfernig, A., Friedrich, G., Gula, B., Hitz, M., Kruggel, T., Leitner, G., et al. (2007). Persuasive recommendation: Serial position effects in knowledge-based recommender systems. *Persuasive Technology*, 283–294.
- Felfernig, A., Friedrich, G., Jannach, D., & Zanker, M. (2011). Developing constraint-based recommenders. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 187–215). Boston, MA: Springer US. doi:10.1007/978-0-387-85820-3\_6.
- Fernández-Tobias, I., Cantador, I., Kaminskas, M., & Ricci, F. (2012). Cross-domain recommender systems: A survey of the state of the art. In *Spanish conference on information retrieval* (p. 24).
- Fischer, G. (2012). Context-aware systems: The 'right' information, at the 'right' time, in the 'right' place, in the 'right' way, to the 'right' person. In *Proceedings of the international working conference on advanced visual interfaces*. In *AVI'12* (pp. 287–294). New York, NY, USA: ACM. doi:10.1145/2254556.2254611.
- Friedrich, G., & Zanker, M. (2011). A taxonomy for generating explanations in recommender systems. *AI Magazine*, 32(3), 90–98.
- Gallego, D., Woerndl, W., & Huecas, G. (2013). Evaluating the impact of proactivity in the user experience of a context-aware restaurant recommender for android smartphones. *Journal of Systems Architecture*, 59(9), 748–758. doi:10.1016/j.sysarc.2013.02.004.
- Gama, J. a., Zliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computer Surveys*, 46(4), 44:1–44:37. doi:10.1145/2523813.
- Gansner, E., Hu, Y., Kobourov, S., & Volinsky, C. (2009). Putting recommendations on the map: Visualizing clusters and relations. In *Proceedings of the third ACM conference on recommender systems*. In *RecSys'09* (pp. 345–348). New York, NY, USA: ACM. doi:10.1145/1639714.1639784.
- García, I., Pajares, S., Sebastia, L., & Onaindia, E. (2012). Preference elicitation techniques for group recommender systems. *Information Sciences*, 189, 155–175. doi:10.1016/j.ins.2011.11.037.
- Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014). Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, 39, 319–333. doi:10.1016/j.jnca.2013.04.006.
- Gedikli, F., Jannach, D., & Ge, M. (2014). How should i explain? a comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4), 367–382. doi:10.1016/j.ijhcs.2013.12.007.
- Ghazanfar, M. A., & Prgel-Bennett, A. (2014). Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. *Expert Systems with Applications*, 41(7), 3261–3275. doi:10.1016/j.eswa.2013.11.010.
- Gilbert, J. K., Boulter, C. J., & Elmer, R. (2000). Positioning models in science education and in design and technology education. In J. K. Gilbert, & C. J. Boulter (Eds.), *Developing models in science education* (pp. 3–17). Dordrecht: Springer Netherlands. doi:10.1007/978-94-010-0876-1\_1.
- Glowacka, D. (2017). Bandit algorithms in interactive information retrieval. In *Proceedings of the ACM SIGIR international conference on theory of information retrieval*. In *ICTIR'17* (pp. 327–328). New York, NY, USA: ACM. doi:10.1145/3121050.3121108.
- Golbandi, N., Koren, Y., & Lempel, R. (2010). On bootstrapping recommender systems. In *Proceedings of the 19th ACM international conference on information and knowledge management*. In *CIKM'10* (pp. 1805–1808). New York, NY, USA: ACM. doi:10.1145/1871437.1871734.
- Gomez-Urbe, C. A., & Hunt, N. (2015). The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*, 6(4), 13:1–13:19. doi:10.1145/2843948.
- Gonzalez, G., de la Rosa, J. L., Montaner, M., & Delfin, S. (2007). Embedding emotional context in recommender systems. In *2007 IEEE 23rd international conference on data engineering workshop* (pp. 845–852). doi:10.1109/ICDEW.2007.4401075.
- Grasch, P., Felfernig, A., & Reinfrank, F. (2013). Recoment: Towards critiquing-based recommendation with speech interaction. In *Proceedings of the 7th ACM conference on recommender systems*. In *RecSys'13* (pp. 157–164). New York, NY, USA: ACM. doi:10.1145/2507157.2507161.
- Gretarsson, B., O'Donovan, J., Bostandjiev, S., Hall, C., & Höllerer, T. (2009). Small-worlds: Visualizing social recommendations. *Computer Graphics Forum*, 29(3), 833–842. doi:10.1111/j.1467-8659.2009.01679.x.
- Gunawardana, A., & Shani, G. (2009). A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10, 2935–2962.
