Information & Management 56 (2019) 103143

Contents lists available at ScienceDirect

Information & Management

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

Recommendation strategies in personalization applications

Michele Gorgoglione[a], Umberto Panniello[a,*], Alexander Tuzhilin[b]

[a] Politecnico di Bari, Bari, Italy
[b] Stern School of Business, New York University, New York, USA

ARTICLE INFO

Keywords:
Recommender system
Personalization
Recommendation strategy
Recommender life cycle
User–business interaction

ABSTRACT

While the initial goal of recommender systems (RSes) was to reduce the information overload for Internet users and make the information retrieval more efficient, they have become a crucial strategic tool for companies in the online markets. According to this evolution, research on RSes has produced a wide variety of approaches and algorithms. As a consequence, the companies deploying RSes in their business applications face the decision of how to generate and deliver personalized recommendations to their users by choosing among many options. The problem has been largely treated from the machine learning performance perspective because there is relatively little research done from the business perspective. The decision of what kind of recommender engines should be used in a personalization application, given certain business conditions, has a strategic value because it affects the way customers perceive the company with respect to its competitors. Choosing the wrong way to personalize recommendations may not only require the redesign of the information systems but also to rebuild the relationships with customers and even the entire brand strategic positioning. The research issues addressed by this paper are (i) which recommendation strategies a company can deploy to generate and deliver recommendations to users, and (ii) which specific strategies should be used depending on the current business conditions. We propose taxonomy based on a literature analysis and a framework to associate each strategy with a certain setting. The proposed framework is empirically supported by four case studies.

1. Introduction

Recommender systems (RSes) were first developed by researchers to reduce the information overload for Internet users and make information retrieval more efficient. With the development of online markets, they have quickly become of strategic importance to many different online businesses, such as Netflix, Amazon, Spotify, and Stitch Fix. To study and advance RS technologies, researchers have produced a wide variety of approaches and algorithms. As a consequence, companies can choose among many options. The problem of how to generate and deliver personalized recommendations to users has been largely treated from the machine learning performance perspective. Several kinds of recommendation engines have been compared against each other by using several performance metrics including accuracy, novelty, diversity, and trust [1]. Although RSes have been previously studied from the business perspective by proposing algorithms aiming at maximizing business outcomes, such as profit or sales or revenues [2–4], this view has been underexplored in comparison to the machine-learning-driven methods, which is unfortunate because this approach is of crucial importance to the business world because the ultimate goal of an RS is to provide ways for customers to interact with the business, enhance their experiences, and let businesses succeed in the long run [5].

Furthermore, the business-performance-centric approach to RSes requires adopting a new perspective that is different from the traditional accuracy-maximization approaches that have been predominant in the literature on RSes [5]. For example, consider the evolution of recommendation applications over the last several years. In the early days, accuracy-driven RSes had serious problems with delivering relevant recommendations and were often frustrating for the users, making recommendation tasks marginal compared to other business tasks. For example, Kemp noted that "no set of applications has disappointed as much as personalization has" [6]. The situation changed dramatically ten years later when many companies, including Amazon, Netflix, and LinkedIn, changed their focus to the business performance metrics, resulting in more effective and mission-critical RSes for many of them. For example, 80% of hours streamed by Netflix customers are determined by their recommendation algorithms [5], which has massive impact on the company, given their annual revenues of $8.83 billion in 2016.[1] When the Netflix prize was launched in 2006, the main problem was in developing the best algorithm that would most

* Corresponding author.
E-mail addresses: michele.gorgoglione@poliba.it (M. Gorgoglione), umberto.panniello@poliba.it (U. Panniello), atuzhili@stern.nyu.edu (A. Tuzhilin).

[1] As stated by Wikipedia at en.wikipedia.org/wiki/Netflix

https://doi.org/10.1016/j.im.2019.01.005
Received 4 January 2018; Received in revised form 7 January 2019; Accepted 20 January 2019
Available online 23 January 2019
0378-7206/

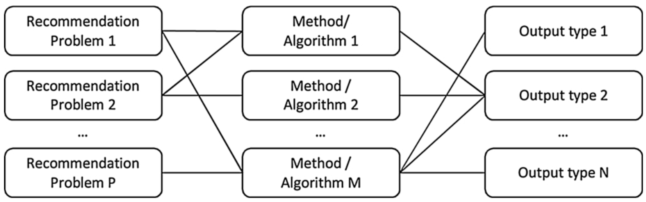

Fig. 1. Recommendation problems, methods, and output in a company.

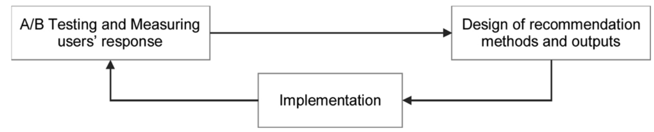

Fig. 2. Design, implementation, and test of recommendation methods and outputs.sssss.

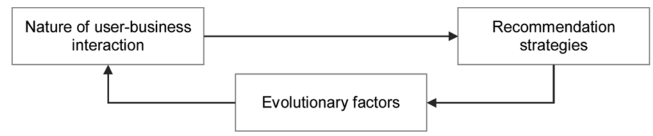

Fig. 3. A model to define recommendation strategies.

accurately estimate the unknown ratings. Since then, the focus shifted significantly at Netflix from the accurate rating prediction problem to the issues of dealing with an increasing number of customers, handling big data, generating recommendations of a constantly improving offering, delivering first-rate experience to customers, and defeating competition. Furthermore, presently, companies have several different recommendation types to provide a wide range of data to process to produce these recommendations, methods that can be used, and types of outputs to deliver, as demonstrated in Fig. 1.[2]

For example, LinkedIn started using collaborative filtering (CF) algorithms only for recommending relevant jobs to individual users [7] while it moved later to also recommend people to link with, people to get recruited to the companies, and segments of users to advertisers. Its recommendation engine processes different types of users' feedback and various types of preferences elicited by analyzing users' navigation patterns and personal profiles in addition to using ratings. The same complexity can be observed at Quora that runs a world-wide operating "question-and-answer" website [7]. The data that its RS handles are often textual and present many relationships among pieces of information. The system can recommend answers, feeds, topics, users, and related questions. Different kinds of users' feedback are combined in order to better represent the long-term goals that the RS has to work for. Several models are used to generate recommendations, including neural networks, logistic regressions, random forests, and matrix factorization, and they are systematically combined using ensemble methods. The performance metrics Quora used to assess the RS overall quality goes beyond accuracy-based metrics and includes truthfulness, reusability, explanations, format, and user retention. In addition, people in companies constantly interact with customers in order to understand whether recommendation methods and outputs work properly and can be improved. They rely heavily on controlled experiments with different groups of customers exposed to different treatment conditions (i.e., A/B test) and on measuring several user-centric performance metrics that are used to improve the design of these methods [5,7], as Fig. 2 describes.

In summary, the key assumption of this paper is that there are many different types of recommendations that an RS can generate, and each of these types is appropriate in some and less appropriate in other *conditions*. In this paper, we introduce and explain these conditions and focus on describing a method to assess which types of recommendations are the most appropriate for the specific conditions.

This paper also presents a framework that is based on the three concepts depicted in Fig. 3. A *"recommendation system selection strategy"* (RSST) is a complex managerial decision that includes the way a company organizes the data about users, how recommendations are generated, and how they are delivered to the users. The *"Nature of user–business interaction"* is the set of conditions describing the way a user and a business interact through an RS and includes the characteristics of the relationship, those of the information exchanged, and those of the device/interface. The *"Evolutionary factors"* constitute those characteristics of the business environment that change in time and can determine a change in the nature of interaction and, in turn, call for a change in the RSSTes.

This paper makes the following contributions. First, it introduces the concept of RSST and classifies these strategies. Second, it presents a framework to guide the decision of which strategies should be adopted based on the nature of interactions with its users. Third, the paper presents four case studies illustrating how the described framework works in practice.

2. Prior Work

In order to introduce an RSST and put it into the context of prior work, we need to identify the appropriate criteria of classification of various approaches to RSes. To do this, we revisit the classical mapping function

$$R: Data \rightarrow Outcomes \quad (1)$$

that generates *Outcomes*, i.e., items that can be recommended to the users, by processing *Data*, such as ratings, that reveal users' preferences of items [8]. For the classical rating-based CF approaches, data in (1) is a set of ratings provided by users to items that are typically organized in a two-dimensional (2D) matrix of "Users x Items." The matrix is very sparse. The goal of the function R is to predict the unknown ratings. Ratings can be explicitly provided by the user or implicitly derived from the user data [9]. Researchers have studied how to use implicit ratings [10] and how to model user-generated content [11]. Several nonrating-based approaches have been proposed, which generate recommendations not from the ratings but from other types of data such as users' reviews or users' purchases [12]. Semantic-based techniques have been applied in the applications with nonnumerical data [13]. The use of "tags" in RS has also been studied [14]. Another classification distinguished between preference-based and behavior-based recommenders [15]. The formers are those based on users' explicitly stated preferences and are further distinguished into rating-based, feature-based (including case-based, utility-based, knowledge-based, and critiquing-based), and personality-based systems. The latter are those based on users' navigation or purchase behaviors. The ratings are stored in a user-item matrix that can be either 2D or multidimensional (MD) depending on whether additional information, such as context [16,17], is collected or not [18]. Context can be important because it can make the behavior of a user change. In certain applications, considering this dimension can significantly affect the quality of recommendations [19].

