Basic Approaches in Recommendation Systems

Konstantin Kundegraber and Sabrina Pletzl

December 4, 2022

Basic Approaches in Recommendation Systems

Konstantin Kundegraber and Sabrina Pletzl

Graz University of Technology, 8010 Graz, Austria

Abstract. This paper aims to identify, collect and combine relevant literature about basic recommendation approaches. It covers four main types of recommenders: Collaborative filtering, content-based filtering, knowledge-based recommendation and hybrid recommendation. In addition, further approaches commonly described in scientific studies are briefly introduced. Furthermore, challenges of recommendation systems are discussed. The paper finally covers cognitive recommendation being a newly developed approach potentially counteracting various of the discussed challenges. The paper concludes with corresponding topics for further research.

Keywords. Collaborative filtering, Item-based, User-based, Content-based filtering, Knowledge-based recommendation, Case-based recommendation, Constraint-based recommendation, Hybrid recommendation, Demographic-based recommendation, Utility-based recommendation, Critique-based recommendation, Group-based recommendation, Cognitive recommendation

Paper type. Literature summary

1 Introduction

Since the advent of modern communication technologies, consumers are constantly provided with an excessive quantity of information (Fouladi and Navimipour 2017). Thus, consumers are having difficulties in grasping the issue and, consequently, in finding satisfactory solutions. Recommendation systems are information filtering systems that aim to support users in various decision-making processes with personalized item suggestions. In general, the systems predict a user's rating for a particular item based on individual preferences and constraints and output lists of items ranked by their relevance to the user (Ricci, Rokach, and Shapira 2011.

The first recommendation systems have emerged in the early 1990 and, since then, have proven to be effective particularly in the field of e-commerce (Jannach et al. 2010). Recommenders appear to be very useful in situations where enormous amounts of data need to be processed to acquire relevant information (Alyari and Navimipour 2018). There are different approaches a recommendation engine might take for information filtering. Collaborative filtering bases its suggestions on user-item interactions while content-based filtering utilizes characteristic item information to predict ratings of a user (N. Singh et al. 2019). In contrast, knowledge-based recommenders rely on domain knowledge encoded in the system and applied to users' preferences and constraints to generate suggestions (Bouraga et al. 2014). To limit any existing drawbacks of individual systems different hybridization methods can be applied (Jannach et al. 2010). Due to ongoing technological advances, various novel strategies emerged based on the basic approaches presented, such as social network-based or IoT-based recommendation systems (N. Singh et al. 2019).

The main goal of this literature summary is to outline the prevailing findings on basic recommendation algorithms. Additionally, the authors aimed to find potentials for further advancements in the field of recommendation systems. The following paragraph outlines the topics which are covered within this paper.

In chapter 2 collaborative filtering and both its approaches user-based and item-based recommendation is outlined. The following chapter 3 covers content-based filtering. Chapter 4 about knowledge-based recommendation is further split into the two subsections chapter 4.1 case-based recommendation and chapter 4.2 constraint-based recommendation. Different methods of hybridization are discussed in chapter 5. Chapter 6 covers further recommendation algorithms, namely demographic-based recommendation in chapter 6.1, utility-based recommendation in chapter 6.2, critique-based recommendation in chapter 6.3 and group-based recommendation in chapter 6.4. Based on fundamental challenges of recommendation systems outlined in chapter 7, chapter 8 covers topics of further research. The paper summarizes all findings in chapter 9.

2 Collaborative Filtering

With the development of collaborative filtering (CF) in 1990 an important milestone for further research of recommender systems has been set (Ko et al. 2022). The basic functionality of the recommendation method originates from the principle of word of mouth, which usually comes from the immediate environment such as family or friends. In the field of CF, these people can be replaced by so-called *k-nearest neighbors (KNN)*, which are users who behave similarly to the target user in the relevant field (Felfernig, Jeran, Ninaus, Reinfrank, Reiterer, and Stettinger 2014). KNN is a categorization technique that attempts to make different entities comparable. Accordingly, a new case is assigned to the category with which it most closely matches (Kalkar and Chawan 2022). Using these techniques, CF was developed to help users make decisions based on the opinions of similar users (Lu et al. 2015) and assumes that user preferences will remain static (Burke 2002). This popular technique is frequently used in the e-commerce sector (e.g., Amazon.com), but also in the entertainment sector (e.g., Netflix) (N. Singh et al. 2019). The reasons for its use are mainly due to the fact that CF is a relatively simple method with low cost of knowledge acquisition and maintenance (S. Sharma, K. Gupta, and D. Gupta 2021; Shah et al. 2017). Furthermore, it requires no domain knowledge and improves over time with high-quality data (Burke 2002).

The model can be divided into memory-based CF and model-based CF. Further subdivisions of memory-based CF are user-based CF and item-based CF (Ko et al. 2022). In user-based CF, the user's KNN are identified and subsequently used to predict the user's rating (Felfernig, Jeran, Ninaus, Reinfrank, Reiterer, and Stettinger 2014). Item-based CF utilizes similarity between items to predict a user's rating (Ko et al. 2022). Both approaches rely on two different types of data: Users and items. The relationship between user and item is determined by the ratings of different users and subsequently used to predict the ratings of other users (Felfernig, Jeran, Ninaus, Reinfrank, Reiterer, and Stettinger 2014). The data is represented as a two-dimensional matrix R_{ij} (Table 1) in which all ratings of users i for items j are stored. Items without ratings are represented by "?". This matrix forms the basis for personalized recommendations (Amin, Philips, and Tabrizi 2019).