- Gunes, I., Kaleli, C., Bilge, A., & Polat, H. (2014). Shilling attacks against recommender systems: A comprehensive survey. *Artificial Intelligence Review*, 42(4), 767–799. doi:10.1007/s10462-012-9364-9.
- Guzmán, G., Torres-Ruiz, M., Tambonero, V., Lytras, M. D., López-Ramírez, B., Quintero, R., et al. (2018). A collaborative framework for sensing abnormal

- heart rate based on a recommender system: Semantic recommender system for healthcare. *Journal of Medical and Biological Engineering*. doi:10.1007/s40846-018-0421-y.
- Harper, F. M., Xu, F., Kaur, H., Condiff, K., Chang, S., & Terveen, L. (2015). Putting users in control of their recommendations. In *Proceedings of the 9th ACM conference on recommender systems*. In *RecSys'15* (pp. 3–10). New York, NY, USA: ACM. doi:10.1145/2792838.2800179.
- He, C., Parra, D., & Verbert, K. (2016). Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications*, 56, 9–27. doi:10.1016/j.eswa.2016.02.013.
- Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on computer supported cooperative work*. In *CSCW'00* (pp. 241–250). New York, NY, USA: ACM. doi:10.1145/358916.358995.
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5–53. doi:10.1145/963770.963772.
- Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120–127. doi:10.1109/2.947100.
- Hixson, C., & Paretti, M. C. (2014). Texts as tools to support innovation: Using the business model canvas to teach engineering entrepreneurs about audiences. In *2014 IEEE international professional communication conference (IPCC)* (pp. 1–7). doi:10.1109/IPCC.2014.7020368.
- Horling, B., & Lesser, V. (2004). A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, 19(4), 281–316. doi:10.1017/S026988905000317.
- Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *2008 eighth IEEE international conference on data mining* (pp. 263–272). doi:10.1109/ICDM.2008.22.
- Hurley, N., Cheng, Z., & Zhang, M. (2009). Statistical attack detection. In *Proceedings of the third ACM conference on recommender systems*. In *RecSys'09* (pp. 149–156). New York, NY, USA: ACM. doi:10.1145/1639714.1639740.
- International Organization for Standardization (2017). ISO 9000:2015 Quality management systems – fundamentals and vocabulary <https://www.iso.org/standard/45481.html>. Accessed: 2017-09-22.
- Jameson, A., & Smyth, B. (2007). Recommendation to groups. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 596–627). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-72079-9\_20.
- Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender systems: An introduction*. Cambridge University Press.
- Jannach, D., Zanker, M., Ge, M., & Gröning, M. (2012). Recommender systems in computer science and information systems—a landscape of research. In C. Huemer, & P. Lops (Eds.), *E-commerce and web technologies* (pp. 76–87). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Jensen, P. D., Basson, L., Hellowell, E. E., Bailey, M. R., & Leach, M. (2011). Quantifying “geographic proximity”: Experiences from the united kingdom’s national industrial symbiosis programme. *Resources, Conservation and Recycling*, 55(7), 703–712. doi:10.1016/j.resconrec.2011.02.003.
- Jugovac, M., & Jannach, D. (2017). Interacting with recommenders: Overview and research directions. *ACM Transactions on Interactive Intelligent Systems*, 7(3), 10:1–10:46. doi:10.1145/3001837.
- Jugovac, M., Jannach, D., & Lercche, L. (2017). Efficient optimization of multiple recommendation quality factors according to individual user tendencies. *Expert Systems with Applications*, 81, 321–331. doi:10.1016/j.eswa.2017.03.055.
- Justi, R. S., & Gilbert, J. K. (2002). Modelling, teachers’ views on the nature of modelling, and implications for the education of modellers. *International Journal of Science Education*, 24(4), 369–387. doi:10.1080/09500690110110142.
- Karatzoglou, A., Baltrunas, L., & Shi, Y. (2013). Learning to rank for recommender systems. In *Proceedings of the 7th ACM conference on recommender systems*. In *RecSys'13* (pp. 493–494). New York, NY, USA: ACM. doi:10.1145/2507157.2508063.
- Kataraya, R., & Verma, O. P. (2016). Recent developments in affective recommender systems. *Physica A: Statistical Mechanics and its Applications*, 461, 182–190. doi:10.1016/j.physa.2016.05.046.
- Kaur, H., Kumar, N., & Batra, S. (2018). An efficient multi-party scheme for privacy preserving collaborative filtering for healthcare recommender system. *Future Generation Computer Systems*. doi:10.1016/j.future.2018.03.017.