The function *R* in (1) can be broadly assumed to be an algorithm used to generate the recommendations. Several classifications of these algorithms have been proposed [8,20], including that of content-based, CF, and hybrid trichotomy [20]. A number of recent studies have incorporated firm-specific measures in the design of recommendation algorithms, such as profit and inventory efficiency [21], surplus value [3], item popularity [2], prices, valuations, saturation effects and competition amongst products [4], profitability [22], cross-selling effects [23], and reduction of energy consumption [24]. Recommendations can be generated by querying and search or by preference

[2] Adapted from AMATRAIN, X. & AGARWAL, D. 2016. "Tutorial: Lessons Learned from Building Real-life Recommender Systems." 10th ACM Conference on Recommender Systems, Boston, Massachusetts, USA. ACM, 433-433.

elicitation and estimation [25]. Some authors [26] proposed a framework of recommendation goals and purpose based on a general distinction between consumer-centric and provider-centric viewpoint.

The *Outcomes* of an RS is the list of items which the user is predicted to like. The outcomes can be classified based on the goal an RS is designed for, such as recommending good items, optimizing utility, and predicting ratings [27]. The first category includes two tasks, recommending "all good items" items and recommending the "top-k items" [28]. In the first case, the list of recommended items contains all the items predicted as relevant to the user. In the second case, the recommendable items are ranked by the predicted score and the list includes only the top-k items. The other two categories are based on models that try to optimize a utility function or to predict a rating for all the items in the dataset. The output can also explain the reasons why those recommended products were suggested [29]. Recent research has proposed a way to rearrange the list of recommended items in order to make the less popular items appearing higher in the list, thus promoting items from the long tail [30].

The topic of adopting the right type of recommendation has been studied by researchers mainly from the perspective of choosing the right performance metric and an algorithm that maximizes it. All of these studies have faced the problem from a machine learning performance perspective and only looking at technology [31,32]. They have not faced it from a business perspective and, in turn, do not explain dramatic evolutions that these systems have undergone over time. For instance, both CF and content-based recommendation algorithms were reviewed and compared to each other in this light [31,32]. In general, the authors of these studies review the metrics to evaluate an RS, highlight weaknesses and strengths of some of the existing recommendation algorithms, and open the way for new approaches. Only few works have tried to identify specific recommendation strategies for specific conditions, but none of these works has proposed a general taxonomy and a framework to associate types of recommendations to specific setting [1,33]. As a result, this remains an open issue that researchers are challenged to tackle. Recently, Jugovac and Jannach [34] reviewed the existing literature on user interaction aspects in RSes. Some interesting insights about recommendation strategies were proposed but they were only focused on the human–computer interaction aspects, and the authors argued that "the intended utility of the recommender should be kept in mind when designing the user interaction mechanisms for the system … this might require the consideration of domain-specific and application-specific solutions." Our research is an attempt to fill this gap in the literature by proposing a framework to associate types of recommendations with the characteristics of the user–business interaction.

3. Recommendation strategies

Based on the definition of RS presented in the previous section, we identify the following three classification criteria differentiating various types of RSes: *Centricity*, *Dimensions,* and *Delivery*. Centricity refers to the design focus of the recommendation algorithm, Dimensions to the way rating data is structured, and Delivery to the way the recommendations are presented. The first criterion *Centricity* refers to the way function R is designed. It can be designed to match either users' utility ("user-centric") or the utility of different actors, including the business itself ("nonuser-centric"). Most of the RS research focused on maximizing user's utility and on making recommended items relevant for the user. Further, researchers combined relevance with other characteristics to increase users' interest [35], such as enhancing diversity of recommended items to make users discover new relevant items they did not know before [36]. We call this type of RSes "user-centric" because the design is centered on the user side. The alternative "nonuser-centric" approach includes both "business-centric" methods [4] and the designs that focus on the goals of various third parties [16,21,37,38]. As an example of such a nonuser-centric method, LinkedIn can recommend groups of users to advertisers where the RS design is focused not on the users or business but on advertisers.

The second criterion *Dimensions* refers to the way *ratings data* are structured, which can be done with a traditional 2D *User x Item* matrix where element *(i,j)* represents the rating that user *i* provided for item *j*. Estimation of unknown ratings can be done using numerous CF methods [39]. Alternatively, the ratings data can be structured using additional dimensions that can define other aspects of the recommendation process, including contextual, social, and other aspects. For example, Adomavicius and Tuzhilin [18] proposed the MD "context-aware" (CARS) approach that represents the ratings data as an MD cube (tensor) having dimensions $User \times Item \times Dimension(k) \times Dimension(h) \times \ldots$ [16]. The additional dimensions can represent situations that may affect users' ratings for an item, such as user's location, intent of purchase, mood, time, and other contextual factors. The MD approach can be generalized beyond context because any factor may be used as an additional dimension of the ratings cube if ratings vary with that factor, such as the factor specifying if the user is in an early or a late stage of a purchasing process [40], or who shares the experience with the user (e.g., social network or a household). Table 2 shows possible examples of 2D and MD RSes.

The third criterion *Delivery* refers to the way the recommendation output is delivered to the users either as a recommendation list or in the form of a complex interface that one can browse and navigate. We call the list-based method "fix/static" and the browsing-based method "navigational/dynamic." Once the data are processed by function R and the items to be recommended are ordered based on their estimated relevance, the most relevant items are presented to the user. The two most popular output methods include the "finding all the good items" approach that delivers those items with a score above a certain threshold level and the "top-k" method that outputs only the most relevant *k* items. An alternative approach may sometimes also deliver low-scored items in those cases when the risk of annoying users with "bad" recommendation is minimal and when the users want to see something different [41]. This happens when the overall number of items are small (e.g., recommending movies by a certain director) or when users' tastes change quickly. Another example is delivering a list of "items recommended to you by other people." In this case, users may want to know both the products their friends or a celebrity would recommend and those they would "not" recommend. The two main tasks in this case include delivering only those items which the user has never seen before and delivering all the items (both seen and unseen). The first task is useful for the applications where products are normally bought only once (e.g., books), while the second task may be used for repetitive purchases (e.g., grocery). As specified in Table 1, each of the aforementioned criteria assumes two values, and the Cartesian product of these values produces eight possible combinations. We associate a particular *RSST* with each of these eight possible combinations, resulting in the following recommendation strategies in our framework:

1.User-centric, 2D, Fix/static	5.User-centric, MD, Fix/static
2.User-centric, 2D, Navigational/Dynamic	6.User-centric, MD, Navigational/Dynamic
3.Nonuser-centric, 2D, Fix/static	7.Nonuser-centric, MD, Fix/static
4.Nonuser-centric, 2D, Navigational/Dynamic	8.Nonuser-centric, MD, Navigational/Dynamic

Table 2 shows examples of RSes classified by centricity, dimensions, and delivery. Once the types of RSes are classified, the second step is to discuss their appropriateness with respect to the business conditions that depend mainly on the nature of user–business interactions. In

Table 1
Taxonomy of recommendation strategies.

RS component	Criterion	Classification of the approach
R function	Centricity	User-centric vs. Nonuser-centric
Data	Dimensions	Two-dimensional vs. Multidimensional
Outcomes	Delivery	Fix/static vs. Navigational/Dynamic

Table 2
Examples of RSes classified by centricity, dimensions, and delivery.

User-centric RSes (goals)	Nonuser-centric RSes' goals
Increasing relevance and/or diversity Increasing trust Improving customer experience Improving customer satisfaction	Increasing profit Preventing churn, increasing retention Increasing word of mouth Increasing revenue from advertising
Two-dimensional RSes	**Multidimensional RSes**
Content-based Collaborative-filtering Hybrid	Context-awareness Type of activity (browsing, buying…) and feedback (textual, ratings…) Household and Social networks
Fixed/Static RSes	**Navigational/Dynamic RSes**
Top k recommendations All items with positive score	Deliver all recommended items never seen before User interface for browsing all items ranked by score

particular, each of the criteria described in this section is driven by one of the three characteristics of the nature of user–business interaction (see Section 4) following the framework we propose (see Section 5).

4. Nature of the interaction between user and Business

A large part of the research investigating factors affecting performance of an RS is based on the idea that the features related to users' experience are the most important ones [15]. For instance, the ability to help the users to decide, compare, discover, and explore [42] and the stages in an RS life cycle [43] were identified as particularly important. Our approach is based on the idea that an *RSST has to fit the characteristics of the interaction between a user and a business*. This interaction can change, even in the short run, and the challenge for companies is to promptly change their RS strategies. Based on the literature discussed in the rest of the section, we identified three main characteristics of a user–business interaction which make an RSST appropriate or not: (a) "maturity" of the relationship, (b) its "complexity," and (c) the "technological enablers" facilitating the interaction.