	Item1	Item2	Item3	Item4	Item5
Sabrina	1	4	5	2	?
User1	3	3	1	4	2
User2	4	2	4	4	5
User3	4	1	2	5	4
User4	5	3	4	3	2

Table 1. User-item ratings matrix

Similarity between users or items can be measured by the respective relationships. The range of similarity is defined from -1 (opposite) to 1 (equal). This step represents an essential part of CF (S. Sharma, K. Gupta, and D. Gupta 2021). There are several variants for calculating the similarity, such as Pearson correlation-based similarity, constrained Pearson correlation-based similarity, cosine-based similarity, or cosine-based adjusted measurement (Lu et al. 2015).

The calculation of the Pearson correlation similarity $sim(m,n)$ for user-based CF is shown in equation 1 (Su and Khoshgoftaar 2009).

1. a and b; users
2. $p \in P_{(a,b)}$; sum of items that users a and b have rated
3. $R_{(a,p)}$ and $R_{(b,p)}$; ratings of users a and b for item p
4. $\overline{R_{(a)}}$ and $\overline{R_{(b)}}$; mean ratings of users a and b for all items

$$sim(a,b) = \frac{\sum p \in P_{(a,b)} (R_{(a,p)} - \overline{R_{(a)}})(R_{(b,p)} - \overline{R_{(b)}})}{\sqrt{\sum p \in P_{(a,b)} (R_{(a,p)} - \overline{R_{(a)}})^2} \sqrt{\sum p \in P_{(a,b)} (R_{(b,p)} - \overline{R_{(b)}})^2}} \tag{1}$$

Based on the similarity, equation 2 predicts the rating of user a for item p (S. Sharma, K. Gupta, and D. Gupta 2021).

1. NN; nearest neighbors of user a
2. $R_{(b,p)}$; rating of user b for item p
3. $\overline{R_{(a)}}$ and $\overline{R_{(b)}}$; mean ratings of users a and b for all items

$$prediction(a,p) = \overline{R_{(a)}} + \frac{\sum b \in NN sim(a,b) * (R_{(b,p)} - \overline{R_{(b)}})}{\sum b \in NN sim(a,b)} \tag{2}$$

The calculation of the Pearson correlation similarity sim(m,n) for item-based CF is shown in equation 3 (Su and Khoshgoftaar 2009).

1. m and n; items
2. $u \in U$; sum of users who rated items m and n
3. $R_{(u,m)}$ and $R_{(u,n)}$; ratings of user u for items m and n
4. $\overline{R_{(m)}}$ and $\overline{R_{(n)}}$; mean ratings of items m and n of all users

$$sim(m,n) = \frac{\sum u \in U(R_{(u,m)} - \overline{R_{(m)}})(R_{(u,n)} - \overline{R_{(n)}})}{\sqrt{\sum u \in U(R_{(u,m)} - \overline{R_{(m)}})^2}\sqrt{\sum u \in U(R_{(u,n)} - \overline{R_{(n)}})^2}} \tag{3}$$

Equation 4 subsequently predicts the rating of user a for item p based on the calculated similarity (S. Sharma, K. Gupta, and D. Gupta 2021).

1. NN; nearest neighbors of user a
2. $R_{(b,p)}$; rating of user b for item p
3. $\overline{R_{(a)}}$ and $\overline{R_{(b)}}$; mean ratings of users a and b for all items

$$prediction(a,p) = \overline{R_{(a)}} + \frac{\sum b \in NN sim(a,b) * (R_{(b,p)} - \overline{R_{(b)}})}{\sum b \in NN sim(a,b)} \tag{4}$$

In cosine similarity, items m and n are represented as vectors in an x-dimensional user space. In equation 5, the similarity is calculated based on the angles between the two vectors (Ricci, Rokach, and Shapira 2015).

1. m and n; items
2. $\cdot$; to calculate the dot product of vectors m and n

$$sim_{(m,n)} = cos(\overrightarrow{m}, \overrightarrow{n}) = \frac{\overrightarrow{m} \cdot \overrightarrow{n}}{||\overrightarrow{m}|| * ||\overrightarrow{n}||} \tag{5}$$

The cosine similarity between users is calculated as shown in equation 6 (Ricci, Rokach, and Shapira 2015).

1. $\cdot$; to calculate the dot product
2. $R_{(a,p)}$ and $R_{(b,p)}$; ratings of users a and b for item p

$$sim(a,b)^{cos} = \frac{\sum p \in P_{(a,b)} (R_{(a,p)}) \cdot (R_{(b,p)})}{\sqrt{\sum p \in P_{(a,b)} (R_{(a,p)})^2} \cdot \sqrt{\sum p \in P_{(a,b)} (R_{(b,p)})^2}} \tag{6}$$

To further deal with problems such as different rating scales and patterns in the cosine similarity of users, the adjusted cosine similarity can be used (Su and Khoshgoftaar 2009). An alternative to the Pearson correlation similarity is the constrained Pearson correlation similarity which uses the median instead of the mean for normalization (Bag, Ghadge, and Tiwari 2019).

3 Content-Based Filtering

Content-based filtering (CBF) recommends items that have already been rated well by the user (Lu et al. 2015). CBF assumes that users' personal interests do not change but rather remain constant (Felfernig, Jeran, Ninaus, Reinfrank, Reiterer, and Stettinger 2014). An example could be a bookstore where a book is characterized by its genre, subject and author. Here, other fantasy novels could be suggested to a user who likes to read fantasy novels (Jannach et al. 2010). This also leads to recommendations being comprehensible for users (Thorat, Goudar, and Barve 2015), but can also result in overspecialization (Burke 2002). Furthermore, no domain knowledge is required (Burke 2002), yet, defining attributes for items is necessary (Shah et al. 2017). Like CF, CBF improves over time with high-quality data (Burke 2002) and has relatively low costs for knowledge acquisition and maintenance (Jannach et al. 2010). Of all the models, CBF is one of the simplest and was, thus, widely used in early recommender systems (Ko et al. 2022).