- Kim, J. K., Kim, H. K., & Cho, Y. H. (2008). A user-oriented contents recommendation system in peer-to-peer architecture. *Expert Systems with Applications*, 34(1), 300–312. doi:10.1016/j.eswa.2006.09.034.
- Knijnenburg, B. P., Bostandjiev, S., O'Donovan, J., & Kobsa, A. (2012a). Inspectability and control in social recommenders. In *Proceedings of the sixth ACM conference on recommender systems*. In *RecSys'12* (pp. 43–50). New York, NY, USA: ACM. doi:10.1145/2365952.2365966.
- Knijnenburg, B. P., & Kobsa, A. (2013). Making decisions about privacy: Information disclosure in context-aware recommender systems. *ACM Transactions on Interactive Intelligent Systems*, 3(3), 20:1–20:23. doi:10.1145/2499670.
- Knijnenburg, B. P., Reijmer, N. J., & Willemsen, M. C. (2011a). Each to his own: How different users call for different interaction methods in recommender systems. In *Proceedings of the fifth ACM conference on recommender systems*. In *RecSys'11* (pp. 141–148). New York, NY, USA: ACM. doi:10.1145/2043932.2043960.
- Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., & Newell, C. (2012b). Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4–5), 441–504. doi:10.1007/s11257-011-9118-4.
- Knijnenburg, B. P., Willemsen, M. C., & Kobsa, A. (2011b). A pragmatic procedure to support the user-centric evaluation of recommender systems. In *Proceedings of the fifth ACM conference on recommender systems*. In *RecSys'11* (pp. 321–324). New York, NY, USA: ACM. doi:10.1145/2043932.2043993.
- Koenigstein, N., Dror, G., & Koren, Y. (2011). Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the fifth ACM conference on recommender systems*. In *RecSys'11* (pp. 165–172). New York, NY, USA: ACM. doi:10.1145/2043932.2043964.
- Kohavi, R., Deng, A., Frasca, B., Walker, T., Xu, Y., & Pohlmann, N. (2013). Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining*. In *KDD'13* (pp. 1168–1176). New York, NY, USA: ACM. doi:10.1145/2487575.2488217.
- Konstan, J. A., & Riedl, J. (2012). Recommender systems: From algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22(1), 101–123. doi:10.1007/s11257-011-9112-x.
- Koren, Y. (2010). Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4), 89–97. doi:10.1145/1721654.1721677.
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30–37. doi:10.1109/MC.2009.263.
- Koychev, I., & Schwab, I. (2000). Adaptation to drifting user’s interests. In *Proceedings of ECML 2000 workshop: Machine learning in new information age* (pp. 39–46).
- Lam, S. K., & Riedl, J. (2004). Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on world wide web*. In *WWW'04* (pp. 393–402). New York, NY, USA: ACM. doi:10.1145/988672.988726.
- Lee, N., Jung, J. J., Selamat, A., & Hwang, D. (2014a). Black-box testing of practical movie recommendation systems: A comparative study. *Computer Science and Information Systems*, 11(1), 241–249. doi:10.2298/CSIS130226006L.
- Lee, T. Q., Park, Y., & Park, Y.-T. (2008). A time-based approach to effective recommender systems using implicit feedback. *Expert Systems with Applications*, 34(4), 3055–3062. doi:10.1016/j.eswa.2007.06.031.
- Lee, W.-P., Kao, C., & Huang, J.-Y. (2014b). A smart tv system with body-gesture control, tag-based rating and context-aware recommendation. *Knowledge-Based Systems*, 56, 167–178. doi:10.1016/j.knosys.2013.11.007.
- Li, J., & Zaiane, O. (2004). Combining usage, content, and structure data to improve web site recommendation. *E-Commerce and Web Technologies*, 313–315.
- Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on world wide web*. In *WWW'10* (pp. 661–670). New York, NY, USA: ACM. doi:10.1145/1772690.1772758.
- Li, L., Zheng, L., & Li, T. (2011). Logo: A long-short user interest integration in personalized news recommendation. In *Proceedings of the fifth ACM conference on recommender systems*. In *RecSys'11* (pp. 317–320). New York, NY, USA: ACM. doi:10.1145/2043932.2043992.
- Lika, B., Kolomvatsos, K., & Hadjiethymiadis, S. (2014). Facing the cold start problem in recommender systems. *Expert Systems with Applications*, 41(4, Part 2), 2065–2073. doi:10.1016/j.eswa.2013.09.005.
- Lin, W., Alvarez, S. A., & Ruiz, C. (2002). Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6(1), 83–105. doi:10.1023/A:1013284820704.