4.1. Maturity

Factors related to user experiences with an RS, such as familiarity and trust, evolve during the RS life cycle, and this evolution can be measured by the "maturity" of the relationship between the user and the business. Much research has demonstrated the influence of users' perceptions, such as trust, familiarity, and ease of use, on the user decision to follow or not RS recommendations [44,45]. Further, credibility, objectivity, and transparency affect the adoption of an RS, which influences shoppers only when it is perceived to be objective and credible [46]. Other users' perceptions affect the adoption of an RS, such as privacy [47], familiarity [48], risk, usefulness and ease of use [49], attitude [50], and personal characteristics [51,52]. These perceptions change over time and evolve according to the life cycle model [53,54]. Several studies showed that recommendations have to change as maturity and familiarity change along the RS life cycle [55–57]. Based on this research, we maintain that RSSTes should change when maturity of an RS changes. When an RS is initially introduced in a business application, maturity is minimal. Users may be unfamiliar with the application, and they may have doubts about usefulness, credibility, and the ease of use of the system. Further, the underlying RS technologies may exhibit flaws, and users' trust may be low. At this stage, improving trust may be the most important goal for a company, and the companies should design their algorithms to maximize relevance and accuracy of an RS. In the subsequent stages, flaws are fixed and users' familiarity and trust in an RS increase. When maturity is high enough, users are willing to accept new recommendation options [58]. As an example, users prefer to be recommended more novel and less popular items after gaining experience with an RS [59]. When maturity is high, new algorithms and interfaces can then be introduced, and the business can further differentiate its technology from that of competitors.

4.2. Complexity

A user–business interaction can occur in a variety of situations, independently of the system's maturity. For instance, the interaction can occur in different contexts (e.g., location, intent, and mood) and the users' activity can vary (e.g., browsing, buying items, providing ratings and feedback, etc.). The variety and diversity of these situations and activities can be measured by their *complexity*. The influence of context, such as users' locations, their mood and intent of a purchase, on an RS performance has been widely demonstrated [60]. Complexity accounts for other factors beyond context, such as user's experience [55], stage in a customer decision-making process [61], purchase type [62] and frequency [56], and social connectivity. These factors have a significant impact on acceptance of RSes [63].

Recommendations should change depending on complexity. In a complex interaction, users have many touch points with the application and spend much time, effort, and money. They buy often and purchase different product categories, evaluate several product attributes [40], perform different tasks (e.g., browsing, evaluating, and comparing). When complexity is high, users have frequent and active interactions with the business, feedback are rich and diverse, and the MD methods can produce better recommendations in this case. When an interaction is simple, users have infrequent touch points with the company and they spend very few resources. Their activities are simple and feedback are relatively poor. In these cases, adding new dimensions to the *User x Item* matrix is very hard, if not impossible, because the sparsity of an MD matrix would make recommendations inaccurate.

4.3. Technological enablers

Research has largely demonstrated the effect of the interface-related factors on the acceptance of an RS, such as the way items are presented and organized [64], the way they are ordered [65], and their number [66]. Explanation of recommendations has a significant effect [67], as well as the possibility of providing feedback [15]. Further, the effectiveness of recommendations depends on the device and its interface properties [68]. Recommendations may be delivered through different interfaces by which users can view the items, organize them, and provide various kinds of feedback. The interfaces' properties vary depending on the technological enabler [54]. For instance, if recommendations are delivered to a desktop, many items can be displayed, and the user may browse them or search by keywords and provide feedback. If recommendations are delivered through an emailed newsletter, only few items can be displayed with the very basic information, and users may provide little feedback. On the other end, a message texted to a user's smartphone may just alert him/her on the possibility of getting recommendations.

In summary, the nature of interactions between users and businesses can be characterized by a certain degree of maturity, complexity, and technological enablers. The combination of these factors determines the most appropriate RSST, and Section 5 describes how this association is done in our framework.

5. Recommendation strategies: a framework

We propose the following framework capturing various recommendation strategies for particular types of user–business interactions that is characterized by a 3-dimensional space defined by "maturity," "complexity," and "technological enablers" dimensions introduced in Section 4. For the sake of clarity, each dimension is represented by only two values, i.e., immature vs. mature user-business relationships, simple vs. complex

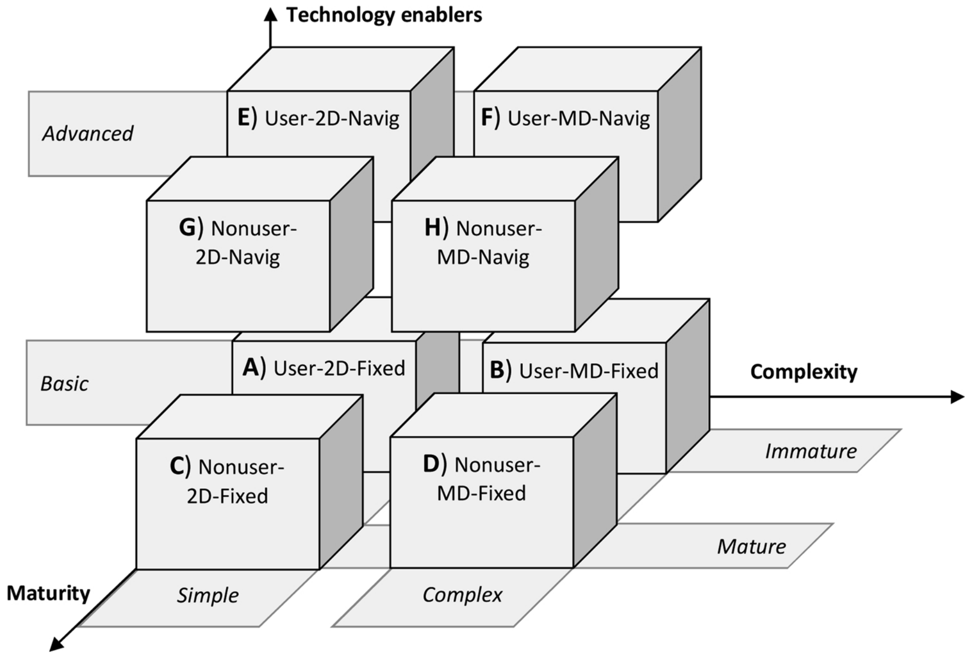

Fig. 4. A framework of recommendation strategy.

user–business interactions, basic vs. advanced technological enablers. For instance, if the relationship is immature, the interaction is simple, and technological enablers are basic, then the nature of the interaction is classified at the origin of the plot, corresponding to the cube labeled “A” in Fig. 4 corresponding to the case when users are not familiar with the RS, interactions are infrequent, and users connect through a basic device. As another example, if maturity is high, the interaction is complex and the technological enablers are advanced, the user–business interaction is defined by the cube labeled “H” in Fig. 4. In this case, users' trust is high, touch points are frequent and devices are advanced. We next discuss which RSST should be adopted for each of the eight cubes (A) through (H) in Fig. 4.

The “maturity” of the relationship determines the RSST “Centricity.” As we argued above, when maturity is low, the technology may have flaws, users may be unfamiliar with the application, they may have doubts about the RS, and their trust in it may be low. In this case, companies should adopt a user-centric RSST and design their algorithms to maximize relevance and usefulness of recommendations. This typically occurs in the initial stage of a recommender life cycle, such as when an RS is recently introduced in the application, when new users start interacting with it, or when the company is targeting a new segment of users. Once the relationship becomes more mature, users are willing to accept nonuser-centric strategies, including exploring less popular items and companies differentiating their algorithms from those of the competitors, as discussed in Section 4. Therefore, nonuser-centric RSSTes can be adopted when the life cycle reaches the maturity stage. All this means that a business may use *both* user-centric *and* nonuser-centric RSSTes *at the same time* in certain cases, e.g., a nonuser-centric RSST for customers who have an old relationship with the company while a user-centric RSST for new users.

The “complexity” of the user–business interactions determines the RSST “Dimensions” criterion that can be 2D or MD depending on certain settings discussed in Section 4. We define an interaction to be *simple* when users interact infrequently, spend little time on the Website, browse few categories, and provide no feedback. In this case, using MD approach is hard, if not impossible, because sparsity of the MD ratings cube would make recommendations inaccurate. Therefore, the 2D RSSTes are more appropriate when user interactions are simple. Similarly, we define an interaction to be *complex* when users connect often, spend much time on the Website, buy different product categories, and provide rich feedback. For complex and frequent interactions, businesses can gather enough data to accurately learn users' behavior in various contextual and other settings. For instance, customers may spend little on weekdays and much on weekends, and they may prefer classic novels in the winter and detective stories in the summer. It has been demonstrated that sparsity is offset by data homogeneity in this case [17]. Based on these observations, we can conclude that MD RSSTes are appropriate when user–business interactions are complex. Moreover, companies may have simple interactions with some segments of the users and complex interactions with other segments at the same time. As an example, an Internet-connected TV may be tuned on animation movies in certain hours of the day when children watch TV, and on a variety of shows and different movie genres in other hours when parents watch. Therefore, the application can learn little about children preferences and only adopt a 2D content-based RS in certain hours and days of the week, while it can learn much about parents and use an MD RSST at these times when parents connect.

The “technological enablers” by which users interact with the business determines the RSST “Delivery” criterion (see Table 1). An interaction is *basic* if it occurs through the interface that bounds a user's activity to the simplest operations, such as reading or watching basic information with the limited possibility of the feedback. In such case, the user is unlikely to browse a long recommendation list, and may be attracted only to a small piece of information. This makes a *fixed/static* RSST (e.g., top-1) to be more appropriate. An interaction is *advanced* if it occurs through the interface that provides the user with ability of both choosing how to browse extensive amount(s) of information and providing rich feedback. In this case, the user can browse, evaluate, and compare more options, even considering the less relevant ones, ask for more information, look for other users' reviews, and provide suggestions to others. In this case, *navigational/dynamic* RSSTes are more appropriate. As in the other case, recommendations can be delivered differently to the same user depending on the technological enabler. A simple “top-1” recommendation may be delivered to a user as a reminder through a text message, and a navigable full list of recommended items can be delivered to the desktop screen.