The similarity between already consumed and potentially consumed items is captured from the description of the items or directly via a category indicator in the metadata (Felfernig, Jeran, Ninaus, Reinfrank, Reiterer, and Stettinger 2014). For example, to provide users with a recommendation for specific articles, CBF uses two types of data. These include information about a collection of articles and information about the profile of the target user (N. Singh et al. 2019). User profile information typically includes the purchase history (Safavi, Jalali, and Houshmand 2022).

Unlike other methods, CBF is not able to suggest truly new products, but always recommends similar products. The use case is therefore mostly limited to text data, where a simple recommendation based on item and user information is possible (Ko et al. 2022). It is also worth mentioning that items that fall into multiple categories also receive higher preferences (N. Singh et al. 2019).

4 Knowledge-Based Recommendation

In contrast to CF and CBF, knowledge-based recommenders (KBR) do not require historic user ratings to make proper recommendations (Safavi, Jalali, and Houshmand 2022). Instead, the system takes user specifications and infers how an item meets the requirements using the respective domain knowledge (Felfernig, Friedrich, et al. 2011). Therefore, intelligent methods like neural networks, fuzzy logic, genetic algorithms, or decision trees are applied (Champiri, Shahamiri, and Salim 2015). The process of recommendation is very interactive. User specifications are collected to derive the best item fit. If the system cannot find a solution, the user has the possibility to modify the requirements (Jannach et al. 2010). In addition, the system provides explanations for the recommended items (Felfernig and Burke 2008). At this point, it should be briefly noted that recommendations do not improve over time, but rather remain static (Burke 2002).

KBR draws on heterogenous, domain-specific sources of knowledge (Bouraga et al. 2014). Ameen (2019) highlights the importance of domain knowledge for KBR by stating the exploitation of deep knowledge in the product domain as one of the system's main tasks to provide valid recommendations. Potential fields of application are infrequently bought products and services that require a high amount of domain-specific knowledge such as financial services, real estate, or health decision support (Safavi, Jalali, and Houshmand 2022). Felfernig, Jeran, Ninaus, Reinfrank, Reiterer, and Stettinger (2014) provide similar examples and denote these products and services as high-involvement items.

There are two approaches of KBR: Case-based and constraint-based recommendation. While both approaches provide the same inputs to the knowledge base, the recommendations are calculated differently (Felfernig, Friedrich, et al. 2011). Case-based recommenders aim to optimally meet the user's desired target (or case) by applying similarity metrics. Constraint-based systems consider rules (or constraints) connecting user specifications and item attributes when searching the knowledge base (Ameen 2019).

4.1 Case-Based Recommendation

For case-based recommendation the knowledge base contains cases, comprising problem descriptions and related solutions (Zhang, Lu, and Jin 2021). As soon as the user files a new requirement, the knowledge base searches for similar specifications, ranks the results according to similarity and provides the best fit to the user (Bouraga et al. 2014).

$$sim(C,Q) = \sum_{a \in A_Q} w_a sim_a(C,Q) \tag{7}$$

In equation 7, $sim_a(C,Q)$ states the similarity between case C to query Q for the attribute a. The similarity is weighted by the importance of the attribute to the customer w_a (Jannach et al. 2010).

While case-based recommendation systems initially followed a query-based approach, many new versions apply a browsing-based approach to retrieve items. Consequently, users must not (re-)specify an exact requirement until a recommendation is valid but define one target that must be fulfilled by the recommended item (e.g., the camera must be cheaper, or the hotel must be closer to the beach). This concept is also known as *critiquing* (Jannach et al. 2010). Case-based recommendation is especially well-applicable in the ethics domain as related situations are usually of high complexity and ambiguity and can be well captured in form of cases (Alyari and Navimipour 2018).

4.2 Constraint-Based Recommendation

In constraint-based or alternatively called rule-based systems, the recommendation of an item depends on the satisfaction of system-internal rules. The rules represent the mapping between user requirements and item attributes and are stored in the knowledge base (Ameen 2019). According to Cena, Console, and Vernero (2021), rules cannot only be related to user preferences but also context related. The authors provide the example of a person enjoying alcoholic drinks, but only in the evening. Next to customer properties (V_C) and product properties (V_{PROD}) the knowledge base of a constraint-based recommendation system consists of three different sets of constraints (C_R, C_F, C_{PROD}). Constraints (C_R) are restricting customer properties and C_{PROD} product properties. C_F represents filter conditions essential to satisfy customer requirements with the available product assortment. Accordingly, a recommendation task can be solved by the constraint satisfaction problem outlined in equation 8 where C_C defines a set of unary rules specifying concrete requirements (Felfernig, Friedrich, et al. 2011).

$$V_C, V_{PROD}, C_C \cup C_F \cup C_R \cup C_{PROD} \tag{8}$$

5 Hybrid Recommendations

As the name of the system suggests, hybrid recommenders are combinations of two or more different recommendation systems (Siswipraptini et al. 2022). Typically, they aim to mitigate respective limitations and, thus, improve the recommendation performance (Alyari and Navimipour 2018). To provide an example, CF may be complemented with KBR to diminish the "cold-start" problem (chapter 7) (Cena, Console, and Vernero 2021).