- Liu, J., Dolan, P., & Pedersen, E. R. (2010). Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on intelligent user interfaces*. In *IUI'10* (pp. 31–40). New York, NY, USA: ACM. doi:10.1145/1719970.1719976.
- Liu, Q., Ma, H., Chen, E., & Xiong, H. (2013). A survey of context-aware mobile recommendations. *International Journal of Information Technology & Decision Making*, 12(01), 139–172. doi:10.1142/S0219622013500077.
- Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 73–105). Boston, MA: Springer US. doi:10.1007/978-0-387-85820-3\_3.
- Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems*, 74, 12–32. doi:10.1016/j.dss.2015.03.008.
- Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., & Zhou, T. (2012). Recommender systems. *Physics Reports*, 519(1), 1–49.
- Ma, H., Zhou, D., Liu, C., Lyu, M. R., & King, I. (2011). Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on web search and data mining*. In *WSDM'11* (pp. 287–296). New York, NY, USA: ACM. doi:10.1145/1935826.1935877.
- Ma, T., Zhou, J., Tang, M., Tian, Y., Al-Dhelaan, A., Al-Rodhaan, M., & Lee, S. (2015). Social network and tag sources based augmenting collaborative recommender system. *IEICE Transactions on Information and Systems*, 98, 902–910.
- Macedo, A. Q., Marinho, L. B., & Santos, R. L. (2015). Context-aware event recommendation in event-based social networks. In *Proceedings of the 9th ACM conference on recommender systems*. In *RecSys'15* (pp. 123–130). New York, NY, USA: ACM. doi:10.1145/2792838.2800187.
- Mahmood, T., & Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM conference on hypertext and hypermedia*. In *HT'09* (pp. 73–82). New York, NY, USA: ACM. doi:10.1145/1557914.1557930.
- Manouselis, N., & Costopoulou, C. (2007). Analysis and classification of multi-criteria recommender systems. *World Wide Web*, 10(4), 415–441. doi:10.1007/s11280-007-0019-8.
- Manouselis, N., Vuorikari, R., & Van Assche, F. (2010). Collaborative recommendation of e-learning resources: an experimental investigation. *Journal of Computer Assisted Learning*, 26(4), 227–242. doi:10.1111/j.1365-2729.2010.00362.x.

- Massa, P., & Avesani, P. (2007). Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on recommender systems*. In *RecSys'07* (pp. 17–24). New York, NY, USA: ACM. doi:10.1145/1297231.1297235.
- Matyas, C., & Schlieder, C. (2009). A spatial user similarity measure for geographic recommender systems. In K. Janowicz, M. Raubal, & S. Levashkin (Eds.), *Geospatial semantics* (pp. 122–139). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-10436-7\_8.
- McNee, S., Lam, S., Konstan, J., & Riedl, J. (2003). Interfaces for eliciting new user preferences in recommender systems. *User Modeling 2003*. 148–148. doi:10.1007/3-540-44963-9\_24.
- McNee, S. M., Riedl, J., & Konstan, J. A. (2006a). Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on human factors in computing systems*. In *CHI EA'06* (pp. 1097–1101). New York, NY, USA: ACM. doi:10.1145/1125451.1125659.
- McNee, S. M., Riedl, J., & Konstan, J. A. (2006b). Making recommendations better: An analytic model for human-recommender interaction. In *CHI'06 extended abstracts on human factors in computing systems*. In *CHI EA'06* (pp. 1103–1108). New York, NY, USA: ACM. doi:10.1145/1125451.1125660.
- McSherry, F., & Mironov, I. (2009). Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining*. In *KDD'09* (pp. 627–636). New York, NY, USA: ACM. doi:10.1145/1557019.1557090.
- Mobasher, B., Burke, R., Bhaumik, R., & Williams, C. (2007). Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology*, 7(4). doi:10.1145/1278366.1278372.
- Mun, M., Hao, S., Mishra, N., Shilton, K., Burke, J., Estrin, D., et al. (2010). Personal data vaults: A locus of control for personal data streams. In *Proceedings of the 6th international conference*. In *Co-NEXT'10* (pp. 17:1–17:12). New York, NY, USA: ACM. doi:10.1145/1921168.1921191.
- Nanou, T., Lekakos, G., & Fouskas, K. (2010). The effects of recommendations' presentation on persuasion and satisfaction in a movie recommender system. *Multimedia Systems*, 16(4), 219–230. doi:10.1007/s00530-010-0190-0.
- Neidhardt, J., Seyfang, L., Schuster, R., & Werthner, H. (2015). A picture-based approach to recommender systems. *Information Technology & Tourism*, 15(1), 49–69. doi:10.1007/s40558-014-0017-5.