In the next sections, we discuss four cases to demonstrate how RSSTes depend on the nature of the user–business interactions according to our framework presented in Fig. 4.

5.1. Evolutionary factors in the user–business interaction

The nature of the interaction between a user and a business can change because of factors that are external to them. Daft [69] defined the external environment as the set of factors that have no direct influence on organization daily activities but can indirectly affect them. These factors comprise evolution of technology, industry structure (e.g., competitors, suppliers, and financial system), and societal trends (e.g., culture, economic conditions, and governmental policies). This definition has been used widely across different areas, such as strategy [70], innovation [71], and e-business [72]. Therefore, we adopt it for our case. Evolution of technology can modify the nature of user–business interaction by making new technology enablers available with better interfaces. For instance, the evolution of mobile devices has suddenly increased the possibility of users to interact with TV broadcasters [73]. An industry structure can also modify the interaction. For instance, the entry of a new player delivering sophisticated recommendations can make users' trust in the established companies decrease [74]. Finally, societal trends can affect the interaction between users and businesses either positively or negatively [75]. For instance, interactions may be more complex during a positive economic cycle when consumers increase their transactions and are willing to be influenced by people's advice in their social network [4].

We want to point out that we do not investigate if and how these evolutionary factors affect the nature of user–business interaction for the sake of brevity and because it is not the main goal of this paper. Our goal is to demonstrate that the RSST depends on the current business condition, and we also propose a framework for this strategy selection. We just want to point out that some factors (i.e., evolutionary factors) may lead to changes in the business condition and, in turn, on the recommendation strategy, but we do not want to explore what these factors are and how do they affect the business condition. Therefore, in the rest of the paper and especially in the case studies discussion, we will not further investigate about the evolutionary factors.

6. Case study 1: the Netflix recommender system

As was pointed out in Section 1, Netflix recommendation engine is of strategic importance to the company because it influences users' choice in about 80% of streaming cases. Further, recommendations delivered by Netflix are not just the output of an algorithm but "rather a collection of different algorithms serving different use cases" [5] that are activated depending on various interaction settings between a user and the Netflix RS. Further, the Netflix RS has evolved over time. When Netflix was shipping DVDs, the company relied on a simple 2D algorithm designed to predict users' ratings because "a star rating was the main feedback" to make an RS work at that time [5]. In 2006, the Netflix Prize was launched in order to improve rating predictions. When the main business became the Internet TV, interactions between the company and its users became more complex, Netflix offers became richer, the customers became more familiar with the RS, and the company could rely on "vast amounts of data that describe what each Netflix member watches, how each member watches…, the place in the product in which each video was discovered" and so on [5]. It became clear that the RS had to be improved. As a consequence, Netflix introduced social recommendations in 2011[3] and a Context-Aware RS in 2012. A 2012 blog reported that Netflix "needs to take into account factors such as context, title popularity, interest, evidence, novelty, diversity, and freshness" and "supporting all the different contexts in which we want to make recommendations requires a range of algorithms that are tuned to the needs of those contexts."[4] In 2013, Netflix introduced the possibility to personalize recommendations for the household and not only for the individual users.[5] Evolution of the Netflix RS shows that the company has changed the way recommendations are generated and delivered in response to changes in the users' needs, the way they interact with the business, and the users' familiarity with the system. These changes do not depend just on the availability of more sophisticated algorithms. They are part of a deliberate strategy according to which recommendations are tuned to the specific ways users interact with the business. In the next paragraphs, we show how this strategy can be explained by our framework (see examples in Fig. 5).

6.1. Centricity and maturity

The Netflix case shows a variety of approaches to centricity, including those of user- and nonuser-centered recommendations. Referring to the first approach, Netflix 2D algorithms are mainly designed to increase relevance and discovery. The construction of every recommendation page takes "into account the relevance of each row as well as the diversity of the page" and in certain cases the design tries to optimize relevance and diversity [5]. Cube "A" in Fig. 5 provides an example of the situation in which Netflix uses a user-centered RSST. Relevance and diversity are not the only user-centered goals, customer experience is another. For instance, Gomez-Uribe and Hunt [5] state that the Netflix "algorithm does not use a template, thus is freer to optimize the experience." The Netflix RS also deploys several nonuser-centered approaches: Some algorithms are designed to increase retention rates, prevent customer churn, improve users' engagement, and increase revenues and viewing spread. Again Gomez-Uribe and Hunt [5] state that "the main measurement target of changes to our recommendation algorithms is to improve member retention," and "our recommender system helps us win moments of truth: When a member starts a session, and we help that member find something engaging within a few seconds, preventing abandonment of our service." Further, the authors state that "maximizing revenue through product changes is fairly equivalent to maximizing the value that members derive" from Netflix service, implying that personalization can help to increase revenues. Another important business metric for the Netflix RS is how spread video viewing is across the catalog [5].

Maturity of the relationship between Netflix and its users plays a crucial role in selecting the user-centric or nonuser-centric RSSTes. The algorithms designed for inexperienced users (e.g., new members accepting a one-month trial) are different from those designed for experienced users. This is not just a matter of using cold-start techniques. The goals that Netflix pursues with these two groups of users are different and, as a consequence, the RS centricity is different. When interacting with new members, Netflix adopts a user-centered RSST and its algorithms are designed to improve relevance (cube "A" in Fig. 5). When interacting with experienced members, the algorithms are designed to improve loyalty and word-of-mouth, according to a nonuser-centric approach: "all members with an improved experience … may be more enthusiastic when describing Netflix to their friends, strongly influencing new subscriber acquisition through word-of-mouth effects" Gomez-Uribe and Hunt [5] (cube "C" in Fig. 5).

6.2. Dimensions and complexity

Beyond the 2D content-based and CF RSes, Netflix incorporates several dimensions into design of its RS algorithms, including the *contextual* dimension. For example, adding such contextual information as the time of the day and location improves performance of Netflix recommendation algorithms up to 3% according to the CEO of Netflix.[6] The household is another important contextual variable because "most members have different moods from session to session, and many accounts are shared by more than one member of a household" [5]. When recommending movie to a child, a good algorithm suggests seeing

[3] http://blog.netflix.com/2011/09/watch-this-now-netflix-facebook.html.
[4] http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html.
[5] http://www.techhive.com/article/2024221/netflix-adds-personalized-profiles-with-recommendations-for-your-entire-household.html.

[6] See the interview with Reed Hastings of Netflix at www.youtube.com/watch?v=8FJ5DBLSFe4&feature=youtu.be

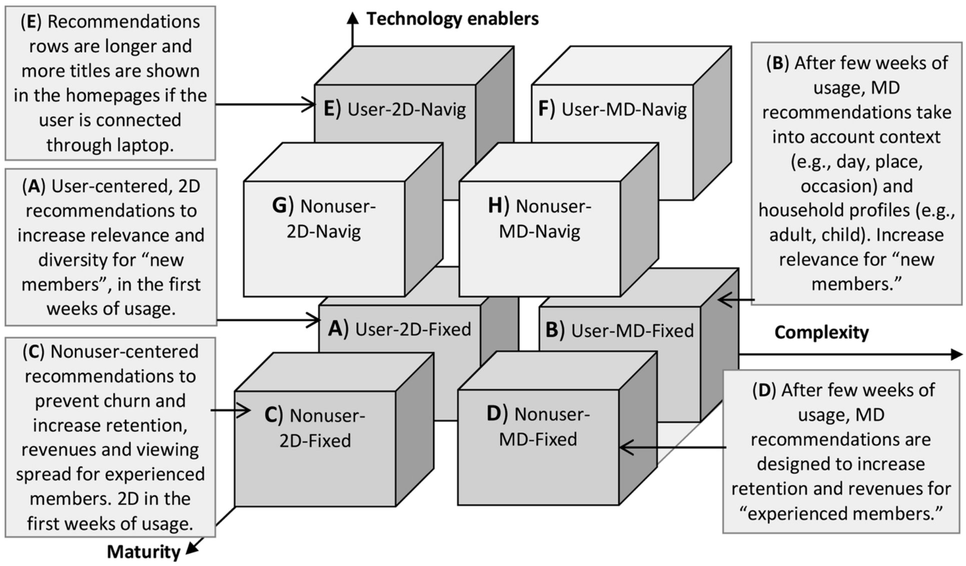

Fig. 5. How the recommendations strategies in Netflix change depending on the settings.

movies already seen in the past with high ratings because children love to see the same movie several times. On the contrary, when recommending movie to an adult, a different algorithm suggests seeing only the movies never seen before by the person.

The choice of a 2D vs. an MD RS depends on the complexity of the user–business interaction. For instance, with new users and with users that watch movies only occasionally, Netflix uses 2D RSSTes based only on the ratings provided by users. They may even bias recommendations toward popularity-based items. As the complexity of the relationship increases, Netflix starts using MD recommendations (cubes "B" and "D" in Fig. 5). Because all new members are offered a one-month-free trial, it is easy to recognize this kind of RSST by registering to the Netflix service: After few weeks of usage, the initial simple recommendations "fade away" once a member has enough data. Also, when the activity level of the member drops, recommendations become popularity-biased again.

6.3. Delivery and technological enablers

The way recommendations are personalized and organized for the users depends also on the device they are using to interact with Netflix. Recommendations are organized in homepages and rows. The number of rows on each page and their length (i.e., number of recommended items) as well as their position on the homepage "vary somewhat across devices because of hardware and user experience considerations" [5]. Recommendations are navigational in some cases (e.g., when using a laptop) whereas fixed in other cases (e.g., when using a smartphone). The most relevant items are put at the very first positions when users are delivered fixed recommendations while these are mixed with less relevant ones when receiving navigational recommendations (cube "E" in Fig. 5).