In 2002, Burke provided a taxonomy of hybrid recommendation to which various authors refer until today. In his paper he states that there are seven hybridization methods: Weighted, switching, mixed, feature combination, cascade, feature augmentation, and meta-level hybrids (Burke 2002). Jannach et al. (2010) further categorized these methods into three hybridization designs which are monolithic, parallelized, and pipelined. As visualized in fig. 1, monolithic hybrids exploit different sources of knowledge as inputs for involved recommendation systems and combine the results into a single recommendation (Jalal and Altun 2016). Feature combination and feature augmentation are part of this category (R. Sharma and R. Singh 2016). Feature combination derives features from all involved recommenders and merges it into one recommendation algorithm (Alyari and Navimipour 2018). Feature augmentation follows the same intent but includes complex transformation steps augmenting the feature space from one recommender system to another (Jannach et al. 2010).

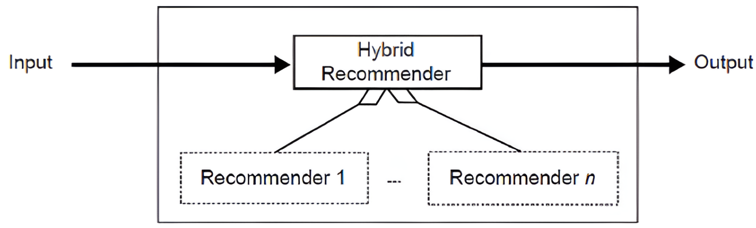

Fig. 1. Monolithic hybridization design (Jannach et al. 2010)

In parallelized hybridization designs all involved recommender systems simultaneously generate recommendation outputs which are subsequently aggregated by a specific hybridization mechanism (Fig. 2). Following Burke's taxonomy, this category includes weighted, mixed, and switching hybrids (R. Sharma and R. Singh 2016). Using weighted hybrids, all involved recommenders are implemented separately predicting the rating of an item. The sum over all ratings determines the recommended item (Alyari and Navimipour 2018). Mixed hybrids provide the top-scoring items per involved recommendation system. If a single recommendation output is desired, conflict resolution such as predefined precedence rules must be applied in addition (Jannach et al. 2010). Switching hybrids are intelligent in a way that they can select the most appropriate algorithm based on the strengths and weaknesses of involved systems (Ghazanfar and Prugel-Bennett 2010).

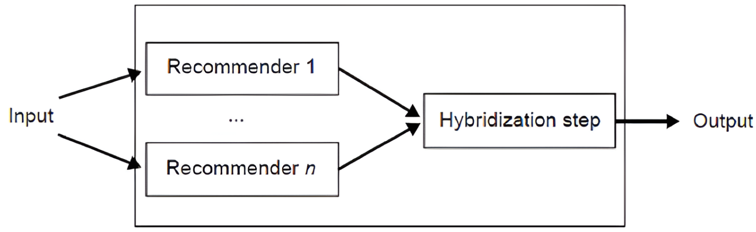

Fig. 2. Parallelized hybridization design (Jannach et al. 2010)

In pipelined hybridization designs the involved recommendation systems operate one after another. Fig. 3 displays the systems as a queue, with the last system producing the final recommendation output (Jalal and Altun 2016). The different parts of the recommender chain either preprocess input data to build a model or deliver recommendations for further refinement (Jannach et al. 2010). This category includes cascade and meta-level hybrids (R. Sharma and R. Singh 2016). When applying cascade hybrids, the higher-priority recommender produces a ranking of items, which is subsequently refined by the second recommendation system. This technique is very efficient as the second system focuses only on items where additional discrimination is required (Alyari and Navimipour 2018). Burke (2002) highlights that cascade and feature augmentation hybrids are often confused. In feature augmentation the recommended features of the second system include the ones of the first recommender. Contrastingly, the result of cascade hybridization is a combination of the prioritized outputs of all involved recommenders, i.e., the successor does not use any output of the predecessor. In meta-level hybrids the predecessor produces a model which is exploited by the successor to generate a recommendation (Jannach et al. 2010).

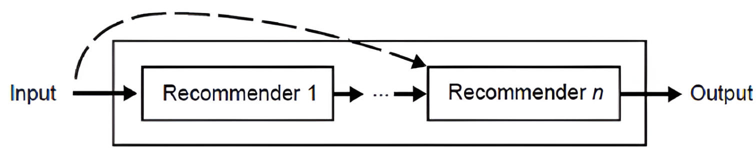

Fig. 3. Pipelined hybridization design (Jannach et al. 2010)

6 Further Approaches of Recommendation Systems

Apart from the basic approaches introduced in the previous chapters, the following section describe further approaches that have been widely applied in the literature.

6.1 Demographic-Based Recommendation

The algorithm of demographic-based recommenders takes demographic data of users, such as gender, postcode or occupation, as input and matches it with all user profiles in the data base (Alyari and Navimipour 2018). It aims to find demographically similar users and recommends items well-rated by the respective demographic user group (Tintarev and Masthoff 2011). Choenyi et al. (2021) emphasize the advantage that no historical data is required and thus the problem of "cold start" does not occur.

6.2 Utility-Based Recommendation

Utility-based recommenders are KBR (Zihayat et al. 2019). Thus, the system follows the approach of recommending items that satisfy customer, product and filtering constraints. In addition, utility-based recommenders consider dimensions that might influence a user's decision, such as quality or cost (Ricci, Rokach, and Shapira 2011). This is particularly useful if the recommender system outputs more than one valid solution, requiring a ranking of items. The ranking can be done by applying the concept of the multi-attribute utility theory (MAUT), where the degree of fit between user requirements and item is determined based on the user's interest in the dimension as well as the item's contribution to the dimension (Jannach et al. 2010).

$$utility(P) = \sum_{j=1}^{(dimensions)} interest(j) * contribution(p, j) \tag{9}$$

In equation 9, $utility(P)$ states the utility of the item p, $interest(j)$ the user's interest in dimension j and $contribution(p, j)$ the contribution of item p to dimension j (Jannach et al. 2010).