- O'Donovan, J., Smyth, B., Gretarsson, B., Bostandjiev, S., & Höllerer, T. (2008). Peerchooser: Visual interactive recommendation. In *Proceedings of the SIGCHI conference on human factors in computing systems*. In *CHI'08* (pp. 1085–1088). New York, NY, USA: ACM. doi:10.1145/1357054.1357222.
- O'Mahony, M., Hurley, N., Kushmerick, N., & Silvestre, G. (2004). Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology*, 4(4), 344–377. doi:10.1145/1031114.1031116.
- O'Mahony, M. P., Hurley, N. J., & Silvestre, G. C. (2006). Detecting noise in recommender system databases. In *Proceedings of the 11th international conference on intelligent user interfaces*. In *IUI'06* (pp. 109–115). New York, NY, USA: ACM. doi:10.1145/1111449.1111477.
- Osterwalder, A., et al. (2004). *The business model ontology: A proposition in a design science approach* Ph.D. thesis.
- O'Mahony, M. P., Hurley, N. J., & Silvestre, G. C. M. (2005). *Recommender systems: Attack types and strategies*. In *proceedings of the 20th national conference on artificial intelligence (AAAI-05)* (pp. 334–339). AAAI Press.
- Palopoli, L., Rosaci, D., & Sarné, G. M. L. (2013). A multi-tiered recommender system architecture for supporting e-commerce. In G. Fortino, C. Badica, M. Malgeri, & R. Unland (Eds.), *Intelligent distributed computing VI* (pp. 71–81). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Papadimitriou, A., Symeonidis, P., & Manolopoulos, Y. (2012). A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining and Knowledge Discovery*, 24(3), 555–583. doi:10.1007/s10618-011-0215-0.
- Park, D. H., Kim, H. K., Choi, I. Y., & Kim, J. K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11), 10059–10072. doi:10.1016/j.eswa.2012.02.038.
- Park, S.-T., Pennock, D., Madani, O., Good, N., & DeCoste, D. (2006). Naïve filterbots for robust cold-start recommendations. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*. In *KDD'06* (pp. 699–705). New York, NY, USA: ACM. doi:10.1145/1150402.1150490.
- Parra, D., Brusilovsky, P., & Trattner, C. (2014). See what you want to see: Visual user-driven approach for hybrid recommendation. In *Proceedings of the 19th international conference on intelligent user interfaces*. In *IUI'14* (pp. 235–240). New York, NY, USA: ACM. doi:10.1145/2557500.2557542.
- Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5), 393–408. doi:10.1023/A:1006544522159.
- Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 325–341). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-72079-9\_10.
- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. doi:10.2753/MIS0742-1222240302.
- Pereira, C. K., Campos, F., Ströbele, V., David, J. M. N., & Braga, R. (2018). Broad-educational recommender system using social networks interactions and linked data. *Journal of Internet Services and Applications*, 9(1), 7. doi:10.1186/s13174-018-0076-5.
- Pham, X. H., Luong, T. N., & Jung, J. J. (2013). An black-box testing approach on user modeling in practical movie recommendation systems. In C. Bădică, N. T. Nguyen, & M. Brezovan (Eds.), *Computational collective intelligence. Technologies and applications: 5th International conference, ICCCI 2013, Craiova, Romania, September 11–13, 2013, proceedings* (pp. 72–79). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-40495-5\_8.
- Pielot, M., Dinger, T., Pedro, J. S., & Oliver, N. (2015). When attention is not scarce - detecting boredom from mobile phone usage. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*. In *UbiComp'15* (pp. 825–836). New York, NY, USA: ACM. doi:10.1145/2750858.2804252.
- Pommeranz, A., Broekens, J., Wiggers, P., Brinkman, W.-P., & Jonker, C. M. (2012). Designing interfaces for explicit preference elicitation: A user-centered investigation of preference representation and elicitation process. *User Modeling and User-Adapted Interaction*, 22(4), 357–397. doi:10.1007/s11257-011-9116-6.
- Popescul, A., Pennock, D. M., & Lawrence, S. (2001). Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the seventeenth conference on uncertainty in artificial intelligence*. In *UAI'01* (pp. 437–444). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc..
- Pu, P., Chen, L., & Hu, R. (2011). A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on recommender systems*. In *RecSys'11* (pp. 157–164). New York, NY, USA: ACM. doi:10.1145/2043932.2043962.
- Pu, P., Chen, L., & Hu, R. (2012). Evaluating recommender systems from the user's perspective: Survey of the state of the art. *User Modeling and User-Adapted Interaction*, 22(4), 317–355. doi:10.1007/s11257-011-9115-7.