In summary, Netflix changes the way recommendations are generated and delivered depending on the nature of the interaction with the user in a way that can be explained by our framework. They are designed for different goals (e.g., relevance, diversity, and WOM) depending on maturity. Complexity (e.g., duration of the relationship) affects the amount of data Netflix can use to define multiple dimensions in its RS. The device a user is connected with makes the recommendation output changes in terms of its organization and layout. For the sake of brevity and clarity, we only described four fixed and one navigational strategy starting from case A and moving toward each dimension at a time in Fig. 5. However, Netflix also varies its recommendation output depending on simultaneous variations of more than one single dimension (namely cube G, H, and F). For example, as the complexity and maturity of the relationship increase and users decide to receive navigational recommendations (e.g., connecting through laptop or through smart TV), the recommendations are generated using a nonuser (e.g., aiming at preventing churn and increasing retention or viewing spread) MD (e.g., taking into account context and household profiles) navigational (e.g., longer rows and more titles shown) recommendation strategy (i.e., cube H). Similarly, if other combinations of changes in the three dimensions of the relationship happen, recommendations are generated using other strategies (i.e., cube F and G).

7. Case study 2: the Spotify recommender system

Spotify was launched on 2008 as a music streaming platform, and it then added the radio, podcast, and video streaming services. It is available in many countries and on all the devices (laptops, tablets, and smartphones). Songs can be searched by several different parameters, such as artist, album, genre, playlist, or record label. Users can create and share their own playlist by accessing to more than 30 million songs. Because of this huge collection of songs, Spotify integrated an RS into its platform since the beginning. The first RSST used when the system was at an early stage was user-based, 2D, and navigational,[7] represented by cube "E" in Fig. 6. The main aim at that time was to help users retrieving relevant songs by using only the few data available. After few years, the complexity of the interaction between Spotify and its customers increased. Users increased their usage of the service in terms of access over time and time spent on it and started providing richer feedback, such as ratings to listened songs (explicit in the form of "thumb up" or "thumb down", implicit in the form of listening time) or consumption information (such as mood or activity done while listening music through sensors installed on their mobile devices). The increase in complexity is witnessed by a session at Venture Beat's Mobile Summit in 2015, when Spotify vice president of North American advertising Brian Benedik said that "thanks to its requirement that every user — paying or nonpaying — sign in to use the service," Spotify collects an "enormous amount of data on what people are listening to, where, and in what context. It really gives us insight into what these people are doing." During the same year, during a press conference with Billboard, the CEO Daniel Ek stated that "People don't look at things

[7] http://benanne.github.io/2014/08/05/spotify-cnns.html.

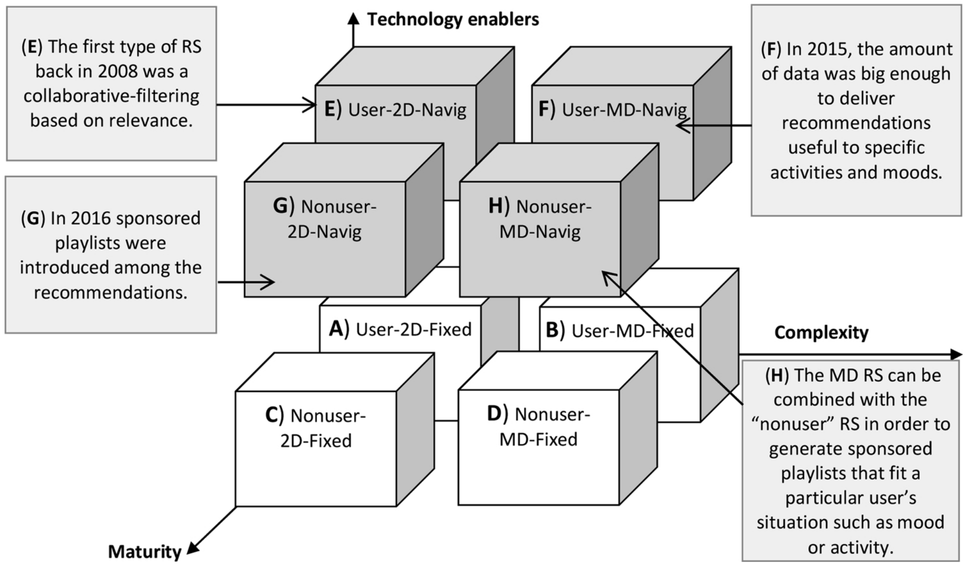

Fig. 6. How settings and RSSTes changed over time in Spotify.

like hip-hop or country anymore—they are looking at things based on events and activities…We need to be able to deliver the right music based on who we are, how we're feeling and what we're doing, day-by-day." He also stated that "if we truly want Spotify to be the soundtrack of our lives, we need it to be able to deliver the right music based on who we are, what we're doing, and how we're feeling." As a consequence, an MD RS was introduced in order to suggest playlists for different activities, such as workout at gym, sleeping, or specific moods.[8] This kind of MD-User-Navigational RSST is represented by cube "F" in Fig. 6. An additional and very recent evolution of the Spotify's RS happened in May 2016 when Spotify introduced the sponsored playlists.[9] An example of this strategy is the Branded Playlist which is sponsored by a particular brand. The RSST is "Non-user-2D-Navigational" (cube "G" in Fig. 6). Finally, the MD RS is combined with the sponsored playlist to generate personalized contextual advertising services based on recommendations of ads (such as Branded Moments, Sponsored Sessions, Video Takeovers, Audio, Display, Overlay, Home-page Takeovers, Branded Playlists, and Advertiser Pages).[10] Fig. 6 shows how Spotify started deploying MD and Nonuser RSSTes when the interaction with its users became complex and mature enough.

8. Case study 3: the amazon recommender system

Amazon.com started as one of the largest online bookstores, but soon diversified, selling DVDs, CDs, MP3 downloads/streaming, video games, electronics, apparel, furniture, food, toys, jewelry, and it is a major provider of cloud computing services. Amazon is now the largest Internet retailer in the world as measured by revenue and market capitalization. Amazon has deployed recommendations since the beginning. Currently, the company uses several types of RSSTes that we briefly review in the light of the framework in Fig. 4.

8.1. Delivery and technological enablers

Recommendations are delivered to customers by both sending newsletters through email and displaying a variety of recommended products on a personalized home page on the Web site. In addition, recommendations are also shown on the application available for smartphones. Various types of recommendation newsletters are sent to customers. In the simplest one, only one product is shown, the most relevant for the user, introduced by a sentence explaining that, based on the user's recent activity, Amazon believes the item could be interesting. The user can click on the item and be directed to the product page on Amazon's Web site. This kind of newsletter represents a User-2D-Fixed RSST corresponding to the cube "A" in Fig. 7. A slightly more advanced newsletter can show a list of several recommended items. The user can click on each item in the list and go to the item's page or he/she can click on a link at the bottom of the email and go to a Web page containing multiple recommendations. The smartphone app is more advanced with respect to the newsletters. The user can visualize several types of recommendations, such as "novelties," "suggestions based on your interests," and "similar to items you have already seen," by scrolling down the page. The most advanced kind of technological enabler is the home page on the Web site where recommendations can be navigated by scrolling up and down, right and left, and clicking on the items. This RSST is User-2D-Navigational, corresponding to the cube "E" in Fig. 7.

8.2. Dimension and complexity

Both the recommendations delivered by newsletters and those shown on the Web site can be organized by product categories. For instance, a newsletter can show several recommended items, all belonging to a product category (such as books, cables, teas…). The Amazon's home page organizes recommendations on multiple rows in order to let the user navigate them. Some of these rows show items belonging to a specific category (such as, "recommended for you in CD and vinyl"). These recommendations represent MD RSST and can be used only if the interaction between Amazon and a customer is complex enough. In fact, only the data pertaining one specific product category are used by the Amazon RS to generate relevant recommendations. These RSSTes are User-MD-fixed when sent through newsletter (cube "B" in Fig. 7), User-MD-navigational when displayed on the Web (cube "F" in Fig. 7).

8.3. Centricity and maturity

Amazon may include various kinds of "sponsored items" in a recommendation list. These products can be easily identified by the word "sponsored" next to the symbol "i," which explains that those items are

[8] https://spotifyforbrands.com/en-US/audiences/.
[9] http://adage.com/article/digital/spotify-expands-ad-formats-sponsored-playlists/304174/.
[10] https://spotifyforbrands.com/en-US/ad-experiences.