6.3 Critique-Based Recommendation

As briefly mentioned in the chapter about case-based recommendation, critiquing allows users to specify feedback on the recommendation. Similar as to case-based recommendation, the user is asked to state requirements on the item. Based on that, the system provides one or more recommendations to the user. At this stage, the user can either chose the preferred item and terminate the process or provide feedback on the system's recommendation (Simran, Pande, and Desai 2019). The feedback is defined as targets on an individual item feature that must be achieved by the recommended item, e.g., the camera must be cheaper (Jannach et al. 2010). The interactive approach is particularly useful in more complex domains where users usually require additional support in the decision process (Ricci, Rokach, and Shapira 2011).

6.4 Group-Based Recommendation

All previously discussed recommendation systems aim to make predictions for individual users. In this paragraph, group-based recommenders or collaborative recommenders are introduced. Group recommendation has increasingly gained in importance in recent years. It can be applied in any domain (Cao et al. 2019). The algorithm attempts to use aggregated information from individuals to make recommendations to groups (Masthoff 2011). Group-based recommenders can follow memory-based and model-based approaches. Memory-based algorithms collect and aggregate user preferences without taking interactions within the group into account (Huang et al. 2021). All available data is taken from the database to provide predictions. In contrast, the model-based algorithm takes all available data and derives a model which is subsequently used from its predictions (Burke, O'Mahony, and Hurley 2015). This model aims at exploiting interactions among group members. For example, the consensus model of Yuan, Cong, and Lin (2014) takes into account the increased influence of individual experts in topics that are relevant for the entire group. Similarly, the deep learning-based algorithm of Wu et al. (2019) computes the weight of an individual's influence in a group and calculates the group recommendation by aggregating the preferences of the group members with different weights (Cao et al. 2019).

7 Challenges

Depending on the type of recommendation systems there are different challenges to tackle. A common problem is the so-called *cold-start* (Table

2). It is defined by the fact that the recommendation engine cannot calculate predictions due to the lack of primary ratings which form the basis for the computation . This problem exists in CF and CBF (Bouraga et al. 2014). The insufficiently available information can concern item-data or user-data. Regarding the former, items with little to no ratings are unlikely to be suggested by the system. Here, it is essential to activate users to provide ratings (N. Singh et al. 2019). A user-cold-start is inevitable when new users are integrated in the system. One potential way to mitigate this issue is to select demographic-based recommenders leveraging demographic similarities between users (Fayyaz et al. 2020).

	Item1	Item2	Item3	Item4	Item5
New User	?	?	?	?	?
User1	3	3	1	4	2
User2	4	2	4	4	5
User3	4	1	2	5	4
User4	5	3	4	3	2

Table 2. Cold-start

A similar limitation is *data sparsity* (Table 3) in the user-item-matrix (Safavi, Jalali, and Houshmand 2022). This results from the fact that only a small proportion of users is leaving ratings (N. Singh et al. 2019). Fayyaz et al. (2020) suggest to apply singular value decomposition (SVD) to reduce the dimensionality of the matrix or to apply a so-called trust-based approach. The authors cite a paper of O'Donovan and Smyth (2005) who introduce a computational model of users' trustworthiness leading to an improved predictive accuracy of the recommendation.

	Item1	Item2	Item3	Item4	Item5
Sabrina	?	4	3	2	?
User1	?	?	?	?	?
User2	?	?	?	?	?
User3	?	?	2	5	4
User4	?	3	?	3	2

Table 3. Data sparsity

A third challenge concerns domain knowledge. While CF and CBF do not require a knowledge base, KBR with all its subcategories relies on deep knowledge about an item's domain (Safavi, Jalali, and Houshmand 2022). This carries many advantages discussed in the chapters above, like a high degree of transparency in what items are recommended (Felfernig, Jeran, Ninaus, Reinfrank, Reiterer, and Stettinger 2014). Yet, a reasonable problem is the so-called *knowledge acquisition bottleneck* (Fig. 4) covering the acquisition, representation and storage of the expert knowledge in the knowledge base (Alyari and Navimipour 2018). In particular, Felfernig, Friedrich, et al. (2011) mention the difficulty to convert the knowledge of the domain experts into formal, executable representations. Yet, not only the construction of the knowledge base but also its maintenance is a complicated task that requires high expertise and domain knowledge (Bouraga et al. 2014).

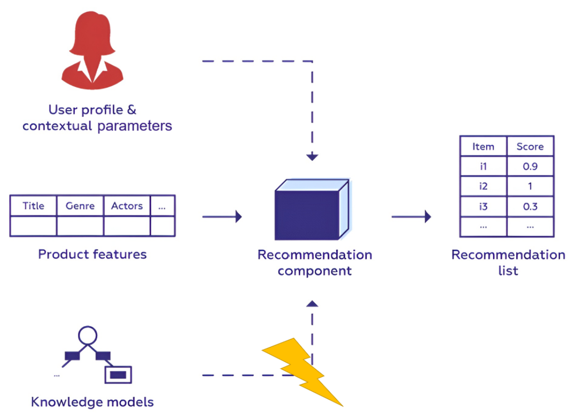

Fig. 4. Knowledge acquisition bottleneck (Venkatesan and Thangadurai 2017)

The authors of the reviewed papers also mention several issues concerning the *user requirements specification* (Fig. 5) in the knowledge base. Accordingly, Safavi, Jalali, and Houshmand (2022) mention the "black box" about offline activities to be challenging in their study about recommendation of attractive places to visit. Furthermore, Felfernig, Jeran, Ninaus, Reinfrank, and Reiterer (2013) highlight the necessity to understand the context in which an item gets recommended, for example if a cinema might only be suggested because it is located close to home. Lastly, recommendations of family members, friends or acquaintances on social media might impact the item selection (He and Chu 2010).