- Rafailidis, D., Kefalas, P., & Manolopoulos, Y. (2017). Preference dynamics with multimodal user-item interactions in social media recommendation. *Expert Systems with Applications*, 74, 11–18. doi:10.1016/j.eswa.2017.01.005.
- Ramakrishnan, N., Keller, B. J., Mirza, B. J., Grama, A. Y., & Karypis, G. (2001). Privacy risks in recommender systems. *IEEE Internet Computing*, 5(6), 54–62. doi:10.1109/4236.968832.
- Rodriguez, M., Posse, C., & Zhang, E. (2012). Multiple objective optimization in recommender systems. In *Proceedings of the sixth ACM conference on recommender systems*. In *RecSys'12* (pp. 11–18). New York, NY, USA: ACM. doi:10.1145/2365952.2365961.
- Ruffo, G., & Schifanella, R. (2009). A peer-to-peer recommender system based on spontaneous affinities. *ACM Transactions on Internet Technology*, 9(1), 4:1–4:34. doi:10.1145/1462159.1462163.
- Sahoo, N., Krishnan, R., Duncan, G., & Callan, J. (2012). Research note - the halo effect in multicomponent ratings and its implications for recommender systems: The case of yahoo! movies. *Information Systems Research*, 23(1), 231–246. doi:10.1287/isre.1100.0336.
- Said, A., & Bellogin, A. (2014). Comparative recommender system evaluation: Benchmarking recommendation frameworks. In *Proceedings of the 8th ACM conference on recommender systems*. In *RecSys'14* (pp. 129–136). New York, NY, USA: ACM. doi:10.1145/2645710.2645746.
- Said, A., Fields, B., Jain, B. J., & Albayrak, S. (2013). User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm. In *Proceedings of the 2013 conference on computer supported cooperative work*. In *CSCW'13* (pp. 1399–1408). New York, NY, USA: ACM. doi:10.1145/2441776.2441933.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Application of dimensionality reduction in recommender system—a case study. *Technical Report*. Minnesota Univ Minneapolis Dept of Computer Science.
- SAS Institute Inc. (2013). *Data mining using SAS enterprise miner: A case study approach* (3rd ed.). SAS Publishing.
- Saveski, M., & Mantrach, A. (2014). Item cold-start recommendations: Learning local collective embeddings. In *Proceedings of the 8th ACM conference on recommender systems*. In *RecSys'14* (pp. 89–96). New York, NY, USA: ACM. doi:10.1145/2645710.2645751.
- Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 291–324). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-72079-9\_9.
- Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1), 115–153. doi:10.1023/A:1009804230409.
- Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 257–297). Boston, MA: Springer US. doi:10.1007/978-0-387-85820-3\_8.
- Sharebox (2019). Sharebox platform. <https://www.sharebox-project.eu/>.
- Shi, X., Luo, X., Shang, M., & Gu, L. (2017). Long-term performance of collaborative filtering based recommenders in temporally evolving systems. *Neurocomputing*, 267, 635–643. doi:10.1016/j.neucom.2017.06.026.
- Sinha, R., & Swearingen, K. (2002). The role of transparency in recommender systems. In *CHI'02 extended abstracts on human factors in computing systems*. In *CHI EA'02* (pp. 830–831). New York, NY, USA: ACM. doi:10.1145/506443.506619.
- Smyth, B. (2007). Case-based recommendation. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 342–376). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-72079-9\_11.

- Sparling, E. I., & Sen, S. (2011). Rating: How difficult is it? In *Proceedings of the fifth ACM conference on recommender systems In RecSys'11* (pp. 149–156). New York, NY, USA: ACM. doi:10.1145/2043932.2043961.
- Steck, H. (2010). Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining. In KDD'10* (pp. 713–722). New York, NY, USA: ACM. doi:10.1145/1835804.1835895.
- Su, H., Zheng, K., Huang, J., Jeung, H., Chen, L., & Zhou, X. (2014). Crowdplanner: A crowd-based route recommendation system. In *2014 IEEE 30th international conference on data engineering* (pp. 1144–1155). doi:10.1109/ICDE.2014.6816730.
- Swearingen, K., & Sinha, R. (2001). Beyond algorithms: An hci perspective on recommender systems. In *ACM SIGIR 2001 workshop on recommender systems: 13* (pp. 1–11).
- Swearingen, K., & Sinha, R. (2002). Interaction design for recommender systems. *Designing interactive systems*. ACM Press.
- Tang, J., Hu, X., & Liu, H. (2013). Social recommendation: A review. *Social Network Analysis and Mining*, 3(4), 1113–1133. doi:10.1007/s13278-013-0141-9.