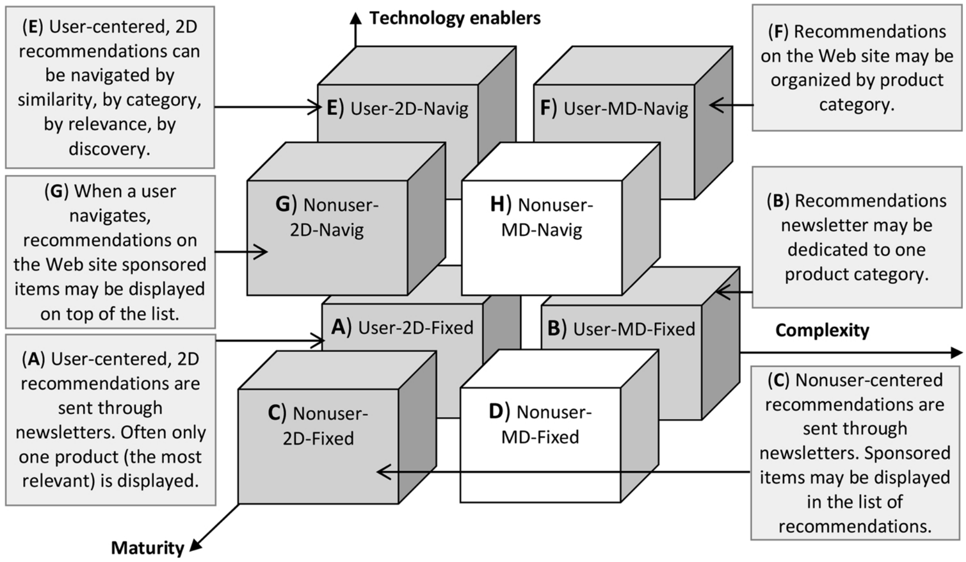

Fig. 7. The most typical recommendation strategies deployed by Amazon.

advertised by Amazon. These recommendations can be seen by both clicking on a newsletter and navigating on the Web site. In the first case, the RSST is Nonuser-2D-Fixed (cube "C" in Fig. 7), while in the second case the RSST is Nonusers-2D-Navigational (cube "G" in Fig. 7). An interesting change in the Amazon recommendation strategy goes back to 2009 when the "Amazon Basics" products were introduced in the catalogue. "Amazon Basics" mainly included electric commodities. From the beginning of 2009: When a user searched for products on the Amazon website, the "Amazon Basics" products were shown at the beginning of the list of relevant items. This change was possible because the relationship with customers was mature enough. The RS had been in use for long enough that most of its initial faults and inherent problems had been removed or reduced by further development, and users had become familiar with the system and found the system easy-to-use. As stated by David Martin, President of Interbrand New York, "Amazon has managed their brand into the role of a 'confidence' brand. Loyalists think of the brand first for sourcing a wide variety of products, especially electronics. It is logical to leverage this "confidence" to stand apart in the commodity segment of everyday electronics." This improvement in the maturity of the relationship with customers allowed Amazon to change its strategy, thus boosting its profits, strengthening its position with suppliers, enhancing the brand, and providing more value to the customer. In fact, as stated by Mark Ritson, visiting Associate Professor of Marketing at MIT Sloan School of Management and brand consultant to LVMH and Ericsson "low supply costs will ensure that Amazon will likely make more money per product on their line than from selling other suppliers' products." This shows how Amazon changed its RSST when the setting changed. Following our framework in Fig. 7, this change happened around 2009 when Amazon added "nonuser" RSST moving from A to C and from E to G. Currently, Amazon makes use of both user-centric and nonuser-centric RSSTes. According to Fig. 1, the company does not just employ one recommendation algorithm but rather a complex set of recommendations strategies, which have to deal with different problems and customers. The number of Amazon's customers is so high that the company has to deal with both customers who have a mature relationship with the provider and those who do not. Moreover, Amazon sells a huge number of different product categories characterized by remarkably different types of purchasing behavior. Therefore, even the same customer who has developed a mature relationship when buying certain product categories may show the behavior of an immature customer when buying for the first time a different category. This requires the use of both user-centric (A and E in Fig. 7) and nonuser-centric RSSTes (C and G in Fig. 7) at the same time.

9. Case study 4: from user-centric to nonuser-centric rsst

The last case study presents a live controlled experiment performed with a European company selling online comic books and related products, such as DVDs, stickers, and T-shirts. The company sent a weekly nonpersonalized newsletter to approximately 23,000 customers and agreed to send personalized recommendations of comic books through e-mail to a sample of this customer base as a part of our project. Upon the receipt of the personalized newsletter, the study participants were required to read it, click on a link, and rate each recommended item. More details about this study (i.e., number of customers involved into the experiment, how many recommendations were sent, etc.), where the authors followed the "user-centric, 2D, fixed" and the "nonuser-centric, 2D, fixed" RSSTes for 9 weeks, can be found in Panniello et al. [38] and Panniello et al. [19]. In this section, we will demonstrate that the appropriateness of these two strategies changes when maturity changes, i.e., a user-centric RSST is more appropriate when the user–-business interactions are immature and a nonuser-centric RSST is more appropriate when these interactions are mature.

9.1. Recommendation strategies

In Panniello et al. [38], the authors used a content-based RS to deliver user-centric recommendations and a profit-maximizing RS of the type described by Hosanagar et al. [21] to deliver nonuser-centric recommendations. In the former RS, recommendations focus exclusively on being relevant to the users, while in the latter the algorithm incorporates the item's unit profit (together with relevance) to generate and deliver recommendations. The user-centric content-based 2D algorithm described in Panniello et al. [38] recommends top-k items that are similar to the ones that users preferred in the past. It computes rating $u(i,s)$ of item s for user i based on the ratings $u(i,s_j)$ assigned by user i to items $s_j \in S$ that are similar to item s [8]. In summary, this method follows the "user-centric, 2D, fixed" RSST strategy.

For the nonuser-centric RSST, the authors chose a profit-maximizing RS described in Panniello et al. [38] and resembling the method presented in Hosanagar et al. [21], the key idea being that the firm should harvest for profitability when the system has high reputation and restore reputation (i.e., only maximize relevance) when the reputation is low. More specifically, the authors assume the following in their method:

- If the average rating provided by the user after reading the last newsletter is above a threshold (2.5 on a 0–5 scale in our case), or if the user purchased at least one recommended item during the last week, then they assume the system reputation is high and recommend the item with the highest unit profit.
- If the average rating provided by the user the last time is below the threshold and the user did not purchase the recommended item last time, then they assume that the reputation of the system is low and they recommend the item with the highest relevance.
- If a user has not received any recommendations previously, then they toss a fair coin to recommend either the most relevant item or the item with the highest unit profit.

The authors use a 2D RS and the "top-k" method and thus follow the "nonuser-centric, 2D, fixed" RSST.

9.2. Nature of user–business interaction

The company involved in the study had never sent personalized recommendations to its customers and, therefore, the author's endeavor constitutes the initial stage in an RS life cycle. In this early stage, the users are not familiar with the system, may have doubts about its value, and may expect flaws. This means that the user–business relationship is *immature* in week 1. In addition, the interaction is *simple* and the technological enablers *basic*. In fact, the users participating in the study where only requested to read the recommendations and rate them were not asked to perform any other kind of activity (e.g., browsing or buying), and the contextual information was not included. The interaction occurred through emailed newsletter, displaying only the top-9 items. The list was not navigable, and the users could only provide one type of feedback. Thus, according to Fig. 4, the user–business interaction at the beginning of the experiment (week 1) is immature, simple, and basic, corresponding to the cube labeled "A."

The nature of interaction changed in week 9 only with respect to "maturity" when users received recommendations for 9 weeks and became familiar with them. Note that the average rating users provided to the recommendations from week 1 to 9 increased as the RS learned users' preferences (from 3 to 3.5 on the [0,5] scale [38]). Furthermore, the recommendation accuracy, as measured by the precision measure $Precision = N_{RLV}/N_{REC}$, where N_{REC} is the number of the items recommended to the user and N_{RLV} is the number of those items that proved "relevant" for the user [27], where "relevant" means that users rated it as 3, 4, or 5 on the [0, 5] scale, has also increased from week 1 to 9, as shown in Fig. 8.[11] On the contrary, neither the technology enablers nor the complexity varied significantly. The users' activity did not increase, the type of activity did not change, the amount of information exchanged remained the same, and the interaction continued to occur through the same personalized newsletter.

In conclusion, following our framework, the settings moved from "A" to "C" (see Fig. 4) because of the increase in maturity during the study.

9.3. Discussion

In order to study which one of the two RSSTes is more appropriate in the two different settings A and C (in Fig. 4), the recommendation accuracy of the two RSSTes was calculated using Precision measure

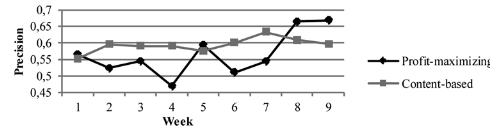

Fig. 8. Precision over time of profit-maximizing and content-based recommendation strategies.

over the period of 9 weeks, as explained above. Fig. 8 presents Precision of the content-based (user-centric) vs. profit-maximizing (nonuser-centric) strategies over 9 weeks and shows that the best performing RSST changes over time: the user-centric, 2D, fixed RSST outperforms the nonuser-centric, 2D, fixed RSST from week 1 to week 7 and vice versa for weeks 8 and 9, and these differences are significant [38]. This result is consistent with the framework discussed in Section 5: the user-centric RSST is appropriate when the maturity of user-business relationship is low, i.e., corresponding to the "A" cube in Fig. 4 and the nonuser-centric RSST is appropriate when maturity is higher. After 9 weeks of interactions with the RS, users' familiarity with the system is significantly higher, assuming that it delivers credible recommendations to them. Therefore, the user–business interaction is significantly more mature in week 9, and the setting moved from cube "A" to "C" (in Fig. 9). The best strategy for the company in this new setting is to use a nonuser-centric RSST, such as the profit-maximizing RS. We can assume that the customers' behavior changed because of the maturity of user–business interaction and, in particular, that the maturity increased over time since Panniello et al. [38] statistically demonstrated that the customers' trust was high at the end of the experiment (i.e., week 9). In addition, we can claim that the reason why users rated this nonuser-centric RSST higher than the user-centric one is the fact that it is the best one in balancing recommendations' accuracy and diversity with customers' trust which, once becoming familiar with the RS, becomes a relevant aspect to take into account. This point was also statistically demonstrated by Panniello et al. [38].