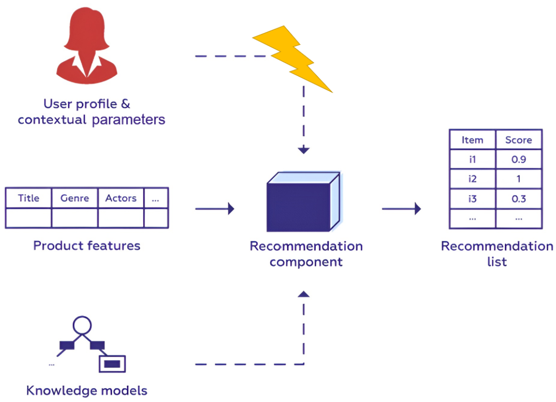

Fig. 5. User requirements specification (Venkatesan and Thangadurai 2017)

Another important factor to consider when implementing recommendation systems is the *serendipity*. It aims to provide the user a reasonable amount of novel suggestions (N. Singh et al. 2019). Fayyaz et al. (2020) mention this as a common challenge as many systems base their recommendation calculations on overlapping instead of differences. This way niche products are less likely to be suggested by the recommendation engine. Fig. 6 displays serendipitous recommendations in dark red, representing unexpected but relevant items to the user.

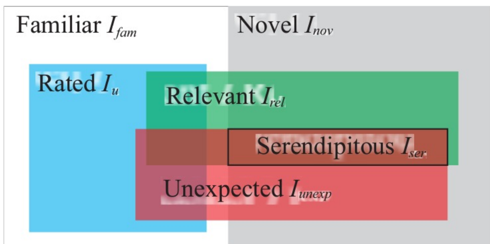

Fig. 6. Serendipity (Kotkov, Wang, and Veijalainen 2016)

Lastly, the challenge of *scalability* (Fig. 7) is increasingly faced due to the increase in users, products and reviews, especially in the fields of e-commerce and entertainment. Mitigation metrics suggested in the literature are to apply dimensionality reduction by clustering or SVD (Fayyaz et al. 2020). N. Singh et al. (2019) recommend focusing on an efficient and effective data model built to handle massive amounts of data or to perform computation on multiple machines in parallel.

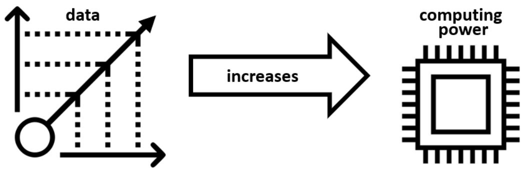

Fig. 7. Scalability

To provide an enhanced overview table 4 contains a comparison of main advantages and disadvantages of basic recommendation approaches presented within this literature summary.

	Advantages	Disadvantages
Collaborative Filtering	- no domain knowledge required [0][1][2] - improves over time [0][1] - simple method [1][2] - popular and widely used [1] - low cost of knowledge acquisition and maintenance [2][4]	- cold-start problem [0] - data set is crucial for quality [0] - changing preferences problem [0] - scalability problem [3][4] - data sparsity [3][4]
Content-Based Filtering	- no domain knowledge required [0] - improves over time [0] - low cost of knowledge acquisition and maintenance [2] - independent user profiles [3][4] - explainable recommendations [3][4]	- cold-start problem [0] - data set is crucial for quality [0] - changing preferences problem [0] - overspecialization problem [3][4] - difficult attribute setting for items [4]
Knowledge-Based Recommendation	- no cold-start problem [0][2] - adaptive to changing preferences [0] - includes non-product features [0] - identifies items by user needs [0]	- no improvement over time [0] - requires knowledge engineering [0]

Table 4. Advantages & disadvantages of basic approaches of recommendation systems; [0] - (Burke 2002); [1] - (S. Sharma, K. Gupta, and D. Gupta 2021); [2] - (Jannach et al. 2010); [3] - (Thorat, Goudar, and Barve 2015); [4] - (Shah et al. 2017)

8 Future Direction

Originally, recommender systems were used primarily in the e-commerce sector with the goal to increase revenue (N. Singh et al. 2019). Today, the systems are increasingly often utilized to achieve business goals. For example, for requirements prioritization in software engineering, point of interest detection in the tourism industry or help services in the field of financial services. Yet, authors emphasize that the viewpoint of the customer is rarely in the focus (Felfernig, Tran, and Le 2021; Felfernig, Jeran, Ninaus, Reinfrank, and Reiterer 2013; Martin et al. 2011; Chung, Sundaram, and Srinivasan 2007). This criticism also matches with the challenge of *user requirements specification* covered in chapter 7. The difficulties may arise from building static recommendation models as well as missing context parameters like the influence of family and friends (Gao et al. 2021; Felfernig, Tran, and Le 2021; Beheshti et al. 2020). The study of Beheshti et al. (2020) about cognitive recommender systems (CRS) addresses multiple of the presented challenges at once. The authors created a framework for "a new type of data-driven, knowledge-driven and cognition-driven recommender system" which is visualized in fig. 8. *Data-driven* implies leveraging artificial intelligence and machine learning techniques to transform massive amounts of raw data into actionable insights. *Knowledge-driven* covers crowdsourcing techniques to counteract the knowledge acquisition bottleneck, while *cognition-driven* aims to overcome the challenge of user requirements specification in the knowledge base. For more detailed information on CRS the papers of Beheshti et al. (2020), Angulo et al. (2020) and HamlAbadi et al. (2017) are recommended.