- Teo, C. H., Nassif, H., Hill, D., Srinivasan, S., Goodman, M., Mohan, V., & Vishwanathan, S. (2016). Adaptive, personalized diversity for visual discovery. In *Proceedings of the 10th ACM conference on recommender systems. In RecSys'16* (pp. 35–38). New York, NY, USA: ACM. doi:10.1145/2959100.2959171.
- The ISDATA project (2015). The industrial symbiosis data repository. Accessed: 2017-01-31 <https://isdata.org/>.
- Tintarev, N., & Masthoff, J. (2007). A survey of explanations in recommender systems. In *2007 IEEE 23rd international conference on data engineering workshop* (pp. 801–810). doi:10.1109/ICDEW.2007.4401070.
- Tintarev, N., & Masthoff, J. (2011). Designing and evaluating explanations for recommender systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 479–510). Boston, MA: Springer US. doi:10.1007/978-0-387-85820-3\_15.
- Tintarev, N., & Masthoff, J. (2012). Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22(4), 399–439. doi:10.1007/s11257-011-9117-5.
- Tkalcic, M., Kosir, A., & Tasic, J. (2011). Affective recommender systems: The role of emotions in recommender systems. In *Proceedings of the Re Cysys 2011 workshop on human decision making in recommender systems* (pp. 9–13).
- Tong, C., Yin, X., Li, J., Zhu, T., Lv, R., Sun, L., & Rodrigues, J. J. P. C. (2018). A shilling attack detector based on convolutional neural network for collaborative recommender system in social aware network. *The Computer Journal*. doi:10.1093/comjnl/bxy008.
- Tsymbal, A. (c). 2004 The problem of concept drift: Definitions and related work. Computer Science Department, Trinity College Dublin.
- Uschold, M., & Gruninger, M. (1996). *Ontologies: Principles, methods and applications. The knowledge engineering review*, 11(2), 93–136.
- van Capelleveen, G., Amrit, C., & Yazan, D. M. (2018a). A literature survey of information systems facilitating the identification of industrial symbiosis. In B. Otjacques, P. Hitzelberger, S. Naumann, & V. Wohlgemuth (Eds.), *From science to society: New trends in environmental informatics* (pp. 155–169). Cham: Springer International Publishing. doi:10.1007/978-3-319-65687-8\_14.
- van Capelleveen, G., Amrit, C., Yazan, D. M., & Zijm, H. (2018b). The influence of knowledge in the design of a recommender system to facilitate industrial symbiosis markets. *Environmental Modelling & Software*. doi:10.1016/j.envsoft.2018.04.004.
- van der Spoel, S. (2016). *Prediction instrument development for complex domains*. University of Twente Ph.D. thesis. doi:10.3990/19789036541749.
- Vargas, S., & Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on recommender systems. In RecSys'11* (pp. 109–116). New York, NY, USA: ACM. doi:10.1145/2043932.2043955.
- Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachler, H., Bosnic, I., & Duval, E. (2012). Context-aware recommender systems for learning: A survey and future challenges. *IEEE Transactions on Learning Technologies*, 5(4), 318–335. doi:10.1109/TLT.2012.11.
- Verbert, K., Parra, D., & Brusilovsky, P. (2014). *The effect of different set-based visualizations on user exploration of recommendations* (1253, pp. 37–44).
- Verbert, K., Parra, D., Brusilovsky, P., & Duval, E. (2013). Visualizing recommendations to support exploration, transparency and controllability. In *Proceedings of the 2013 international conference on intelligent user interfaces. In IUI'13* (pp. 351–362). New York, NY, USA: ACM. doi:10.1145/2449396.2449442.
- Walsh, G., & Golbeck, J. (2010). Curator: A game with a purpose for collection recommendation. In *Proceedings of the SIGCHI conference on human factors in computing systems. In CHI'10* (pp. 2079–2082). New York, NY, USA: ACM. doi:10.1145/1753326.1753643.
- Wang, J., & Zhang, Y. (2013). Opportunity model for e-commerce recommendation: Right product; right time. In *Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval. In SIGIR'13* (pp. 303–312). New York, NY, USA: ACM. doi:10.1145/2484028.2484067.
- Wang, Y., Chan, S. C.-F., & Ngai, G. (2012). Applicability of demographic recommender system to tourist attractions: A case study on trip advisor. In *Proceedings of the 2012 IEEE/WIC/ACM international joint conferences on web intelligence and intelligent agent technology - vol. 03. In WI-IAT'12* (pp. 97–101). Washington, DC, USA: IEEE Computer Society. doi:10.1109/WI-IAT.2012.133.