10. Conclusions

The main contribution of this paper lies in introducing the concept of RSST, presenting a framework to associate each RSST with the nature of the user–business interaction, and in supporting the proposed framework with empirical evidence in the form of four case studies. An RSST constitutes a complex managerial decision, including the way a company chooses to organize and process the data about users, the way these data are organized in order to generate recommendations, and the way these recommendations are delivered. By classifying the approaches to RS design based on three criteria, "centricity," "dimensions," and "delivery", we defined eight possible RSSTes. Then, we argued that a particular RSST is appropriate if it fits the characteristics of the interaction between a user and a business, such as relationship *maturity*, interaction *complexity,* and *technological enablers*. The framework we propose is the 3-dimensional space defined by maturity, complexity, and technological enablers. Each of the eight RSSTes fits a portion of this space. These results have also some relevant implications for practitioners. Companies deploying RSes in their business applications should carefully evaluate the characteristics of the interaction with the users and, as a consequence, choose the appropriate RSST to adopt. In addition, they should also continuously monitor their business conditions in order to timely react to any change by adopting an RSST, which is consistent with the new conditions. We also discussed four case studies to demonstrate how RSST depends on the nature of the user–business interaction according to our framework. The first three cases demonstrate how Netflix, Amazon, and Spotify modified their RSST over time according to the presented framework. The last case is a live controlled study that shows that the RSST centricity depends on the maturity of the relationship. The main limitation of this paper is that

[11] The authors measured only precision because of the missing ratings for the unseen items necessary to compute recall and F-measure.

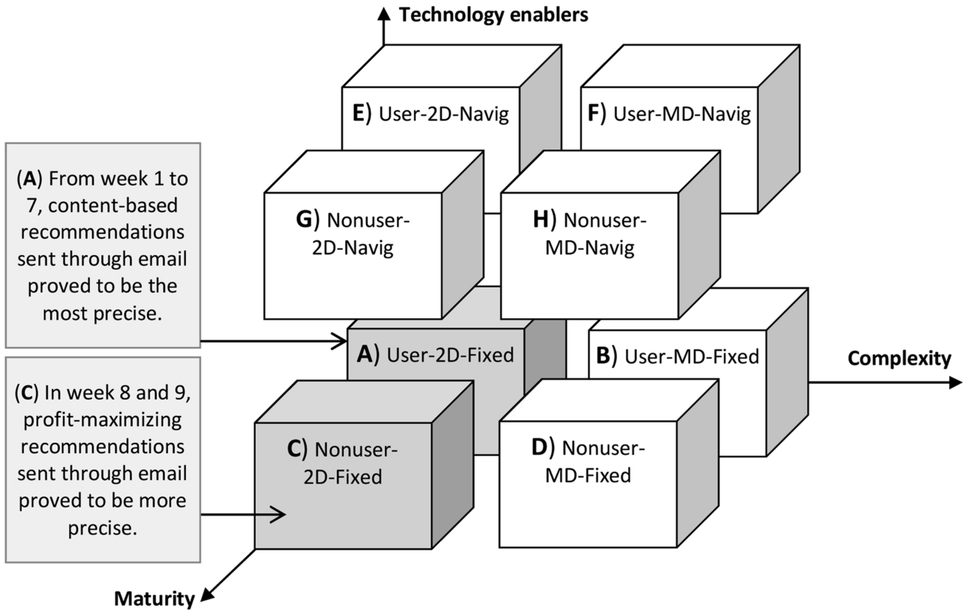

Fig. 9. How settings and RSSTes changed over time in case study 4.

the supporting evidence is restricted to only these four case studies, and therefore, we plan to enhance our study by providing additional empirical evidences in the future. In addition, another important limitation is pertaining to the binary values we set for each dimension in our framework. We plan to enhance our framework by considering also additional values for the three dimensions of the proposed framework in order to cover a broader set of business conditions. Finally, we plan to explore what are the main evolutionary factors and how do they affect the business conditions.

References

[1] A. Gunawardana, G. Shani, Evaluating recommender systems, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer, New York, 2015.
[2] D. Fleder, K. Hosanagar, Blockbuster culture's next rise or fall: the impact of recommender systems on sales diversity, Manage. Sci. 55 (5) (2009) 697–712.
[3] A. Ghose, P.G. Ipeirotis, B. Li, Designing ranking systems for hotels on travel search engines by mining user-generated and crowdsourced content, Mark. Sci. 31 (3) (2012) 493–520.
[4] W. Lu, S. Ioannidis, S. Bhagat, L.V.S. Lakshmanan, Optimal recommendations under attraction, aversion, and social influence, 20th ACM Sigkdd International Conference on Knowledge Discovery and Data Mining (2014) 811–820.
[5] C.A. Gomez-Uribe, N. Hunt, The netflix recommender system: algorithms, business value, and innovation, ACM Trans. Manage. Inf. Syst. 6 (4) (2015) 1–19.
[6] T. Kemp, Personalization isn't a product, Internet Week 864 (1) (2001) 1–2.
[7] X. Amatriain, D. Agarwal, Tutorial: lessons learned from building real-life recommender systems, 10th ACM Conference on Recommender Systems (2016) 433-433.
[8] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
[9] D. Oard, J. Kim, Implicit feedback for recommender systems, Year. Published, AAAI Workshop on Recommender Systems, (2019), pp. 81–83.
[10] Y. Fang, L. Si, A latent pairwise preference learning approach for recommendation from implicit feedback, 21st International Conference on Information and Knowledge Management (2012) 2567–2570.
[11] S. Garcia Esparza, M.P. O'mahony, B. Smyth, A multi-criteria evaluation of a user generated content based recommender system, 3rd Workshop on Recommender Systems and the Social Web (2011).
[12] L. Chen, G. Chen, F. Wang, Recommender systems based on user reviews: the state of the art, User Model. Useradapt. Interact. 25 (2) (2015) 99–154.
[13] I. Cantador, P. Castells, Semantic contextualisation in a news recommender system, Workshop on Context-Aware Recommender Systems (2009).
[14] I. Cantador, A. Bellogín, I. Fernández-Tobías, S. López-Hernández, Semantic contextualisation of social tag-based profiles and item recommendations, International Conference on Electronic Commerce and Web Technologies (2011) 101–113.
[15] P. Pu, L. Chen, R. Hu, Evaluating recommender systems from the user's perspective: survey of the state of the art, User Model. Useradapt. Interact. 22 (4-5) (2012) 317–355.
[16] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, Incorporating contextual information in recommender systems using a multidimensional approach, ACM Trans. Inf. Syst. 23 (1) (2005) 103–145.
[17] U. Panniello, A. Tuzhilin, M. Gorgoglione, Comparing context-aware recommender systems in terms of accuracy and diversity, User Model. Useradapt. Interact. 24 (1–2) (2014) 35–65.
[18] G. Adomavicius, A. Tuzhilin, Multidimensional recommender systems: a data warehousing approach, Electron. Comm. 2232 (1) (2001) 180–192.
[19] U. Panniello, M. Gorgoglione, A. Tuzhilin, Research note—in carss we trust: how context-aware recommendations affect customers' trust and other business performance measures of recommender systems, Inf. Syst. Res. 27 (1) (2016) 182–196.
[20] P. Resnick, H.R. Varian, Recommender systems, Commun. ACM 40 (3) (1997) 56–58.
[21] K. Hosanagar, K. Ramayya, L. Ma, Recommended for you: the impact of profit incentives on the relevance of online recommendations, International Conference on Information Systems (2008).
[22] T. Iwata, K. Saito, T. Yamada, Recommendation method for improving customer lifetime value, IEEE Trans. Knowl. Data Eng. 20 (9) (2008) 1254–1263.
[23] L.S. Chen, F.H. Hsu, M.C. Chen, Y.C. Hsu, Developing recommender systems with the consideration of product profitability for sellers, Inf. Sci. 178 (4) (2008) 1032–1048.
[24] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, M. Pazzani, An energy-efficient Mobile recommender system, 16th International Conference on Knowledge Discovery and Data Mining (2010).
[25] D. Dzyabura, A. Tuzhilin, Not by search alone: how recommendations complement search results, 7th ACM Conference On Recommender Systems (2013) 371–374.
[26] D. Jannach, G. Adomavicius, Boston, Massachusetts, USA. ACM"Recommendations With A Purpose." 10th ACM Conference on Recommender Systems2016, "Recommendations With A Purpose." 10th ACM Conference on Recommender Systems (2016) 7–10.
[27] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (1) (2004) 5–53.
[28] A. Gunawardana, G. Shani, A survey of accuracy evaluation metrics of recommendation tasks, J. Mach. Learn. Res. 10 (January) (2009) 2935–2962.
[29] N. Tintarev, J. Masthoff, Explaining recommendations: design and evaluation, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer, New York, 2015.
[30] Y.-J. Park, A. Tuzhilin, The long tail of recommender systems and how to leverage it, ACM Conference on Recommender Systems (2008) 11–18.
[31] F. Cacheda, C. Carneiro, D. Fern, V. Formoso, Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems, ACM Trans. Web 5 (1) (2011) 1–33 Ndez.
[32] P. Lops, M. Gemmis, G. Semeraro, Content-Based Recommender Systems: State Of The Art and Trends Recommender Systems Handbook, Springer Us, 2011.
[33] F. Hernández Del Olmo, E. Gaudioso, Evaluation of recommender systems: a new approach, Expert Syst. Appl. 35 (3) (2008) 790–804.
[34] M. Jugovac, D. Jannach, Interacting with recommenders— overview and research directions, ACM Trans. Interact. Intell. Syst. 7 (3) (2017) 1–46.
[35] C. Than, S. Ha, Improving recommender systems by incorporating similarity, trust and reputation, J. Internet Serv. Inf. Secur. 4 (1) (2014) 64–76.
[36] Y.C. Zhang, D.O. Seaghdha, D. Quercia, T. Jambor, Auralist: introducing serendipity into music recommendation, 5th International Conference on Web Search and Data Mining (2012) 13–22.
[37] H. Abdollahpouri, R. Burke, B. Mobasher, Recommender systems as