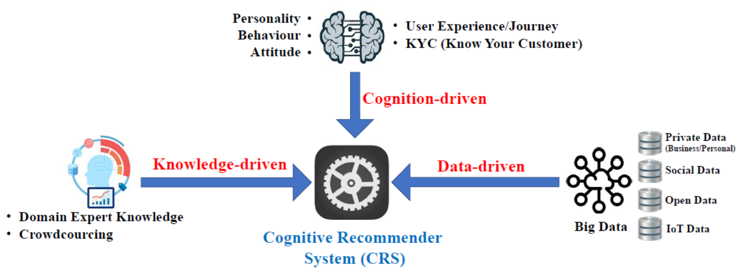

Fig. 8. The three dimensions of cognitive recommender systems (Beheshti et al. 2020)

As outlined in the previous chapters of this paper, KBR provides some supportive features to the user while other approaches represent little to no user-centered recommendation. CRS not only aims to understand user's preferences, but also detect changes over time (*user requirements specification*) and predict unknown favourites (*serendipity*) (Beheshti et al. 2020). Yet, additional research is still required to further enhance the performance of the new recommendation paradigm.

Based on the present literature summary, the following areas are suggested to be addressed in future research.

- Contextualization of user requirements (social relations, time, location, etc.) including its implications on the user's decision-making process.
- Application of social-aware recommendation by considering the influence of a user's social environment (family members, friends, acquaintances on social media) on item selection.
- Application of time-aware recommendation by considering changes in user requirements over time.
- Transfer of a user's rating behavior across domains (e.g., movie/book domain).

9 Conclusion

Based on the available literature, this paper covers four classic approaches of recommendation, namely collaborative filtering, content-based filtering, knowledge-based recommendation and hybrid recommendation. In addition, four further mechanisms are described which emerged based on limitations of the main approaches. This includes demographic-based, utility-based, critique-based and group-based recommendation. While the classic approaches are introduced in detail regarding its algorithms, metrics and fields of application, the additional approaches are briefly covered. Furthermore, general challenges within the implementation of recommendation systems are outlined. Another useful outcome of this paper is the brief review of cognitive recommender systems from the perspective of the previously presented challenges. This new type of recommendation resulted from ongoing socio-technical developments as well as limitations of existing recommendation systems. Accordingly, it provides potential to overcome existing challenges in recommender systems such as the knowledge acquisition bottleneck and user requirements specification.

10 References

Alyari, Fatemeh and Nima Jafari Navimipour (2018). "Recommender systems: a systematic review of the state of the art literature and suggestions for future research". In: *Kybernetes*.

Ameen, Ayesha (2019). "Knowledge based recommendation system in semantic web-a survey". In: *International Journal of Computer Applications* 182.43, pp. 20–25.

Amin, Sana Abida, James Philips, and Nasseh Tabrizi (2019). "Current trends in collaborative filtering recommendation systems". In: *World Congress on Services*. Springer, pp. 46–60.

Angulo, Cecilio et al. (2020). "Bridging cognitive models and recommender systems". In: *Cognitive Computation* 12.2, pp. 426–427.

Bag, Sujoy, Abhijeet Ghadge, and Manoj Kumar Tiwari (2019). "An integrated recommender system for improved accuracy and aggregate diversity". In: *Computers & Industrial Engineering* 130, pp. 187–197.

Beheshti, Amin et al. (2020). "Towards cognitive recommender systems". In: *Algorithms* 13.8, p. 176.

Bouraga, Sarah et al. (2014). "Knowledge-based recommendation systems: A survey". In: *International Journal of Intelligent Information Technologies (IJIIT)* 10.2, pp. 1–19.

Burke, Robin (2002). "Hybrid recommender systems: Survey and experiments". In: *User modeling and user-adapted interaction* 12.4, pp. 331–370.

Burke, Robin, Michael P O'Mahony, and Neil J Hurley (2015). "Robust collaborative recommendation". In: *Recommender systems handbook*. Springer, pp. 961–995.

Cao, Da et al. (2019). "Social-enhanced attentive group recommendation". In: *IEEE Transactions on Knowledge and Data Engineering* 33.3, pp. 1195–1209.

Cena, Federica, Luca Console, and Fabiana Vernero (2021). "Logical foundations of knowledge-based recommender systems: A unifying spectrum of alternatives". In: *Information Sciences* 546, pp. 60–73.

Champiri, Zohreh Dehghani, Seyed Reza Shahamiri, and Siti Salwah Binti Salim (2015). "A systematic review of scholar context-aware recommender systems". In: *Expert Systems with Applications* 42.3, pp. 1743–1758.

Choenyi, T et al. (2021). "A review on filtering techniques used in restaurant recommendation system". In: *Int. J. Comput. Sci. Mob. Comput* 10.4, pp. 113–117.

Chung, Ronald, David Sundaram, and Ananth Srinivasan (2007). "Integrated personal recommender systems". In: *Proceedings of the ninth international conference on Electronic commerce*, pp. 65–74.

Fayyaz, Zeshan et al. (2020). "Recommendation systems: Algorithms, challenges, metrics, and business opportunities". In: *applied sciences* 10.21, p. 7748.

Felfernig, Alexander and Robin Burke (2008). "Constraint-based recommender systems: technologies and research issues". In: *Proceedings of the 10th international conference on Electronic commerce*, pp. 1–10.

Felfernig, Alexander, Gerhard Friedrich, et al. (2011). "Developing constraint-based recommenders". In: *Recommender systems handbook*. Springer, pp. 187–215.

Felfernig, Alexander, Michael Jeran, Gerald Ninaus, Florian Reinfrank, and Stefan Reiterer (2013). "Toward the next generation of recommender systems: applications and research challenges". In: *Multimedia services in intelligent environments*, pp. 81–98.

Felfernig, Alexander, Michael Jeran, Gerald Ninaus, Florian Reinfrank, Stefan Reiterer, and Martin Stettinger (2014). "Basic approaches in recommendation systems". In: *Recommendation Systems in Software Engineering*. Springer, pp. 15–37.