- Williams, C. A., Mobasher, B., & Burke, R. (2007). Defending recommender systems: Detection of profile injection attacks. *Service Oriented Computing and Applications*, 1(3), 157–170. doi:10.1007/s11761-007-0013-0.
- William of Occam (c.1320). Entities should not be multiplied unnecessarily. Comment regarding superfluous elaboration.
- Woerndl, W., Huebner, J., Bader, R., & Gallego-Vico, D. (2011). A model for proactivity in mobile, context-aware recommender systems. In *Proceedings of the fifth ACM conference on recommender systems. In RecSys'11* (pp. 273–276). New York, NY, USA: ACM. doi:10.1145/2043932.2043981.
- Wörndl, W., Weicker, J., & Lamche, B. (2013). Selecting gestural user interaction patterns for recommender applications on smartphones. In *Decisions@ recsys* (pp. 17–20).
- Wu, D., Im, I., Tremaine, M., Instone, K., & Turoff, M. (2003). A framework for classifying personalization scheme used on e-commerce websites. In *36th Annual hawaii international conference on system sciences, 2003. Proceedings of the*. pp.–12. doi:10.1109/HICSS.2003.1174586.
- Wu, Z., Wu, J., Cao, J., & Tao, D. (2012). Hysad: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining. In KDD'12* (pp. 985–993). New York, NY, USA: ACM. doi:10.1145/2339530.2339684.
- Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., & Sun, J. (2010). Temporal recommendation on graphs via long- and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining. In KDD'10* (pp. 723–732). New York, NY, USA: ACM. doi:10.1145/1835804.1835896.
- Xiao, B., & Benbasat, I. (2007). E-commerce product recommendation agents: Use, characteristics, and impact. *MIS Quarterly*, 31(1), 137–209. doi:10.2307/25148784.
- Xu, H., Luo, X. R., Carroll, J. M., & Rosson, M. B. (2011). The personalization privacy paradox: An exploratory study of decision making process for location-aware marketing. *Decision Support Systems*, 51(1), 42–52. doi:10.1016/j.dss.2010.11.017.
- Yang, W.-S., & Hwang, S.-Y. (2013). itravel: A recommender system in mobile peer-to-peer environment. *Journal of Systems and Software*, 86(1), 12–20. doi:10.1016/j.jss.2012.06.041.
- Yang, X., Guo, Y., Liu, Y., & Steck, H. (2014). A survey of collaborative filtering based social recommender systems. *Computer Communications*, 41, 1–10. doi:10.1016/j.comcom.2013.06.009.
- Yi, X., Hong, L., Zhong, E., Liu, N. N., & Rajan, S. (2014). Beyond clicks: Dwell time for personalization. In *Proceedings of the 8th ACM conference on recommender systems* (pp. 113–120). ACM.
- Yin, H., Cui, B., Li, J., Yao, J., & Chen, C. (2012). Challenging the long tail recommendation. *Proceedings of the VLDB Endowment*, 5(9), 896–907. doi:10.14778/2311906.2311916.
- Zachary, W. (1986). A cognitively based functional taxonomy of decision support techniques. *Human-Computer Interaction*, 2(1), 25–63. doi:10.1207/s15327051hci0201\_2.
- Zeng, W., Zeng, A., Liu, H., Shang, M.-S., & Zhou, T. (2014). Uncovering the information core in recommender systems. *Scientific Reports*, 4, 6140. doi:10.1038/srep06140.
- Zhang, C.-J., & Zeng, A. (2012). Behavior patterns of online users and the effect on information filtering. *Physica A: Statistical Mechanics and its Applications*, 391(4), 1822–1830. doi:10.1016/j.physa.2011.09.038.
- Zhang, J., & Curley, S. P. (2017). Exploring explanation effects on consumers' trust in online recommender agents. *International Journal of Human-Computer Interaction*, 1–12. doi:10.1080/10447318.2017.1357904.
- Zhang, M., & Hurley, N. (2008). Avoiding monotony: Improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on recommender systems. In RecSys'08* (pp. 123–130). New York, NY, USA: ACM. doi:10.1145/1454008.1454030.
- Zhao, Q., Chang, S., Harper, F. M., & Konstan, J. A. (2016). Gaze prediction for recommender systems. In *Proceedings of the 10th ACM conference on recommender systems. In RecSys'16* (pp. 131–138). New York, NY, USA: ACM. doi:10.1145/2959100.2959150.
- Zhu, H., Xiong, H., Ge, Y., & Chen, E. (2014). Mobile app recommendations with security and privacy awareness. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. In KDD'14* (pp. 951–960). New York, NY, USA: ACM. doi:10.1145/2623330.2623705.