multistakeholder environments, Proceedings of The 25th Conference On User Modeling, Adaptation And Personalization (2017) 347–348 3079657: ACM.
[38] U. Panniello, S. Hill, M. Gorgoglione, The impact of profit incentives on the relevance of online recommendations, Electron. Commer. Res. Appl. 20 (C) (2016) 87–104.
[39] F. Ricci, L. Rokach, B. Shapira, Recommender Systems Handbook, Springer-Verlag New York, Inc., 2015.
[40] J.D. Xu, I. Benbasat, R.T. Cenfetelli, The influences of online service technologies and task complexity on efficiency and personalization, Inf. Syst. Res. 25 (2) (2014) 420–436.
[41] S.M. Mcnee, J. Riedl, J.A. Konstan, Being accurate is not enough: how accuracy metrics have hurt recommender systems, Chi '06 Extended Abstracts on Human Factors In Computing Systems, ACM, Montréal, Québec, Canada, 2006, pp. 1097–1101 1125659.
[42] F. Meyer, F. Fessant, F. Clérot, E. Gaussier, Toward a new protocol to evaluate recommender systems, Workshop on Recommendation Utility Evaluation (2012).
[43] J.A. Konstan, J. Riedl, Recommender systems: from algorithms to user experience, User Model. Useradapt. Interact. 22 (1–2) (2012) 101–123.
[44] Y.C. Cho, E. Sagynov, "Exploring factors that affect usefulness, ease of use, trust, and purchase intention in the online environment, Int. J. Manag. Inf. Syst. 19 (1) (2015) 21–37.
[45] P. Pu, L. Chen, R. Hu, A user-centric evaluation framework for recommender systems, Proceedings Of The Fifth ACM Conference On Recommender Systems (2011) 157–164.
[46] B. Pathak, R. Garfinkel, R. Gopal, R. Venkatesan, F. Yin, Empirical analysis of the impact of recommender systems on sales, J. Manage. Inf. Syst. 27 (2) (2010) 159–188.
[47] D.-J. Lee, J.-H. Ahn, Y. Bang, Managing consumer privacy concerns in personalization: a strategic analysis of privacy protection, MIS Q. 35 (2) (2011) 423–444.
[48] A. Cooke, H. Sujan, M. Sujan, B. Weitz, Marketing the unfamiliar: the role of context and item-specific information in electronic agent recommendations, J. Mark. Res. 39 (4) (2002) 488–497.
[49] D. Gefen, E. Karahanna, D.W. Straub, Trust and tam in online shopping: an integrated model, Mis Q. 27 (1) (2003) 51–90.
[50] W. Wang, I. Benbasat, Recommendation agents for electronic commerce: effects of explanation facilities on trusting beliefs, J. Manag. Inf. Syst. 23 (4) (2007) 217–246.
[51] S.Y.X. Komiak, I. Benbasat, The effects of personalization and familiarity on trust and adoption of recommendation agents, Mis Q. 30 (4) (2006) 941–960.
[52] B. Xiao, I. Benbasat, E-Commerce product recommendation agents: use, characteristics, and impact, Mis Q. 31 (1) (2007) 137–209.
[53] P. Jiang, Y. Zhu, Y. Zhang, Q. Yuan, Life-stage prediction for product recommendation in e-commerce, 21th International Conference On Knowledge Discovery And Data Mining (2015) 1879–1888.
[54] J.A. Konstan, Tutorial/HCI for recommender systems: a tutorial, 16th International Conference On Intelligent User Interfaces (2011) 463.
[55] S. Mukherjee, H. Lamba, G. Weikum, Experience-aware item recommendation in evolving review communities, IEEE International Conference On Data Mining, (2015), pp. 925–930.
[56] W. Wang, Recommendation strategies analysis in e-commerce, The 2nd International Conference On Information Science And Engineering, 4-6 December, (2010), pp. 1394–1397.
[57] C.L. Zheng, K.R. Hao, Y.S. Ding, A collaborative filtering recommendation algorithm incorporated with life cycle, Adv. Mat. Res. 765-767 (1) (2013) 630–633.
[58] K. Hu, W. Hsu, M.L. Lee, Utilizing users' tipping points in e-commerce recommender systems, 29th International Conference On Data Engineering (2013) 494–504.
[59] M.D. Ekstrand, D. Kluver, F.M. Harper, J.A. Konstan, Letting users choose recommender algorithms: an experimental study, 9th ACM Conference On Recommender Systems (2015) 11–18.
[60] C. Palmisano, A. Tuzhilin, M. Gorgoglione, Using context to improve predictive modeling of customers in personalization applications, IEEE Trans. Knowl. Data Eng. 20 (11) (2008) 1535–1549.
[61] S.Y. Ho, K.Y. Tam, An empirical examination of the effects of web personalization at different stages of decision making, Int. J. Hum. Interact. 19 (1) (2005) 95–112.
[62] I. Simonson, Determinants of customers' responses to customized offers: conceptual framework and research propositions, J. Mark. 69 (1) (2005) 32–45.
[63] T. Mahmood, F. Ricci, A. Venturini, Improving recommendation effectiveness: adapting a dialogue strategy in online travel planning, Inf. Technol. Tour. 11 (4) (2009) 285–302.
[64] B. Knijnenburg, M. Willemsen, Z. Gantner, H. Soncu, C. Newell, Explaining the user experience of recommender systems, User Model. Useradapt. Interact. 22 (4-5) (2012) 441–504.
[65] K. Diehl, L.J. Kornish, J.G. Lynch, Smart agents: when lower search costs for quality information increase price sensitivity, J. Consum. Res. 30 (1) (2003) 56–71.
[66] T.-P. Liang, H.-J. Lai, Y.-C. Ku, Personalized content recommendation and user satisfaction: theoretical synthesis and empirical findings, J. Manage. Inf. Syst. 23 (3) (2007) 45–70.
[67] H. Cramer, V. Evers, S. Ramlal, M. Van Someren, L. Rutledge, N. Stash, L. Aroyo, B. Wielinga, The effects of transparency on trust in and acceptance of a content-based art recommender, User Model. Useradapt. Interact. 18 (5) (2008) 455–496.
[68] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, Mobile recommender systems in tourism, J. Netw. Comput. Appl. 39 (March) (2014) 319–333.
[69] R.L. Daft, Organization Theory and Design, West Publishing Company, 1992.
[70] M. Porter, How competitive forces shape strategy, Harv. Bus. Rev. 57 (2) (1979) 137–145.
[71] L.G. Tornatzky, M. Fleischer, A.K. Chakrabarti, The Processes of Technological Innovation, Lexington Books, Lexington, Mass, 1990.
[72] K. Zhu, K.L. Kraemer, E-commerce metrics for net-enhanced organizations: assessing the value of E-commerce to firm performance in the manufacturing sector, Inf. Syst. Res. 13 (3) (2002) 275–295.
[73] Google, The New Multi-Screen World: Understanding Cross-Platform Consumer Behavior, Available: (2012) Http://Www.Google.Com/Think/Research-Studies/The-New-Multi-Screen-World-Study.Html..
[74] B. Schafer, J. Konstan, J. Riedl, E-commerce recommendation applications, Data Min. Knowl. Discov. 5 (1) (2001) 115–153.
[75] E. Ioanăs, I. Stoica, Social media and its impact on consumers behavior, Int. J. Econ. Pract. Theor. 4 (2) (2014) 295–303.

Michele Gorgoglione is professor at the Politecnico di Bari, where he teaches marketing and e-business models. He earned the Ph.D. degree in management from the University of "Roma-Tor Vergata." He currently does research on personalization technologies and customer relationship management. Research projects include context-aware and business-centric recommender systems, customer experience, predictive models of customer churn. He is member of the program committees of ACM RecSys and EC-Web scientific international conferences.

Umberto Panniello received Ph.D. degree in business engineering from the Politecnico di Bari. He has been a visiting scholar at the Wharton Business School of University of Pennsylvania. He is assistant professor in management at the Politecnico di Bari, where he teaches e-business models and business intelligence. His current main research interest is on customer modeling, consumer behavior, social TV, context-aware, and profit-based recommender systems.

Alexander Tuzhilin is professor of Information Systems and the NEC faculty fellow at the Stern School of Business. He received Ph.D. degree in computer science from the Courant Institute of Mathematical Sciences. His current research interests include data mining, recommender systems, and personalization. Dr. Tuzhilin has produced more than 100 research publications on these and other topics. He has served on the organizing and program committees of numerous CS and IS conferences and on the Editorial Boards of numerous international journals.