Felfernig, Alexander, Thi Ngoc Trang Tran, and Viet-Man Le (2021). "Recommender Systems Beyond E-Commerce: Presence and Future". In: *Consumer Happiness: Multiple Perspectives*. Springer, pp. 203–230.

Fouladi, Parisa and Nima Jafari Navimipour (2017). "Human resources ranking in a cloud-based knowledge sharing framework using the quality control criteria". In: *Kybernetes*.

Gao, Chongming et al. (2021). "Advances and challenges in conversational recommender systems: A survey". In: *AI Open* 2, pp. 100–126.

Ghazanfar, Mustansar and Adam Prugel-Bennett (2010). "An improved switching hybrid recommender system using naive bayes classifier and collaborative filtering". In.

HamlAbadi, Kamran Gholizadeh et al. (2017). "A framework for cognitive recommender systems in the Internet of Things (IoT)". In: *2017 IEEE 4th international conference on knowledge-based engineering and innovation (KBEI)*. IEEE, pp. 0971–0976.

He, Jianming and Wesley W Chu (2010). "A social network-based recommender system (SNRS)". In: *Data mining for social network data*. Springer, pp. 47–74.

Huang, Zhenhua et al. (2021). "A novel group recommendation model with two-stage deep learning". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Jalal, Ahmed Adeeb and Oğuz Altun (2016). "Feature Engineering in Hybrid Recommender Systems". In: *Feature Engineering in Hybrid Recommender Systems*, p. 1.

Jannach, Dietmar et al. (2010). *Recommender systems: an introduction.* Cambridge University Press.

Kalkar, Shailesh D and Pramila M Chawan (2022). "Recommendation System using Machine Learning Techniques". In.

Ko, Hyeyoung et al. (2022). "A survey of recommendation systems: recommendation models, techniques, and application fields". In: *Electronics* 11.1, p. 141.

Kotkov, Denis, Shuaiqiang Wang, and Jari Veijalainen (2016). "A survey of serendipity in recommender systems". In: *Knowledge-Based Systems* 111, pp. 180–192.

Lu, Jie et al. (2015). "Recommender system application developments: a survey". In: *Decision Support Systems* 74, pp. 12–32.

Martin, Francisco J et al. (2011). "The big promise of recommender systems". In: *AI Magazine* 32.3, pp. 19–27.

Masthoff, Judith (2011). "Group recommender systems: Combining individual models". In: *Recommender systems handbook*. Springer, pp. 677–702.

O'Donovan, John and Barry Smyth (2005). "Trust in recommender systems". In: *Proceedings of the 10th international conference on Intelligent user interfaces*, pp. 167–174.

Ricci, Francesco, Lior Rokach, and Bracha Shapira (2011). "Introduction to recommender systems handbook". In: *Recommender systems handbook*. Springer, pp. 1–35.

— (2015). "Recommender systems: introduction and challenges". In: *Recommender systems handbook*. Springer, pp. 1–34.

Safavi, Sadaf, Mehrdad Jalali, and Mahboobeh Houshmand (2022). "Toward point-of-interest recommendation systems: A critical review on deep-learning Approaches". In: *Electronics* 11.13, p. 1998.

Shah, Kunal et al. (2017). "Recommender systems: An overview of different approaches to recommendations". In: *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. IEEE, pp. 1–4.

Sharma, Richa and Rahul Singh (2016). “Evolution of recommender systems from ancient times to modern era: a survey”. In: *Indian Journal of Science and Technology* 9.20, pp. 1–12.
Sharma, Sheetal, Kamali Gupta, and Deepali Gupta (2021). “Recommender System: A bibliometric analysis”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 1022. 1. IOP Publishing, p. 012057.
Simran, Shristi, Akash Pande, and Payal Desai (2019). “Preference-Search based Recommendation System for Accommodation Facilitator: A Review”. In: *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 5.2, pp. 951–956.
Singh, Neelam et al. (2019). “Recommendation Systems in the Big Data Era”. In: *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*.
Siswipraptini, Puji Catur et al. (2022). “Trends and Characteristics of Career Recommendation Systems for Fresh Graduated Students”. In: *2022 10th International Conference on Information and Education Technology (ICIET)*. IEEE, pp. 355–361.
Su, Xiaoyuan and Taghi M Khoshgoftaar (2009). “A survey of collaborative filtering techniques”. In: *Advances in artificial intelligence* 2009.
Thorat, Poonam B, Rajeshwari M Goudar, and Sunita Barve (2015). “Survey on collaborative filtering, content-based filtering and hybrid recommendation system”. In: *International Journal of Computer Applications* 110.4, pp. 31–36.
Tintarev, Nava and Judith Masthoff (2011). “Designing and evaluating explanations for recommender systems”. In: *Recommender systems handbook*. Springer, pp. 479–510.
Venkatesan, M and K Thangadurai (2017). “History and overview of the recommender systems”. In: *Collaborative Filtering Using Data Mining and Analysis*. IGI Global, pp. 74–99.
Wu, Di et al. (2019). “A deep latent factor model for high-dimensional and sparse matrices in recommender systems”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51.7, pp. 4285–4296.
Yuan, Quan, Gao Cong, and Chin-Yew Lin (2014). “COM: a generative model for group recommendation”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 163–172.
Zhang, Qian, Jie Lu, and Yaochu Jin (2021). “Artificial intelligence in recommender systems”. In: *Complex & Intelligent Systems* 7.1, pp. 439–457.

Zihayat, Morteza et al. (2019). “A utility-based news recommendation system”. In: *Decision Support Systems* 117, pp. 14–27.